

Contents

[Azure Video Indexer documentation](#)

[Overview](#)

[What is Azure Video Indexer?](#)

[Compliance privacy security](#)

[Account types](#)

[Language support](#)

[Quickstarts](#)

[Sign up and upload your first video](#)

[Invite users](#)

[Tutorials](#)

[Create an ARM-based account](#)

[Deploy using Bicep](#)

[Deploy using ARM template](#)

[Develop with API](#)

[Samples](#)

[Azure Video Indexer samples](#)

[Azure Video Indexer widgets](#)

[Concepts](#)

[Terminology & concepts overview](#)

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

[Manage multiple tenants](#)

[Using at scale](#)

[Customizing content models](#)

[Overview](#)

[Animated characters](#)

[Brands](#)

[Language](#)

[Person](#)

[Responsible use of AI](#)

Limited access features

Transparency notes, use cases

Transparency notes

Audio effects detection

Faces detection

OCR

Keywords extraction

Transcription, translation, language

Labels identification

Named entities

Observed people tracking & matched faces

Topics inference

How to guides

Manage accounts

Manage access to your account

Connect an existing account to ARM

Add contributor role to Media Services

Import content from the trial account

Create a classic account

Upload and index media

Upload and index videos

Index a video from OneDrive

Examine Azure Video Indexer output

Use insights

View and edit insights

Find exact moments within videos

Detect scenes, shots, keyframes

Insert or remove transcript lines

Identify and transcribe multi-language content

Use editor to create projects

Automatically identify spoken languages

Observed people (preview)

- [Detected clothing \(preview\)](#)
- [Featured clothing \(preview\)](#)
- [Matched person \(preview\)](#)
- [Audio effects detection \(preview\)](#)
- [Customize content models](#)
 - [Animated characters](#)
 - [Person
 - \[using the website\]\(#\)
 - \[using the API\]\(#\)](#)
 - [Brands
 - \[using the website\]\(#\)
 - \[using the API\]\(#\)](#)
 - [Language
 - \[using the website\]\(#\)
 - \[using the API\]\(#\)](#)
 - [Embed widgets into your application](#)
- [Logic Apps connector
 - \[with ARM-based AVI accounts\]\(#\)
 - \[with classic-based AVI accounts\]\(#\)](#)
- [Use live stream analysis](#)
- [Monitor Video Indexer](#)
- [Network Security](#)
- [Disaster recovery](#)
- [Troubleshooting
 - \[Account connection issues\]\(#\)
 - \[Switch between tenants in portal\]\(#\)](#)
- [Reference
 - \[Azure Video Indexer API\]\(#\)
 - \[Azure Video Indexer ARM REST API\]\(#\)
 - \[Monitor Video Indexer data reference\]\(#\)](#)
- [Resources
 - \[Azure Roadmap\]\(#\)](#)

[Pricing](#)

[Regional availability](#)

[Regions](#)

[Microsoft Q&A](#)

[FAQ](#)

[Compliance](#)

[Release notes](#)

[Stack Overflow](#)

[Azure ideas](#)

[Blogs](#)

What is Azure Video Indexer?

9/22/2022 • 9 minutes to read • [Edit Online](#)

WARNING

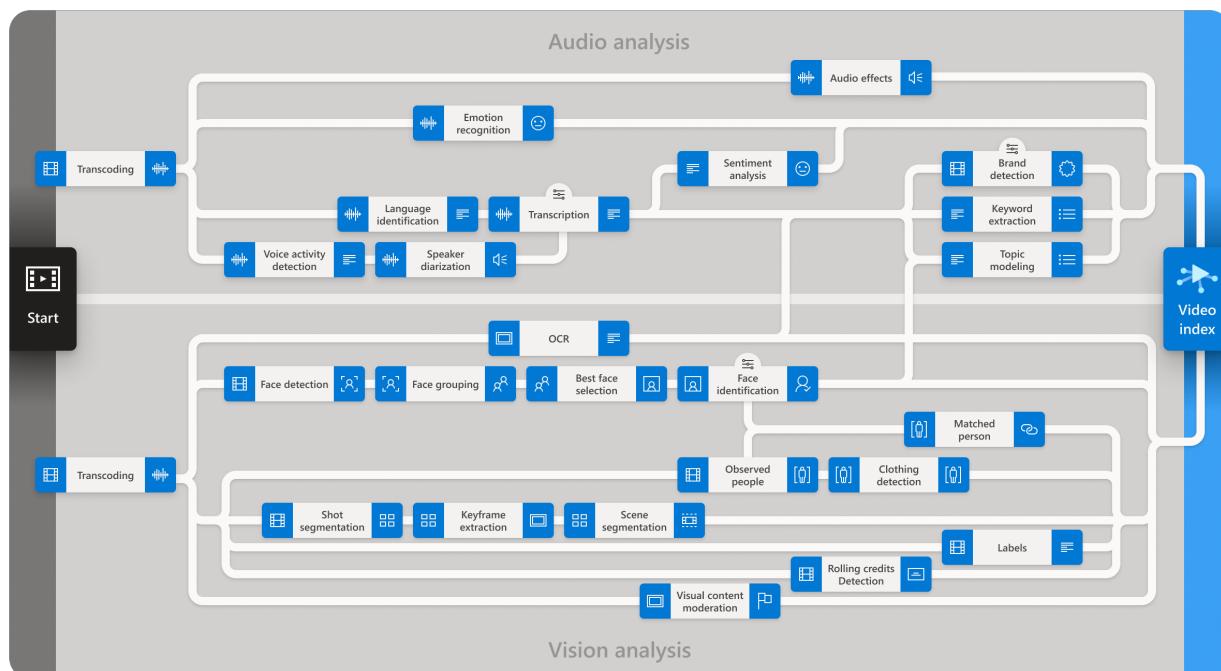
On June 11, 2020, Microsoft announced that it will not sell facial recognition technology to police departments in the United States until strong regulation, grounded in human rights, has been enacted. As such, customers may not use facial recognition features or functionality included in Azure Services, such as Face or Azure Video Indexer, if a customer is, or is allowing use of such services by or for, a police department in the United States.

IMPORTANT

Face identification, customization and celebrity recognition features access is limited based on eligibility and usage criteria in order to support our Responsible AI principles. Face identification, customization and celebrity recognition features are only available to Microsoft managed customers and partners. Use the [Face Recognition intake form](#) to apply for access.

Azure Video Indexer is a cloud application, part of Azure Applied AI Services, built on Azure Media Services and Azure Cognitive Services (such as the Face, Translator, Computer Vision, and Speech). It enables you to extract the insights from your videos using Azure Video Indexer video and audio models.

Azure Video Indexer analyzes the video and audio content by running 30+ AI models, generating rich insights. Below is an illustration of the audio and video analysis performed by Azure Video Indexer in the background.



To start extracting insights with Azure Video Indexer, see the [how can I get started](#) section below.

Compliance, Privacy and Security

IMPORTANT

Before you continue with Azure Video Indexer, read [Compliance, privacy and security](#).

What can I do with Azure Video Indexer?

Azure Video Indexer's insights can be applied to many scenarios, among them are:

- Deep search: Use the insights extracted from the video 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 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: Create trailers, highlight reels, social media content, or news clips based on the insights Azure Video Indexer extracts from your content. Keyframes, scenes markers, and timestamps of the people and label appearances make the creation process smoother and easier, enabling you to easily get to the parts of the video you need when creating content.
- Accessibility: Whether you want to make your content available for people with disabilities or if you want your content to be distributed to different regions using different languages, you can use the transcription and translation provided by Azure Video Indexer in multiple languages.
- Monetization: Azure Video Indexer can help increase the value of videos. For example, industries that rely on ad revenue (news media, social media, and so on) can deliver relevant ads by using the extracted insights as additional signals to the ad server.
- Content moderation: Use textual and visual content moderation models to keep your users safe from inappropriate content and validate that the content you publish matches your organization's values. You can automatically block certain videos or alert your users about the content.
- Recommendations: Video insights can be used to improve user engagement by highlighting the relevant video moments to users. By tagging each video with additional metadata, you can recommend to users the most relevant videos and highlight the parts of the video that will match their needs.

Video/audio AI features

The following list shows the insights you can retrieve from your video/audio files using Azure Video Indexer video and audio AI features (models).

Unless specified otherwise, a model is generally available.

Video models

- **Face detection:** Detects and groups faces appearing in the video.
- **Celebrity identification:** Identifies over 1 million celebrities—like world leaders, actors, artists, athletes, researchers, business, and tech leaders across the globe. The data about these celebrities can also be found on various websites (IMDB, Wikipedia, and so on).
- **Account-based face identification:** 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 Azure Video Indexer website](#) and [Customize a Person model with the Azure Video Indexer API](#).
- **Thumbnail extraction for faces:** Identifies the best captured face in each group of faces (based on quality, size, and frontal position) and extracts it as an image asset.
- **Optical character recognition (OCR):** Extracts text from images like pictures, street signs and products in media files to create insights.
- **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's 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:** Identifies the beginning and end of the rolling credits in the end of TV shows and movies.
- **Animated characters detection :** Detects, groups, and recognizes characters in animated content via integration with [Cognitive Services custom vision](#). For more information, see [Animated character detection](#).
- **Editorial shot type detection:** Tags shots based on their type (like wide shot, medium shot, close up, extreme close up, two shot, multiple people, outdoor and indoor, and so on). For more information, see [Editorial shot type detection](#).
- **Observed people tracking** (preview): Detects observed people in videos and provides information such as the location of the person in the video frame (using bounding boxes) and the exact timestamp (start, end) and confidence when a person appears. For more information, see [Trace observed people in a video](#).
 - **People's detected clothing** (preview): Detects the clothing types of people appearing in the video and provides information such as long or short sleeves, long or short pants and skirt or dress. The detected clothing is associated with the people wearing it and the exact timestamp (start, end) along with a confidence level for the detection are provided. For more information, see [detected clothing](#).
 - **Featured clothing** (preview): captures featured clothing images appearing in a video. You can improve your targeted ads by using the featured clothing insight. For information on how the featured clothing images are ranked and how to get the insights, see [featured clothing](#).
- **Matched person** (preview): Matches people that were observed in the video with the corresponding faces detected. The matching between the observed people and the faces contain a confidence level.

Audio models

- **Audio transcription:** Converts speech to text over 50 languages and allows extensions. For more information, see [Azure Video Indexer language support](#).
- **Automatic language detection:** Identifies the dominant spoken language. For more information, see [Azure Video Indexer language support](#). If the language can't be identified with confidence, Azure Video Indexer assumes the spoken language is English. For more information, see [Language identification model](#).
- **Multi-language speech identification and transcription:** Identifies the spoken language in different segments from audio. It sends each segment of the media file to be transcribed and then combines the transcription back to one unified transcription. For more information, see [Automatically identify and transcribe multi-language content](#).
- **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 Azure Video Indexer website](#) and [Customize a Language model with the Azure Video Indexer APIs](#).
- **Speaker enumeration:** Maps and understands which speaker spoke which words and when. Sixteen speakers can be detected in a single audio-file.
- **Speaker statistics:** Provides statistics for speakers' speech ratios.
- **Textual content moderation:** Detects explicit text in the audio transcript.
- **Emotion detection:** Identifies emotions based on speech (what's being said) and voice tonality (how it's being said). The emotion could be joy, sadness, anger, or fear.
- **Translation:** Creates translations of the audio transcript to many different languages. For more

information, see [Azure Video Indexer language support](#).

- **Audio effects detection** (preview): Detects the following audio effects in the non-speech segments of the content: alarm or siren, dog barking, crowd reactions (cheering, clapping, and booing), gunshot or explosion, laughter, breaking glass, and silence.

The detected acoustic events are in the closed captions file. The file can be downloaded from the Azure Video Indexer portal. For more information, see [Audio effects detection](#).

NOTE

The full set of events is available only when you choose **Advanced Audio Analysis** when uploading a file, in upload preset. By default, only silence is detected.

Audio and video models (multi-channels)

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

- **Keywords extraction**: Extracts keywords from speech and visual text.
- **Named entities extraction**: Extracts brands, locations, and people from speech and visual text via natural language processing (NLP).
- **Topic inference**: Extracts topics based on various keywords (that is, keywords 'Stock Exchange', 'Wall Street' will produce the topic 'Economics'). The model uses three different ontologies ([IPTC](#), [Wikipedia](#) and the Video Indexer hierarchical topic ontology). The model uses transcription (spoken words), OCR content (visual text), and celebrities recognized in the video using the Video Indexer facial recognition model.
- **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.

How can I get started with Azure Video Indexer?

Prerequisite

Before creating a new account, review [Account types](#).

Supported browsers

The following list shows the supported browsers that you can use for the Azure Video Indexer website and for your apps that embed the widgets. The list also shows the minimum supported browser version:

- Edge, version: 16
- Firefox, version: 54
- Chrome, version: 58
- Safari, version: 11
- Opera, version: 44
- Opera Mobile, version: 59
- Android Browser, version: 81
- Samsung Browser, version: 7
- Chrome for Android, version: 87
- Firefox for Android, version: 83

Supported file formats

See the [input container/file formats](#) article for a list of file formats that you can use with Azure Video Indexer.

Start using Azure Video Indexer

You can access Azure Video Indexer capabilities in three ways:

- Azure Video Indexer portal: An easy-to-use solution that lets you evaluate the product, manage the account, and customize models.

For more information about the portal, see [Get started with the Azure Video Indexer website](#).

- API integration: All of Azure Video Indexer's capabilities are available through a REST API, which lets you integrate the solution into your apps and infrastructure.

To get started as a developer, see [Use Azure Video Indexer REST API](#).

- Embeddable widget: Lets you embed the Azure Video Indexer insights, player, and editor experiences into your app.

For more information, see [Embed visual widgets in your application](#). If you're using the website, the insights are added as metadata and are visible in the portal. If you're using APIs, the insights are available as a JSON file.

Next steps

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

- [Pricing](#)
- [Get started with the Azure Video Indexer website](#).
- [Process content with Azure Video Indexer REST API](#).
- [Embed visual widgets in your application](#).

For the latest updates, see [Azure Video Indexer release notes](#).

Compliance, Privacy and Security

9/22/2022 • 2 minutes to read • [Edit Online](#)

As an important reminder, you must comply with all applicable laws in your use of Azure Video Indexer, and you may not use Azure Video Indexer or any Azure service in a manner that violates the rights of others, or that may be harmful to others.

Before uploading any video/image to Azure Video Indexer, You must have all the proper rights to use the video/image, including, where required by law, all the necessary consents from individuals (if any) in the video/image, for the use, processing, and storage of their data in Azure Video Indexer and Azure. Some jurisdictions may impose special legal requirements for the collection, online processing and storage of certain categories of data, such as biometric data. Before using Azure Video Indexer and Azure for the processing and storage of any data subject to special legal requirements, You must ensure compliance with any such legal requirements that may apply to You.

To learn about compliance, privacy and security in Azure Video Indexer please visit the Microsoft [Trust Center](#). For Microsoft's privacy obligations, data handling and retention practices, including how to delete your data, please review Microsoft's [Privacy Statement](#), the [Online Services Terms](#) ("OST") and [Data Processing Addendum](#) ("DPA"). By using Azure Video Indexer, you agree to be bound by the OST, DPA and the Privacy Statement.

Next steps

[Azure Video Indexer overview](#)

Azure Video Indexer account types

9/22/2022 • 3 minutes to read • [Edit Online](#)

This article gives an overview of Azure Video Indexer accounts types and provides links to other articles for more details.

A trial account

The first time you visit the [Azure Video Indexer](#) website, a trial account is automatically created. The trial Azure Video Indexer account has limitation on number of indexing minutes, support, and SLA.

With a trial, account Azure Video Indexer provides:

- up to 600 minutes of free indexing to the [Azure Video Indexer](#) website users and
- up to 2400 minutes of free indexing to users that subscribe to the Azure Video Indexer API on the [developer portal](#).

When using the trial account, you don't have to set up an Azure subscription.

The trial account option is not available on the Azure Government cloud. For other Azure Government limitations, see [Limitations of Azure Video Indexer on Azure Government](#).

A paid (unlimited) account

You can later create a paid account where you're not limited by the quota. Two types of paid accounts are available to you: Azure Resource Manager (recommended) and classic. The main difference between the two is account management platform.

- ARM-based accounts management is built on Azure, which enables using [Azure RBAC](#) (recommended).
- Classic accounts are built on the API Management.

With the paid option, you pay for indexed minutes, for more information, see [Azure Video Indexer pricing](#).

When creating a new paid account, you need to connect the Azure Video Indexer account to your Azure subscription and an Azure Media Services account.

Create accounts

NOTE

It is recommended to use Azure Video Indexer ARM-based accounts.

- [Create an ARM-based \(paid\) account in Azure portal](#). To create an account with an API, see [Accounts](#)

TIP

Make sure you are signed in with the correct domain to the [Azure Video Indexer website](#). For details, see [Switch tenants](#).

- [Upgrade a trial account to an ARM-based \(paid\) account and import your content for free](#).

- If you have a classic (paid) account, [connect an existing classic Azure Video Indexer account to an ARM-based account](#).

Unless your code or infrastructure prevents you from moving from existing classic accounts, you should start using ARM-based accounts.

To get access to your account

	ARM-BASED	CLASSIC	TRIAL
Get access token	ARM REST API	Get access token	Same as classic
Share account	Azure RBAC(role based access control)	Invite users	Same as classic

Limited access features

This section talks about limited access features in Azure Video Indexer.

WHEN DID I CREATE THE ACCOUNT?	TRIAL ACCOUNT (FREE)	PAID ACCOUNT (CLASSIC OR ARM-BASED)
Existing VI accounts created before June 21, 2022	Able to access face identification, customization and celebrities recognition till June 2023. Recommended: Move to a paid account and afterward fill in the intake form and based on the eligibility criteria we will enable the features also after the grace period.	Able to access face identification, customization and celebrities recognition till June 2023*. Recommended: fill in the intake form and based on the eligibility criteria we will enable the features also after the grace period. We proactively sent emails to these customers + AEs.
New VI accounts created after June 21, 2022	Not able the access face identification, customization and celebrities recognition as of today. Recommended: Move to a paid account and afterward fill in the intake form . Based on the eligibility criteria we will enable the features (after max 10 days).	Azure Video Indexer disables the access face identification, customization and celebrities recognition as of today by default, but gives the option to enable it. Recommended: Fill in the intake form and based on the eligibility criteria we will enable the features (after max 10 days).

*In Brazil South we also disabled the face detection.

For more information, see [Azure Video Indexer limited access features](#).

Next steps

Make sure to review [Pricing](#).

Language support in Azure Video Indexer

9/22/2022 • 4 minutes to read • [Edit Online](#)

This article provides a comprehensive list of language support by service features in Azure Video Indexer. For the list and definitions of all the features, see [Overview](#).

NOTE

The list below contains the source languages for transcription that are supported by the Video Indexer API. Some languages are supported only through the API and not through the Video Indexer website or widgets.

To make sure a language is supported for search, transcription, or translation by the Azure Video Indexer website and widgets, see the [frontend language support table](#) further below.

General language support

This section describes language support in Azure Video Indexer.

- Transcription (source language of the video/audio file)
- Language identification (LID)
- Multi-language identification (MLID)
- Translation

The following insights are translated, otherwise will remain in English:

- Transcript
 - Keywords
 - Topics
 - Labels
 - Frame patterns (Only to Hebrew as of now)
- Language customization

LANGUAGE	CODE	TRANSCRIPTI ON	LID	MLID	TRANSLATIO N	CUSTOMIZATI ON (LANGE UAGE MODEL)
Afrikaans	af-ZA				✓	
Arabic (Israel)	ar-IL	✓			✓	✓
Arabic (Jordan)	ar-JO	✓			✓	✓
Arabic (Kuwait)	ar-KW	✓			✓	✓
Arabic (Lebanon)	ar-LB	✓			✓	✓

LANGUAGE	CODE	TRANSCRIPTI ON	LID	MLID	TRANSLATIO N	CUSTOMIZATI ON (LANGUAGE MODEL)
Arabic (Oman)	ar-OM	✓			✓	✓
Arabic (Palestinian Authority)	ar-PS	✓			✓	✓
Arabic (Qatar)	ar-QA	✓			✓	✓
Arabic (Saudi Arabia)	ar-SA	✓			✓	✓
Arabic (United Arab Emirates)	ar-AE	✓			✓	✓
Arabic Egypt	ar-EG	✓			✓	✓
Arabic Modern Standard (Bahrain)	ar-BH	✓			✓	✓
Arabic Syrian Arab Republic	ar-SY	✓			✓	✓
Bangla	bn-BD				✓	
Bosnian	bs-Latn				✓	
Bulgarian	bg-BG				✓	
Catalan	ca-ES				✓	
Chinese (Cantonese Traditional)	zh-HK	✓		✓	✓	✓
Chinese (Simplified)	zh-Hans	✓			✓	✓
Chinese (Traditional)	zh-Hant				✓	
Croatian	hr-HR				✓	
Czech	cs-CZ	✓			✓	✓
Danish	da-DK	✓			✓	✓

LANGUAGE	CODE	TRANSCRIPTION	LID	MLID	TRANSLATION	CUSTOMIZATION (LANGUAGE MODEL)
Dutch	nl-NL	✓			✓	✓
English Australia	en-AU	✓			✓	✓
English United Kingdom	en-GB	✓			✓	✓
English United States	en-US	✓	✓	✓	✓	✓
Estonian	et-EE				✓	
Fijian	en-FJ				✓	
Filipino	fil-PH				✓	
Finnish	fi-FI	✓			✓	✓
French	fr-FR	✓	✓	✓	✓	✓
French (Canada)	fr-CA	✓			✓	✓
German	de-DE	✓	✓	✓	✓	✓
Greek	el-GR				✓	
Haitian	fr-HT				✓	
Hebrew	he-IL	✓			✓	✓
Hindi	hi-IN	✓			✓	✓
Hungarian	hu-HU				✓	
Indonesian	id-ID				✓	
Italian	it-IT	✓	✓		✓	✓
Japanese	ja-JP	✓	✓		✓	✓
Kiswahili	sw-KE				✓	
Korean	ko-KR	✓			✓	✓
Latvian	lv-LV				✓	

LANGUAGE	CODE	TRANSCRIPTION	LID	MLID	TRANSLATION	CUSTOMIZATION (LANGUAGE MODEL)
Lithuanian	lt-LT				✓	
Malagasy	mg-MG				✓	
Malay	ms-MY				✓	
Maltese	mt-MT				✓	
Norwegian	nb-NO	✓			✓	✓
Persian	fa-IR	✓			✓	✓
Polish	pl-PL	✓			✓	✓
Portuguese	pt-BR	✓	✓		✓	✓
Portuguese (Portugal)	pt-PT	✓			✓	✓
Romanian	ro-RO				✓	
Russian	ru-RU	✓	✓		✓	✓
Samoan	en-WS				✓	
Serbian (Cyrillic)	sr-Cyrl-RS				✓	
Serbian (Latin)	sr-Latn-RS				✓	
Slovak	sk-SK				✓	
Slovenian	sl-SI				✓	
Spanish	es-ES	✓	✓	✓	✓	✓
Spanish (Mexico)	es-MX	✓			✓	✓
Swedish	sv-SE	✓			✓	✓
Tamil	ta-IN				✓	
Thai	th-TH	✓			✓	✓
Tongan	to-TO				✓	

LANGUAGE	CODE	TRANSCRIPTI ON	LID	MLID	TRANSLATIO N	CUSTOMIZATI ON (LANGUAGE MODEL)
Turkish	tr-TR	✓			✓	✓
Ukrainian	uk-UA	✓			✓	
Urdu	ur-PK				✓	
Vietnamese	vi-VN	✓			✓	

Language support in frontend experiences

The following table describes language support in the Azure Video Indexer frontend experiences.

- website - the website column lists supported languages for the [Azure Video Indexer website](#). For more information, see [Get started](#).
- widgets - the widgets column lists supported languages for translating the index file. For more information, see [Get started](#).

LANGUAGE	CODE	WEBSITE	WIDGETS
Afrikaans	af-ZA		✓
Arabic (Iraq)	ar-IQ		
Arabic (Israel)	ar-IL		
Arabic (Jordan)	ar-JO		
Arabic (Kuwait)	ar-KW		
Arabic (Lebanon)	ar-LB		
Arabic (Oman)	ar-OM		
Arabic (Palestinian Authority)	ar-PS		
Arabic (Qatar)	ar-QA		
Arabic (Saudi Arabia)	ar-SA		
Arabic (United Arab Emirates)	ar-AE		
Arabic Egypt	ar-EG		✓
Arabic Modern Standard (Bahrain)	ar-BH		

LANGUAGE	CODE	WEBSITE	WIDGETS
Arabic Syrian Arab Republic	ar-SY		
Bangla	bn-BD		✓
Bosnian	bs-Latn		✓
Bulgarian	bg-BG		✓
Catalan	ca-ES		✓
Chinese (Cantonese Traditional)	zh-HK		✓
Chinese (Simplified)	zh-Hans	✓	✓
Chinese (Traditional)	zh-Hant		✓
Croatian	hr-HR		
Czech	cs-CZ	✓	✓
Danish	da-DK		✓
Dutch	nl-NL	✓	✓
English Australia	en-AU		✓
English United Kingdom	en-GB		✓
English United States	en-US	✓	✓
Estonian	et-EE		✓
Fijian	en-FJ		✓
Filipino	fil-PH		✓
Finnish	fi-FI		✓
French	fr-FR		✓
French (Canada)	fr-CA	✓	✓
German	de-DE	✓	
Greek	el-GR		✓
Haitian	fr-HT		✓

LANGUAGE	CODE	WEBSITE	WIDGETS
Hebrew	he-IL		✓
Hindi	hi-IN	✓	✓
Hungarian	hu-HU	✓	✓
Indonesian	id-ID		
Italian	it-IT		✓
Japanese	ja-JP	✓	✓
Kiswahili	sw-KE	✓	✓
Korean	ko-KR	✓	✓
Latvian	lv-LV		✓
Lithuanian	lt-LT		✓
Malagasy	mg-MG		✓
Malay	ms-MY		✓
Maltese	mt-MT		
Norwegian	nb-NO		✓
Persian	fa-IR		
Polish	pl-PL	✓	✓
Portuguese	pt-BR	✓	✓
Portuguese (Portugal)	pt-PT		✓
Romanian	ro-RO		✓
Russian	ru-RU	✓	✓
Samoan	en-WS		
Serbian (Cyrillic)	sr-Cyrl-RS		✓
Serbian (Latin)	sr-Latn-RS		
Slovak	sk-SK		✓

LANGUAGE	CODE	WEBSITE	WIDGETS
Slovenian	sl-SI		✓
Spanish	es-ES	✓	✓
Spanish (Mexico)	es-MX		✓
Swedish	sv-SE	✓	✓
Tamil	ta-IN		✓
Thai	th-TH		✓
Tongan	to-TO		✓
Turkish	tr-TR	✓	✓
Ukrainian	uk-UA	✓	✓
Urdu	ur-PK		✓
Vietnamese	vi-VN	✓	✓

Next steps

- [Overview](#)
- [Release notes](#)

Quickstart: How to sign up and upload your first video

9/22/2022 • 2 minutes to read • [Edit Online](#)

IMPORTANT

Face identification, customization and celebrity recognition features access is limited based on eligibility and usage criteria in order to support our Responsible AI principles. Face identification, customization and celebrity recognition features are only available to Microsoft managed customers and partners. Use the [Face Recognition intake form](#) to apply for access.

This quickstart shows you how to sign in to the Azure Video Indexer [website](#) and how to upload your first video.

When you visit the [Azure Video Indexer](#) website for the first time, a trial account is automatically created for you. With the trial account, you get some free indexing minutes. You can later add a paid account. With the paid option, you pay for indexed minutes. For details about available accounts (trial and paid options), see [Azure Video Indexer account types](#).

Sign up for Azure Video Indexer

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

Once you start using Azure Video Indexer, all your stored data and uploaded content are encrypted at rest with a Microsoft managed key.

NOTE

Review [planned Azure Video Indexer website authentication changes](#).

Upload a video using the Azure Video Indexer website

Supported browsers

For more information, see [supported browsers](#).

Supported file formats for Azure Video Indexer

See the [input container/file formats](#) article for a list of file formats that you can use with Azure Video Indexer.

Upload a video

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

NOTE

The name of the video must be no greater than 80 characters.

The screenshot shows the Azure Video Indexer web interface. On the left is a sidebar with options: Media files (selected), Model customizations, and Account settings. The main area has a title 'Extract insights and enhance your content' and a search bar with placeholder 'Search for any text, person, insight, or object in your videos'. Below the search bar are 'Filters' and a green 'Upload' button. At the bottom of the main area are tabs for Library (selected), Projects, and Samples, along with a 'Upload' button.

- Once your video has been uploaded, Azure Video Indexer starts indexing and analyzing the video. As a result a JSON output with insights is produced.

You see the progress.

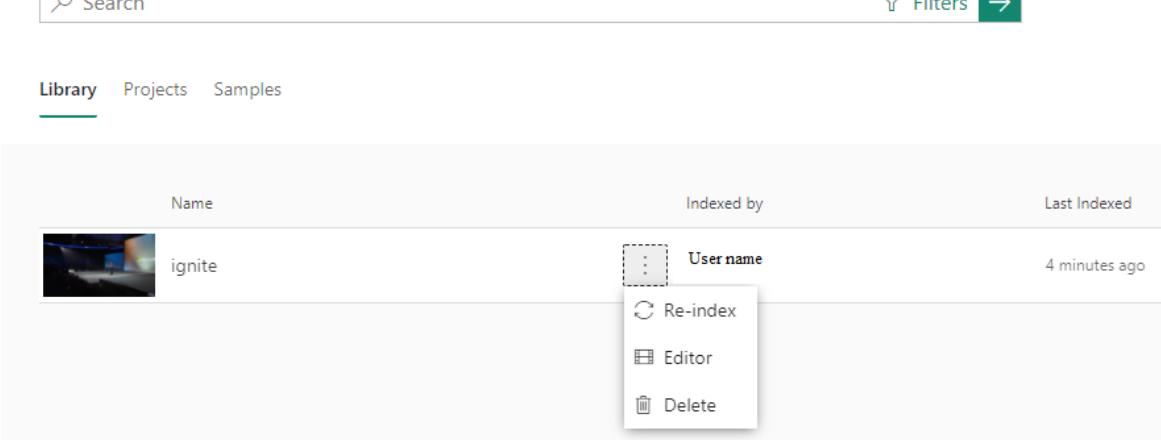
The screenshot shows a progress dialog for uploading a media file. It displays '15%' completion for a file named 'ignite'. Below the progress bar are fields for 'Video name' (set to 'ignite'), 'Video source language' (set to 'English'), and 'Privacy' (set to 'Private'). There is also an 'Advanced options' dropdown. At the bottom are 'Run in background' and 'Cancel' buttons.

The produced JSON output contains `Insights` and `summarizedInsights` elements. We highly recommend using `Insights` and not using `SummarizedInsights` (which is present for backward compatibility).

- Once Azure Video Indexer is done analyzing, you'll get an email with a link to your video and a short description of what was found in your video. For example: people, spoken and written words, topics, and named entities.
- You can later find your video in the library list and perform different operations. For example: search, reindex, edit.

Extract insights and enhance your content

Search for any text, person, insight, or object in your videos



The screenshot shows the Azure Video Indexer library interface. At the top, there is a search bar with a magnifying glass icon and the word "Search", followed by a "Filters" button and a green "→" button. Below the search bar, there are three tabs: "Library" (which is selected and underlined), "Projects", and "Samples". The main area displays a table with columns: "Name", "Indexed by", and "Last Indexed". A single row is visible for a video named "ignite", which was indexed by "User name" 4 minutes ago. To the right of the "ignite" row, a context menu is open, showing options: "User name" (selected), "Re-index", "Editor", and "Delete".

After you upload and index a video, you can continue using [Azure Video Indexer website](#) or [Azure Video Indexer Developer Portal](#) to see the insights of the video (see [Examine the Azure Video Indexer output](#)).

For more details, see [Upload and index videos](#).

To start using the APIs, see [use APIs](#)

Next steps

For detailed introduction please visit our [introduction lab](#).

At the end of the workshop, you'll have a good understanding of the kind of information that can be extracted from video and audio content, you'll be more prepared to identify opportunities related to content intelligence, pitch video AI on Azure, and demo several scenarios on Azure Video Indexer.

Quickstart: Invite users to Azure Video Indexer

9/22/2022 • 2 minutes to read • [Edit Online](#)

To collaborate with your colleagues, you can invite them to your Azure Video Indexer account.

NOTE

Only the account's admin can add or remove users.

When using paid accounts, you are only able to invite Azure AD users.

Invite new users

1. Sign in on the [Azure Video Indexer](#) website. Make sure you are connected with an admin account.
2. If you are the admin, you see the **Share account** button in the top-right corner. Click on the button and you can invite users.

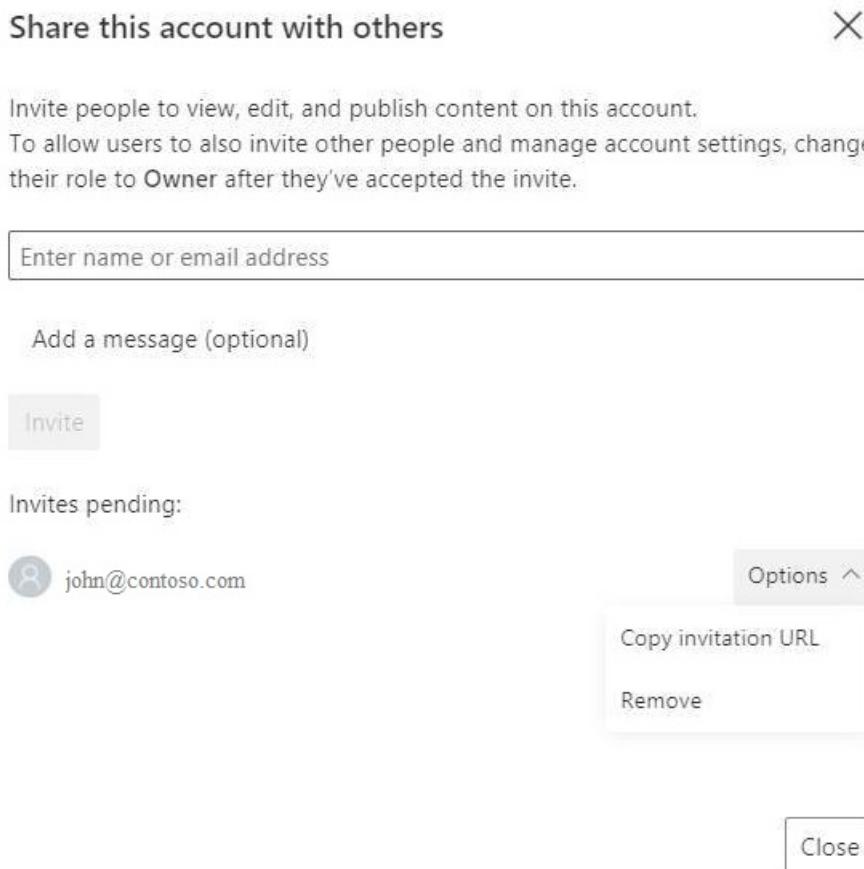
The screenshot shows the Azure Video Indexer dashboard. On the left, there is a sidebar with options: Media files, Model customizations, and Account settings. The main area has a title 'Extract insights and enhance your content' and a search bar. Below the search bar is a table with columns 'Name' and 'Indexed by'. A single row is visible, showing 'ignite' and 'User name'. In the top right corner, there is a 'Video Indexer accounts' section with a list of accounts. One account, 'viaccount eastus', is shown with its GUID. Next to the account name is a 'Share account' button, which is highlighted with a red box. Other buttons in this section include 'Settings' and '+ Create unlimited account'.

3. In the **Share this account with others** dialog, enter an email addresses of a person you want to invite to your Azure Video Indexer account:

The screenshot shows the 'Share this account with others' dialog. At the top, it says 'Share this account with others' and has a close button. Below that, there is a note: 'Invite people to view, edit, and publish content on this account. To allow users to also invite other people and manage account settings, change their role to Owner after they've accepted the invite.' There is a large input field labeled 'Enter name or email address' where 'ignite' is typed. Below the input field is a message input field labeled 'Add a message (optional)'. At the bottom left is a 'Invite' button, and at the bottom right is a 'Close' button.

4. After you press **Invite**, the person will be added to the list of pending invites.

You can choose from two options for each invitee who didn't yet join the account: **remove invitation** or **copy invitation URL**.



5. Once the invitee joins the account, you will see three options to choose from. Two options for roles: **contributor** (default) or **owner**. Or, you can choose to remove the invitee by pressing **Remove**.

Share this account with others

X

Invite people to view, edit, and publish content on this account.
To allow users to also invite other people and manage account settings, change their role to Owner after they've accepted the invite.

Enter name or email address

Add a message (optional)

Invite

Account shared with 1 person:



John Doe

Contributor ^

Owner

Contributor

Remove

Close

Users do not receive a notification upon removal. Once removed, users will not be authorized to log in.

Manage roles, invite more users

In addition to bringing up the **Share this account with others** dialog by clicking on **Share account** (as described above), you can do it from **Settings**.

1. Press the **Settings** button in the open account.

The screenshot shows the Video Indexer interface. On the left, there's a sidebar with icons for Library (highlighted with a red box), Projects, and Samples. The main area has a heading "Extract insights and enhance your content" and a search bar labeled "Search". To the right, a modal window titled "Video Indexer accounts" is open. It shows a single account entry: "viaccount Trial" with a GUID below it. There are buttons for "Share account" and "Settings" (which is highlighted with a red box). At the bottom of the modal is a button for "+ Create unlimited account".

2. Click the **Manage roles** button.
3. To invite another user, click **Invite more people to this account**.

Share this account with others



Invite people to view, edit, and publish content on this account.
To allow users to also invite other people and manage account settings, change their role to Owner after they've accepted the invite.

Invite more people to this account

Account shared with 1 person:



John Doe

Contributor ▾

Close

- Once you press **invite more people to this account**, invite dialog appears

Share this account with others



Invite people to view, edit, and publish content on this account.
To allow users to also invite other people and manage account settings, change their role to Owner after they've accepted the invite.

Enter name or email address

Add a message (optional)

Invite

Close

Next steps

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

See also

- [Azure Video Indexer overview](#)
- [How to sign up and upload your first video](#)
- [Start using APIs](#)

Tutorial: create an ARM-based account with Azure portal

9/22/2022 • 4 minutes to read • [Edit Online](#)

IMPORTANT

Face identification, customization and celebrity recognition features access is limited based on eligibility and usage criteria in order to support our Responsible AI principles. Face identification, customization and celebrity recognition features are only available to Microsoft managed customers and partners. Use the [Face Recognition intake form](#) to apply for access.

To start using unlimited features and robust capabilities of Azure Video Indexer, you need to create an Azure Video Indexer unlimited account.

This tutorial walks you through the steps of creating the Azure Video Indexer account and its accompanying resources by using the Azure portal. The account that gets created is ARM (Azure Resource Manager) account. For information about different account types, see [Overview of account types](#).

Prerequisites

- You should be a member of your Azure subscription with either an **Owner** role, or both **Contributor** and **User Access Administrator** roles. You can be added twice, with two roles, once with **Contributor** and once with **User Access Administrator**. For more information, see [View the access a user has to Azure resources](#).
- 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, select **Register**. It takes a couple of minutes to register.

- Have an **Owner** role (or **Contributor** and **User Access Administrator** roles) assignment on the associated Azure Media Services (AMS). You select the AMS account during the Azure Video Indexer account creation, as described below.
- Have an **Owner** role (or **Contributor** and **User Access Administrator** roles) assignment on the related managed identity.

Use the Azure portal to create an Azure Video Indexer account

1. Sign into the [Azure portal](#).

Alternatively, you can start creating the **unlimited** account from the [videoindexer.ai](#) website.

2. Using the search bar at the top, enter "Video Indexer".
3. Select **Video Indexer** under **Services**.
4. Select **Create**.
5. In the Create an Azure Video Indexer resource section, enter required values (the descriptions follow after the image).

Create a Video Indexer resource

Basics Tags Review + create

Create an Azure Video Indexer resource and get business insights into your video and audio files in a shared timeline and readable format. Easily extract your insights, no heavy lifting integration or machine learning knowledge needed.

[Learn more](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Resource group * ⓘ

[Create new](#)

Resource name *

Region * ⓘ

 East US

Existing content

Connect all content from an existing classic account

 All the videos, files, and data associated with the classic account will be referred to this new account.

Instance details

Media Services account name ⓘ

[Create a Media Services resource](#)

Managed identities

The managed identity will be used by the Video Indexer account to access the Media Services account. Select at least one type of managed identity.

System-assigned

User-assigned

I have all the rights to use the content/file, and agree that it will be handled per the [Online Services Terms](#) and the [Microsoft Privacy Statement](#).*

[Review + create](#)

[Previous](#)

[Next: Tags >](#)

Here are the definitions:

NAME	DESCRIPTION
Subscription	Choose the subscription to use. If you're a member of only one subscription, you'll see that name. If there are multiple choices, choose a subscription in which your user has the required role.
Resource group	Select an existing resource group or create a new one. A resource group is a collection of resources that share lifecycle, permissions, and policies. Learn more here .
Resource name	This will be the name of the new Azure Video Indexer account. The name can contain letters, numbers and dashes with no spaces.
Region	Select the Azure region that will be used to deploy the Azure Video Indexer account. The region matches the resource group region you chose. If you'd like to change the selected region, change the selected resource group or create a new one in the preferred region. Azure region in which Azure Video Indexer is available

NAME	DESCRIPTION
Existing content	If you have existing classic Video Indexer accounts, you can choose to have the videos, files, and data associated with an existing classic account connected to the new account. See the following article to learn more Connect the classic account to ARM
Available classic accounts	Classic accounts available in the chosen subscription, resource group, and region.
Media Services account name	Select a Media Services that the new Azure Video Indexer account will use to process the videos. You can select an existing Media Services or you can create a new one. The Media Services must be in the same region you selected for your Azure Video Indexer account.
Storage account (appears when creating a new AMS account)	Choose or create a new storage account in the same resource group.
Managed identity	Select an existing user-assigned managed identity or system-assigned managed identity or both when creating the account. The new Azure Video Indexer account will use the selected managed identity to access the Media Services associated with the account. If both user-assigned and system assigned managed identities will be selected during the account creation the default managed identity is the user-assigned managed identity. A contributor role should be assigned on the Media Services.

6. Select **Review + create** at the bottom of the form.

Review deployed resource

You can use the Azure portal to validate the Azure Video Indexer account and other resources that were created. After the deployment is finished, select **Go to resource** to see your new Azure Video Indexer account.

The Overview tab of the account

This tab enables you to view details about your account.

The screenshot shows the Azure Video Indexer portal. On the left, there's a navigation sidebar with sections like Overview, Activity log, Access control (IAM), Tags, Settings, Properties, Locks, Management (Management API, Identity), Monitoring (Diagnostic settings, Logs), Automation (Tasks (preview), Export template), and Support + troubleshooting (New Support Request). The main area is titled 'Essentials' and contains fields for Resource group, Location, Subscription, Subscription ID, Status, and Tags. It also has a 'Properties' section with an 'Account ID' field and a 'Management API' tab. Below that is a 'Monitoring' section with 'Diagnostic settings' and 'Logs' tabs. At the bottom, there's a note about people models and a link to the Limited Access Review form.

Select [Explore Azure Video Indexer's portal](#) to view your new account on the [Azure Video Indexer website](#).

Essential details

NAME	DESCRIPTION
Status	When the resource is connected properly, the status is Active . When there's a problem with the connection between the managed identity and the Media Service instance, the status will be <i>Connection to Azure Media Services failed</i> . Contributor role assignment on the Media Services should be added to the proper managed identity.
Managed identity	The name of the default managed identity, user-assigned or system-assigned. The default managed identity can be updated using the Change button.

The Management tab of the account

This tab contains sections for:

- getting an access token for the account
- managing identities

Management API

Use the **Management API** tab to manually generate access tokens for the account. This token can be used to authenticate API calls for this account. Each token is valid for one hour.

To get the access token

Choose the following:

- Permission type: **Contributor** or **Reader**
- Scope: **Account**, **Project** or **Video**
 - For **Project** or **Video** you should also insert the matching ID.
- Select **Generate**

Identity

Use the **Identity** tab to manually update the managed identities associated with the Azure Video Indexer resource.

Add new managed identities, switch the default managed identity between user-assigned and system-assigned or set a new user-assigned managed identity.

Next steps

Learn how to [Upload a video using C#](#).

Tutorial: deploy Azure Video Indexer by using Bicep

9/22/2022 • 2 minutes to read • [Edit Online](#)

In this tutorial, you create an Azure Video Indexer account by using [Bicep](#).

NOTE

This sample is *not* for connecting an existing Azure Video Indexer classic account to an ARM-based Azure Video Indexer account. For full documentation on Azure Video Indexer API, visit the [Developer portal](#) page. For the latest API version for Microsoft.VideoIndexer, see the [template reference](#).

Prerequisites

- An Azure Media Services (AMS) account. You can create one for free through the [Create AMS Account](#).

Review the Bicep file

The Bicep file used in this tutorial is:

```
param location string = resourceGroup().location

@description('The name of the AVAM resource')
param accountName string

@description('The managed identity Resource Id used to grant access to the Azure Media Service (AMS) account')
param managedIdentityResourceId string

@description('The media Service Account Id. The Account needs to be created prior to the creation of this template')
param mediaServiceAccountResourceId string

@description('The AVAM Template')
resource avamAccount 'Microsoft.VideoIndexer/accounts@2022-08-01' = {
    name: accountName
    location: location
    identity:{
        type: 'UserAssigned'
        userAssignedIdentities : {
            '${managedIdentityResourceId}' : {}
        }
    }
    properties: {
        mediaServices: {
            resourceId: mediaServiceAccountResourceId
            userAssignedIdentity: managedIdentityResourceId
        }
    }
}
```

One Azure resource is defined in the bicep file:

- [Microsoft.videoIndexer/accounts](#)

Check [Azure Quickstart Templates](#) for more updated Bicep samples.

Deploy the sample

1. Save the Bicep file as main.bicep to your local computer.
2. Deploy the Bicep file using either Azure CLI or Azure PowerShell

- [CLI](#)
- [PowerShell](#)

```
az group create --name exampleRG --location eastus
az deployment group create --resource-group exampleRG --template-file main.bicep --parameters
accountName=<account-name> managedIdentityResourceId=<managed-identity>
mediaServiceAccountResourceId=<media-service-account-resource-id>
```

The location must be the same location as the existing Azure media service. You need to provide values for the parameters:

- Replace <account-name> with the name of the new Azure video indexer account.
- Replace <managed-identity> with the managed identity used to grant access between Azure Media Services(AMS).
- Replace <media-service-account-resource-id> with the existing Azure media service.

Reference documentation

If you're new to Azure Video Indexer, see:

- [Azure Video Indexer Documentation](#)
- [Azure Video Indexer Developer Portal](#)
- After completing this tutorial, head to other Azure Video Indexer samples, described on [README.md](#)

If you're new to Bicep deployment, see:

- [Azure Resource Manager documentation](#)
- [Deploy Resources with Bicep and Azure PowerShell](#)
- [Deploy Resources with Bicep and Azure CLI](#)

Next steps

[Connect an existing classic paid Azure Video Indexer account to ARM-based account](#)

Tutorial: Deploy Azure Video Indexer by using an ARM template

9/22/2022 • 2 minutes to read • [Edit Online](#)

IMPORTANT

Face identification, customization and celebrity recognition features access is limited based on eligibility and usage criteria in order to support our Responsible AI principles. Face identification, customization and celebrity recognition features are only available to Microsoft managed customers and partners. Use the [Face Recognition intake form](#) to apply for access.

In this tutorial, you'll create an Azure Video Indexer account by using the Azure Resource Manager template (ARM template, which is in preview). The resource will be deployed to your subscription and will create the Azure Video Indexer resource based on parameters defined in the *avam.template* file.

NOTE

This sample is *not* for connecting an existing Azure Video Indexer classic account to a Resource Manager-based Azure Video Indexer account.

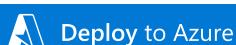
For full documentation on the Azure Video Indexer API, visit the [developer portal](#). For the latest API version for *Microsoft.VideoIndexer*, see the [template reference](#).

Prerequisites

You need an Azure Media Services account. You can create one for free through [Create a Media Services account](#).

Deploy the sample

Option 1: Select the button for deploying to Azure, and fill in the missing parameters



Option 2: Deploy by using a PowerShell script

1. Open the [template file](#) and inspect its contents.
2. Fill in the required parameters.
3. Run the following PowerShell commands:

- Create a new resource group on the same location as your Azure Video Indexer account by using the [New-AzResourceGroup](#) cmdlet.

```
New-AzResourceGroup -Name myResourceGroup -Location eastus
```

- Deploy the template to the resource group by using the [New-AzResourceGroupDeployment](#) cmdlet.

```
New-AzResourceGroupDeployment -ResourceGroupName myResourceGroup -TemplateFile  
./avam.template.json
```

NOTE

If you want to work with Bicep format, see [Deploy by using Bicep](#).

Parameters

name

- Type: string
- Description: The name of the new Azure Video Indexer account.
- Required: true

location

- Type: string
- Description: The Azure location where the Azure Video Indexer account should be created.
- Required: false

NOTE

You need to deploy your Azure Video Indexer account in the same location (region) as the associated Azure Media Services resource.

mediaServiceAccountId

- Type: string
- Description: The resource ID of the Azure Media Services resource.
- Required: true

managedIdentityId

NOTE

User assigned managed Identity must have at least Contributor role on the Media Service before deployment, when using System Assigned Managed Identity the Contributor role should be assigned after deployment.

- Type: string
- Description: The resource ID of the managed identity that's used to grant access between Azure Media Services resource and the Azure Video Indexer account.
- Required: true

tags

- Type: object
- Description: The array of objects that represents custom user tags on the Azure Video Indexer account.
- Required: false

Reference documentation

If you're new to Azure Video Indexer, see:

- [Azure Video Indexer documentation](#)
- [Azure Video Indexer developer portal](#)

After you complete this tutorial, head to other Azure Video Indexer samples described in [README.md](#).

If you're new to template deployment, see:

- [Azure Resource Manager documentation](#)
- [Deploy resources with ARM templates](#)
- [Deploy resources with Bicep and the Azure CLI](#)

Next steps

Connect a [classic paid Azure Video Indexer account](#) to a Resource Manager-based account.

Tutorial: Use the Azure Video Indexer API

9/22/2022 • 7 minutes to read • [Edit Online](#)

Azure Video Indexer consolidates various audio and video artificial intelligence (AI) technologies offered by Microsoft into 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 platforms. You can use the API to upload your files, get detailed video insights, get URLs of embeddable insight and player widgets, and more.

When you visit the [Azure Video Indexer](#) website for the first time, a trial account is automatically created for you. With the trial account, you get some free indexing minutes. You can later add a paid account. With the paid option, you pay for indexed minutes. For details about available accounts (trial and paid options), see [Azure Video Indexer account types](#).

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

Prerequisite

Before you start, see the [Recommendations](#) section (that follows later in this article).

Subscribe to the API

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

IMPORTANT

- You must use the same provider you used when you signed up for Azure 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 email. If a user tries to sign in with user@gmail.com for LinkedIn and later with user@gmail.com for Google, the latter will display an error page, saying the user already exists.

Choose an account to continue:

 Work account (AAD)

 Personal Microsoft account

 Google

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

[I was using LinkedIn or Facebook—where did it go?](#)

2. Subscribe.

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

NOTE

New users are automatically subscribed to Authorization.

After you subscribe, you can find your subscription under [Products](#) -> **Profile**. In the subscriptions section, you'll find the primary and secondary keys. The keys should be protected. The keys should only be used by your server code. They shouldn't be available on the client side (js, .html, and so on).

Your subscriptions

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

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

Obtain access token using the Authorization API

Once you subscribe to the Authorization API, you can 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, and so on.
- Video level: Video level access tokens let you perform operations on a specific **video**. For example, get video insights, download captions, get widgets, and so on.

You can control the permission level of tokens in two ways:

- For **Account** tokens, you can use the [Get Account Access Token With Permission API](#) and specify the permission type (**Reader/Contributor/MyAccessManager/Owner**).
- For all types of tokens (including **Account** tokens), you can specify `allowEdit=true/false`. `false` is the equivalent of a **Reader** permission (read-only) and `true` is the equivalent of a **Contributor** permission (read-write).

For most server-to-server scenarios, you'll probably use the same **account** token since it covers both **account** operations and **video** operations. However, if you're planning to make client side calls to Azure 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's also the reason that when embedding Azure Video Indexer 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 it expires, call the Authorization API again to get a new access token.

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

Operational API calls

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

The screenshot shows the 'Settings' page for an account named 'account-name'. It displays the account name, email (john@contoso.com), account ID (00000000-0000-0000-0000-000000000000), upload limit (175.12/2400min), and a delete link.

- Use [Azure Video Indexer Developer Portal](#) to programmatically get the Account ID.

Use the [Get account](#) 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-f77dacb0a368/videos/d45bf160b5/
```

Recommendations

This section lists some recommendations when using Azure Video Indexer API.

Uploading

- If you're planning to upload a video, it's recommended to place the file in some public network location (for example, an Azure Blob Storage account). Get the link to the video and provide the URL as the upload file param.

The URL provided to Azure Video Indexer must point to a media (audio or video) file. An easy verification for the URL (or SAS URL) is to paste it into a browser, if the file starts playing/downloading, it's likely a

good URL. If the browser is rendering some visualization, it's likely not a link to a file but to an HTML page. When you're uploading videos by using the API, you have the following options:

- Upload your video from a URL (preferred).
- Send the video file as a byte array in the request body.
- Use existing an Azure Media Services asset by providing the [asset ID](#). This option is supported in paid accounts only.

Getting JSON output

- 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 article](#).
- The JSON output produced by the API contains `Insights` and `SummarizedInsights` elements. We highly recommend using `Insights` and not using `SummarizedInsights` (which is present for backward compatibility).
- We don't recommend that you use data directly from the artifacts folder for production purposes. Artifacts are intermediate outputs of the indexing process. They're essentially raw outputs of the various AI engines that analyze the videos; the artifacts schema may change over time.

It's recommended that you use the [Get Video Index](#) API, as described in [Get insights and artifacts produced by the API](#) and not [Get-Video-Artifact-Download-Url](#).

Code sample

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

NOTE

The following sample is intended for classic accounts only and not compatible with ARM-based accounts. For an updated sample for ARM (recommended), see [this ARM sample repo](#).

```
var apiUrl = "https://api.videoindexer.ai";
var accountId = "...";
var location = "westus2"; // replace with the account's location, or with "trial" if this is a trial account
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 = new byte[video.Length];
//video.Read(buffer, 0, buffer.Length);
//content.Add(new ByteArrayContent(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);
```

Clean up resources

After you're done with this tutorial, delete resources that you aren't planning to use.

See also

- [Azure Video Indexer overview](#)
- [Regions](#)

Next steps

- [Examine details of the output JSON](#)
- Check out the [sample code](#) that demonstrates important aspect of uploading and indexing a video. Following the code will give you a good idea of how to use our API for basic functionalities. Make sure to read the inline comments and notice our best practices advice.

Azure Video Indexer terminology & concepts

9/22/2022 • 2 minutes to read • [Edit Online](#)

This article gives a brief overview of Azure Video Indexer terminology and concepts.

Confidence scores

The confidence score indicates the confidence in an insight. It is a number between 0.0 and 1.0. The higher the score the greater the confidence in the answer. For example:

```
"transcript": [
  {
    "id":1,
    "text":"Well, good morning everyone and welcome to",
    "confidence":0.8839,
    "speakerId":1,
    "language":"en-US",
    "instances":[
      {
        "adjustedStart":"0:00:10.21",
        "adjustedEnd":"0:00:12.81",
        "start":"0:00:10.21",
        "end":"0:00:12.81"
      }
    ]
  },
]
```

Content moderation

Use textual and visual content moderation models to keep your users safe from inappropriate content and validate that the content you publish matches your organization's values. You can automatically block certain videos or alert your users about the content. For more information, see [Insights: visual and textual content moderation](#).

Insights

Insights contain an aggregated view of the data: faces, topics, emotions. Azure Video Indexer analyzes the video and audio content by running 30+ AI models, generating rich insights. For more information about available models, see [overview](#).

TIP

The JSON output produced by the website or API contains `Insights` and `SummarizedInsights` elements. We highly recommend using `Insights` and not using `SummarizedInsights` (which is present for backward compatibility).

The [Azure Video Indexer](#) website enables you to use your video's 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 Azure Video Indexer and be used in your own editing applications or downstream workflows. For more information, see [Use editor to create projects](#).

Keyframes

Azure Video Indexer selects the frame(s) that best represent each shot. Keyframes are the representative frames selected from the entire video based on aesthetic properties (for example, contrast and stakeness). For more information, see [Scenes, shots, and keyframes](#).

Time range vs. adjusted time range

Time range is the time period in the original video. Adjusted time range 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 adjusted time range is 00:00-00:15.

Widgets

Azure Video Indexer supports embedding widgets in your apps. For more information, see [Embed Azure Video Indexer widgets in your apps](#).

Next steps

- [overview](#)
- [Insights](#)

Compare Azure Media Services v3 presets and Azure Video Indexer

9/22/2022 • 2 minutes to read • [Edit Online](#)

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

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

Compare

FEATURE	AZURE 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
Pricing	AVI pricing	Media Services pricing
Compliance	For the most current compliance updates, visit Azure Compliance Offerings.pdf and search for "Azure Video Indexer" to see if it complies with a certificate of interest.	For the most current compliance updates, visit Azure Compliance Offerings.pdf and search for "Media Services" to see if it complies with a certificate of interest.
Trial	East US	Not available
Region availability	See Cognitive Services availability by region	See Media Services availability by region .

Next steps

[Azure Video Indexer overview](#)

[Media Services v3 overview](#)

Manage multiple tenants

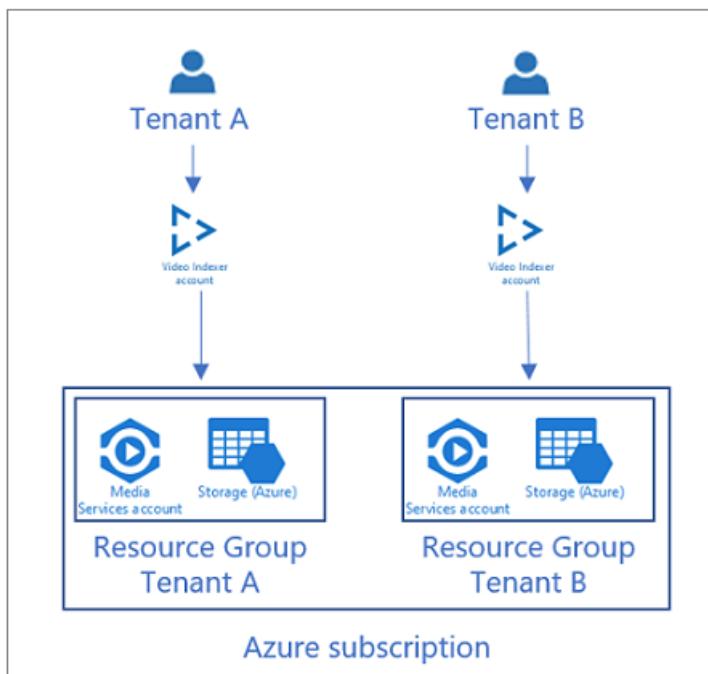
9/22/2022 • 2 minutes to read • [Edit Online](#)

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

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

Azure Video Indexer account per tenant

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



Considerations

- Customers don't share storage accounts (unless manually configured by the customer).
- Customers don't 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 Azure Video Indexer account.
- There's no ability to share custom models between tenants.

Make sure there's no business requirement to share custom models.

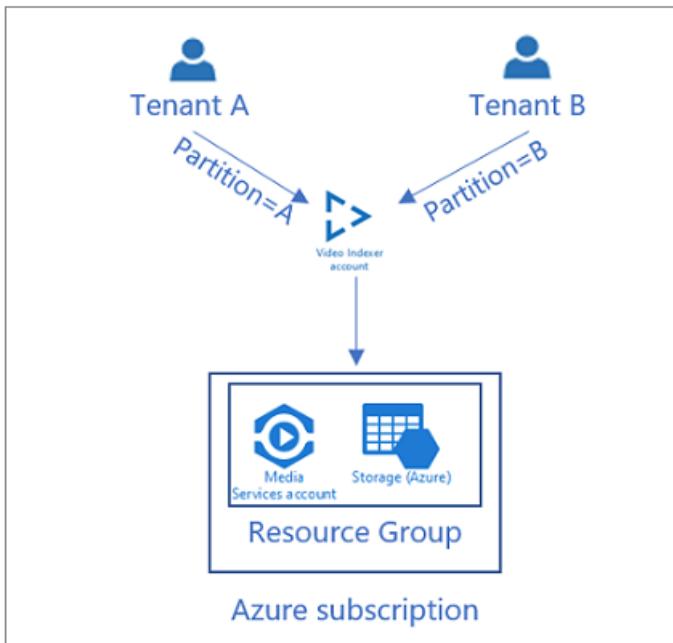
- Harder to manage due to multiple Azure Video Indexer (and associated Media Services) accounts per tenant.

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 Azure Video Indexer account for all users

When using this architecture, the customer is responsible for tenants isolation. All tenants have to use a single Azure 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'll only get results of the specified partition.

Considerations

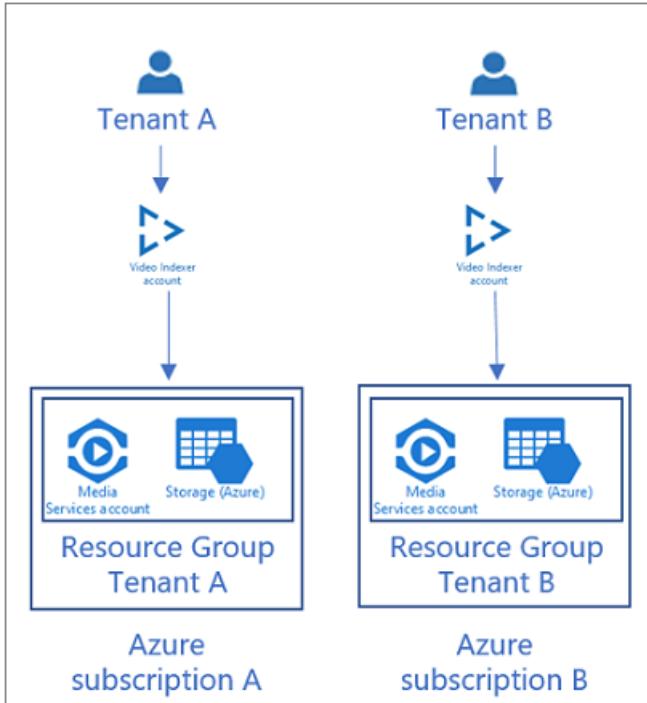
- Ability to share content and customization models between tenants.
- One tenant impacts the performance of other tenants.
- Customer needs to build a complex management layer on top of Azure Video Indexer.

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'll create a new Azure Video Indexer account in the tenant subscription.



Considerations

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

Next steps

[Overview](#)

Things to consider when using Azure Video Indexer at scale

9/22/2022 • 4 minutes to read • [Edit Online](#)

When using Azure Video Indexer to index videos and your archive of videos is growing, consider scaling.

This article answers questions like:

- Are there any technological constraints I need to take into account?
- Is there a smart and efficient way of doing it?
- Can I prevent spending excess money in the process?

The article provides six best practices of how to use Azure Video Indexer at scale.

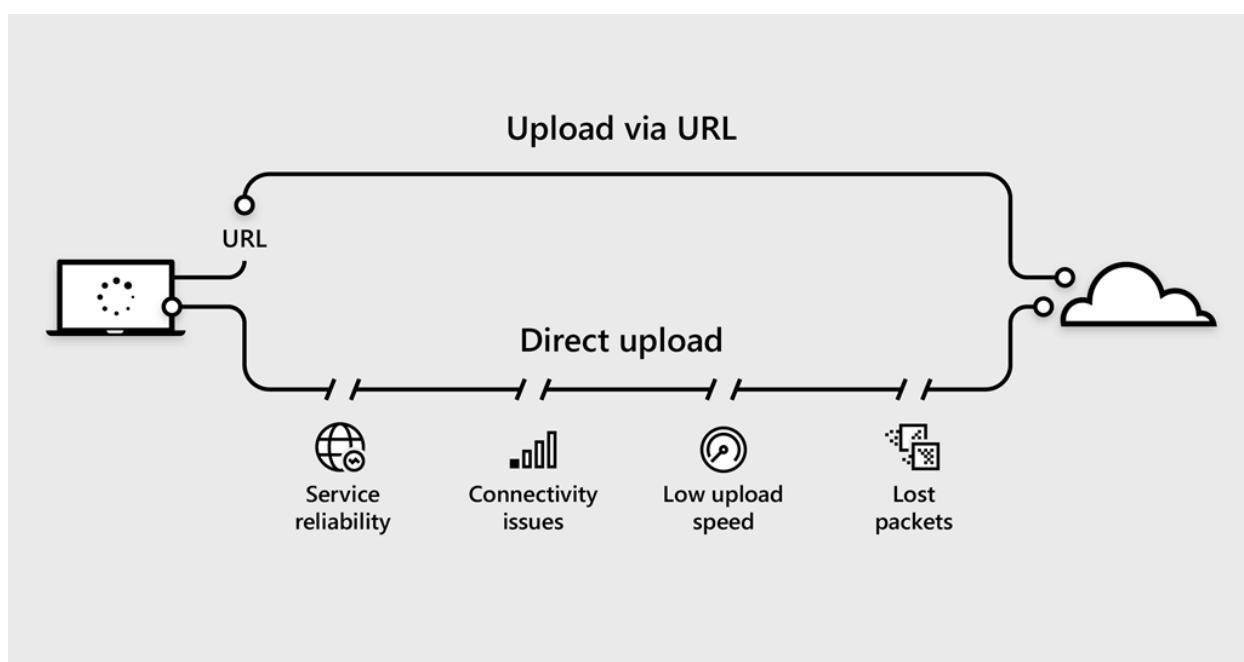
When uploading videos consider using a URL over byte array

Azure Video Indexer does give you the choice to upload videos from URL or directly by sending the file as a byte array, the latter comes with some constraints. For more information, see [uploading considerations and limitations](#))

First, it has file size limitations. The size of the byte array file is limited to 2 GB compared to the 30-GB upload size limitation while using URL.

Second, consider just some of the issues that can affect your performance and hence your ability to scale:

- Sending files using multi-part means high dependency on your network,
- service reliability,
- connectivity,
- upload speed,
- lost packets somewhere in the world wide web.



When you upload videos using URL, you just need to provide a path to the location of a media file and Video Indexer takes care of the rest (see the `videoUrl` field in the [upload video API](#)).

TIP

Use the `videourl` optional parameter of the upload video API.

To see an example of how to upload videos using URL, check out [this example](#). Or, you can use [AzCopy](#) for a fast and reliable way to get your content to a storage account from which you can submit it to Azure Video Indexer using [SAS URL](#). Azure Video Indexer recommends using *readonly* SAS URLs.

Automatic Scaling of Media Reserved Units

Starting August 1st 2021, Azure Video Indexer enabled [Reserved Units](#)(MRUs) auto scaling by [Azure Media Services](#) (AMS), as a result you do not need to manage them through Azure Video Indexer. That will allow price optimization, e.g. price reduction in many cases, based on your business needs as it is being auto scaled.

Respect throttling

Azure Video Indexer is built to deal with indexing at scale, and when you want to get the most out of it you should also be aware of the system's capabilities and design your integration accordingly. You don't want to send an upload request for a batch of videos just to discover that some of the movies didn't upload and you are receiving an HTTP 429 response code (too many requests). It can happen due to the fact that you sent more requests than the [limit of movies per minute we support](#). Azure Video Indexer adds a `retry-after` header in the HTTP response, the header specifies when you should attempt your next retry. Make sure you respect it before trying your next request.

429 Too many requests

Too many requests were sent, use Retry-After response header to decide when to send the next request

Response headers

<code>Retry-After</code> (optional)	integer	A non-negative decimal integer indicating the seconds to delay after the response is received
<code>x-ms-request-id</code> (optional)	string	A globally unique identifier (GUID) for the request which is assigned by the server for instrumentation purposes. The server makes sure all logs associated with handling the request can be linked to the server request id so a client can provide this request id in support tickets so support engineers could find the logs linked to this particular request. The server makes sure this request id will never repeat itself.

Use callback URL

We recommend that instead of polling the status of your request constantly from the second you sent the upload request, you can add a [callback URL](#), and wait for Azure Video Indexer to update you. As soon as there is any status change in your upload request, you get a POST notification to the URL you specified.

You can add a callback URL as one of the parameters of the [upload video API](#). Check out the code samples in [GitHub repo](#).

For callback URL you can also use Azure Functions, a serverless event-driven platform that can be triggered by HTTP and implement a following flow.

callBack URL definition

A callback URL is used to notify the customer (through 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

Use the right indexing parameters for you

When making decisions related to using Azure Video Indexer at scale, look at how to get the most out of it with the right parameters for your needs. Think about your use case, by defining different parameters you can save money and make the indexing process for your videos faster.

Before uploading and indexing your video read this short [documentation](#), check the [indexingPreset](#) and [streamingPreset](#) to get a better idea of what your options are.

For example, don't set the preset to streaming if you don't plan to watch the video, don't index video insights if you only need audio insights.

Index in optimal resolution, not highest resolution

You might be asking, what video quality do you need for indexing your videos?

In many cases, indexing performance has almost no difference between HD (720P) videos and 4K videos. Eventually, you'll get almost the same insights with the same confidence. The higher the quality of the movie you upload means the higher the file size, and this leads to higher computing power and time needed to upload the video.

For example, for the face detection feature, a higher resolution can help with the scenario where there are many small but contextually important faces. However, this will come with a quadratic increase in runtime and an increased risk of false positives.

Therefore, we recommend you to verify that you get the right results for your use case and to first test it locally. Upload the same video in 720P and in 4K and compare the insights you get.

Next steps

Examine the Azure Video Indexer output produced by API

Customizing content models in Azure Video Indexer

9/22/2022 • 2 minutes to read • [Edit Online](#)

IMPORTANT

Face identification, customization and celebrity recognition features access is limited based on eligibility and usage criteria in order to support our Responsible AI principles. Face identification, customization and celebrity recognition features are only available to Microsoft managed customers and partners. Use the [Face Recognition intake form](#) to apply for access.

Azure 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 Azure Video Indexer website or API.

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

Animated characters

- [Animated character detection](#)

Brands model

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

Language model

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

Person model

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

Next steps

[Azure Video Indexer overview](#)

Animated character detection

9/22/2022 • 2 minutes to read • [Edit Online](#)

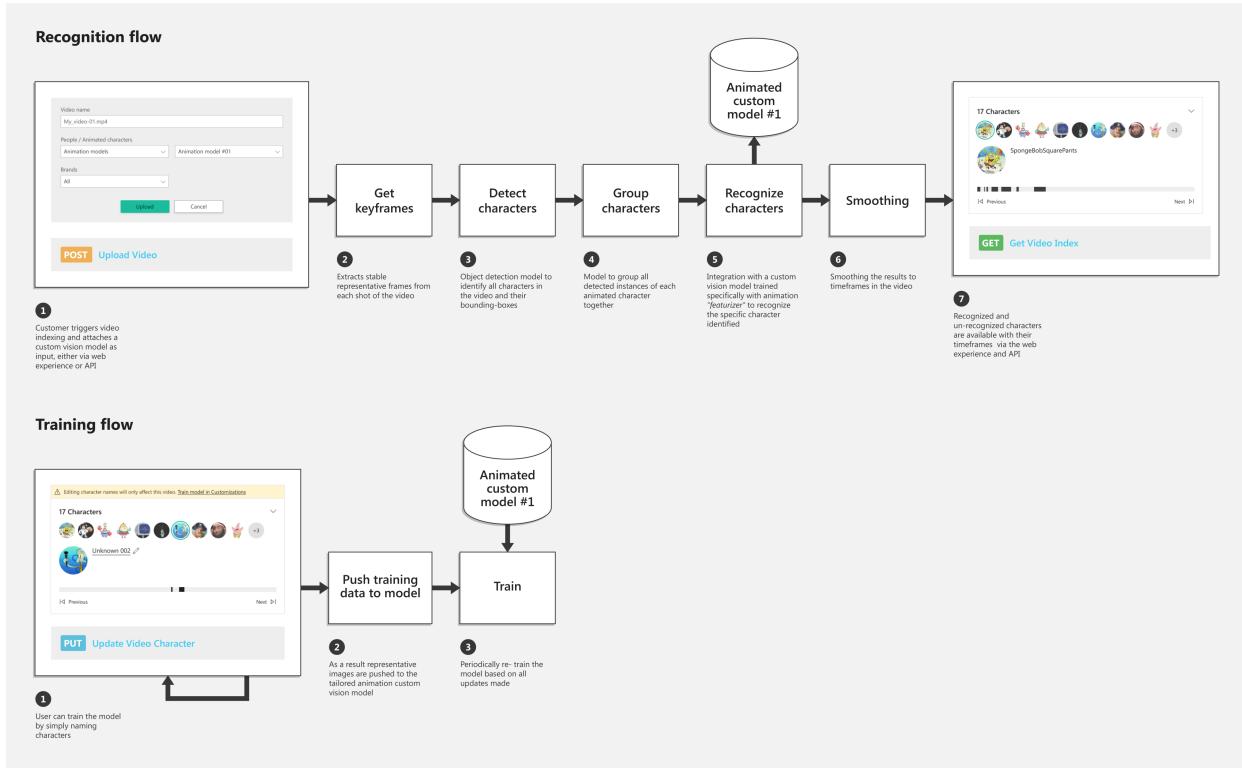
Azure Video Indexer supports detection, grouping, and recognition of characters in animated content via integration with [Cognitive Services custom vision](#). This functionality is available both through the portal and through the API.

After uploading an animated video with a specific animation model, Azure Video Indexer extracts keyframes, detects animated characters in these frames, groups similar character, and chooses the best sample. Then, it sends the grouped characters to Custom Vision that identifies characters based on the models it was trained on.

Before you start training your model, the characters are detected namelessly. As you add names and train the model the Azure Video Indexer will recognize the characters and name them accordingly.

Flow diagram

The following diagram demonstrates the flow of the animated character detection process.



Accounts

Depending on a type of your Azure Video Indexer account, different feature sets are available. For information on how to connect your account to Azure, see [Create an Azure Video Indexer account connected to Azure](#).

- The trial account: Azure Video Indexer uses an internal Custom Vision account to create model and connect it to your Azure Video Indexer account.
- The paid account: you connect your Custom Vision account to your Azure Video Indexer account (if you don't already have one, you need to create an account first).

Trial vs. paid

FUNCTIONALITY	TRIAL	PAID
Custom Vision account	Managed behind the scenes by Azure Video Indexer.	Your Custom Vision account is connected to Azure Video Indexer.
Number of animation models	One	Up to 100 models per account (Custom Vision limitation).
Training the model	Azure Video Indexer trains the model for new characters additional examples of existing characters.	The account owner trains the model when they're ready to make changes.
Advanced options in Custom Vision	No access to the Custom Vision portal.	You can adjust the models yourself in the Custom Vision portal.

Use the animated character detection with portal and API

For details, see [Use the animated character detection with portal and API](#).

Limitations

- Currently, the "animation identification" capability isn't supported in East-Asia region.
- Characters that appear to be small or far in the video may not be identified properly if the video's quality is poor.
- The recommendation is to use a model per set of animated characters (for example per an animated series).

Next steps

[Azure Video Indexer overview](#)

Customize a Brands model in Azure Video Indexer

9/22/2022 • 2 minutes to read • [Edit Online](#)

Azure 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, Azure 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. Azure 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 Azure Video Indexer account. The custom Brands model feature allows you to select whether or not Azure Video Indexer will detect brands from the Bing brands database, exclude certain brands from being detected (essentially creating a list of unapproved brands), and include brands that should be part of your model that might not be in Bing's brands database (essentially creating a list of approved 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. Azure 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, Azure 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 window displaying a man speaking on stage at a Microsoft Build event. The video progress bar shows 0:02 / 1:51:00. Below the video player are social sharing icons and a 'Report' link. To the right of the video player is a sidebar titled 'Insights' which includes a transcript search bar, a list of topic clusters, and a list of annotations. At the bottom of the sidebar is a 'Brands' section with several brand names listed in buttons.

This screenshot provides a detailed look at the 'Brands' section. It shows a list of brands: Microsoft Windows, Microsoft, Windows 10, Autodesk, iOS, Android, OneDrive, Microsoft Visual Studio, Microsoft Cortana, and Windows Store. Below this list is a specific entry for 'Microsoft Windows' with a 'Show description' link and a timeline visualization.

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.

Next steps

To bring your own brands, check out these topics:

[Customize Brands model using APIs](#)

[Customize Brands model using the website](#)

Customize a Language model with Azure Video Indexer

9/22/2022 • 2 minutes to read • [Edit Online](#)

Azure 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. See the list of supported by Azure Video Indexer languages in [supported languages](#).

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 Azure 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 Azure 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

Azure 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 Azure Video Indexer

9/22/2022 • 2 minutes to read • [Edit Online](#)

IMPORTANT

Face identification, customization and celebrity recognition features access is limited based on eligibility and usage criteria in order to support our Responsible AI principles. Face identification, customization and celebrity recognition features are only available to Microsoft managed customers and partners. Use the [Face Recognition intake form](#) to apply for access.

Azure 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 Azure Video Indexer are still detected but are left unnamed. Customers can build custom Person models and enable Azure 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, Azure Video Indexer will use the default Person model in your account.

You can use the Azure 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](#).

Limited Access features of Azure Video Indexer

9/22/2022 • 2 minutes to read • [Edit Online](#)

IMPORTANT

Face service access is limited based on eligibility and usage criteria in order to support our Responsible AI principles. Face service is only available to Microsoft managed customers and partners. Use the [Face Recognition intake form](#) to apply for access. For more information, see the [Face limited access](#) page.

Our vision is to empower developers and organizations to leverage AI to transform society in positive ways. We encourage responsible AI practices to protect the rights and safety of individuals. Microsoft facial recognition services are Limited Access in order to help prevent the misuse of the services in accordance with our [AI Principles](#) and [facial recognition](#) principles. The Face Identify and Celebrity Recognition operations in Azure Video Indexer are Limited Access features that require registration.

Since the announcement on June 11th, 2020, customers may not use, or allow use of, any Azure facial recognition service by or for a police department in the United States.

Application process

Limited Access features of Azure Video Indexer are only available to customers managed by Microsoft, and only for use cases selected at the time of registration. Other Azure Video Indexer features do not require registration to use.

Customers and partners who wish to use Limited Access features of Azure Video Indexer are required to [submit an intake form](#). Access is subject to Microsoft's sole discretion based on eligibility criteria and a vetting process. Microsoft may require customers and partners to reverify this information periodically.

The Azure Video Indexer service is made available to customers and partners under the terms governing their subscription to Microsoft Azure Services (including the [Service Specific Terms](#)). Please review these terms carefully as they contain important conditions and obligations governing your use of Azure Video Indexer.

Limited access features

This section talks about limited access features in Azure Video Indexer.

WHEN DID I CREATE THE ACCOUNT?	TRIAL ACCOUNT (FREE)	PAID ACCOUNT (CLASSIC OR ARM-BASED)
Existing VI accounts created before June 21, 2022	Able to access face identification, customization and celebrities recognition till June 2023. Recommended: Move to a paid account and afterward fill in the intake form and based on the eligibility criteria we will enable the features also after the grace period.	Able to access face identification, customization and celebrities recognition till June 2023*. Recommended: fill in the intake form and based on the eligibility criteria we will enable the features also after the grace period. We proactively sent emails to these customers + AEs.

WHEN DID I CREATE THE ACCOUNT?	TRIAL ACCOUNT (FREE)	PAID ACCOUNT (CLASSIC OR ARM-BASED)
New VI accounts created after June 21, 2022	<p>Not able the access face identification, customization and celebrities recognition as of today.</p> <p>Recommended: Move to a paid account and afterward fill in the intake form. Based on the eligibility criteria we will enable the features (after max 10 days).</p>	<p>Azure Video Indexer disables the access face identification, customization and celebrities recognition as of today by default, but gives the option to enable it.</p> <p>Recommended: Fill in the intake form and based on the eligibility criteria we will enable the features (after max 10 days).</p>

*In Brazil South we also disabled the face detection.

Help and support

FAQ about Limited Access can be found [here](#).

If you need help with Azure Video Indexer, find support [here](#).

[Report Abuse](#) of Azure Video Indexer.

Next steps

Learn more about the legal terms that apply to this service [here](#).

Manage access with the Video Indexer Restricted Viewer role

9/22/2022 • 2 minutes to read • [Edit Online](#)

Azure Video Indexer enables managing user access to the Azure Video Indexer resource at the account level with the following built-in role: **Video Indexer Restricted Viewer**.

NOTE

The **Video Indexer Restricted Viewer** built-in role is only available for ARM-based (recommended) accounts, not classic accounts. If you have an existing classic account, see [Connect an existing classic paid account to an ARM-based account](#).

Users with the owner or administrator Azure Active Directory (Azure AD) permissions can assign the **Video Indexer Restricted Viewer** role to Azure AD users or security groups for an account. For information on how to assign roles, see [Assign Azure roles using the Azure portal](#).

The limited access Video Indexer Restricted Viewer role is intended for the [Azure Video Indexer website](#) users as its permitted actions relate to the [Azure Video Indexer website](#) experience.

Azure Video Indexer Restricted Viewer permissions

Users with this role are **able** to perform the following tasks:

- View and play videos in the account
- Search through insights in the account
- Translate the transcription of a specific video

Users with this role are **unable** to perform the following tasks:

- Upload/Index/Re-index a video
- Download/Embed video/insights
- Change account settings
- Edit insights
- Create/update customized models
 - Language
 - People
 - Brands
- Assign roles
- Generate an access token

Using an ARM API

To generate a Video Indexer restricted viewer access token via API, see [documentation](#).

Restricted Viewer Video Indexer website experience

When using the [Azure Video Indexer](#) website with a Video Indexer Restricted Viewer access, disabled features are greyed out. If a user with the restricted viewer role attempts to access an unauthorized page, they'll

encounter the pop-up message below:

The screenshot shows the Azure Video Indexer interface. At the top, there's a navigation bar with icons for Home, Library, and Sample videos, along with a search bar and upload buttons. Below the navigation is a table header with columns: Name ↑, Last indexed, and Indexed by. The main content area displays a list of video entries, each with a thumbnail, name, last indexed date, and indexed by user. A prominent white modal dialog box is centered over the list, displaying the message "You don't have access to this page" and explaining that the account has view-only access. It includes a "Close" button.

Name ↑	Last indexed	Indexed by
Video name		
Video name		
Video name		
CREATORS Video name		
Video name	Mar 23	Josh Barnhill
Video name	Mar 12	Stephanie Bourne
Video name	Mar 5	Nicole Holliday
Video name	Feb 26	John Kane
Video name	Feb 18	Jean Trenary
Video name	Jan 20	John Kane
Video name	Jan 14	Stephanie Bourne
Video name	Dec 19, 2018	Rachel Valdez

Next steps

[Overview](#)

Connect an existing classic paid Azure Video Indexer account to ARM-based account

9/22/2022 • 4 minutes to read • [Edit Online](#)

This article shows how to connect an existing classic paid Azure Video Indexer account to an Azure Resource Manager (ARM)-based (recommended) account. To create a new ARM-based account, see [create a new account](#). To understand the Azure Video Indexer account types, review [account types](#).

In this article, we demonstrate options of connecting your **existing** Azure Video Indexer account to an **ARM-based** account.

Prerequisites

1. Unlimited paid Azure Video Indexer account (classic account).
 - a. To perform the connect to the ARM (Azure Resource Manager) action, you should have owner's permissions on the Azure Video Indexer classic account.
2. Azure Subscription with Owner permissions or Contributor with Administrator Role assignment.
 - a. Same level of permission for the Azure Media Service associated with the existing Azure Video Indexer Classic account.
3. User assigned managed identity (can be created along the flow).

Transition state

Connecting a classic account to be ARM-based triggers a 30 days of a transition state. In the transition state, an existing account can be accessed by generating an access token using both:

- Access token [generated through API Management](#)(classic way)
- Access token [generated through ARM](#)

The transition state moves all account management functionality to be managed by ARM and will be handled by [Azure RBAC](#).

The [invite users](#) feature in the Azure Video Indexer portal gets disabled. The invited users on this account lose their access to the Azure Video Indexer account Media in the portal.

However, this can be resolved by assigning the right role-assignment to these users through Azure RBAC, see [How to assign RBAC](#).

Only the account owner, who performed the connect action, is automatically assigned as the owner on the connected account. When [Azure policies](#) are enforced, they override the settings on the account.

If users are not added through Azure RBAC to the account after 30 days, they will lose access through API as well as Azure Video Indexer portal.

After the transition state ends, users will only be able to generate a valid access token through ARM, making Azure RBAC the exclusive way to manage role-based access control on the account.

NOTE

If there are invited users you wish to remove access from, do it before connecting the account to ARM.

Before the end of the 30 days of transition state, you can remove access from users through the Azure Video Indexer portal on the account settings page.

Get started

Browse to Azure Video Indexer portal

1. Sign in using your Azure AD account.
2. On the top right bar press *User account* to open the side pane account list.
3. Select the Azure Video Indexer classic account you wish to connect to ARM (classic accounts will be tagged with a *classic tag*).
4. Click **Settings**.

The screenshot shows the 'Accounts' section of the Azure Video Indexer portal. At the top, there's a navigation bar with a 'Create unlimited account' button and icons for help and settings. Below the bar, the word 'Accounts' is centered. A list of accounts is shown, with 'My-classic-account' and 'eastus' listed. To the right of the account name is a small circular badge with the word 'Classic' in it, which is highlighted with a red box. Below the account name are two buttons: 'Share account' and 'Settings', with 'Settings' also highlighted with a red box.

5. Click **Connect to an ARM-based account**.

The screenshot shows the 'Account settings' page for the selected account. The left sidebar has icons for Home, Accounts, and Settings, with 'Accounts' currently selected. The main area shows account details: Display name (with an 'Edit' button), Email, Account ID (with a 'Copy' button), Account region (set to 'eastus'), and Options (Manage roles, Update connection to Azure Media Services, Delete this account). Below these, a section for 'Media service used' is shown, with a note that 'Azure Media Services now manages Media Reserved Units for you.' At the bottom of the page, there are two buttons: 'Manage in Azure portal' and 'Connect to an ARM-based account', with a red arrow pointing to the 'Connect to an ARM-based account' button.

6. Sign to Azure portal.
7. The Azure Video Indexer create blade will open.
8. In the **Create Azure Video Indexer account** section enter required values.

If you followed the steps the fields should be auto-populated, make sure to validate the eligible values.

Basics Tags Review + create

Create an Azure Video Indexer resource and get business insights into your video and audio files in a shared timeline and readable format. Easily extract your insights, no heavy lifting integration or machine learning knowledge needed.
[Learn more](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * My-subscription

Resource group * Resource group * Create new

Resource name *

Region * East US

Existing content

Connect all content from an existing classic account

Info All the videos, files, and data associated with the classic account will be referred to this new account.

Existing account ID *

Instance details

Media Services account name Create a Media Services resource

Managed identities

User assigned Create a user-assigned managed identity

I have all the rights to use the content/file, and agree that it will be handled per the [Online Services Terms](#) and the [Microsoft Privacy Statement](#) *

[Review + create](#) [Previous](#) [Next: Tags >](#)

Here are the descriptions for the resource fields:

NAME	DESCRIPTION
Subscription	The subscription currently contains the classic account and other related resources such as the Media Services.
Resource Group	Select an existing resource or create a new one. The resource group must be the same location as the classic account being connected
Azure Video Indexer account (radio button)	Select the "Connecting an existing classic account".

NAME	DESCRIPTION
Existing account ID	Enter the ID of existing Azure Video Indexer classic account.
Resource name	Enter the name of the new Azure Video Indexer account. Default value would be the same name the account had as classic.
Location	The geographic region can't be changed in the connect process, the connected account must stay in the same region.
Media Services account name	The original Media Services account name that was associated with classic account.
User-assigned managed identity	Select a user-assigned managed identity, or create a new one. Azure Video Indexer account will use it to access the Media services. The user-assignment managed identity will be assigned the roles of Contributor for the Media Service account.

9. Click **Review + create** at the bottom of the form.

After connecting to ARM is complete

After successfully connecting your account to ARM, it is recommended to make sure your account management APIs are replaced with [Azure Video Indexer REST API](#). As mentioned in the beginning of this article, during the 30 days of the transition state, "[Get-access-token](#)" will be supported side by side the ARM-based "[Generate-Access token](#)". Make sure to change to the new "Generate-Access token" by updating all your solutions that use the API.

APIs to be changed:

- Get Access token for each scope: Account, Project & Video.
- Get account – the account's details.
- Get accounts – List of all account in a region.
- Create paid account – would create a classic account.

For a full description of [Azure Video Indexer REST API](#) calls and documentation, follow the link.

For code sample generating an access token through ARM see [C# code sample](#).

Next steps

Learn how to [Upload a video using C#](#).

Add contributor role to Media Services

9/22/2022 • 2 minutes to read • [Edit Online](#)

This article describes how to assign contributor role on the Media Services account.

NOTE

If you are creating your Azure Video Indexer through the Azure portal UI, the selected Managed identity will be automatically assigned with a contributor permission on the selected Media Service account.

Prerequisites

1. Azure Media Services (AMS)
2. User-assigned managed identity

NOTE

You'll need an Azure subscription where you have access to both the [Contributor](#) role and the [User Access Administrator](#) role to the Azure Media Services and the User-assigned managed identity. If you don't have the right permissions, ask your account administrator to grant you those permissions. The associated Azure Media Services must be in the same region as the Azure Video Indexer account.

Add Contributor role on the Media Services

- [Azure portal](#)

Add Contributor role to Media Services using Azure portal

1. Sign in at the [Azure portal](#).
 - Using the search bar at the top, enter **Media Services**.
 - Find and select your Media Service resource.
2. In the pane to the left, click **Access control (IAM)**.
 - Click **Add > Add role assignment**. If you don't have permissions to assign roles, the **Add role assignment** option will be disabled.
3. In the Role list, select [Contributor](#) role and click **Next**.
4. In the **Assign access to**, select *Managed identity* radio button.
 - Click **+Select members** button and **Select managed identities** pane should be pop up.
5. **Select** the following:
 - In the **Subscription**, the subscription where the managed identity is located.
 - In the **Managed identity**, select *User-assigned managed identity*.
 - In the **Select** section, search for the Managed identity you'd like to grant contributor permissions on the Media services resource.
6. Once you have found the security principal, click to select it.
7. To assign the role, click **Review + assign**

Next steps

Create a new Azure Resource Manager based account

Import your content from the trial account

9/22/2022 • 2 minutes to read • [Edit Online](#)

When creating a new ARM-based account, you have an option to import your content from the trial account into the new ARM-based account free of charge.

NOTE

Make sure to review the following considerations.

Considerations

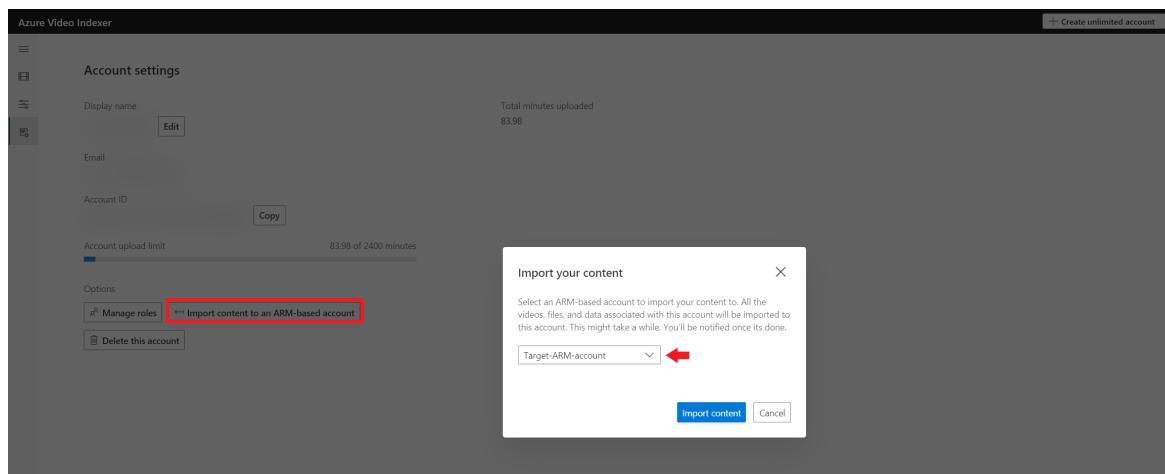
Review the following considerations.

- Import from trial can be performed only once per trial account.
- The target ARM-based account needs to be created and available before import is assigned.
- Target ARM-based account has to be an empty account (never indexed any media files).

Import your data

To import your data, follow the steps:

1. Go to [Azure Video Indexer portal](#)
2. Select your trial account and go to the **Account settings** page.
3. Click the **Import content to an ARM-based account**.
4. From the dropdown menu choose the ARM-based account you wish to import the data to.
 - If the account ID isn't showing, you can copy and paste the account ID from Azure portal or the account list, on the side blade in the Azure Video Indexer Portal.
5. Click **Import content**



All media and content model customizations will be copied from the trial account into the new ARM-based account.

Next steps

You can programmatically interact with your trial account and/or with your Azure 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.

Create a classic Azure Video Indexer account

9/22/2022 • 7 minutes to read • [Edit Online](#)

IMPORTANT

Face identification, customization and celebrity recognition features access is limited based on eligibility and usage criteria in order to support our Responsible AI principles. Face identification, customization and celebrity recognition features are only available to Microsoft managed customers and partners. Use the [Face Recognition intake form](#) to apply for access.

This topic shows how to create a new classic account connected to Azure using the [Azure Video Indexer website](#). You can also create an Azure Video Indexer classic account through our [API](#).

The topic discusses prerequisites that you need to connect to your Azure subscription and how to configure an Azure Media Services account.

A few Azure Video Indexer account types are available to you. For detailed explanation, review [Account types](#).

For the pricing details, see [pricing](#).

Prerequisites for connecting to Azure

- An Azure subscription.

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

- An Azure Active Directory (Azure 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 AD](#)

- A user in your Azure AD domain with an **Application administrator** role. You'll use this member when connecting your Azure Video Indexer account to Azure.

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

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

- A user and member in your Azure AD domain.

You'll use this member when connecting your Azure 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 two roles. Once with Contributor and once with User Access Administrator. For more information, see [View the access a user has to Azure resources](#).

- Register the Event Grid 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, select **Register**. It takes a couple of minutes to register.

Connect to Azure

NOTE

Use the same Azure AD user you used when connecting to Azure.

It's strongly recommended to have the following three accounts located in the same region:

- The Azure Video Indexer account that you're creating.
- The Azure Video Indexer account that you're connecting with the Media Services account.
- The Azure storage account connected to the same Media Services account.

When you create an Azure Video Indexer account and connect it to Media Services, the media and metadata files are stored in the Azure storage account associated with that Media Services account.

If your storage account is behind a firewall, see [storage account that is behind a firewall](#).

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](#).

NOTE

Make sure to write down the Media Services resource and account names.

2. Before you can play your videos in the Azure Video Indexer web app, you must start the default **Streaming Endpoint** of the new Media Services account.

In the new Media Services account, select **Streaming endpoints**. Then select the streaming endpoint and press start.

NAME	STATUS	CDN	TYPE	STREAMING UNITS	HOST NAME
default	Stopped	Enabled	Standard	N/A	testmediaaccountv1-usea.streaming.media.azure.net

3. For Azure Video Indexer to authenticate with Media Services API, an AD app 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](#):

- a. In the new Media Services account, select **API access**.
- b. Select **Service principal authentication method**.
- c. Get the client ID and client secret

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

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

NOTE

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

Azure Media Services considerations

The following Azure Media Services related considerations apply:

- If you connect to a new Media Services account, Azure 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 Azure Video Indexer web app.

- If you connect to an existing Media Services account, Azure Video Indexer doesn't change the default Streaming Endpoint configuration. If there's no running **Streaming Endpoint**, you can't watch videos from this Media Services account or in Azure Video Indexer.

Create a classic account

1. On the [Azure Video Indexer website](#), select **Create unlimited account** (the paid account).
2. To create a classic account, select **Switch to manual configuration**.

In the dialog, provide the following information:

SETTING	DESCRIPTION
Azure Video Indexer account region	The name of the Azure Video Indexer account region. For better performance and lower costs, it's highly recommended to specify the name of the region where the Azure Media Services resource and Azure Storage account are located.
Azure AD 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 .
Subscription ID	The Azure subscription under which this connection should be created. The subscription ID can be retrieved from the Azure portal. Select 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.

SETTING	DESCRIPTION
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.

Import your content from the trial account

See [Import your content from the trial account](#).

Automate creation of the Azure Video Indexer account

To automate the creation of the account is a two steps process:

1. Use Azure Resource Manager to create an Azure Media Services account + Azure AD application.

See an example of the [Media Services account creation template](#).

2. Call [Create-Account with the Media Services and Azure AD application](#).

Azure Video Indexer in Azure Government

Prerequisites for connecting to Azure Government

- An Azure subscription in [Azure Government](#).
- An Azure AD account in Azure Government.
- All pre-requirements of permissions and resources as described above in [Prerequisites for connecting to Azure](#).

Create new account via the Azure Government portal

NOTE

The Azure Government cloud does not include a *trial* experience of Azure Video Indexer.

To create a paid account via the Azure Video Indexer portal:

1. Go to <https://videoindexer.ai.azure.us>
2. Sign-in with your Azure Government Azure AD account.
3. If you don't have any Azure Video Indexer accounts in Azure Government that you're an owner or a contributor to, you'll get an empty experience from which you can start creating your account.

The rest of the flow is as described in above, only the regions to select from will be Government regions in which Azure Video Indexer is available

If you already are a contributor or an admin of an existing one or more Azure Video Indexer accounts in Azure Government, you'll be taken to that account and from there you can start a following steps for creating an additional account if needed, as described above.

Create new account via the API on Azure Government

To create a paid account in Azure Government, follow the instructions in [Create-Paid-Account](#). This API end point only includes Government cloud regions.

Limitations of Azure Video Indexer on Azure Government

- Only paid accounts (ARM or classic) are available on Azure Government.
- No manual content moderation available in Azure Government.

In the public cloud when content is deemed offensive based on a content moderation, the customer can ask for a human to look at that content and potentially revert that decision.

- Bing description - in Azure Government we won't present a description of celebrities and named entities identified. This is a UI capability only.

Clean up resources

After you're done with this tutorial, delete resources that you aren't planning to use.

Delete an Azure Video Indexer account

If you want to delete an Azure Video Indexer account, you can delete the account from the Azure Video Indexer website. To delete the account, you must be the owner.

Select the account -> **Settings** -> **Delete this account**.

The account will be permanently deleted in 90 days.

Next steps

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

Upload and index your videos

9/22/2022 • 14 minutes to read • [Edit Online](#)

This article shows how to upload and index videos by using the Azure Video Indexer website (see [get started with the website](#)) and the Upload Video API (see [get started with API](#)).

After you upload and index a video, you can use [Azure Video Indexer website](#) or [Azure Video Indexer Developer Portal](#) to see the insights of the video (see [Examine the Azure Video Indexer output](#)).

Supported file formats

For a list of file formats that you can use with Azure Video Indexer, see [Standard Encoder formats and codecs](#).

Storage of video files

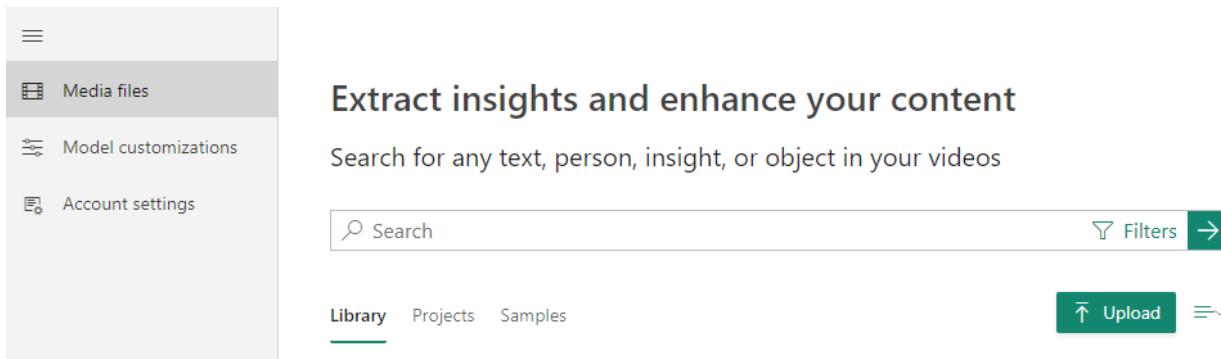
When you use Azure Video Indexer, video files are stored in Azure Storage through Media Services. The limits are 30 GB in size and 4 hours in length.

You can always delete your video and audio files, along with any metadata and insights that Azure Video Indexer has extracted from them. After you delete a file from Azure Video Indexer, the file and its metadata and insights are permanently removed from Azure Video Indexer. However, if you've implemented your own backup solution in Azure Storage, the file remains in Azure Storage.

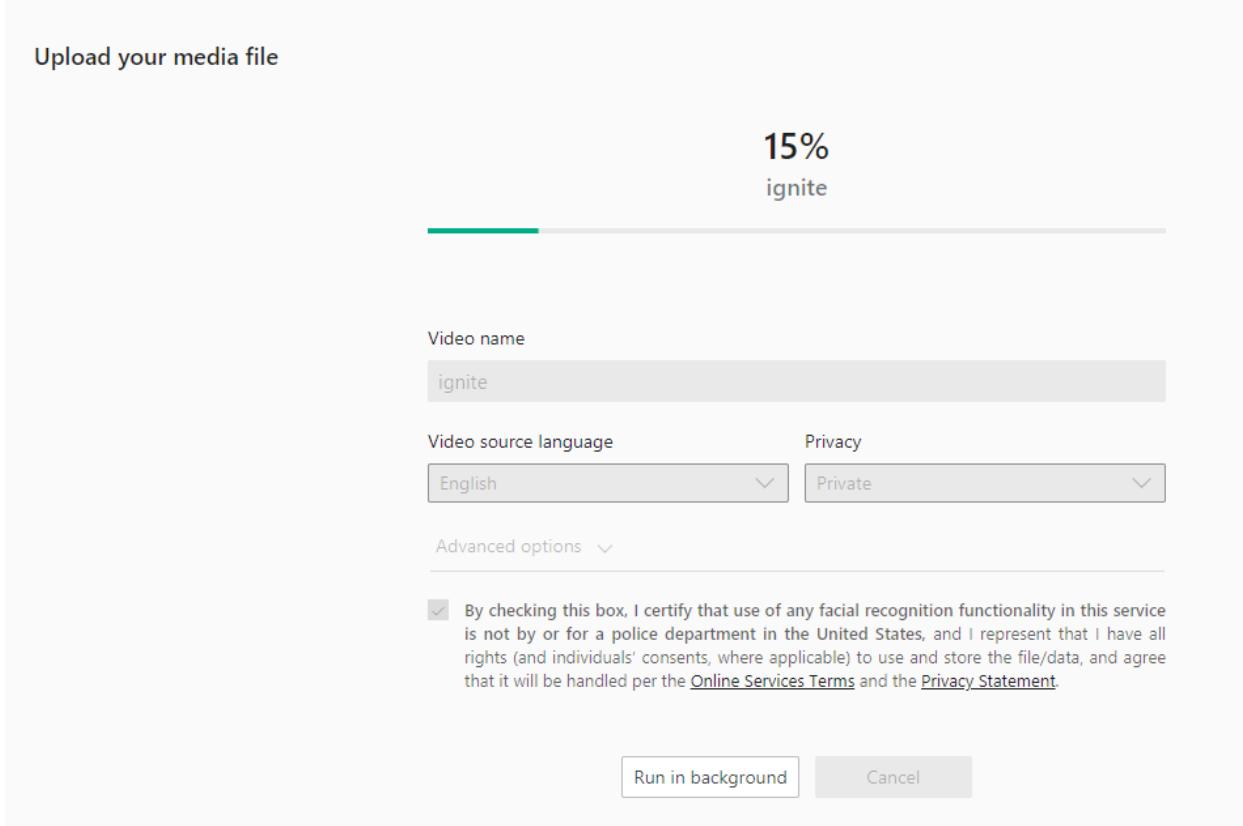
The persistence of a video is identical whether you upload by using the Azure Video Indexer website or by using the Upload Video API.

Upload and index a video by using the website

Sign in on the [Azure Video Indexer](#) website, and then select **Upload**.



After your video is uploaded, Azure Video Indexer starts indexing and analyzing the video.



After Azure Video Indexer is done analyzing, you get an email with a link to your video. The email also includes a short description of what was found in your video (for example: people, topics, optical character recognition).

Upload and index a video by using the API

You can use the [Upload Video](#) API to upload and index your videos based on a URL. The code sample that follows includes the commented-out code that shows how to upload the byte array.

NOTE

Before you proceed, make sure to review [API recommendations](#).

Configurations and parameters

This section describes some of the optional parameters and when to set them. For the most up-to-date info about parameters, see the [Azure Video Indexer portal](#).

externalID

Use this parameter to specify an ID that will be associated with the video. The ID can be applied to integration into an external video content management (VCM) system. The videos that are in the Azure Video Indexer portal can be searched via the specified external ID.

callbackUrl

Use this parameter to specify a callback URL.

A callback URL is used to notify the customer (through 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

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

indexingPreset

Use this parameter to define an AI bundle that you want to apply on your audio or video file. This parameter is used to configure the indexing process. You can specify the following values:

- `AudioOnly` : Index and extract insights by using audio only (ignoring video).
- `VideoOnly` : Index and extract insights by using video only (ignoring audio).
- `Default` : Index and extract insights by using both audio and video.
- `DefaultWithNoiseReduction` : Index and extract insights from both audio and video, while applying noise reduction algorithms on the audio stream.

The `DefaultWithNoiseReduction` value is now mapped to a default preset (deprecated).

- `BasicAudio` : Index and extract insights by using audio only (ignoring video). Include only basic audio features (transcription, translation, formatting of output captions and subtitles).
- `AdvancedAudio` : Index and extract insights by using audio only (ignoring video). Include advanced audio features (such as audio event detection) in addition to the standard audio analysis.
- `AdvancedVideo` : Index and extract insights by using video only (ignoring audio). Include advanced video features (such as observed people tracing) in addition to the standard video analysis.
- `AdvancedVideoAndAudio` : Index and extract insights by using both advanced audio and advanced video analysis.

NOTE

The preceding advanced presets include models that are in public preview. When these models reach general availability, there might be implications for the price.

Azure Video Indexer covers up to two tracks of audio. If the file has more audio tracks, they're treated as one track. If you want to index the tracks separately, you need to extract the relevant audio file and index it as `AudioOnly`.

Price depends on the selected indexing option. For more information, see [Media Services pricing](#).

priority

Azure Video Indexer indexes videos according to their priority. Use the `priority` parameter to specify the index priority. The following values are valid: `Low`, `Normal` (default), and `High`.

This parameter is supported only for paid accounts.

streamingPreset

After your video is uploaded, Azure Video Indexer optionally encodes the video. It then proceeds to indexing and analyzing the video. When Azure Video Indexer is done analyzing, you get a notification with the video ID.

When you're 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.

After 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.

For `SingleBitrate`, the standard encoder cost will apply for the output. If the video height is greater than or equal to 720, Azure Video Indexer encodes it as 1280 x 720. Otherwise, it's encoded as 640 x 468. The default setting is [content-aware encoding](#).

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

videoUrl

This parameter specifies the URL of the video or audio file to be indexed. If the `videoUrl` parameter is not specified, Azure Video Indexer expects you to pass the file as multipart/form body content.

Code sample

The following C# code snippets demonstrate the usage of all the Azure Video Indexer APIs together.

- [Classic account](#)
- [Azure Resource Manager account](#)

After you copy the following code into your development platform, you'll need to provide two parameters:

- API key (`apiKey`): Your personal API management subscription key. It allows you to get an access token in order to perform operations on your Azure Video Indexer account.

To get your API key:

1. Go to the [Azure Video Indexer portal](#).
 2. Sign in.
 3. Go to **Products > Authorization > Authorization subscription**.
 4. Copy the **Primary key** value.
- Video URL (`videoUrl`): A URL of the video or audio file to be indexed. Here are the requirements:
 - The URL must point at a media file. (HTML pages are not supported.)

- The file can be protected by an access token that's provided as part of the URI. The endpoint that serves the file must be secured with TLS 1.2 or later.
- The URL must be encoded.

The result of successfully running the code sample includes an insight widget URL and a player widget URL. They allow you to examine the insights and the uploaded video, respectively.

```
public async Task Sample()
{
    var apiUrl = "https://api.videoindexer.ai";
    var apiKey = "..."; // Replace with API key taken from https://aka.ms/viapiv2

    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<AccountContractSlim[]>(json);

    // Take the relevant account. Here we simply take the first.
    // You can also get the account via accounts.First(account => account.Id == <GUID>);
    var accountInfo = accounts.First();

    // We'll use the access token from here on, so there's no need for the APIM key
    client.DefaultRequestHeaders.Remove("Ocp-Apim-Subscription-Key");

    // Upload a video
    MultipartFormDataContent content = null;
    Console.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 parameters below and add the following lines:
    //content = new MultipartFormDataContent();
    //FileStream video = File.OpenRead(@"c:\videos\democratic3.mp4");
    //byte[] buffer = new byte[video.Length];
    //video.Read(buffer, 0, buffer.Length);
    //content.Add(new ByteArrayContent(buffer), "MyVideo", "MyVideo");

    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();
}
```

```

var uploadResult = await uploadRequestResult.Content.ReadAsStringAsync();

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

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

    // Job is finished
    if (processingState != "Uploaded" && processingState != "Processing")
    {
        Console.WriteLine("");
        Console.WriteLine("Full JSON:");
        Console.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();
Console.WriteLine("");
Console.WriteLine("Search:");
Console.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;  

    Console.WriteLine("Insights Widget url:");  

    Console.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;  

    Console.WriteLine("");  

    Console.WriteLine("Player Widget url:");  

    Console.WriteLine(playerWidgetLink);  

    Console.WriteLine("\nPress Enter to exit...");  

    String line = Console.ReadLine();  

    if (line == "enter")  

    {  

        System.Environment.Exit(0);  

    }  

}  

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; }  

}

```

Index your videos stored on OneDrive

9/22/2022 • 14 minutes to read • [Edit Online](#)

This article shows how to index videos stored on OneDrive by using the Azure Video Indexer website.

Supported file formats

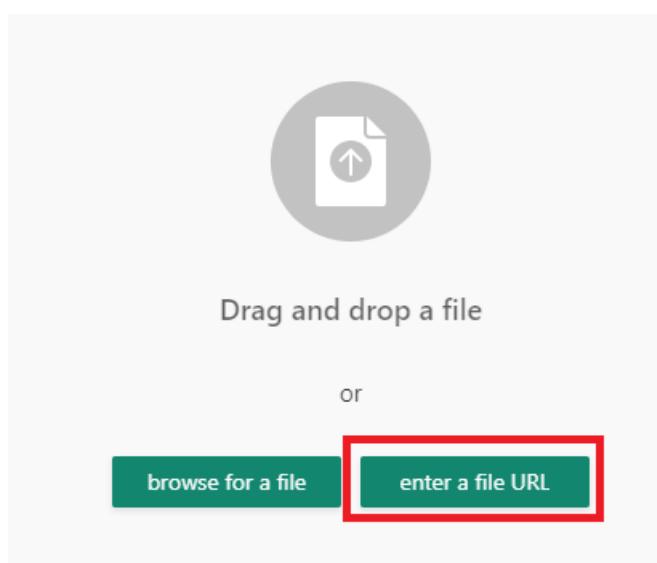
For a list of file formats that you can use with Azure Video Indexer, see [Standard Encoder formats and codecs](#).

Index a video by using the website

1. Sign into the [Azure Video Indexer](#) website, and then select **Upload**.



2. Click on **enter a file URL** button



3. Next, go to your video/audio file located on your OneDrive using a web browser. Select the file you want to index, at the top select **embed**



4. On the right click on **Generate** to generate an embed url.

Embed "AVAM-Video" in a blog or webpage.



Generate HTML code to embed this file

Generate

Note: Anyone who visits the blog or webpage with this embedded file will be able to view it without signing in.

5. Copy the embed code and extract only the URL part including the key. For example:

```
https://onedrive.live.com/embed?
cid=5BC591B7C713B04F&resid=5DC518B6B713C40F%2110126&authkey=HnsodidN_50oA31Lfk
```

Replace **embed** with **download**. You will now have a url that looks like this:

```
https://onedrive.live.com/download?
cid=5BC591B7C713B04F&resid=5DC518B6B713C40F%2110126&authkey=HnsodidN_50oA31Lfk
```

6. Now enter this URL in the Azure Video Indexer portal in the URL field.

The screenshot shows the Azure Video Indexer upload interface. It includes fields for 'Enter URL' containing the modified URL, 'Video name' with the value 'AVAM-Onedrive', 'Video source language' set to 'English', 'Privacy' set to 'Private', and an 'Advanced options' dropdown. Below the form is a note about uploading online files and a checkbox agreement. At the bottom are 'Upload' and 'Cancel' buttons.

Enter URL
https://onedrive.live.com/download?cid=5BC591B7C713B04F&resid=5DC518B6B713C40F%2110126&authkey=HnsodidN_50oA31Lfk

Video name
AVAM-Onedrive

Video source language
English

Privacy
Private

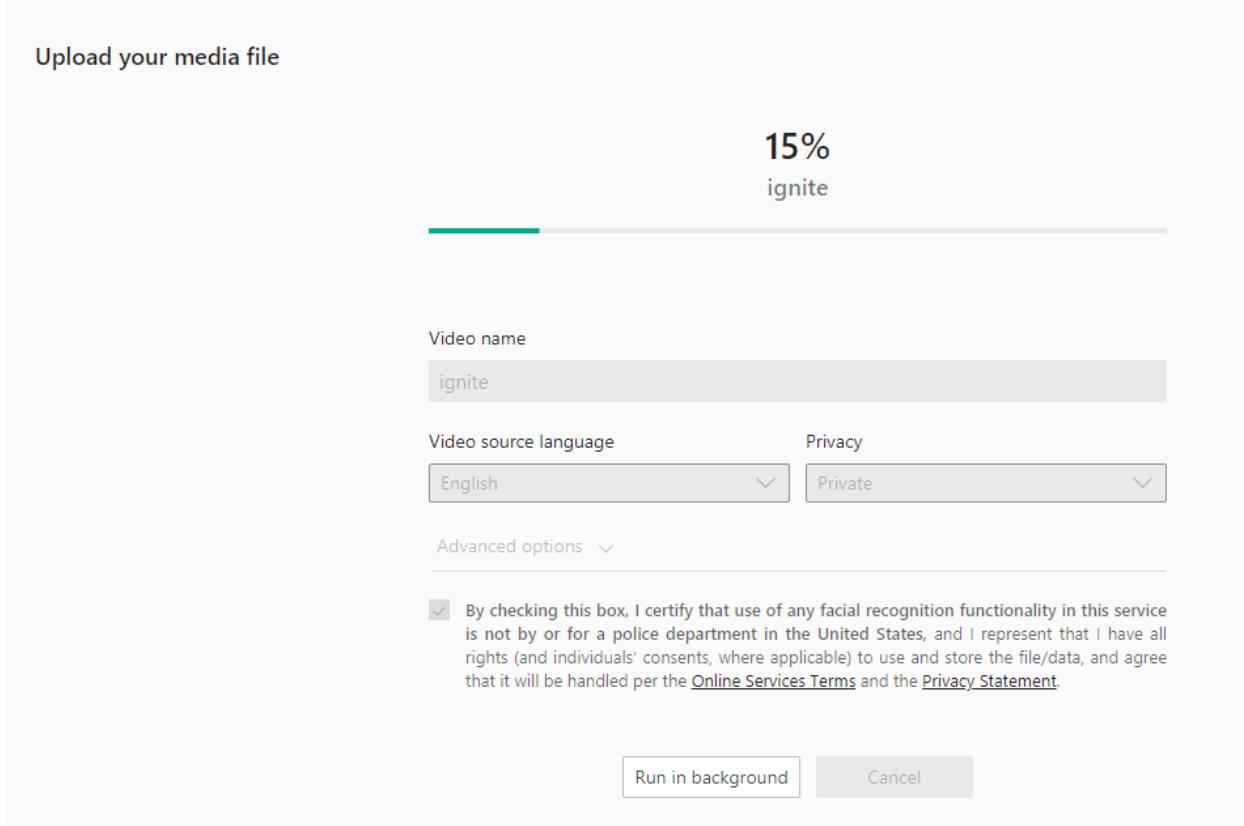
Advanced options ▾

[Learn more about uploading online files](#)

By submitting my file/data, I represent that I have all rights (and individuals' consents, where applicable) to use and store the file/data, and agree that it will be handled per the [Online Services Terms](#) and the [Privacy Statement](#).

Upload **Cancel**

After your video is downloaded from OneDrive, Azure Video Indexer starts indexing and analyzing the video.



Once Azure Video Indexer is done analyzing, you will receive an email with a link to your indexed video. The email also includes a short description of what was found in your video (for example: people, topics, optical character recognition).

Upload and index a video by using the API

You can use the [Upload Video](#) API to upload and index your videos based on a URL. The code sample that follows includes the commented-out code that shows how to upload the byte array.

Configurations and parameters

This section describes some of the optional parameters and when to set them. For the most up-to-date info about parameters, see the [Azure Video Indexer portal](#).

externalID

Use this parameter to specify an ID that will be associated with the video. The ID can be applied to integration into an external video content management (VCM) system. The videos that are in the Azure Video Indexer portal can be searched via the specified external ID.

callbackUrl

Use this parameter to specify a callback URL.

A callback URL is used to notify the customer (through 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

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

indexingPreset

Use this parameter to define an AI bundle that you want to apply on your audio or video file. This parameter is used to configure the indexing process. You can specify the following values:

- `AudioOnly` : Index and extract insights by using audio only (ignoring video).
- `VideoOnly` : Index and extract insights by using video only (ignoring audio).
- `Default` : Index and extract insights by using both audio and video.
- `DefaultWithNoiseReduction` : Index and extract insights from both audio and video, while applying noise reduction algorithms on the audio stream.

The `DefaultWithNoiseReduction` value is now mapped to a default preset (deprecated).

- `BasicAudio` : Index and extract insights by using audio only (ignoring video). Include only basic audio features (transcription, translation, formatting of output captions and subtitles).
- `AdvancedAudio` : Index and extract insights by using audio only (ignoring video). Include advanced audio features (such as audio event detection) in addition to the standard audio analysis.
- `AdvancedVideo` : Index and extract insights by using video only (ignoring audio). Include advanced video features (such as observed people tracing) in addition to the standard video analysis.
- `AdvancedVideoAndAudio` : Index and extract insights by using both advanced audio and advanced video analysis.

NOTE

The preceding advanced presets include models that are in public preview. When these models reach general availability, there might be implications for the price.

Azure Video Indexer covers up to two tracks of audio. If the file has more audio tracks, they're treated as one track. If you want to index the tracks separately, you need to extract the relevant audio file and index it as `AudioOnly`.

Price depends on the selected indexing option. For more information, see [Media Services pricing](#).

priority

Azure Video Indexer indexes videos according to their priority. Use the `priority` parameter to specify the index priority. The following values are valid: `Low`, `Normal` (default), and `High`.

This parameter is supported only for paid accounts.

streamingPreset

After your video is uploaded, Azure Video Indexer optionally encodes the video. It then proceeds to indexing and analyzing the video. When Azure Video Indexer is done analyzing, you get a notification with the video ID.

When you're 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.

After 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.

For `SingleBitrate`, the standard encoder cost will apply for the output. If the video height is greater than or equal to 720, Azure Video Indexer encodes it as 1280 x 720. Otherwise, it's encoded as 640 x 468. The default setting is [content-aware encoding](#).

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

videoUrl

This parameter specifies the URL of the video or audio file to be indexed. If the `videoUrl` parameter is not specified, Azure Video Indexer expects you to pass the file as multipart/form body content.

Code sample

NOTE

The following sample is intended for Classic accounts only and isn't compatible with ARM accounts. For an updated sample for ARM, see [this ARM sample repo](#).

The following C# code snippets demonstrate the usage of all the Azure Video Indexer APIs together.

- [Classic account](#)
- [Azure Resource Manager account](#)

After you copy the following code into your development platform, you'll need to provide two parameters:

- API key (`apiKey`): Your personal API management subscription key. It allows you to get an access token in order to perform operations on your Azure Video Indexer account.

To get your API key:

1. Go to the [Azure Video Indexer portal](#).
2. Sign in.
3. Go to **Products > Authorization > Authorization subscription**.
4. Copy the **Primary key** value.

- Video URL (`videoUrl`): A URL of the video or audio file to be indexed. Here are the requirements:

- The URL must point at a media file. (HTML pages are not supported.)
- The file can be protected by an access token that's provided as part of the URI. The endpoint that serves the file must be secured with TLS 1.2 or later.
- The URL must be encoded.

The result of successfully running the code sample includes an insight widget URL and a player widget URL. They allow you to examine the insights and the uploaded video, respectively.

```
public async Task Sample()
{
    var apiUrl = "https://api.videoindexer.ai";
    var apiKey = "..."; // Replace with API key taken from https://aka.ms/viapi

    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<AccountContractSlim[]>(json);

    // Take the relevant account. Here we simply take the first.
    // You can also get the account via accounts.First(account => account.Id == <GUID>);
    var accountInfo = accounts.First();

    // We'll use the access token from here on, so there's no need for the APIM key
    client.DefaultRequestHeaders.Remove("Ocp-Apim-Subscription-Key");

    // Upload a video
    var content = new MultipartFormDataContent();
    Console.WriteLine("Uploading...");
    // Get the video from URL
    var videoUrl = "VIDEO_URL"; // Replace with the video URL from OneDrive

    // As an alternative to specifying video URL, you can upload a file.
    // Remove the videoUrl parameter from the query parameters below and add the following lines:
    //FileStream video =File.OpenRead(Globals.VIDEOFILE_PATH);
    //byte[] buffer =new byte[video.Length];
    //video.Read(buffer, 0, buffer.Length);
    //content.Add(new ByteArrayContent(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"];
    Console.WriteLine("Uploaded");
    Console.WriteLine("Video ID:");
    Console.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"];

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

// Job is finished
if (processingState != "Uploaded" && processingState != "Processing")
{
    Console.WriteLine("");
    Console.WriteLine("Full JSON:");
    Console.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();
Console.WriteLine("");
Console.WriteLine("Search:");
Console.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;
Console.WriteLine("Insights Widget url:");
Console.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;
Console.WriteLine("");
Console.WriteLine("Player Widget url:");
Console.WriteLine(playerWidgetLink);
Console.WriteLine("\nPress Enter to exit...");
String line = Console.ReadLine();
if (line == "enter")
{
    System.Environment.Exit(0);
}

}

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; }
}

```

Examine the Azure Video Indexer output

9/22/2022 • 15 minutes to read • [Edit Online](#)

When a video is indexed, Azure Video Indexer produces the JSON content that contains details of the specified video insights. The insights include transcripts, optical character recognition elements (OCRs), faces, topics, and similar details. Each insight type includes instances of time ranges that show when the insight appears in the video.

To visually examine the video's insights, press the **Play** button on the video on the [Azure Video Indexer](#) website.

The screenshot shows the Azure Media Services | Video Indexer interface. On the left is a video player displaying a video of Satya Nadella speaking at Ignite 2016. The video player shows a timestamp of 2:05 / 14:32. Below the video player are controls for video editor, share, and download. To the right of the video player are several sections of insights:

- 18 People:** A grid of 18 small profile pictures. One is highlighted for "Satya Nadella" with the title "Microsoft CEO". Below it is a timeline bar with a playhead.
- 17 Keywords:** A list of keywords including graph, azure, microsoft, app services, cloud, location services, etc. Below it is a timeline bar.
- Emotions:** A chart showing emotion percentages: 3.5% Joy (orange dot) and 1.7% Sadness (blue dot). Below it is a timeline bar.
- 11 Scenes:** A grid of 11 thumbnail images showing different scenes from the video.

Below these sections, there are links to "More videos with similar people and keywords" showing thumbnails for various other video clips.

When indexing with an API and the response status is OK, you get a detailed JSON output as the response content. When calling the [Get Video Index](#) API, we recommend passing `&includeSummarizedInsights=false`.

TIP

The JSON output produced by the website or API contains `Insights` and `SummarizedInsights` elements. We highly recommend using `Insights` and not using `SummarizedInsights` (which is present for backward compatibility).

This article examines the Azure Video Indexer output (JSON content). For information about what features and insights are available to you, see [Azure Video Indexer insights](#).

NOTE

All the access tokens in Azure Video Indexer expire in one hour.

Get the insights using the website

To get insights produced on the website or the Azure portal:

1. Browse to the [Azure Video Indexer](#) website and sign in.
2. Find a video whose output you want to examine.
3. Press **Play**.
4. Choose the **Insights** tab.
5. Select which insights you want to view (under the **View** drop-down).
6. Go to the **Timeline** tab to see timestamped transcript lines.
7. Select **Download > Insights (JSON)** to get the insights output file.
8. If you want to download artifacts, beware of the following:

WARNING

We do not recommend that you use data directly from the artifacts folder for production purposes. Artifacts are intermediate outputs of the indexing process. They are essentially raw outputs of the various AI engines that analyze the videos; the artifacts schema may change over time. It is recommended that you use the [Get Video Index API](#), as described in [Get insights and artifacts produced by the API](#).

For more information, see [View and edit video insights](#).

Get insights produced by the API

To retrieve the JSON file (OCR, face, keyframe, etc.) or an artifact type, call the [Get Video Index API](#).

This API returns a URL only with a link to the specific resource type you request. An additional GET request must be made to this URL for the specific artifact. The file types for each artifact type vary depending on the artifact:

JSON

- OCR
- Faces
- VisualContentModeration
- LanguageDetection
- MultiLanguageDetection
- Metadata
- Emotions
- TextualContentModeration
- AudioEffects
- ObservedPeople
- Labels

Zip file containing JPG images

- KeyframesThumbnails
- FacesThumbnails

Root elements of the insights

NAME	DESCRIPTION
accountId	The playlist's VI account ID.
id	The playlist's ID.
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 (<code>Private</code> or <code>Public</code>).
state	The playlist's state (<code>Uploaded</code> , <code>Processing</code> , <code>Processed</code> , <code>Failed</code> , or <code>Quarantined</code>).
isOwned	Indicates whether the current user created the playlist.
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 summarized insight .
videos	A list of videos that construct 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": [{ . . . }]
}
```

Summary of the insights

This section shows a summary of the insights.

TIP

The produced JSON output contains `Insights` and `SummarizedInsights` elements. We highly recommend using `Insights` and not using `SummarizedInsights` (which is present for backward compatibility).

ATTRIBUTE	DESCRIPTION
<code>name</code>	The name of the video. For example: <code>Azure Monitor</code> .
<code>id</code>	The ID of the video. For example: <code>63c6d532ff</code> .
<code>privacyMode</code>	Your breakdown can have one of the following modes: A <code>Public</code> video is visible to everyone in your account and anyone who has a link to the video. A <code>Private</code> video is visible to everyone in your account.
<code>duration</code>	The time when an insight occurred, in seconds.
<code>thumbnailVideoId</code>	The ID of the video from which the thumbnail was taken.
<code>thumbnailId</code>	The video's thumbnail ID. To get the actual thumbnail, call Get-Thumbnail and pass it <code>thumbnailVideoId</code> and <code>thumbnailId</code> .
<code>faces/animatedCharacters</code>	Contains zero or more faces. For more information, see faces/animatedCharacters .
<code>keywords</code>	Contains zero or more keywords. For more information, see keywords .
<code>sentiments</code>	Contains zero or more sentiments. For more information, see sentiments .
<code>audioEffects</code>	Contains zero or more audio effects. For more information, see audioEffects .
<code>labels</code>	Contains zero or more labels. For more information, see labels .
<code>brands</code>	Contains zero or more brands. For more information, see brands .
<code>statistics</code>	For more information, see statistics .
<code>emotions</code>	Contains zero or more emotions. For more information, see emotions .
<code>topics</code>	Contains zero or more topics. For more information, see topics .

videos

NAME	DESCRIPTION
<code>accountId</code>	The video's VI account ID.
<code>id</code>	The video's ID.
<code>name</code>	The video's name.
<code>state</code>	The video's state (<code>Uploaded</code> , <code>Processing</code> , <code>Processed</code> , <code>Failed</code> , or <code>Quarantined</code>).
<code>processingProgress</code>	The progress during processing. For example: <code>20%</code> .
<code>failureCode</code>	The failure code if the video failed to process. For example: <code>UnsupportedFileType</code> .
<code>failureMessage</code>	The failure message if the video failed to process.
<code>externalId</code>	The video's external ID (if the user specifies one).
<code>externalUrl</code>	The video's external URL (if the user specifies one).
<code>metadata</code>	The video's external metadata (if the user specifies one).
<code>isAdult</code>	Indicates whether the video was manually reviewed and identified as an adult video.
<code>insights</code>	The insights object. For more information, see insights .
<code>thumbnailId</code>	The video's thumbnail ID. To get the actual thumbnail, call Get-Thumbnail and pass it the video ID and thumbnail ID.
<code>publishedUrl</code>	A URL to stream the video.
<code>publishedUrlProxy</code>	A URL to stream the video on Apple devices.
<code>viewToken</code>	A short-lived view token for streaming the video.
<code>sourceLanguage</code>	The video's source language.
<code>language</code>	The video's actual language (translation).
<code>indexingPreset</code>	The preset used to index the video.
<code>streamingPreset</code>	The preset used to publish the video.
<code>linguisticModelId</code>	The transcript customization (CRIS) model used to transcribe the video.
<code>statistics</code>	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://videvmediaseservices.streaming.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

Each insight (for example, transcript lines, faces, or brands) contains a list of unique elements (for example, `face1`, `face2`, `face3`). Each element has its own metadata and a list of its instances, which are time ranges with additional 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 metadata can include the face's rectangle coordinates (`20,230,60,60`).

VERSION	THE CODE VERSION
<code>sourceLanguage</code>	The video's source language (assuming one master language), in the form of a BCP-47 string.
<code>language</code>	The insights language (translated from the source language), in the form of a BCP-47 string.
<code>transcript</code>	The transcript insight.
<code>ocr</code>	The OCR insight.
<code>keywords</code>	The keywords insight.
<code>transcripts</code>	Might contain one or more transcript .
<code>faces/animatedCharacters</code>	The faces/animatedCharacters insight.
<code>labels</code>	The labels insight.
<code>shots</code>	The shots insight.

VERSION	THE CODE VERSION
<code>brands</code>	The brands insight.
<code>audioEffects</code>	The audioEffects insight.
<code>sentiments</code>	The sentiments insight.
<code>visualContentModeration</code>	The visualContentModeration insight.
<code>textualContentModeration</code>	The textualContentModeration insight.
<code>emotions</code>	The emotions insight.
<code>topics</code>	The topics insight.
<code>speakers</code>	The speakers insight.

Example:

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

transcript

NAME	DESCRIPTION
<code>id</code>	The line ID.
<code>text</code>	The transcript itself.
<code>confidence</code>	The confidence level for transcript accuracy.
<code>speakerId</code>	The ID of the speaker.
<code>language</code>	The transcript language. It's intended to support transcripts where each line can have a different language.
<code>instances</code>	A list of time ranges where this line appeared. If the instance is in a transcript, it will have only one instance.

Example:

```

"transcript": [
{
  "id":1,
  "text":"Well, good morning everyone and welcome to",
  "confidence":0.8839,
  "speakerId":1,
  "language":"en-US",
  "instances":[
    {
      "adjustedStart":"0:00:10.21",
      "adjustedEnd":"0:00:12.81",
      "start":"0:00:10.21",
      "end":"0:00:12.81"
    }
  ]
},
{
  "id":2,
  "text":"ignite 2016. Your mission at Microsoft is to empower every",
  "confidence":0.8944,
  "speakerId":2,
  "language":"en-US",
  "instances":[
    {
      "adjustedStart":"0:00:12.81",
      "adjustedEnd":"0:00:17.03",
      "start":"0:00:12.81",
      "end":"0:00:17.03"
    }
  ]
}
]
}

```

ocr

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

NAME	DESCRIPTION
angle	The angle of the OCR rectangle, from <code>-180</code> to <code>180</code> . A value of <code>0</code> means left-to-right horizontal. A value of <code>90</code> means top-to-bottom vertical. A value of <code>180</code> means right-to-left horizontal. A value of <code>-90</code> means bottom-to-top vertical. A value of <code>30</code> means from top left to bottom right.

```
"ocr": [
  {
    "id": 0,
    "text": "LIVE FROM NEW YORK",
    "confidence": 675.971,
    "height": 35,
    "language": "en-US",
    "left": 31,
    "top": 97,
    "width": 400,
    "angle": 30,
    "instances": [
      {
        "start": "00:00:26",
        "end": "00:00:52"
      }
    ]
  },
],
```

keywords

NAME	DESCRIPTION
id	The keyword's ID.
text	The keyword's text.
confidence	Recognition confidence in the keyword.
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/animatedCharacters

The `animatedCharacters` element replaces `faces` if the video was indexed with an animated characters model. This indexing is done through a custom model in Custom Vision. Azure Video Indexer runs it on keyframes.

If faces (not animated characters) are present, Azure Video Indexer uses the Face API on all the video's frames to detect faces and celebrities.

NAME	DESCRIPTION
<code>id</code>	The face's ID.
<code>name</code>	The name of the face. It can be <code>Unknown #0</code> , an identified celebrity, or a customer-trained person.
<code>confidence</code>	The level of confidence in face identification.
<code>description</code>	A description of the celebrity.
<code>thumbnailId</code>	The ID of the thumbnail of the face.
<code>knownPersonId</code>	If it's a known person, the internal ID.
<code>referenceId</code>	If it's a Bing celebrity, the Bing ID.
<code>referenceType</code>	Currently, just Bing.
<code>title</code>	If it's a celebrity, the person's title. For example: <code>Microsoft's CEO</code> .
<code>imageUrl</code>	If it's a celebrity, the image URL.
<code>instances</code>	Instances of where the face appeared in the time range. Each instance also has a <code>thumbnailsIds</code> value.

```

"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": "00:00:07.240000",
                "adjustedEnd": "00:00:45.678000",
                "start": "00:00:07.240000",
                "end": "00:00:45.678000"
            },
            {
                "thumbnailsIds": ["00000000-51e5-4260-91a5-890fa05c68b0"],
                "adjustedStart": "00:10:23.957000",
                "adjustedEnd": "00:10:39.239000",
                "start": "00:10:23.957000",
                "end": "00:10:39.239000"
            }
        ]
    }
]

```

labels

NAME	DESCRIPTION
<code>id</code>	The label's ID.
<code>name</code>	The label's name. For example: <code>Computer</code> or <code>TV</code> .
<code>language</code>	The language of the label's name (when translated), in the form of a BCP-47 string.
<code>instances</code>	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
<code>id</code>	The scene's ID.
<code>instances</code>	A list of time ranges for this scene. (A scene can have only one 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
<code>id</code>	The shot's ID.
<code>keyFrames</code>	A list of keyframes within the shot. Each has an ID and a list of instance time ranges. Each keyframe instance has a <code>thumbnailId</code> field, which holds the keyframe's thumbnail ID.
<code>instances</code>	A list of time ranges for this shot. (A shot can have only one 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": "0:00:00",
                "end": "0:00:06.34",
                "duration": "0:00:06.34"
            }
        ]
    }
]

```

brands

Azure Video Indexer detects business and product brand names in the speech-to-text transcript and/or video OCR. This information does not include visual recognition of brands or logo detection.

NAME	DESCRIPTION
<code>id</code>	The brand's ID.
<code>name</code>	The brand's name.
<code>referenceId</code>	The suffix of the brand's Wikipedia URL. For example, <code>Target_Corporation</code> is the suffix of https://en.wikipedia.org/wiki/Target_Corporation .
<code>referenceUrl</code>	The brand's Wikipedia URL, if exists. For example: https://en.wikipedia.org/wiki/Target_Corporation .
<code>description</code>	The brand's description.
<code>tags</code>	A list of predefined tags that were associated with this brand.

NAME	DESCRIPTION
confidence	The confidence value of the Azure Video Indexer brand detector (0 - 1).
instances	A list of time ranges for this brand. Each instance has a brandType value, which indicates whether this brand appeared in the transcript or in an OCR.

```

"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"
    }
  ]
}
]

```

statistics

NAME	DESCRIPTION
CorrespondenceCount	The number of correspondences in the video.
SpeakerWordCount	The number of words per speaker.

NAME	DESCRIPTION
<code>SpeakerNumberOfFragments</code>	The number of fragments that the speaker has in a video.
<code>SpeakerLongestMonolog</code>	The speaker's longest monolog. If the speaker has silence inside the monolog, it's included. Silence at the beginning and the end of the monolog is removed.
<code>SpeakerTalkToListenRatio</code>	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 (preview)

NAME	DESCRIPTION
<code>id</code>	The audio effect's ID.
<code>type</code>	The audio effect's type.
<code>name</code>	The audio effect's type in the language in which the JSON was indexed.
<code>instances</code>	A list of time ranges where this audio effect appeared. Each instance has a confidence field.
<code>start</code> + <code>end</code>	The time range in the original video.
<code>adjustedStart</code> + <code>adjustedEnd</code>	Time range versus adjusted time range.

```
audioEffects: [
  {
    id: 0,
    type: "Laughter",
    name: "Laughter",
    instances: [
      {
        confidence: 0.8815,
        adjustedStart: "0:00:10.2",
        adjustedEnd: "0:00:11.2",
        start: "0:00:10.2",
        end: "0:00:11.2"
      },
      {
        confidence: 0.8554,
        adjustedStart: "0:00:48.26",
        adjustedEnd: "0:00:49.56",
        start: "0:00:48.26",
        end: "0:00:49.56"
      },
      {
        confidence: 0.8492,
        adjustedStart: "0:00:59.66",
        adjustedEnd: "0:01:00.66",
        start: "0:00:59.66",
        end: "0:01:00.66"
      }
    ]
  },
]
```

sentiments

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

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

```
"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` transcript contains time ranges that Azure Video Indexer found to potentially have adult content. If `visualContentModeration` is empty, no adult content was identified.

Videos that 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 that case, the `IsAdult` attribute will contain the result of the human review.

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

```

"VisualContentModeration": [
  {
    "id": 0,
    "adultScore": 0.00069,
    "racyScore": 0.91129,
    "instances": [
      {
        "start": "00:00:25.4840000",
        "end": "00:00:25.5260000"
      }
    ]
  },
  {
    "id": 1,
    "adultScore": 0.99231,
    "racyScore": 0.99912,
    "instances": [
      {
        "start": "00:00:35.5360000",
        "end": "00:00:35.5780000"
      }
    ]
  }
]

```

textualContentModeration

NAME	DESCRIPTION
<code>id</code>	The ID of the textual content moderation.
<code>bannedWordsCount</code>	The number of banned words.
<code>bannedWordsRatio</code>	The ratio of banned words to the total number of words.

emotions

Azure Video Indexer identifies emotions based on speech and audio cues.

NAME	DESCRIPTION
<code>id</code>	The emotion's ID.
<code>type</code>	The type of an identified emotion: <code>Joy</code> , <code>Sadness</code> , <code>Anger</code> , or <code>Fear</code> .
<code>instances</code>	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",
        "start": "0:07:19.57",
        "end": "0:07:23.25"
      }
    ]
  }
]

```

```

        "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

Azure Video Indexer makes an inference of main topics from transcripts. When possible, the second-level [IPTC](#)

taxonomy is included.

NAME	DESCRIPTION
<code>id</code>	The topic's ID.
<code>name</code>	The topic's name. For example: <code>Pharmaceuticals</code> .
<code>referenceId</code>	Breadcrumbs that reflect the topic's hierarchy. For example: <code>HEALTH AND WELLBEING/MEDICINE AND HEALTHCARE/PHARMACEUTICALS</code>
<code>confidence</code>	The confidence score in the range <code>0 - 1</code> . Higher is more confident.
<code>language</code>	The language used in the topic.
<code>iptcName</code>	The IPTC media code name, if detected.
<code>instances</code>	Currently, Azure Video Indexer does not index a topic to time intervals. 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"
        }
    ]
}]

. . .
```

speakers

NAME	DESCRIPTION
<code>id</code>	The speaker's ID.

NAME	DESCRIPTION
name	The speaker's name in the form of <code>Speaker #<number></code> . For example: <code>Speaker #1</code> .
instances	A list of time ranges where this speaker appeared.

```
"speakers": [
{
  "id":1,
  "name":"Speaker #1",
  "instances":[
    {
      "adjustedStart":"0:00:10.21",
      "adjustedEnd":"0:00:12.81",
      "start":"0:00:10.21",
      "end":"0:00:12.81"
    }
  ]
},
{
  "id":2,
  "name":"Speaker #2",
  "instances":[
    {
      "adjustedStart":"0:00:12.81",
      "adjustedEnd":"0:00:17.03",
      "start":"0:00:12.81",
      "end":"0:00:17.03"
    }
  ]
},
```

Next steps

Explore the [Azure Video Indexer Developer Portal](#).

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

View and edit Azure Video Indexer insights

9/22/2022 • 2 minutes to read • [Edit Online](#)

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

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

The page shows the video's insights.

The screenshot displays the Azure Video Indexer interface. On the left is a video player showing a man speaking at a conference. The video is 2:05 / 14:32. Below the player are controls for Public, Created 2 months ago by Video Indexer, Report, Share, and Download. To the right of the video player are several sections: '18 People' (listing Satya Nadella as Microsoft CEO), '17 Keywords' (including graph, azure, microsoft, app services, cloud, location services), 'Emotions' (33% Fear, 28% Anger, 7% Sadness, 12% Joy), and '102 Keyframes' (a grid of thumbnail images). At the bottom left, there is a section titled 'More videos with similar people and keywords' featuring thumbnails for various Microsoft events like Windows Holographic Technical Session, Build 2018 Keynote, and Creators.

4. View the 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.

TIP

The JSON output produced by the website or API contains `Insights` and `SummarizedInsights` elements. We highly recommend using `Insights` and not using `SummarizedInsights` (which is present for backward compatibility).

Select the **Timeline** tab to see transcripts with timelines and other information that you can choose from the **View** drop-down.

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](#).

If you want to download artifact files, beware of the following:

WARNING

We do not recommend that you use data directly from the artifacts folder for production purposes. Artifacts are intermediate outputs of the indexing process. They are essentially raw outputs of the various AI engines that analyze the videos; the artifacts schema may change over time. It is recommended that you use the [Get Video Index API](#), as described in [Get insights and artifacts produced by the API](#).

For more information, see [Insights output](#).

Next steps

[Use your videos' deep insights](#)

See also

[Azure Video Indexer overview](#)

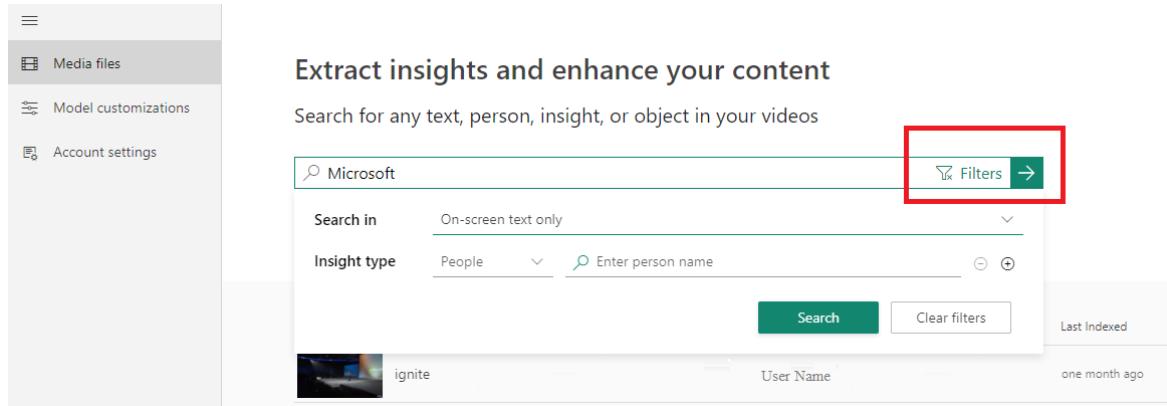
Search for exact moments in videos with Azure Video Indexer

9/22/2022 • 2 minutes to read • [Edit Online](#)

This topic shows you how to use the Azure Video Indexer website to search for exact moments in videos.

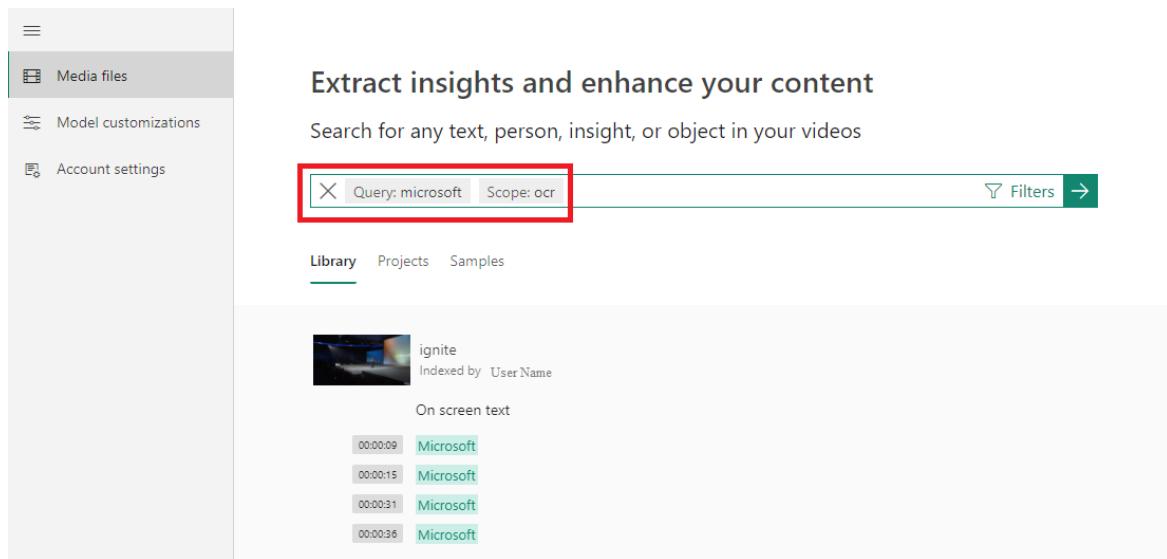
1. Go to the [Azure Video Indexer](#) website and sign in.
2. Specify the search keywords and the search will be performed among all videos in your account's library.

You can filter your search by selecting **Filters**. In below example, we search for "Microsoft" that appears as an on-screen text only (OCR).



The screenshot shows the Azure Video Indexer search interface. On the left is a sidebar with 'Media files' selected. The main area has a heading 'Extract insights and enhance your content' and a search bar containing 'Microsoft'. Below the search bar, 'Search in' is set to 'On-screen text only'. A red box highlights the 'Filters' button. The results section shows a video thumbnail for 'ignite' with the text 'User Name' and 'one month ago'.

3. Press **Search** to see the result.



The screenshot shows the search results for 'microsoft' with 'Scope: ocr' applied. The search bar now shows 'Query: microsoft Scope: ocr'. The results list shows the 'ignite' video with four 'Microsoft' instances highlighted at specific timestamps: 00:00:09, 00:00:15, 00:00:31, and 00:00:36.

If you select one of the results, the player brings you to that exact moment in the video.

4. View and search the summarized insights of the video by clicking **Play** on the video or selecting one of your original search results.

You can view, search, edit the **insights**. When you select one of the insights, the player brings you to that exact moment in the video.

The screenshot shows the Azure Video Indexer interface. On the left is a video player window displaying a man in a red shirt speaking on stage. To the right of the video are several sections of analysis results:

- 19 People:** A grid of 19 profile icons, with one highlighted for "Scott Guthrie".
- 8 topics:** Categories including Technology / Social change, Social innovation, Technology / Information systems, Cloud computing, Enterprise architecture, Environment / Climate and climate, Cloud, and 4 more topics.
- 51 Labels:** Various labels such as person, clothing, man, human face, footware, walk, standing, trousers, red, glasses, text, play, etc.
- 3 Named entities:** Entities including Scott Guthrie, Microsoft, and Microsoft.
- 4 Scenes:** A grid of four small video thumbnail scenes.

If you embed the video through Azure Video Indexer widgets, you can achieve the player/insights view and synchronization in your app. For more information, see [Embed Azure Video Indexer widgets into your app](#).

5. You can view, search, and edit the transcripts by clicking on the **Timeline** tab.

The screenshot shows the Azure Video Indexer interface with the **Timeline** tab selected. On the left is a video player window showing a man in a red shirt speaking. On the right is a transcript pane:

Timecode	Text
00:00:29	doing here at Microsoft to enable you to use technology to deliver amazing solutions.
00:00:34	Solutions
00:00:37	Solutions that
00:00:42	Speaker #9
00:00:49	your organizations. Solutions that take advantage of an intelligent
00:00:59	solutions that enable

Below the transcript are controls for **Edit**, **Download**, and **View**.

To edit the text, select **Edit** from the top-right corner and change the text as you need.

You can also translate and download the transcripts by selecting the appropriate option from the top-right corner.

Embed, download, create projects

You can embed your video by selecting </> **Embed** under your video. For details, see [Embed visual widgets in your application](#).

You can download the source video, insights of the video, transcripts by clicking **Download** under your video.

You can create a clip based on your video of specific lines and moments by clicking **Open in editor**. Then editing the video, and saving the project. For details, see [Use your videos' deep insights](#).



ignite

PRIVATE Created one month ago by User Name

Embed Download Open in editor

Next steps

Process content with Azure Video Indexer REST API

Scenes, shots, and keyframes

9/22/2022 • 4 minutes to read • [Edit Online](#)

Azure 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.



Scene detection

Azure 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. Azure 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

Azure 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

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

Extracting Keyframes

To extract high-resolution keyframes for your video, you must first upload and index the video.

The screenshot shows the AMS Video Indexer interface. At the top, there's a navigation bar with 'Create new account', 'Account' (amsdemo2017-247fc4), 'Trial', 'Download', 'English', 'View', and 'Edit'. Below the navigation is a search bar with 'Search'. The main content area has tabs for 'Insights' (which is selected) and 'Timeline'. A large video thumbnail for 'TODAY IN TECHNOLOGY' is displayed, showing an aerial view of a forest. Below the thumbnail, it says 'WHEN AN AMERICAN AND A RUSSIAN TOOK A "WALK IN THE WOODS"'. A progress bar indicates '0:02 / 5:25'. To the right of the thumbnail, there's a section titled '107 Keyframes' showing a grid of small video frames. At the bottom left, there's a video player control bar with icons for play, pause, volume, and other media controls.

Today in Technology: When an American and a Russian Took a "Walk in the Woods"

Public Created 7 months ago by Ella Ben-Tov

Embed Download Report Open in editor

With the Azure Video Indexer website

To extract keyframes using the Azure Video Indexer website, upload and index your video. Once the indexing job is complete, click on the **Download** button and select **Artifacts (ZIP)**. This will download the artifacts folder to your computer (make sure to view the warning regarding artifacts below). Unzip and open the folder. In the `_KeyframeThumbnail` folder, and you will find all of the keyframes that were extracted from your video.



WARNING

We do not recommend that you use data directly from the artifacts folder for production purposes. Artifacts are intermediate outputs of the indexing process. They are essentially raw outputs of the various AI engines that analyze the videos; the artifacts schema may change over time. It is recommended that you use the [Get Video Index API](#), as described in [Get insights and artifacts produced by the API](#).

With the Azure Video Indexer API

To get keyframes using the Video Indexer API, upload and index your video using the [Upload Video](#) call. Once the indexing job is complete, call [Get Video Index](#). This will give you all of the insights that Video Indexer extracted from your content in a JSON file.

You will get a list of keyframe IDs as part of each shot's metadata.

```

"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"
            }
        ]
    },
    {
        "start": "0:00:06.34"
    }
]

```

You will now need to run each of these keyframe IDs on the [Get Thumbnails](#) call. This will download each of the keyframe images to your computer.

Editorial shot type detection

Keyframes are associated with shots in the output JSON.

The shot type associated with an individual shot in the insights JSON represents its editorial type. You may find these shot type characteristics useful when editing videos into clips, trailers, or when searching for a specific style of keyframe for artistic purposes. The different types are determined based on analysis of the first keyframe of each shot. Shots are identified by the scale, size, and location of the faces appearing in their first keyframe.

The shot size and scale are determined based on the distance between the camera and the faces appearing in the frame. Using these properties, Azure Video Indexer detects the following shot types:

- Wide: shows an entire person's body.
- Medium: shows a person's upper-body and face.
- Close up: mainly shows a person's face.
- Extreme close-up: shows a person's face filling the screen.

Shot types can also be determined by location of the subject characters with respect to the center of the frame. This property defines the following shot types in Azure Video Indexer:

- Left face: a person appears in the left side of the frame.
- Center face: a person appears in the central region of the frame.
- Right face: a person appears in the right side of the frame.
- Outdoor: a person appears in an outdoor setting.
- Indoor: a person appears in an indoor setting.

Additional characteristics:

- Two shots: shows two persons' faces of medium size.
- Multiple faces: more than two persons.

Next steps

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

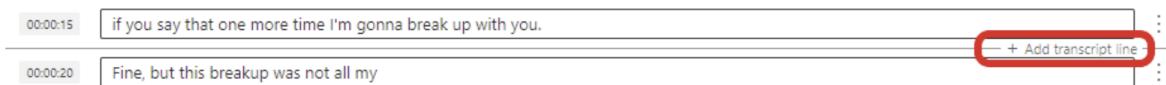
Insert or remove transcript lines in Video Indexer portal

9/22/2022 • 2 minutes to read • [Edit Online](#)

This article explains how to insert or remove a transcript line in Azure Video Indexer portal.

Add new line to the transcript timeline

While in the edit mode, hover between two transcription lines. You'll find a gap between **ending time** of the **transcript line** and the beginning of the following transcript line, user should see the following **add new transcription line** option.



After clicking the add new transcription line, there will be an option to add the new text and the time stamp for the new line. Enter the text, choose the time stamp for the new line, and select **save**. Default timestamp is the gap between the previous and next transcript line.



If there isn't an option to add a new line, you can adjust the end/start time of the relevant transcript lines to fit a new line in your desired place.

Choose an existing line in the transcript line, click the **three dots** icon, select **edit** and change the time stamp accordingly.

NOTE

New lines will not appear as part of the **From transcript edits** in the **Content model customization** under languages.

While using the API, when adding a new line, **Speaker name** can be added using free text. For example, *Speaker 1* can now become *Adam*.

Edit existing line

While in the edit mode, select the three dots icon. The editing options were enhanced, they now contain not just the text but also the timestamp with accuracy of milliseconds.

Delete line

Lines can now be deleted through the same three dots icon.

Example how and when to use this feature

To consolidate two lines which you believe should appear as one.

1. Go to line number 2, select edit.
2. Copy the text
3. Delete the line
4. Go to line 1, edit, paste the text and save.

Next steps

For updating transcript lines and text using API visit [Azure Video Indexer Developer portal](#)

Automatically identify and transcribe multi-language content

9/22/2022 • 2 minutes to read • [Edit Online](#)

Azure Video Indexer supports automatic language identification and transcription in multi-language content. This process involves automatically identifying the spoken language in different segments from audio, sending each segment of the media file to be transcribed and combine the transcription back to one unified transcription.

Choosing multilingual identification on indexing with portal

You can choose **multi-language detection** when uploading and indexing your video. Alternatively, you can choose **multi-language detection** when re-indexing your video. The following steps describe how to reindex:

1. Browse to the [Azure Video Indexer](#) website and sign in.
2. Go to the **Library** page and hover over the name of the video that you want to reindex.
3. On the right-bottom corner, click the **Re-index video** button.
4. In the **Re-index video** dialog, choose **multi-language detection** from the **Video source language** drop-down box.
 - When a video is indexed as multi-language, the insight page will include that option, and an additional insight type will appear, enabling the user to view which segment is transcribed in which language "Spoken language".
 - Translation to all languages is fully available from the multi-language transcript.
 - All other insights will appear in the master language detected – that is the language that appeared most in the audio.
 - Closed captioning on the player is available in multi-language as well.

The screenshot shows the Azure Video Indexer interface. On the left, a video player displays a scene between President Obama and King Don Felipe VI. The video player includes a closed captioning panel with options for Settings, Captions Off, and Multi-language checked. Below the video player, the caption text is shown in Spanish: "Señor presidente permitame nuevamente que le de la más cordial bienvenida a España, quiero agradeceros muy vivamente vuestra visita y sobre todo, gracias por haberla mantenido en las actuales circunstancias tan particulares." To the right of the video player is the "Timeline" tab of the interface. It lists transcription results for three segments. The first segment, at 00:00:00, is in Spanish and reads: "Señor presidente permitame nuevamente que le de la más cordial bienvenida a España, quiero agradeceros muy vivamente vuestra visita y sobre todo, gracias por haberla mantenido en las actuales circunstancias tan particulares." The second segment, at 00:00:18, is in English and reads: "Mister President allow me to agian, wish you the very warmest welcome to Spain. I want to thank you. Dearly, for visiting us and specially for maintaining this visit given the president particular circumstances." The third segment, at 00:00:36, is in Spanish and reads: "En nombre del pueblo español en este Palacio Real, simbolo de nuestra historia una historia y una tradición que hacen de nuestro país una nación fuerte con fuerte vocación americana y universal. Los Estados Unidos y España comparten principios valores e intereses los vínculos entre nuestras dos naciones se han fortalecido intensamente en todos los ámbitos tenemos la voluntad firme y el pleno compromiso de mantener siempre la más estrecha colaboración con los Estados Unidos de América un país, amigo y decisivo en el mundo."

Choosing multilingual identification on indexing with API

When indexing or [reindexing](#) a video using the API, choose the `multi-language detection` option in the `sourceLanguage` parameter.

Model output

The model will retrieve all of the languages detected in the video in one list

```
"sourceLanguage": null,  
"sourceLanguages": [  
    "es-ES",  
    "en-US"  
,
```

Additionally, each instance in the transcription section will include the language in which it was transcribed

```
{  
    "id": 136,  
    "text": "I remember well when my youth Minister took me to hear Doctor King I was a teenager.",  
    "confidence": 0.9343,  
    "speakerId": 1,  
    "language": "en-US",  
    "instances": [  
        {  
            "adjustedStart": "0:21:10.42",  
            "adjustedEnd": "0:21:17.48",  
            "start": "0:21:10.42",  
            "end": "0:21:17.48"  
        }  
    ]  
,},
```

Guidelines and limitations

- Set of supported languages: English, French, German, Spanish.
- Support for multi-language content with up to three supported languages.
- If the audio contains languages other than the supported list above, the result is unexpected.
- Minimal segment length to detect for each language – 15 seconds.
- Language detection offset is 3 seconds on average.
- Speech is expected to be continuous. Frequent alternations between languages may affect the models performance.
- Speech of non-native speakers may affect the model performance (for example, when speakers use their native tongue and they switch to another language).
- The model is designed to recognize a spontaneous conversational speech with reasonable audio acoustics (not voice commands, singing, etc.).
- Project creation and editing is currently not available for multi-language videos.
- Custom language models are not available when using multi-language detection.
- Adding keywords is not supported.
- When exporting closed caption files the language indication will not appear.
- The update transcript API does not support multiple languages files.

Next steps

[Azure Video Indexer overview](#)

Add video clips to your projects

9/22/2022 • 4 minutes to read • [Edit Online](#)

The [Azure Video Indexer](#) website enables you to use your video's 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 Azure 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 and add selected clips from the videos to the project.

Create new project and manage videos

1. Browse to the [Azure 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**.

The screenshot shows the Azure Video Indexer interface. At the top, there is a navigation bar with the text "Azure Media Services | Video Indexer". On the left, there is a sidebar with icons for Library, Projects (which is currently selected), and Samples. The main area has a heading "Extract insights and enhance your content" and a search bar with placeholder text "Search for any text, person, insight, or object in your videos". Below the search bar are three tabs: "Library", "Projects" (underlined in green to indicate it is selected), and "Samples". A table header with columns "Name", "Indexed by", and "Last Indexed" is visible. At the bottom of the table area, there is a button labeled "Create new project" with a plus sign icon. The overall layout is clean and modern, typical of a cloud-based service dashboard.

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.

The screenshot shows the Azure Media Services Video Indexer interface. At the top, it says "Azure Media Services | Video Indexer". Below that, there's a breadcrumb navigation "Untitled project" with a red box around the edit icon. To the right of the breadcrumb are buttons for "+ Add videos" and "View insights". A search bar below the breadcrumb says "Search transcript, on-screen text (OCR), people, and labels". The main area is titled "No videos" with the sub-instruction "Add videos to start editing". A green button at the bottom right says "+ Add videos". On the left side, there's a vertical sidebar with icons for a list, grid, and other settings.

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, the videos that have other languages will be grayed out/disabled.

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". You can 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 were looking for videos that mention "custom vision" in transcript only (use **Filter** if you want to filter your search results).

The screenshot shows the "Add videos" search interface. At the top, it says "Add up to 10 videos to a project. You can only add videos indexed in the same language." Below that is a search bar with "custom vision" typed in, which is also highlighted with a red box. To the right of the search bar are "Clear filters" and a dropdown menu. Underneath the search bar are filter options: "Find in" (with dropdown arrows) and "Visual text only". The main area lists two video results:

	computer-vision-made-easy	English
	cognitive-vision-services	English

At the bottom right are "Add" and "Cancel" buttons.

2. Click **Add** to add videos to the project.

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.

You can 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. To create queries for specific clips, use the search box that says "Search in transcript, visual text, people, and labels".
2. Select **View Insights** to customize which insights you want to see and which you don't want to see.

The screenshot shows the Azure Media Services Video Indexer interface. At the top, there's a navigation bar with 'Azure Media Services | Video Indexer' and a search bar containing 'custom vision project'. Below the search bar, there are buttons for '+ Add videos' and 'View insights'. A search input field is highlighted with a red box, containing the text 'try cognitive services'. On the right, there's a 'Render' and 'Save project' button. Underneath, a list of video clips is displayed. One clip, 'cognitive-vision-services', is selected and its details are shown: Duration: 00:52:12, segments: 00:42:44 - 00:42:45, 00:42:48 - 00:42:49, 00:42:50 - 00:42:54, 00:42:51 - 00:42:53, 00:42:53 - 00:42:53, and 00:42:53 - 00:42:54. A context menu is open over the third segment, with options 'Remove', 'Select all', and 'Move down' highlighted with a red box. Other clips listed include 'computer-vision-made-easy' with a duration of 01:01:49.

3. Add filters to further specify details on what scenes you are looking for by selecting **Filter options**.

You can add multiple filters.

4. Once you are happy with your results, select the clips you want to add to this 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 (or, all that were returned after your search) by clicking on the list menu option next to the video and selecting **Select all**.

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

IMPORTANT

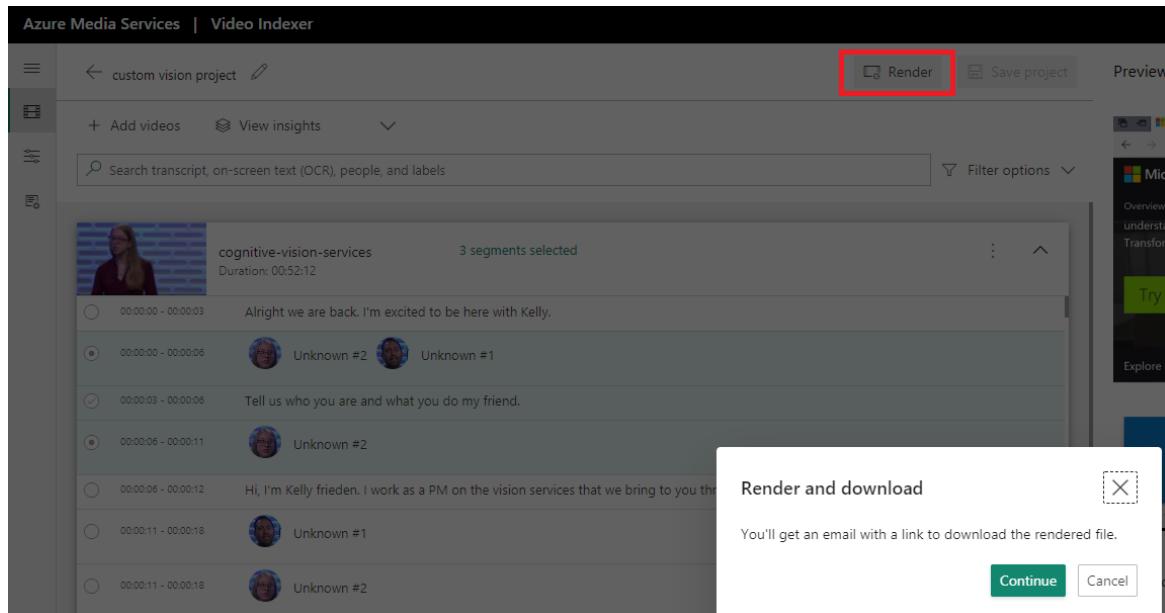
Remember to save your project when you make changes by selecting **Save project** at the top of the page.

Render and download the project

NOTE

For Azure Video Indexer paid accounts, rendering your project has encoding costs. Azure 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. Click **Render**, a popup dialog comes up that tells you that Azure Video Indexer will render a file and then the download link will be sent 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.

- 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.

Create a project from your video

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

- Go to the **Library** tab of the Azure Video Indexer website.
- 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

Automatically identify the spoken language with language identification model

9/22/2022 • 2 minutes to read • [Edit Online](#)

Azure Video Indexer supports automatic language identification (LID), which is the process of automatically identifying the spoken language content from audio and sending the media file to be transcribed in the dominant identified language.

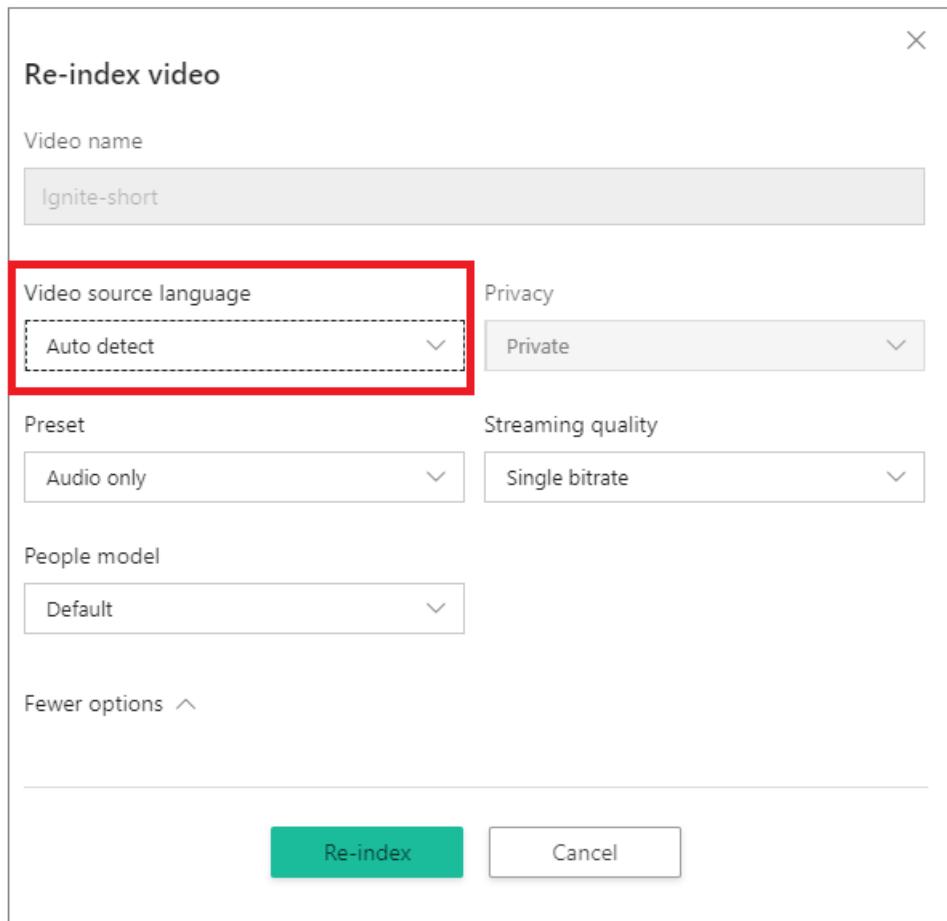
See the list of supported by Azure Video Indexer languages in [supported languages](#).

Make sure to review the [Guidelines and limitations](#) section below.

Choosing auto language identification on indexing

When indexing or [re-indexing](#) a video using the API, choose the `auto detect` option in the `sourceLanguage` parameter.

When using portal, go to your [Account videos](#) on the [Azure Video Indexer](#) home page and hover over the name of the video that you want to re-index. On the right-bottom corner click the re-index button. In the **Re-index video** dialog, choose *Auto detect* from the **Video source language** drop-down box.



Model output

Azure Video Indexer transcribes the video according to the most likely language if the confidence for that language is `> 0.6`. If the language can't be identified with confidence, it assumes the spoken language is

English.

Model dominant language is available in the insights JSON as the `sourceLanguage` attribute (under root/videos/insights). A corresponding confidence score is also available under the `sourceLanguageConfidence` attribute.

```
"insights": {  
    "version": "1.0.0.0",  
    "duration": "0:05:30.902",  
    "sourceLanguage": "fr-FR",  
    "language": "fr-FR",  
    "transcript": [...],  
    ...  
    "sourceLanguageConfidence": 0.8563  
},
```

Guidelines and limitations

- Automatic language identification (LID) supports the following languages:

See the list of supported by Azure Video Indexer languages in [supported langues](#).

- Even though Azure Video Indexer supports Arabic (Modern Standard and Levantine), Hindi, and Korean, these languages are not supported in LID.
- If the audio contains languages other than the supported list above, the result is unexpected.
- If Azure Video Indexer can't identify the language with a high enough confidence (`>0.6`), the fallback language is English.
- Currently, there isn't support for file with mixed languages audio. If the audio contains mixed languages, the result is unexpected.
- Low-quality audio may impact the model results.
- The model requires at least one minute of speech in the audio.
- The model is designed to recognize a spontaneous conversational speech (not voice commands, singing, etc.).

Next steps

- [Overview](#)
- [Automatically identify and transcribe multi-language content](#)

Trace observed people in a video (preview)

9/22/2022 • 2 minutes to read • [Edit Online](#)

Azure Video Indexer detects observed people in videos and provides information such as the location of the person in the video frame and the exact timestamp (start, end) when a person appears. The API returns the bounding box coordinates (in pixels) for each person instance detected, including detection confidence.

Some scenarios where this feature could be useful:

- Post-event analysis—detect and track a person's movement to better analyze an accident or crime post-event (for example, explosion, bank robbery, incident).
- Improve efficiency when creating raw data for content creators, like video advertising, news, or sport games (for example, find people wearing a red shirt in a video archive).
- Create a summary out of a long video, like court evidence of a specific person's appearance in a video, using the same detected person's ID.
- Learn and analyze trends over time, for example—how customers move across aisles in a shopping mall or how much time they spend in checkout lines.

For example, if a video contains a person, the detect operation will list the person's appearances together with their coordinates in the video frames. You can use this functionality to determine the person's path in a video. It also lets you determine whether there are multiple instances of the same person in a video.

The newly added **Observed people tracing** feature is available when indexing your file by choosing the **Advanced option -> Advanced video or Advanced video + audio preset** (under **Video + audio indexing**). Standard indexing will not include this new advanced model.

Video name

2001_ A Space Odyssey - Trailer [1968] HD - YouTube [720p]

Video source language

Privacy

English

Private

Video + audio indexing ⓘ

Streaming quality

Advanced

Single bitrate

Video + audio

Default

Standard

Advanced

Video only

Manage language models

Fewer options ⌂

By checking this box, I certify that use of any facial recognition functionality in this service is not by or for a police department in the United States, and I represent that I have all rights (and individuals' consents, where applicable) to use and store the file/data, and agree that it will be handled per the [Online Services Terms](#) and the [Privacy Statement](#).

When you choose to see **Insights** of your video on the [Video Indexer](#) website, the Observed People Tracing will show up on the page with all detected people thumbnails. You can choose a thumbnail of a person and see where the person appears in the video player.

The following JSON response illustrates what Video Indexer returns when tracing observed people:

```

{
...
"videos": [
{
...
"insights": {
...
"observedPeople": [
{
"id": 1,
"thumbnailId": "560f2cfb-90d0-4d6d-93cb-72bd1388e19d",
"instances": [
{
"adjustedStart": "0:00:01.5682333",
"adjustedEnd": "0:00:02.7027",
"start": "0:00:01.5682333",
"end": "0:00:02.7027"
}
]
},
{
"id": 2,
"thumbnailId": "9c97ae13-558c-446b-9989-21ac27439da0",
"instances": [
{
"adjustedStart": "0:00:16.7167",
"adjustedEnd": "0:00:18.018",
"start": "0:00:16.7167",
"end": "0:00:18.018"
}
]
},]
}
...
}
]
}

```

Limitations and assumptions

It's important to note the limitations of Observed People Tracing, to avoid or mitigate the effects of false negatives (missed detections) and limited detail.

- To optimize the detector results, use video footage from static cameras (although a moving camera or mixed scenes will also give results).
- People are generally not detected if they appear small (minimum person height is 200 pixels).
- Maximum frame size is HD
- People are generally not detected if they're not standing or walking.
- Low quality video (for example, dark lighting conditions) may impact the detection results.
- The recommended frame rate —at least 30 FPS.
- Recommended video input should contain up to 10 people in a single frame. The feature could work with more people in a single frame, but the detection result retrieves up to 10 people in a frame with the detection highest confidence.
- People with similar clothes (for example, people wear uniforms, players in sport games) could be detected as the same person with the same ID number.
- Obstruction – there maybe errors where there are obstructions (scene/self or obstructions by other people).
- Pose: The tracks may be split due to different poses (back/front)

Next steps

[Review overview](#)

Enable detected clothing feature (preview)

9/22/2022 • 2 minutes to read • [Edit Online](#)

Azure Video Indexer detects clothing associated with the person wearing it in the video and provides information such as the type of clothing detected and the timestamp of the appearance (start, end). The API returns the detection confidence level.

Two examples where this feature could be useful:

- Improve efficiency when creating raw data for content creators, like video advertising, news, or sport games (for example, find people wearing a red shirt in a video archive).
- Post-event analysis—detect and track a person's movement to better analyze an accident or crime post-event (for example, explosion, bank robbery, incident).

The newly added clothing detection feature is available when indexing your file by choosing the **Advanced option** -> **Advanced video** or **Advanced video + audio** preset (under Video + audio indexing). Standard indexing will not include this new advanced model.

Video name
2001_ A Space Odyssey - Trailer [1968] HD - YouTube [720p]

Video source language English Privacy Private

Video + audio indexing ⓘ
Advanced ↑
Video + audio
Standard
Advanced
Video only

Streaming quality Single bitrate
Default

[Manage language models](#)

Fewer options ▾

- By checking this box, I certify that use of any facial recognition functionality in this service is not by or for a police department in the United States, and I represent that I have all rights (and individuals' consents, where applicable) to use and store the file/data, and agree that it will be handled per the [Online Services Terms](#) and the [Privacy Statement](#).

When you choose to see **Insights** of your video on the [Azure Video Indexer](#) website, the People's detected clothing could be viewed from the **Observed People** tracing insight. When choosing a thumbnail of a person the detected clothing became available.

4 observed people (preview)

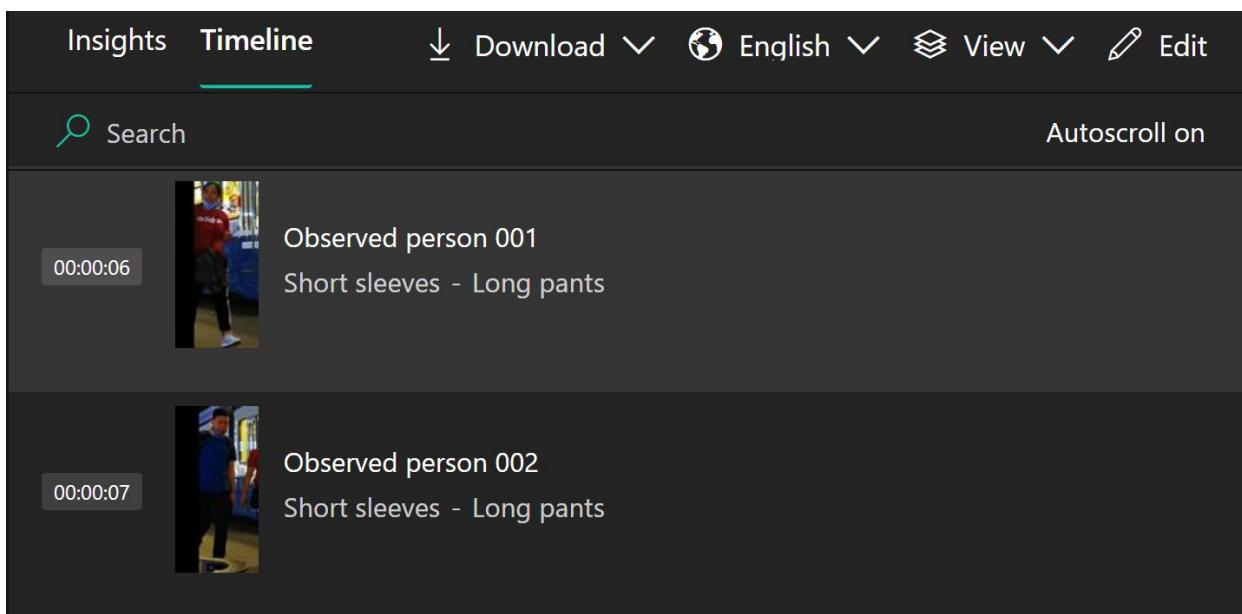


Observed person 002

Appears in 17.48% of video
Short sleeves - Long pants

Play previous ▶ Play next

If you are interested to view People's detected clothing in the Timeline of your video on the Azure Video Indexer website, go to **View -> Show Insights** and select the **All** option or **View -> Custom View** and select **Observed People**.



The screenshot shows the Azure Video Indexer interface with the **Timeline** tab selected. The timeline displays two entries:

- Observed person 001**: Short sleeves - Long pants. This entry includes a thumbnail image and a timestamp of 00:00:06.
- Observed person 002**: Short sleeves - Long pants. This entry includes a thumbnail image and a timestamp of 00:00:07.

At the top, there are navigation links for **Insights**, **Timeline**, **Download**, **English**, **View**, and **Edit**. There is also a search bar and an **Autoscroll on** button.

Searching for a specific clothing to return all the observed people wearing it is enabled using the search bar of either the **Insights** or from the **Timeline** of your video on the Azure Video Indexer website .

The following JSON response illustrates what Azure Video Indexer returns when tracing observed people having detected clothing associated:

```

"observedPeople": [
  {
    "id": 1,
    "thumbnailId": "68bab0f2-f084-4c2b-859b-a951ed03c209",
    "clothing": [
      {
        "id": 1,
        "type": "sleeve",
        "properties": {
          "length": "short"
        }
      },
      {
        "id": 2,
        "type": "pants",
        "properties": {
          "length": "long"
        }
      }
    ],
    "instances": [
      {
        "adjustedStart": "0:00:05.5055",
        "adjustedEnd": "0:00:09.9766333",
        "start": "0:00:05.5055",
        "end": "0:00:09.9766333"
      }
    ]
  },
  {
    "id": 2,
    "thumbnailId": "449bf52d-06bf-43ab-9f6b-e438cde4f217",
    "clothing": [
      {
        "id": 1,
        "type": "sleeve",
        "properties": {
          "length": "short"
        }
      },
      {
        "id": 2,
        "type": "pants",
        "properties": {
          "length": "long"
        }
      }
    ],
    "instances": [
      {
        "adjustedStart": "0:00:07.2072",
        "adjustedEnd": "0:00:10.5105",
        "start": "0:00:07.2072",
        "end": "0:00:10.5105"
      }
    ]
  },
]

```

Limitations and assumptions

It's important to note the limitations of Detected clothing, to avoid or mitigate the effects of false negatives (missed detections).

- To optimize the detector results, use video footage from static cameras (although a moving camera or mixed

scenes will also give results).

- People are not detected if they appear small (minimum person height is 200 pixels).
- Maximum frame size is HD
- People are not detected if they're not standing or walking.
- Low-quality video (for example, dark lighting conditions) may impact the detection results.
- The recommended frame rate at least 30 FPS.
- Recommended video input should contain up to 10 people in a single frame. The feature could work with more people in a single frame, but the detection result retrieves up to 10 people in a frame with the detection highest confidence.
- People with similar clothes (for example, people wear uniforms, players in sport games) could be detected as the same person with the same ID number.
- Obstructions – there maybe errors where there are obstructions (scene/self or obstructions by other people).
- Pose: The tracks may be split due to different poses (back/front)

Next steps

[Trace observed people in a video](#)

Enable featured clothing of an observed person (preview)

9/22/2022 • 2 minutes to read • [Edit Online](#)

When indexing a video using Azure Video Indexer advanced video settings, you can view the featured clothing of an observed person. The insight provides information of key items worn by individuals within a video and the timestamp in which the clothing appears. This allows high-quality in-video contextual advertising, where relevant clothing ads are matched with the specific time within the video in which they are viewed.

This article discusses how to view the featured clothing insight and how the featured clothing images are ranked.

Viewing featured clothing

The featured clothing insight is available when indexing your file by choosing the Advanced option -> Advanced video or Advanced video + audio preset (under Video + audio indexing). Standard indexing will not include this insight.

The screenshot shows the Azure Video Indexer interface for indexing a video. The 'Video name' field contains '2001_ A Space Odyssey - Trailer [1968] HD - YouTube [720p]'. Under 'Video source language', 'English' is selected. Under 'Privacy', 'Private' is selected. In the 'Video + audio indexing' section, 'Advanced' is selected from a dropdown menu. Other options shown are 'Video + audio' (highlighted in green), 'Standard', and 'Video only'. To the right, under 'Streaming quality', 'Single bitrate' is selected from a dropdown menu. Other options shown are 'Default' and 'Custom'. At the bottom, there is a link to 'Manage language models' and a 'Fewer options' button. A checkbox agreement is at the bottom.

Video name

2001_ A Space Odyssey - Trailer [1968] HD - YouTube [720p]

Video source language

English

Privacy

Advanced

Video + audio

Standard

Advanced

Video only

Streaming quality

Single bitrate

Default

Manage language models

Fewer options

By checking this box, I certify that use of any facial recognition functionality in this service is not by or for a police department in the United States, and I represent that I have all rights (and individuals' consents, where applicable) to use and store the file/data, and agree that it will be handled per the [Online Services Terms](#) and the [Privacy Statement](#).

The featured clothing images are ranked based on some of the following factors: key moments of the video, general emotions from text or audio. The `id` property indicates the ranking index. For example, `"id": 1` signifies the most important featured clothing.

NOTE

The featured clothing currently can only be viewed from the artifact file.

1. In the right-upper corner, select to download the artifact zip file: [Download -> Artifact \(ZIP\)](#)
2. Open `featuredclothing.zip`.

The .zip file contains two objects:

- `featuredclothing.map.json` – the file contains instances of each featured clothing, with the following properties:
 - `id` – ranking index (`"id": 1` is the most important clothing).
 - `confidence` – the score of the featured clothing.
 - `frameIndex` – the best frame of the clothing.
 - `timestamp` – corresponding to the frameIndex.
 - `opBoundingBox` – bounding box of the person.
 - `faceBoundingBox` – bounding box of the person's face, if detected.
 - `fileName` – where the best frame of the clothing is saved.

An example of the featured clothing with `"id": 1`.

```
"instances": [
  {
    "confidence": 0.98,
    "faceBoundingBox": {
      "x": 0.50158,
      "y": 0.10508,
      "width": 0.13589,
      "height": 0.45372
    },
    "fileName": "frame_12147.jpg",
    "frameIndex": 12147,
    "id": 1,
    "opBoundingBox": {
      "x": 0.34141,
      "y": 0.16667,
      "width": 0.28125,
      "height": 0.82083
    },
    "timestamp": "00:08:26.6311250"
  },
]
```

- `featuredclothing.frames.map` – this folder contains images of the best frames that the featured clothing appeared in, corresponding to the `fileName` property in each instance in `featuredclothing.map.json`.

Limitations and assumptions

It's important to note the limitations of featured clothing to avoid or mitigate the effects of false detections of images with low quality or low relevancy.

- Pre-condition for the featured clothing is that the person wearing the clothes can be found in the observed people insight.

- If the face of a person wearing the featured clothing wasn't detected, the results won't include the faces bounding box.
- If a person in a video wears more than one outfit, the algorithm selects its best outfit as a single featured clothing image.
- When posed, the tracks are optimized to handle observed people who most often appear on the front.
- Wrong detections may occur when people are overlapping.
- Frames containing blurred people are more prone to low quality results.

For more information, see the [limitations of observed people](#).

Next steps

- [Trace observed people in a video](#)
- [People's detected clothing](#)

Enable the matched person insight (preview)

9/22/2022 • 2 minutes to read • [Edit Online](#)

IMPORTANT

Face identification, customization and celebrity recognition features access is limited based on eligibility and usage criteria in order to support our Responsible AI principles. Face identification, customization and celebrity recognition features are only available to Microsoft managed customers and partners. Use the [Face Recognition intake form](#) to apply for access.

Azure Video Indexer matches observed people that were detected in the video with the corresponding faces ("People" insight). To produce the matching algorithm, the bounding boxes for both the faces and the observed people are assigned spatially along the video. The API returns the confidence level of each matching.

The following are some scenarios that benefit from this feature:

- Improve efficiency when creating raw data for content creators, like video advertising, news, or sport games (for example, find all appearances of a specific person in a video archive).
- Post-event analysis—detect and track specific person's movement to better analyze an accident or crime post-event (for example, explosion, bank robbery, incident).
- Create a summary out of a long video, to include the parts where the specific person appears.

The **Matched person** feature is available when indexing your file by choosing the **Advanced -> Video + audio indexing** preset.

NOTE

Standard indexing does not include this advanced model.

Video name

2001_ A Space Odyssey - Trailer [1968] HD - YouTube [720p]

Video source language

Privacy

English

Private

Video + audio indexing ⓘ

Streaming quality

Advanced

Single bitrate

Video + audio

Default

Standard

Advanced

Video only

Manage language models

Fewer options ⌂

By checking this box, I certify that use of any facial recognition functionality in this service is not by or for a police department in the United States, and I represent that I have all rights (and individuals' consents, where applicable) to use and store the file/data, and agree that it will be handled per the [Online Services Terms](#) and the [Privacy Statement](#).

To view the Matched person on the [Azure Video Indexer](#) website, go to **View** -> **Show Insights** -> select the **All** option or **View** -> **Custom View** -> **Mapped Faces**.

When you choose to see insights of your video on the [Azure Video Indexer](#) website, the matched person could be viewed from the **Observed People tracing** insight. When choosing a thumbnail of a person the matched person became available.



Observed person 025

Appears in 1% of video

Long sleeves

Matched person:



Satya Nadella



◀ Play previous

Play next ▶

If you would like to view people's detected clothing in the **Timeline** of your video on the [Video Indexer website](#), go to **View -> Show Insights** and select the **All** option or **View -> Custom View -> Observed People**.

Searching for a specific person by name, returning all the appearances of the specific person is enables using the search bar of the Insights of your video on the Azure Video Indexer.

JSON code sample

The following JSON response illustrates what Azure Video Indexer returns when tracing observed people having Mapped person associated:

```

"observedPeople": [
  {
    "id": 1,
    "thumbnailId": "d09ad62e-e0a4-42e5-8ca9-9a640c686596",
    "clothing": [
      {
        "id": 1,
        "type": "sleeve",
        "properties": {
          "length": "short"
        }
      },
      {
        "id": 2,
        "type": "pants",
        "properties": {
          "length": "short"
        }
      }
    ],
    "matchingFace": {
      "id": 1310,
      "confidence": 0.3819
    },
    "instances": [
      {
        "adjustedStart": "0:00:34.8681666",
        "adjustedEnd": "0:00:36.0026333",
        "start": "0:00:34.8681666",
        "end": "0:00:36.0026333"
      },
      {
        "adjustedStart": "0:00:36.6699666",
        "adjustedEnd": "0:00:36.7367",
        "start": "0:00:36.6699666",
        "end": "0:00:36.7367"
      },
      {
        "adjustedStart": "0:00:37.2038333",
        "adjustedEnd": "0:00:39.6729666",
        "start": "0:00:37.2038333",
        "end": "0:00:39.6729666"
      }
    ]
  }
]

```

Limitations and assumptions

It's important to note the limitations of Mapped person, to avoid or mitigate the effects of miss matches between people or people who have no matches.

Precondition for the matching is that the person that showing in the observed faces was detected and can be found in the People insight.

Pose: The tracks are optimized to handle observed people who most often appear on the front.

Obstructions: There is no match between faces and observed people where there are obstruction (people or faces overlapping each other).

Spatial allocation per frame: There is no match where different people appear in the same spatial position relatively to the frame in a short time.

See the limitations of Observed people: [Trace observed people in a video](#)

Next steps

[Overview](#)

Enable audio effects detection (preview)

9/22/2022 • 3 minutes to read • [Edit Online](#)

Audio effects detection is one of Azure Video Indexer AI capabilities that detects various acoustics events and classifies them into different acoustic categories (such as dog barking, crowd reactions, laughter and more).

Some scenarios where this feature is useful:

- Companies with a large set of video archives can easily improve accessibility with audio effects detection. The feature provides more context for persons who are hard of hearing, and enhances video transcription with non-speech effects.
- In the Media & Entertainment domain, the detection feature can improve efficiency when creating raw data for content creators. Important moments in promos and trailers (such as laughter, crowd reactions, gunshot, or explosion) can be identified by using **audio effects detection**.
- In the Public Safety & Justice domain, the feature can detect and classify gunshots, explosions, and glass shattering. It can be implemented in a smart-city system or in other public environments that include cameras and microphones to offer fast and accurate detection of violence incidents.

Supported audio categories

Audio effect detection can detect and classify 7 different categories. In the next table, you can find the different categories split in to the different presets, divided to **Standard** and **Advanced**. For more information, see [pricing](#).

INDEXING TYPE	STANDARD INDEXING	ADVANCED INDEXING
Preset Name	"Audio Only" "Video + Audio"	"Advance Audio" "Advance Video + Audio"
Appear in insights pane		V
Crowd Reactions		V
Silence	V	V
Gunshot or explosion		V
Breaking glass		V
Alarm or siren		V
Laughter		V
Dog barking		V

Result formats

The audio effects are retrieved in the insights JSON that includes the category ID, type, name, and set of instances per category along with their specific timeframe and confidence score.

The `name` parameter will be presented in the language in which the JSON was indexed, while the type will always remain the same.

```
audioEffects: [{
    id: 0,
    type: "Gunshot or explosion",
    name: "Gunshot",
    instances: [{
        confidence: 0.649,
        adjustedStart: "0:00:13.9",
        adjustedEnd: "0:00:14.7",
        start: "0:00:13.9",
        end: "0:00:14.7"
    }, {
        confidence: 0.7706,
        adjustedStart: "0:01:54.3",
        adjustedEnd: "0:01:55",
        start: "0:01:54.3",
        end: "0:01:55"
    }
],
},
{
    id: 1,
    type: "CrowdReactions",
    name: "Crowd Reactions",
    instances: [{
        confidence: 0.6816,
        adjustedStart: "0:00:47.9",
        adjustedEnd: "0:00:52.5",
        start: "0:00:47.9",
        end: "0:00:52.5"
    },
    {
        confidence: 0.7314,
        adjustedStart: "0:04:57.67",
        adjustedEnd: "0:05:01.57",
        start: "0:04:57.67",
        end: "0:05:01.57"
    }
]
},
],
```

How to index audio effects

In order to set the index process to include the detection of audio effects, the user should chose one of the **Advanced** presets under **Video + audio indexing** menu as can be seen below.

Enter URL

For example, a OneDrive file URL

Video name

Video source language

English

Privacy

Private

Video + audio indexing ⓘ

Advanced

Video + audio

Standard

Advanced

Video only

Streaming quality

Single bitrate

Closed Caption

When audio effects are retrieved in the closed caption files, they will be retrieved in square brackets the following structure:

TYPE	EXAMPLE
SRT	00:00:00,000 00:00:03,671 [Gunshot or explosion]
VTT	00:00:00.000 00:00:03.671 [Gunshot or explosion]
TTML	Confidence: 0.9047 <code><p begin="00:00:00.000" end="00:00:03.671">[Gunshot or explosion]</p></code>
TXT	[Gunshot or explosion]
CSV	0.9047,00:00:00.000,00:00:03.671, [Gunshot or explosion]

Audio Effects in closed captions file will be retrieved with the following logic employed:

- `Silence` event type will not be added to the closed captions
- Maximum duration to show an event is 5 seconds
- Minimum timer duration to show an event is 700 milliseconds

Adding audio effects in closed caption files

Audio effects can be added to the closed captions files supported by Azure Video Indexer via the [Get video captions API](#) by choosing true in the `includeAudioEffects` parameter or via the video.ai portal experience by selecting Download -> Closed Captions -> Include Audio Effects.

Download closed captions



Select format

SRT



Include audio effects

Download

Cancel

NOTE

When using [update transcript](#) from closed caption files or [update custom language model](#) from closed caption files, audio effects included in those files will be ignored.

Limitations and assumptions

- The audio effects are detected when present in non-speech segments only.
- The model is optimized for cases where there is no loud background music.
- Low quality audio may impact the detection results.
- Minimal non-speech section duration is 2 seconds.
- Music that is characterized with repetitive and/or linearly scanned frequency can be mistakenly classified as Alarm or siren.
- The model is currently optimized for natural and non-synthetic gunshot and explosions sounds.
- Door knocks and door slams can sometimes be mistakenly labeled as gunshot and explosions.
- Prolonged shouting and human physical effort sounds can sometimes be mistakenly detected.
- Group of people laughing can sometime be classified as both Laughter and Crowd reactions.

Next steps

Review [overview](#)

Use the animated character detection with portal and API

9/22/2022 • 7 minutes to read • [Edit Online](#)

Azure Video Indexer supports detection, grouping, and recognition of characters in animated content, this functionality is available through the Azure portal and through API. Review [this overview](#) article.

This article demonstrates to how to use the animated character detection with the Azure portal and the Azure Video Indexer API.

Use the animated character detection with portal

In the trial accounts the Custom Vision integration is managed by Azure Video Indexer, you can start creating and using the animated characters model. If using the trial account, you can skip the following ("Connect your Custom Vision account") section.

Connect your Custom Vision account (paid accounts only)

If you own an Azure Video Indexer paid account, you need to connect a Custom Vision account first. If you don't have a Custom Vision account already, create one. For more information, see [Custom Vision](#).

NOTE

Both accounts need to be in the same region. The Custom Vision integration is currently not supported in the Japan region.

Paid accounts that have access to their Custom Vision account can see the models and tagged images there. Learn more about [improving your classifier in Custom Vision](#).

The training of the model should be done only via Azure Video Indexer, and not via the Custom Vision website.

Connect a Custom Vision account with API

Follow these steps to connect your Custom Vision account to Azure Video Indexer, or to change the Custom Vision account that is currently connected to Azure Video Indexer:

1. Browse to www.customvision.ai and sign in.
2. Copy the keys for the Training and Prediction resources:

NOTE

To provide all the keys you need to have two separate resources in Custom Vision, one for training and one for prediction.

3. Provide other information:
 - Endpoint
 - Prediction resource ID
4. Browse and sign in to the [Azure Video Indexer](#).
5. Select the question mark on the top-right corner of the page and choose **API Reference**.

6. Make sure you're subscribed to API Management by clicking **Products** tab. If you have an API connected you can continue to the next step, otherwise, subscribe.

7. On the developer portal, select the **Complete API Reference** and browse to **Operations**.

8. Select **Connect Custom Vision Account** and select **Try it**.

9. Fill in the required fields and the access token and select **Send**.

For more information about how to get the Video Indexer access token go to the [developer portal](#), and see the [relevant documentation](#).

10. Once the call return 200 OK response, your account is connected.

11. To verify your connection by browse to the [Azure Video Indexer](#) portal:

12. Select the **Content model customization** button in the top-right corner.

13. Go to the **Animated characters** tab.

14. Once you select Manage models in Custom Vision, you'll be transferred to the Custom Vision account you just connected.

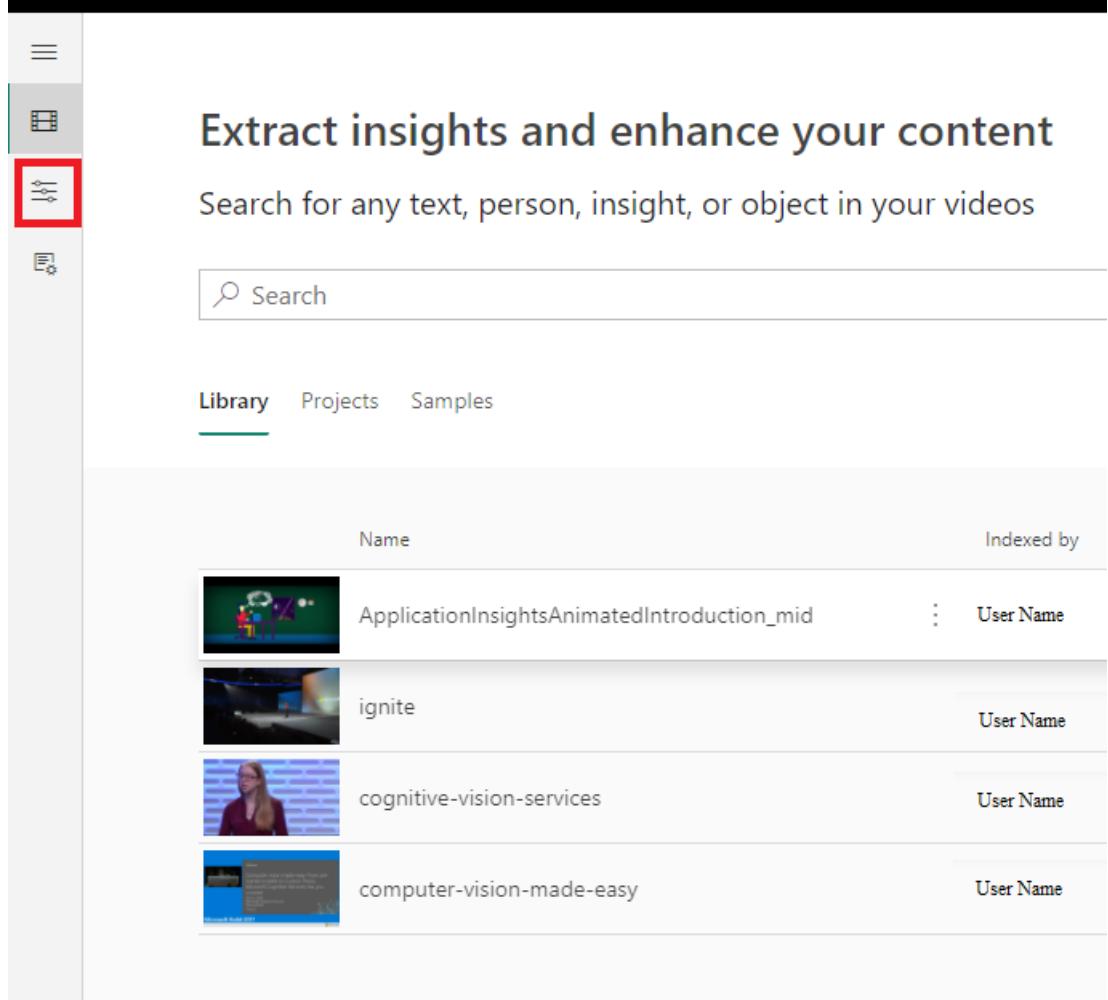
NOTE

Currently, only models that were created via Azure Video Indexer are supported. Models that are created through Custom Vision will not be available. In addition, the best practice is to edit models that were created through Azure Video Indexer only through the Azure Video Indexer platform, since changes made through Custom Vision may cause unintended results.

Create an animated characters model

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

2. To customize a model in your account, select the **Content model customization** button on the left of the page.



Extract insights and enhance your content

Search for any text, person, insight, or object in your videos

Search

Library Projects Samples

Name	Indexed by
ApplicationInsightsAnimatedIntroduction_mid	User Name
ignite	User Name
cognitive-vision-services	User Name
computer-vision-made-easy	User Name

3. Go to the **Animated characters** tab in the model customization section.

4. Select **Add model**.

5. Name your model and select enter to save the name.

NOTE

The best practice is to have one custom vision model for each animated series.

Index a video with an animated model

For the initial training, upload at least two videos. Each should be preferably longer than 15 minutes, before expecting good recognition model. If you have shorter episodes, we recommend uploading at least 30 minutes of video content before training. This will allow you to merge groups that belong to the same character from different scenes and backgrounds, and therefore increase the chance it will detect the character in the following episodes you index. To train a model on multiple videos (episodes), you need to index them all with the same animation model.

1. Select the **Upload** button.
2. Choose a video to upload (from a file or a URL).
3. Select **Advanced options**.
4. Under **People / Animated characters** choose **Animation models**.
5. If you have one model it will be chosen automatically, and if you have multiple models you can choose the relevant one out of the dropdown menu.
6. Select upload.

- Once the video is indexed, you'll see the detected characters in the **Animated characters** section in the **Insights** pane.

Before tagging and training the model, all animated characters will be named "Unknown #X". After you train the model, they'll also be recognized.

Customize the animated characters models

- Name the characters in Azure Video Indexer.
 - After the model created character group, it's recommended to review these groups in Custom Vision.
 - To tag an animated character in your video, go to the **Insights** tab and select the **Edit** button on the top-right corner of the window.
 - In the **Insights** pane, select any of the detected animated characters and change their names from "Unknown #X" to a temporary name (or the name that was previously assigned to the character).
 - After typing in the new name, select the check icon next to the new name. This saves the new name in the model in Azure Video Indexer.
- Paid accounts only: Review the groups in Custom Vision

NOTE

Paid accounts that have access to their Custom Vision account can see the models and tagged images there. Learn more about [improving your classifier in Custom Vision](#). It's important to note that training of the model should be done only via Azure Video Indexer (as described in this topic), and not via the Custom Vision website.

- Go to the **Custom Models** page in Azure Video Indexer and choose the **Animated characters** tab.
- Select the Edit button for the model you're working on to manage it in Custom Vision.
- Review each character group:
 - If the group contains unrelated images, it's recommended to delete these in the Custom Vision website.
 - If there are images that belong to a different character, change the tag on these specific images by selecting the image, adding the right tag and deleting the wrong tag.
 - If the group isn't correct, meaning it contains mainly non-character images or images from multiple characters, you can delete in Custom Vision website or in Azure Video Indexer insights.
 - The grouping algorithm will sometimes split your characters to different groups. It's therefore recommended to give all the groups that belong to the same character the same name (in Azure Video Indexer Insights), which will immediately cause all these groups to appear as one in Custom Vision website.
- Once the group is refined, make sure the initial name you tagged it with reflects the character in the group.

Train the model

- After you finished editing all names you want, you need to train the model.
- Once a character is trained into the model, it will be recognized in the next video indexed with that model.
- Open the customization page and select the **Animated characters** tab and then select the **Train** button to train your model. In order to keep the connection between Video

Indexer and the model, don't train the model in the Custom Vision website (paid accounts have access to Custom Vision website), only in Azure Video Indexer. Once trained, any video that will be indexed or reindexed

with that model will recognize the trained characters.

Delete an animated character and the model

1. Delete an animated character.

- a. To delete an animated character in your video insights, go to the **Insights** tab and select the **Edit** button on the top-right corner of the window.
- b. Choose the animated character and then select the **Delete** button under their name.

NOTE

This will delete the insight from this video but will not affect the model.

2. Delete a model.

- a. Select the **Content model customization** button on the top menu and go to the **Animated characters** tab.
- b. Select the ellipsis icon to the right of the model you wish to delete and then on the delete button.
 - Paid account: the model will be disconnected from Azure Video Indexer and you won't be able to reconnect it.
 - Trial account: the model will be deleted from Customs vision as well.

NOTE

In a trial account, you only have one model you can use. After you delete it, you can't train other models.

Use the animated character detection with API

1. Connect a Custom Vision account.

If you own an Azure Video Indexer paid account, you need to connect a Custom Vision account first. If you don't have a Custom Vision account already, create one. For more information, see [Custom Vision](#).

[Connect your Custom Vision account using API](#).

2. Create an animated characters model.

Use the [create animation model](#) API.

3. Index or reindex a video.

Use the [re-indexing](#) API.

4. Customize the animated characters models.

Use the [train animation model](#) API.

View the output

See the animated characters in the generated JSON file.

```
"animatedCharacters": [
  {
    "videoId": "e867214582",
    "confidence": 0,
    "thumbnailId": "00000000-0000-0000-0000-000000000000",
    "seenDuration": 201.5,
    "seenDurationRatio": 0.3175,
    "isKnownCharacter": true,
    "id": 4,
    "name": "Bunny",
    "appearances": [
      {
        "startTime": "0:00:52.333",
        "endTime": "0:02:02.6",
        "startSeconds": 52.3,
        "endSeconds": 122.6
      },
      {
        "startTime": "0:02:40.633",
        "endTime": "0:03:16.6",
        "startSeconds": 160.6,
        "endSeconds": 196.6
      }
    ]
  }
]
```

Limitations

- Currently, the "animation identification" capability isn't supported in East-Asia region.
- Characters that appear to be small or far in the video may not be identified properly if the video's quality is poor.
- The recommendation is to use a model per set of animated characters (for example per an animated series).

Next steps

[Azure Video Indexer overview](#)

Customize a Person model with the Azure Video Indexer website

9/22/2022 • 8 minutes to read • [Edit Online](#)

IMPORTANT

Face identification, customization and celebrity recognition features access is limited based on eligibility and usage criteria in order to support our Responsible AI principles. Face identification, customization and celebrity recognition features are only available to Microsoft managed customers and partners. Use the [Face Recognition intake form](#) to apply for access.

Azure 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 Azure Video Indexer](#).

You can use the Azure Video Indexer website to edit faces that were detected in a video, as described in this article. 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 Azure Video Indexer website and sign in.
2. Select the content model customization button on the left of the page.

The screenshot shows the Azure Media Services Video Indexer interface. On the left, there's a vertical navigation bar with icons for Home, Library, Projects, Samples, and a search icon (magnifying glass). The 'Library' icon is highlighted with a red box. Below the navigation bar, the main area has a heading 'Extract insights and enhance your content'. A search bar with the placeholder 'Search' is present. Below the search bar, there are tabs for 'Library', 'Projects', and 'Samples', with 'Library' being the active tab. The main content area displays a table with four rows of video entries:

Name	Indexed by
ApplicationInsightsAnimatedIntroduction_mid	User Name
ignite	User Name
cognitive-vision-services	User Name
computer-vision-made-easy	User Name

3. Select the People tab.

You'll 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 didn't specify a custom Person model during indexing.

If you created other Person models, they'll also be listed on this page.

Content model customization

[People](#) [Language](#) [Brands](#) [Animated characters \(preview\)](#)

Click on a person to manage their model and add images to enhance the model.
Uploaded photos should include only one face, and the file format can be .jpg or .png.
You can add up to 50 person models on each account.
[Learn more](#)

The screenshot shows the 'People' tab in the Azure Media Services Video Indexer interface. At the top, there's a search bar labeled 'Search people models...' and a '+ Add model' button. Below the search bar, it says 'Default model 1 person'. The table below lists the person model:

Name	Updated by	Last updated
Scott guthrie	User Name	22 days ago

Create a new Person model

1. Select the + Add model button on the right.
2. Enter the name of the model and select the check button to save the new model created. You can now add

new people and faces to the new Person model.

3. Select the list menu button and choose **+ Add person**.

The screenshot shows a list of Person models. At the top, there's a search bar labeled 'Enter model name'. Below it, a section for 'Soccer' shows '0 people'. A list item for 'Default model' is shown, indicating '1 person'. To the right of this list is a vertical menu with options: '+ Add person' (highlighted with a red box), 'Rename', 'Delete', and 'People model ID:'. Below the list, there's a table with columns 'Name', 'Updated by', and 'Last'. One row shows 'Scott guthrie' as the name, 'User Name' as the updated by, and '22' as the last value.

Add a new person to a Person model

NOTE

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

1. To add a new face to a Person model, select the list menu button next to the Person model that you want to add the face to.
2. Select **+ Add person** from the menu.

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

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

Azure Video Indexer can 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. 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. Select the list menu button next to the Person model that you want to rename.
2. Select **Rename** from the menu.
3. Select the current name of the model and type in your new name.
4. Select 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 can't delete your Default person model.

1. Select **Delete** from the menu.

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 can't be undone.

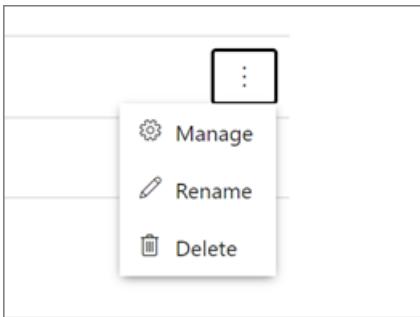
2. If you're sure, select delete again.

NOTE

The existing videos that were indexed using this (now deleted) Person model won't support the ability for you to update the names of the faces that appear in the video. You'll 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, select the arrow next to the name of the Person model. Then you can view all of the people in that particular Person model. If you select 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, select the list menu button and choose **Rename** from the list menu.
2. Select the current name of the person and type in your new name.
3. Select the check button, and the person will be renamed.

Delete a person

1. To delete a person from your Person model, select 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 can't be undone.
3. Select **Delete** again and this will remove the person from the Person model.

Check if a person already exists

You can use the search to check if a person already exists in the model.

Manage a person

If you select **Manage**, you see the **Person's details** window with 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've manually uploaded.

TIP

You can get to the **Person's details** window by clicking on the person's name or by clicking **Manage**, as shown above.

Add a face

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

Delete a face

Select the image you wish to delete and click **Delete**.

Rename and delete a person

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 steps:

1. Select the **Upload** button on the right of the page.
2. Drop your video file or browse for your file.
3. Select the **Advanced options** arrow.
4. Select the drop-down and select the Person model that you created.
5. Select the **Upload** option in the bottom of the page, and your new video will be indexed using your Person model.

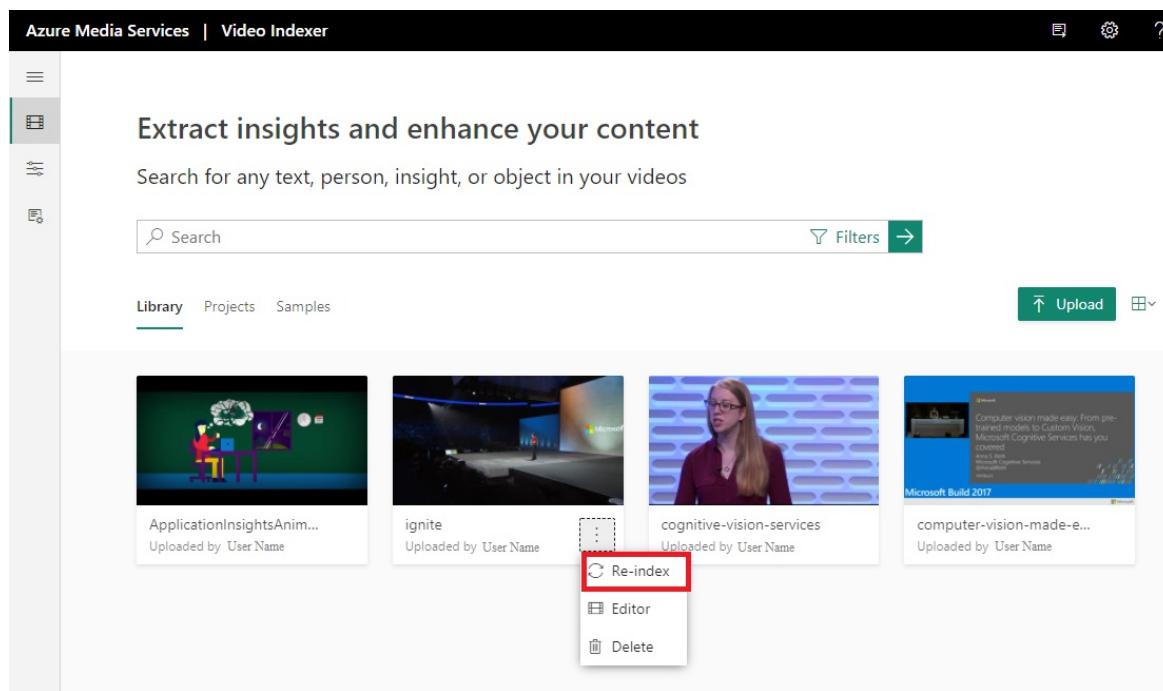
If you don't specify a Person model during the upload, Azure Video Indexer will index the video using the Default Person model in your account.

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 Azure 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. Select the option to reindex your video.



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

2. Select the drop-down and select the Person model that you want to use.
3. Select 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 are 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's 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.

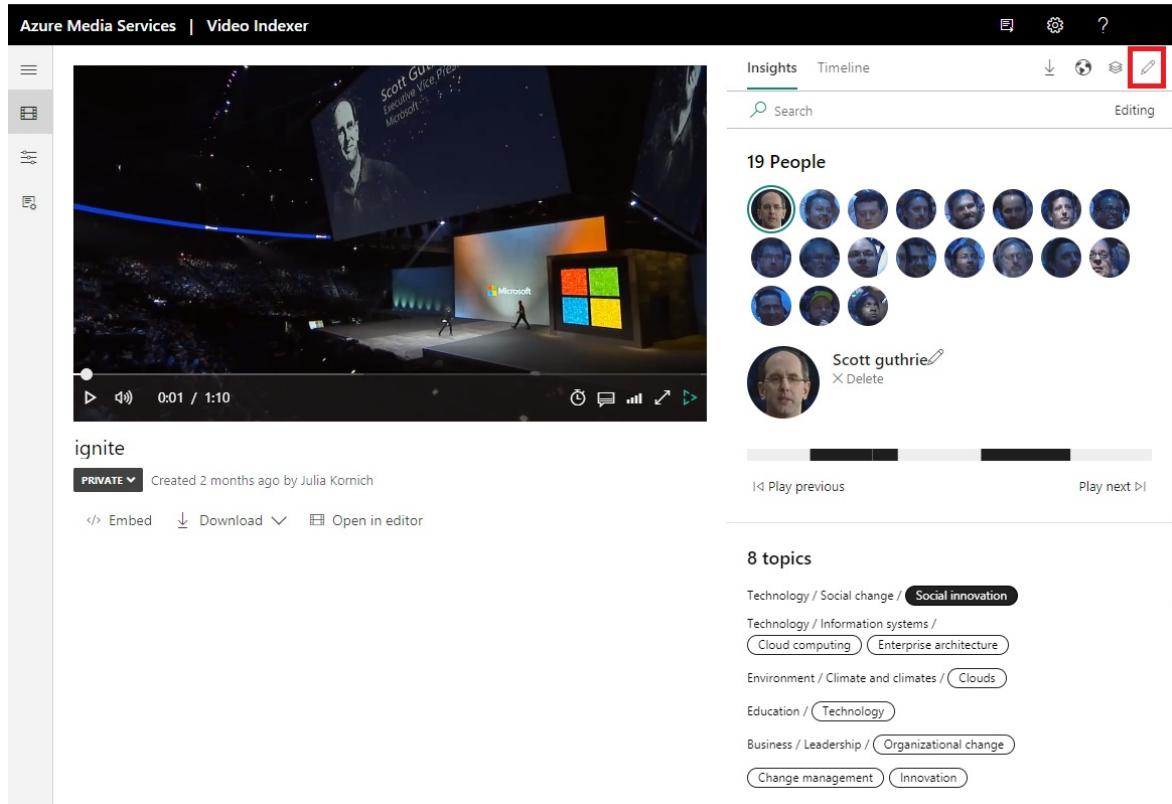
If you don't assign a Person model to the video during upload, your edit is saved in your account's Default person model.

Edit a face

NOTE

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

1. Browse to the Azure 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 select the pencil icon on the top-right corner of the window.



4. Select 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, select the check icon next to the new name. This action 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, the detected face images from this video of that person will merge with what already exists in the model. If you name a face with a new name, a new Person entry is created in the Person model that the video is using.

Delete a face

To delete a detected face in your video, go to the Insights pane and select the pencil icon in the top-right corner of the pane. Select the **Delete** option underneath the name of the face. This action removes the 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've been indexed.

The person, if they 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.

Optimize the ability of your model to recognize a person

To optimize your model ability to recognize the person, upload as many different images as possible and from different angles. To get optimal results, use high resolution images.

Next steps

[Customize Person model using APIs](#)

Customize a Person model with the Azure Video Indexer API

9/22/2022 • 4 minutes to read • [Edit Online](#)

IMPORTANT

Face identification, customization and celebrity recognition features access is limited based on eligibility and usage criteria in order to support our Responsible AI principles. Face identification, customization and celebrity recognition features are only available to Microsoft managed customers and partners. Use the [Face Recognition intake form](#) to apply for access.

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

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

Managing multiple Person models

Azure Video Indexer supports multiple Person models per account. This feature is currently available only through the Azure 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, and so on).

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 don't need the multiple Person model support, don't assign a Person model ID to your video when uploading/indexing or reindexing. In this case, Azure Video Indexer uses the default custom Person model in your account.

Create a new Person model

To create a new Person model in the specified account, use the [create a person model](#) API.

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

Delete a Person model

To delete a custom Person model from the specified account, use the [delete a person model](#) API.

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 won't be recognized by Azure Video Indexer in your current videos that were indexed using that model but the 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.

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

Get all Person models

To get all Person models in the specified account, use the [get a person model](#) API.

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 action 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.

The system then 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 Azure 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.

To update the face, use the [update a video face](#) API.

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

Next steps

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

Customize a Brands model with the Azure Video Indexer website

9/22/2022 • 3 minutes to read • [Edit Online](#)

Azure 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 video or audio content or if it shows up in visual text in a video, Azure Video Indexer detects it as a brand in the content.

A custom Brands model allows you to:

- select if you want Azure Video Indexer to detect brands from the Bing brands database.
- select if you want Azure Video Indexer to exclude certain brands from being detected (essentially creating a blocklist of brands).
- select if you want Azure Video Indexer to include brands that should be part of your model that might not be in Bing's brands database (essentially creating an accept list of brands).

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

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

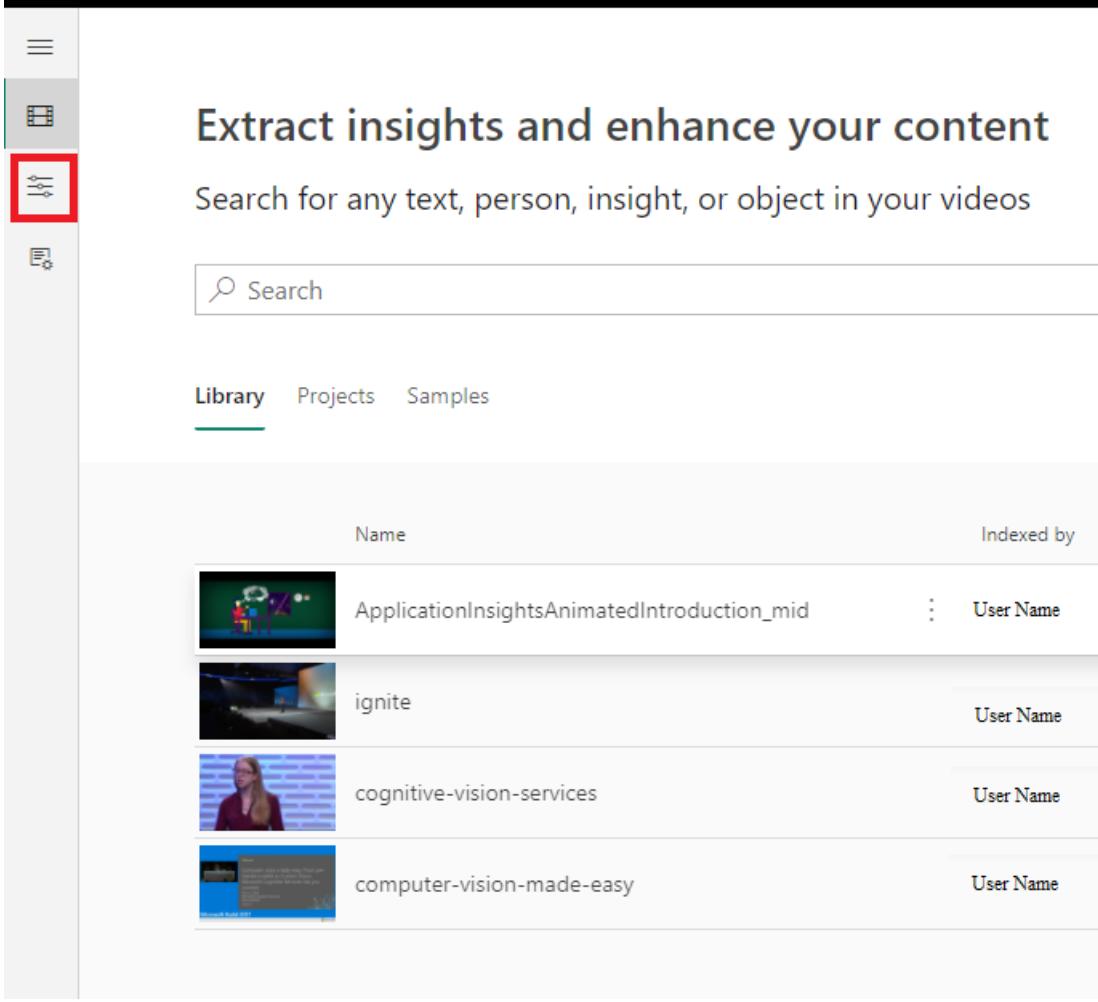
NOTE

If your video was indexed prior to adding a brand, you need to reindex it. You will find **Re-index** item in the drop-down menu associated with the video. Select **Advanced options -> Brand categories** and check **All brands**.

Edit Brands model settings

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

1. Go to the [Azure Video Indexer](#) website and sign in.
2. To customize a model in your account, select the **Content model customization** button on the left of the page.



Extract insights and enhance your content

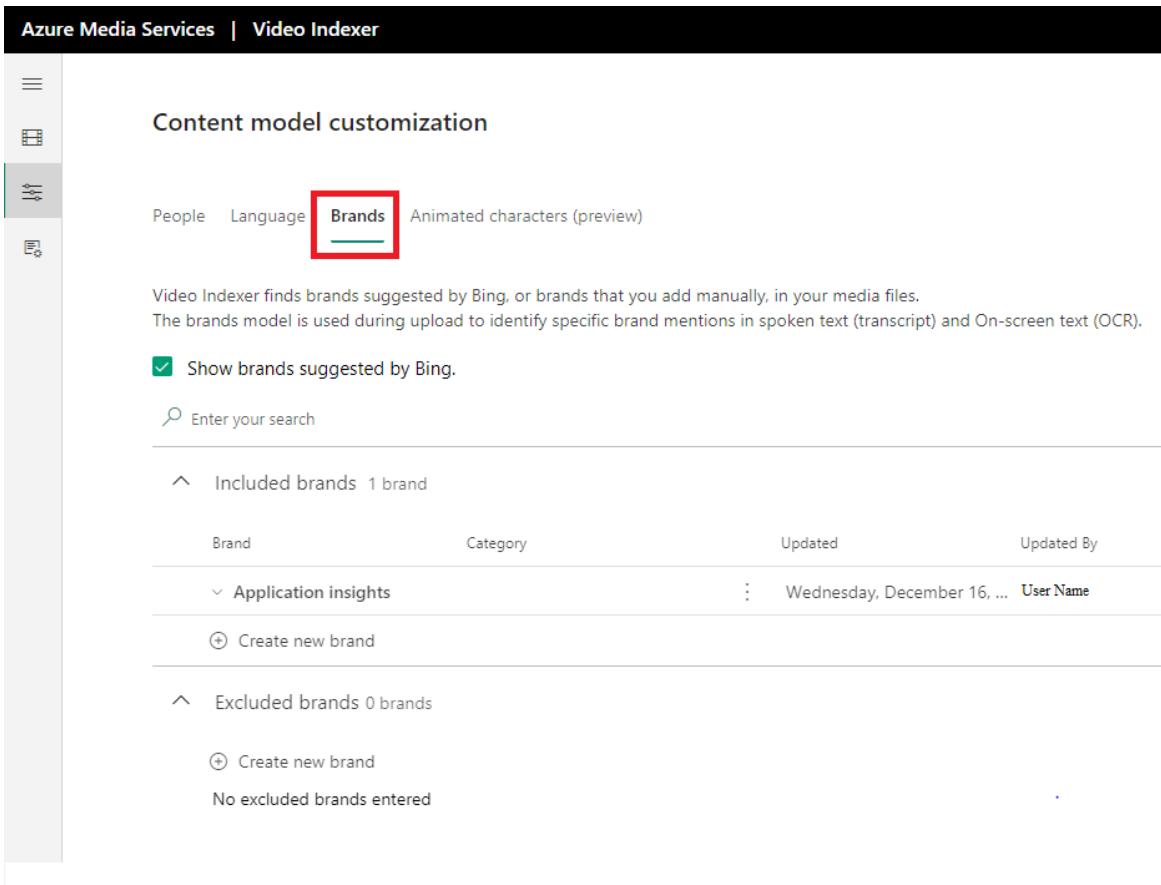
Search for any text, person, insight, or object in your videos

Search

Library Projects Samples

Name	Indexed by
ApplicationInsightsAnimatedIntroduction_mid	User Name
ignite	User Name
cognitive-vision-services	User Name
computer-vision-made-easy	User Name

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



Content model customization

People Language **Brands** Animated characters (preview)

Video Indexer finds brands suggested by Bing, or brands that you add manually, in your media files. The brands model is used during upload to identify specific brand mentions in spoken text (transcript) and On-screen text (OCR).

Show brands suggested by Bing.

Enter your search

^ Included brands 1 brand

Brand	Category	Updated	Updated By
Application insights		Wednesday, December 16, ...	User Name

+ Create new brand

^ Excluded brands 0 brands

+ Create new brand

No excluded brands entered

4. Check the **Show brands suggested by Bing** option if you want Azure Video Indexer to detect brands suggested by Bing—leave the option unchecked if you don't.

Include brands in the model

The **Include brands** section represents custom brands that you want Azure Video Indexer to detect, even if they aren't suggested by Bing.

Add a brand to include list

1. Select **+ Create new brand**.

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 Azure 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 (like a link to its Wikipedia page).

2. Select **Save** and you'll see that the brand has been added to the **Include brands** list.

Edit a brand on the include list

1. Select 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. Select the **Update** button to update the brand with the new information.

Delete a brand on the include list

1. Select the trash icon next to the brand that you want to delete.
2. Select **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 don't want Azure Video Indexer to detect.

Add a brand to exclude list

1. Select **+ Create new brand**.

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

2. Select **Save** and you'll see that the brand has been added to the **Exclude brands** list.

Edit a brand on the exclude list

1. Select 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. Select the **Update** button to update the brand with the new information.

Delete a brand on the exclude list

1. Select the trash icon next to the brand that you want to delete.
2. Select **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 Azure Video Indexer API

9/22/2022 • 4 minutes to read • [Edit Online](#)

Azure 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 video or audio content or if it shows up in visual text in a video, Azure 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 more information, see [Overview](#).

NOTE

If your video was indexed prior to adding a brand, you need to reindex it.

You can use the Azure 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 Azure Video Indexer website, as described in [Customize Brands model using the Azure Video Indexer website](#).

Create a Brand

The [create a brand](#) API creates a new custom brand and adds it to the custom Brands model for the specified account.

NOTE

Setting `enabled` (in the body) to true puts the brand in the *Include* list for Azure Video Indexer to detect. Setting `enabled` to false puts the brand in the *Exclude* list, so Azure Video Indexer won't detect it.

Some other parameters that you can set in the body:

- 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 tag shows up in the brand's *Category* field in the Azure 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

The [delete a brand](#) API 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.

Response

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

Get a specific Brand

The [get a brand](#) API lets you search for the details of a brand in the custom Brands model for the specified account using the brand ID.

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 Azure Video Indexer to detect, and `enabled` being false signifies that the brand is in the *Exclude* list, so Azure Video Indexer won't detect it.

Update a specific brand

The [update a brand](#) API lets you search for the details of a brand in the custom Brands model for the specified account using the brand ID.

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": "2018-04-25T14:59:52.7433333",  
    "lastModified": "2018-04-25T15:37:50.67",  
    "enabled": false,  
    "description": "This is an update example",  
    "tags": [  
        "Tag1",  
        "NewTag2"  
    ]  
}
```

Get all of the Brands

The [get all brands](#) API 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.

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",  
        "accountId": "AccountId",  
        "lastModifierUserName": "UserName",  
        "Created": "2018-04-26T14:59:52.7433333",  
        "LastModified": "2018-04-26T14:59:52.7433333",  
        "enabled": false,  
        "description": "This is another example",  
        "tags": ["Tag1", "Tag2"]  
    },  
]
```

NOTE

The brand named *Example* is in the *Include* list for Azure Video Indexer to detect, and the brand named *Example2* is in the *Exclude* list, so Azure Video Indexer won't detect it.

Get Brands model settings

The [get brands settings](#) API 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 aren't enabled, Azure Video Indexer will only detect brands from the custom Brands model of the specified account.

Response

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

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

NOTE

The `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 because it has been deprecated.

Update Brands model settings

The [update brands](#) API 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 aren't enabled, Azure Video Indexer will only detect brands from the custom Brands model of the specified account.

The `useBuiltIn` flag set to true means that Bing brands are enabled. If `useBuiltin` is false, Bing brands are disabled.

Response

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

Next steps

[Customize Brands model using website](#)

Customize a Language model with the Azure Video Indexer website

9/22/2022 • 5 minutes to read • [Edit Online](#)

Azure 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 Azure Video Indexer](#).

You can use the Azure 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. Go to the [Azure Video Indexer](#) website and sign in.
2. To customize a model in your account, select the **Content model customization** button on the left of the page.

Content model customization

People **Language** Brands Animated characters (preview)

Teach Video Indexer to recognize specific vocabulary or industry terms. Upload a .txt, .utt, .ttml, or .srt file up to 500 KB, and Video Indexer will apply the new language models when videos are uploaded and indexed. [Learn more](#)

Search language models...

- English 0 trained models [Create new model](#)
- German 0 trained models [Create new model](#)
- Spanish 0 trained models [Create new model](#)
- Arabic 0 trained models [Create new model](#)
- French 0 trained models [Create new model](#)
- Hindi 0 trained models [Create new model](#)
- Italian 0 trained models [Create new model](#)

3. Select the **Language** tab.

You see a list of supported languages.

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

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

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

6. To add a text file, select **Add file**. Your file explorer will open.

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 selecting the ... button on the right side of the Language model and selecting **Add file**.

8. Once you're done uploading the text files, select the green **Train** option.

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.

Content model customization

People Language Brands Animated characters (preview)

Teach Video Indexer to recognize specific vocabulary or industry terms.

Upload a .txt, .utt, .ttml, or .srt file up to 500 KB, and Video Indexer will apply the new language models when videos are uploaded and indexed. [Learn more](#)

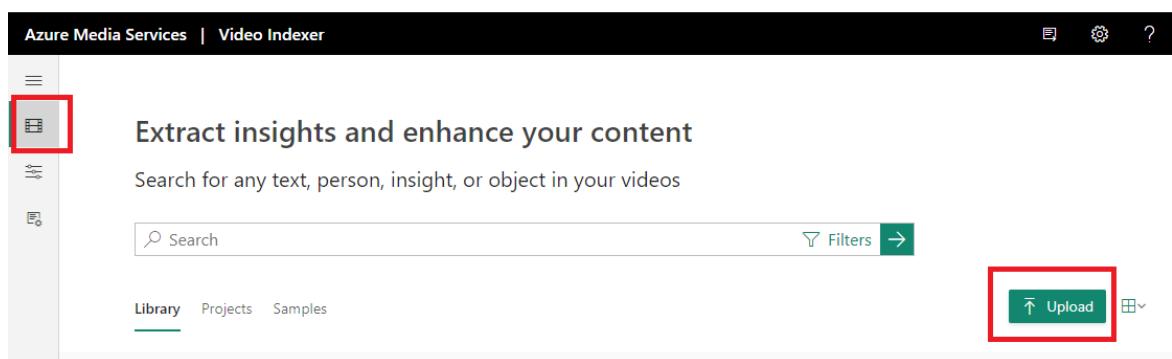
The screenshot shows the 'Language' section of the Azure Media Services Video Indexer interface. At the top, there's a search bar labeled 'Search language models...'. Below it, a list of trained models is shown under three categories: English, German, and Spanish. Each category has a 'Create new model' link. The English category has one trained model named 'Account adaptations'.

- ^ English 1 trained models
 - ^ Account adaptations : ✓ Trained
 - ⊕ Create new model
- ^ German 0 trained models
 - ⊕ Create new model
- ^ Spanish 0 trained models
 - ⊕ Create new model

Using a Language model on a new video

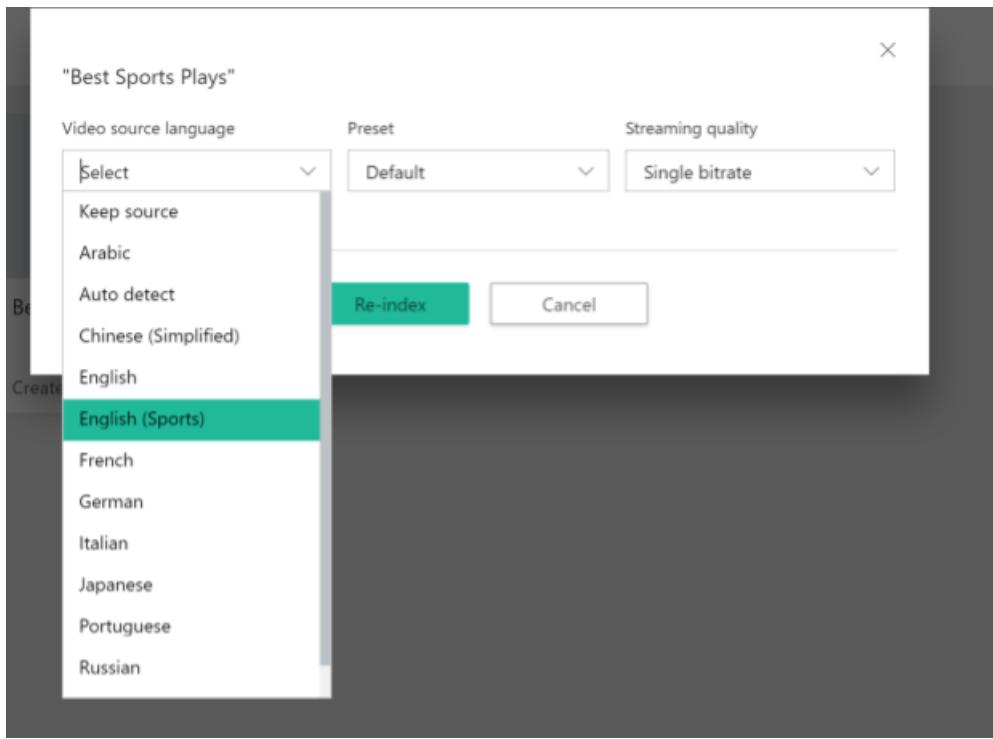
To use your Language model on a new video, do one of the following actions:

- Select the **Upload** button on the top of the page.



- Drop your audio or video file or browse for your file.

You're given the option to select the **Video source language**. Select 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. For example:



Select 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, follow these steps:

1. Sign in to the [Azure Video Indexer](#) home page.
2. Click on ... button on the video and select **Re-index**.
3. You're given the option to select the **Video source language** to reindex your video with. Select the dropdown 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.
4. Select 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'll have to train the model again by selecting the green **Train** option.

Rename the Language model

You can change the name of the Language model by selecting the ellipsis (...) button 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, select **Add file**. Your file explorer will open.

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 selecting the ellipsis (...) button on the right side of the Language model and selecting **Add file**.

Delete files

To delete a file from the Language model, select the ellipsis (...) button on the right side of the text file and select

Delete. A new window pops up telling you that the deletion can't be undone. Select 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, select the ellipsis (...) button on the right side of the Language model and select **Delete**.

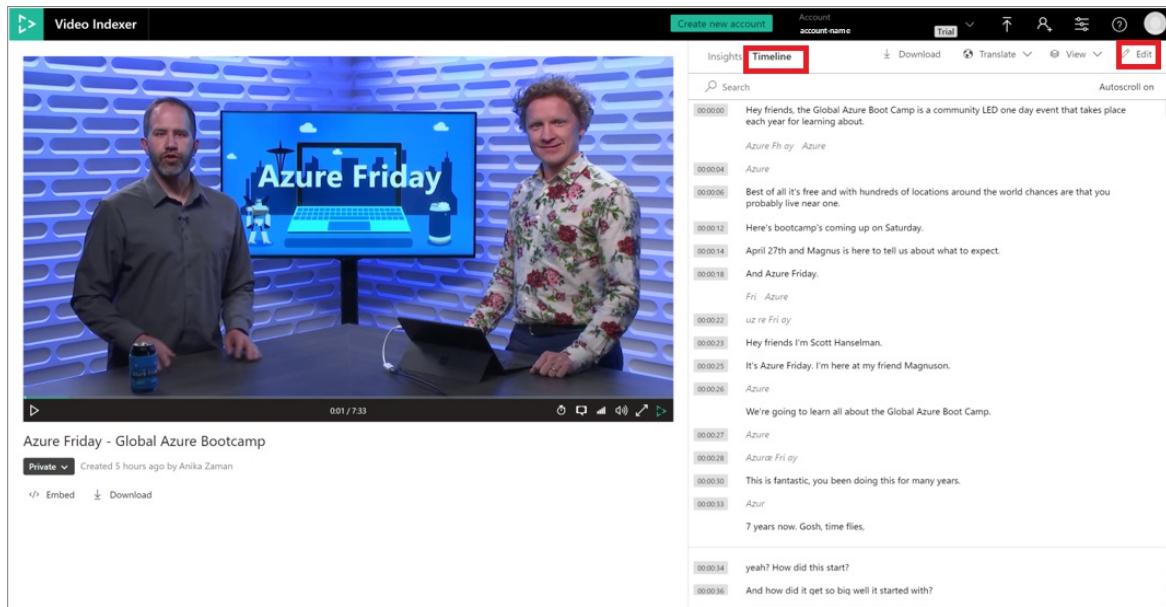
A new window pops up telling you that the deletion can't be undone. Select the **Delete** option in the new window.

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 reindex the video. If you reindex the video, you can assign a new Language model to the video. Otherwise, Azure Video Indexer will use its default model to reindex the video.

Customize Language models by correcting transcripts

Azure 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 Azure Video Indexer web interface. At the top, there's a navigation bar with 'Create new account' and 'Account' (account-name). Below the navigation is a toolbar with 'Trial', 'Download', 'Translate', 'View', and an 'Edit' button, which has a red box around it. The main area is divided into two sections: 'Insights' (disabled) and 'Timeline'. The 'Timeline' tab is active and highlighted with a red box. On the left, there's a video player showing two men in a studio setting. The video title is 'Azure Friday - Global Azure Bootcamp'. Below the video player, there are buttons for 'Private', 'Created 5 hours ago by Anika Zaman', 'Embed', and 'Download'. On the right, the 'Timeline' pane displays a list of transcription entries with timestamps and text. The first few entries are: 'Hey friends, the Global Azure Boot Camp is a community LED one day event that takes place each year for learning about.' (timestamp 00:00:00), 'Azure Friday - Azure' (00:00:04), 'Best of all it's free and with hundreds of locations around the world chances are that you probably live near one.' (00:00:06), 'Here's bootcamp's coming up on Saturday.' (00:00:12), 'April 27th and Magnus is here to tell us what to expect.' (00:00:14), 'And Azure Friday.' (00:00:18), 'Fri - Azure' (00:00:22), 'Azure Fri ay' (00:00:22), 'Hey friends I'm Scott Hanselman.' (00:00:23), 'It's Azure Friday. I'm here at my friend Magnuson.' (00:00:25), 'Azure' (00:00:26), 'We're going to learn all about the Global Azure Boot Camp.' (00:00:27), 'Azure' (00:00:28), 'Azure Fri ay' (00:00:28), 'This is fantastic, you been doing this for many years.' (00:00:30), 'Azur' (00:00:31), '7 years now. Gosh, time flies.' (00:00:32), 'yeah? How did this start?' (00:00:34), and 'And how did it get so big well it started with?' (00:00:36).

2. Select the pencil icon to edit the transcript of your transcription.

The screenshot shows the Azure Video Indexer interface with the 'Timeline' tab selected. A search bar at the top right says 'Edit is on'. Below it, a transcript of a video is displayed with timestamp markers. The transcript includes several lines of text with corresponding timestamp markers:

- 00:00:00 Hey friends, the Global Azure Boot Camp is a community led one day event that takes place each year for learning about.
- 00:00:04 *Azure Friday* *Azure*
- 00:00:06 Best of all, it's free and with hundreds of locations around the world chances are that you probably live near one.
- 00:00:12 Here's bootcamp's coming up on Saturday.
- 00:00:14 April 27th and Magnus is here to tell us about what to expect today
- 00:00:18 on **Azure Friday.**

A callout box is overlaid on the timeline, centered over the last line. It contains the following text:
Changes you make here will be added to your language model as a 'From transcript edits' file.
 Don't show this again

Azure 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 retrain the specific Language model that was used to index this video.

The edits that were done in the [widget's](#) timeline are also included.

If you didn't specify a Language model when indexing this video, 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. Corrections that don't involve actual words (for example, punctuation marks or spaces) aren't included.

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

To look at the "From transcript edits" file for each of your Language models, select 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 Azure Video Indexer API

9/22/2022 • 6 minutes to read • [Edit Online](#)

Azure 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 Azure Video Indexer](#).

You can use the Azure 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 Azure Video Indexer website](#).

Create a Language model

The [create a language model](#) API 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.

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

- Key will be the file name and value will be the txt file.
- 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 this 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 [train a language model](#) API 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 when creating the Language model or by updating the Language model.

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 this 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"
    }
  ]
}
```

The returned `id` is a unique ID used to distinguish between language models, while `languageModelId` is used both for [uploading a video to index](#) and [reindexing a video](#) APIs (also known as `linguisticModelId` in Azure Video Indexer upload/reindex APIs).

Delete a Language model

The [delete a language model](#) API 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 reindex the video. If you reindex the video, you can assign a new Language model to the video. Otherwise, Azure Video Indexer will use its default model to reindex the video.

Response

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

Update a Language model

The [update a Language model](#) API 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.

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

- Key will be the file name and value will be the txt file.
- 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 this 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"  
        }  
    ]  
}
```

Use the `id` of the files returned in the response to download the contents of the file.

Update a file from a Language model

The [update a file](#) allows you to update the name and `enable` state of a file in a custom Language model in the specified account.

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": "2018-04-27T20:10:10.5233333"  
}
```

Use the `id` of the file returned in the response to download the contents of the file.

Get a specific Language model

The [get](#) API returns information on the specified Language model in the specified account such as language and the files that are in the Language model.

Response

The response provides metadata on the specified Language model along with metadata on each of the model's files following the format of this 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"  
        }  
    ]  
}
```

Use the `id` of the file returned in the response to download the contents of the file.

Get all the Language models

The [get all](#) API returns all of the custom Language models in the specified account in a list.

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 this 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"
      }
    ]
  },
  {
    "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 [delete](#) API deletes the specified file from the specified Language model in the specified account.

Response

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

Get metadata on a file from a Language model

The [get metadata of a file](#) API returns the contents of and metadata on the specified file from the chosen Language model in your account.

Response

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

```
{
  "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 [download a file](#) API 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.

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](#)

Embed Azure Video Indexer widgets in your apps

9/22/2022 • 10 minutes to read • [Edit Online](#)

This article shows how you can embed Azure Video Indexer widgets in your apps. Azure Video Indexer supports embedding three types of widgets into your apps: *Cognitive Insights*, *Player*, and *Editor*.

Starting with version 2, the widget base URL includes the region of the specified account. For example, an account in the West US region generates: `https://www.videoindexer.ai/embed/insights/.../?location=westus2`.

Widget types

Cognitive Insights widget

A Cognitive Insights widget includes all visual insights that were extracted from your video indexing process.

The Cognitive Insights widget supports the following optional URL parameters:

NAME	DEFINITION	DESCRIPTION
<code>widgets</code>	Strings separated by comma	Allows you to control the insights that you want to render. Example: <code>https://www.videoindexer.ai/embed/insights/<accountId>/widgets=people,keywords</code> renders only people and keywords UI insights. Available options: people, animatedCharacters, keywords, audioEffects, labels, sentiments, emotions, topics, keyframes, transcript, ocr, speakers, scenes, spokenLanguage, observedPeople and namedEntities.
<code>controls</code>	Strings separated by comma	Allows you to control the controls that you want to render. Example: <code>https://www.videoindexer.ai/embed/insights/<accountId>/controls=search,download</code> renders only search option and download button. Available options: search, download, presets, language.
<code>language</code>	A short language code (language name)	Controls insights language. Example: <code>https://www.videoindexer.ai/embed/insights/<accountId>/language=es-es</code> or <code>https://www.videoindexer.ai/embed/insights/<accountId>/language=spanish</code>
<code>locale</code>	A short language code	Controls the language of the UI. The default value is <code>en</code> . Example: <code>locale=de</code> .
<code>tab</code>	The default selected tab	Controls the Insights tab that's rendered by default. Example: <code>tab=timeline</code> renders the insights with the Timeline tab selected.
<code>search</code>	String	Allows you to control the initial search term. Example: <code>https://www.videoindexer.ai/embed/insights/<accountId>/search=azure</code> renders the insights filtered by the word "azure".

NAME	DEFINITION	DESCRIPTION
<code>sort</code>	Strings separated by comma	<p>Allows you to control the sorting of an insight.</p> <p>Each sort consists of 3 values: widget name, property and order, connected with '_'. <code>sort=name_property_order</code></p> <p>Available options:</p> <ul style="list-style-type: none"> widgets: keywords, audioEffects, labels, sentiments, emotions, keyframes, scenes, namedEntities and spokenLanguage. property: startTime, endTime, seenDuration, name and id. order: asc and desc. <p>Example: <code>https://www.videoindexer.ai/embed/insights/<accountId></code> <code>sort=labels_id_asc,keywords_name_desc</code></p> <p>renders the labels sorted by id in ascending order and keywords sorted by name in descending order.</p>
<code>location</code>		<p>The <code>location</code> parameter must be included in the embedded links, see how to get the name of your region. If your account is in preview, the <code>trial</code> should be used for the location value. <code>trial</code> is the default value for the <code>location</code> parameter.</p>

Player widget

You can use the Player widget to stream video by using adaptive bit rate. The Player widget supports the following optional URL parameters.

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

NAME	DEFINITION	DESCRIPTION
<code>location</code>		The <code>location</code> parameter must be included in the embedded links, see how to get the name of your region . If your account is in preview, the <code>trial</code> should be used for the location value. <code>trial</code> is the default value for the <code>location</code> parameter.
<code>boundingBoxes</code>	Array of bounding boxes options: people (faces) and observed people. Values should be separated by a comma (",").	Controls the option to set bounding boxes on/off when embedding the player. All mentioned option will be turned on. Example: <code>boundingBoxes= observedPeople, people</code> Default value is <code>boundingBoxes= observedPeople</code> (only observed people bounding box are turned on).

Editor widget

You can use the Editor widget to create new projects and manage a video's insights. The Editor widget supports the following optional URL parameters.

NAME	DEFINITION	DESCRIPTION
<code>accessToken</code> *	String	Provides access to videos that are only in the account that's used to embed the widget. The Editor widget requires the <code>accessToken</code> parameter.
<code>language</code>	A language code	Controls the player language. The default value is <code>en-US</code> . Example: <code>language=de-DE</code> .
<code>locale</code>	A short language code	Controls the insights language. The default value is <code>en</code> . Example: <code>language=de</code> .
<code>location</code>		The <code>location</code> parameter must be included in the embedded links, see how to get the name of your region . If your account is in preview, the <code>trial</code> should be used for the location value. <code>trial</code> is the default value for the <code>location</code> parameter.

*The owner should provide `accessToken` with caution.

Embed videos

This section discusses embedding videos by [using the portal](#) or by [assembling the URL manually](#) into apps.

The `location` parameter must be included in the embedded links, see [how to get the name of your region](#). If your account is in preview, the `trial` should be used for the location value. `trial` is the default value for the `location` parameter. For example:

```
https://www.videoindexer.ai/accounts/00000000-0000-0000-000000000000/videos/b2b2c74b8e/?location=trial
```

The portal experience

To embed a video, use the portal as described below:

1. Sign in to the [Azure Video Indexer](#) website.
2. Select the video that you want to work with and press **Play**.
3. Select the type of widget that you want (**Cognitive Insights**, **Player**, or **Editor**).

4. Click </> Embed.
5. Copy the embed code (appears in **Copy the embedded code** in the **Share & Embed** dialog).
6. Add the code to your app.

NOTE

Sharing a link for the **Player** or **Insights** widget will include the access token and grant the read-only permissions to your account.

Assemble the URL manually

Public videos

You can embed public videos assembling the URL as follows:

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

Private videos

To embed a private video, you must pass an access token (use [Get Video Access Token](#) in the `src` attribute of the iframe:

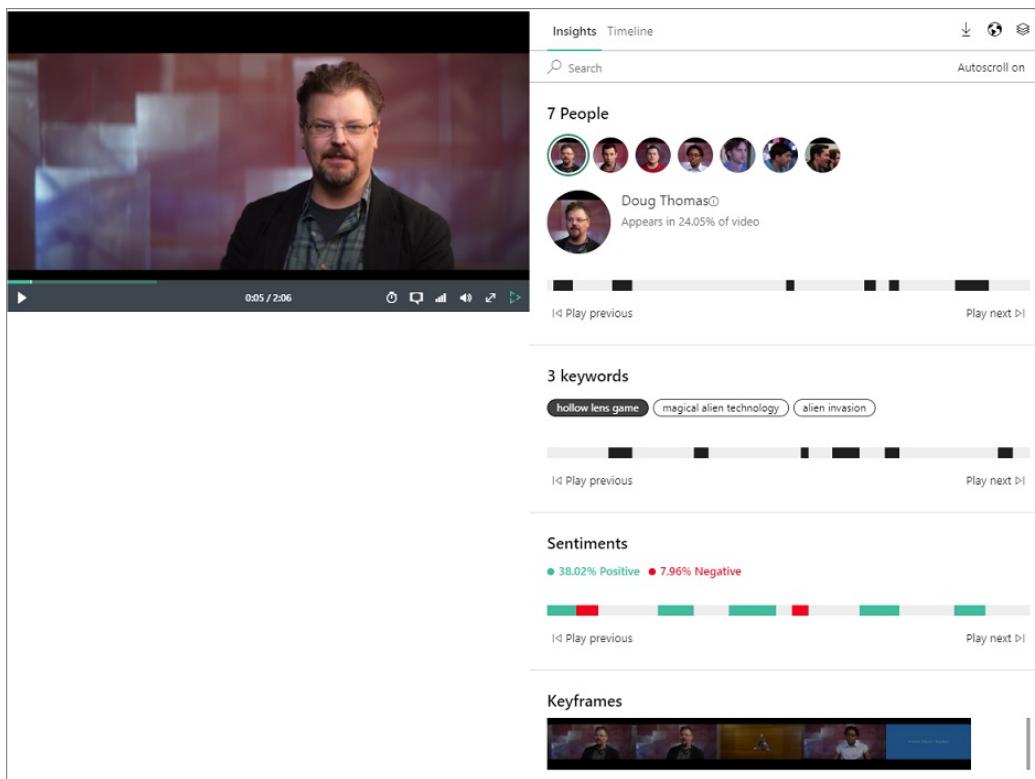
```
https://www.videoindexer.ai/embed/[insights | player]/<accountId>/<videoId>/?accessToken=<accessToken>
```

Provide editing insights capabilities

To provide editing insights capabilities in your embedded widget, you must pass an access token that includes editing permissions. Use [Get Video Access Token](#) with `&allowEdit=true`.

Widgets interaction

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



Flow overview

When you edit the transcripts, the following flow occurs:

1. You edit the transcript in the timeline.
2. Azure Video Indexer gets these updates and saves them in the [from transcript edits](#) in the language model.
3. The captions are updated:
 - If you are using Azure Video Indexer's player widget - it's automatically updated.

- If you are using an external player - you get a new captions file user the **Get video captions** call.

Cross-origin communications

To get Azure Video Indexer widgets to communicate with other components:

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

If you implement your own player code and integrate with Cognitive Insights widgets, it's your responsibility to validate the origin of the message that comes from VideoIndexer.ai.

Embed widgets in your app or blog (recommended)

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

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 selects the insight control on your app, the player jumps to the relevant moment.

For more information, see the [Azure Video Indexer - Embed both Widgets 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 by using the [AMP plug-in](#).

1. Add an Azure Video Indexer plug-in 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 Azure Video Indexer plug-in.

```
// Init the source.
function initSource() {
    var tracks = [
        {
            kind: 'captions',
            // To load vtt from VI, 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-
fabafab6c2206a2/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 plug-in.
    this.videobreakdown({
        videoId: "c4c1ad4c9a",
        syncTranscript: true,
        syncLanguage: true,
        location: "trial" /* location option for paid accounts (default is trial) */
    });

    // Set the source dynamically.
    initSource.call(this);
});
```

3. Copy the Cognitive Insights embed code.

You can now communicate with Azure Media Player.

For more information, see the [Azure Media Player + VI Insights demo](#).

Embed the Azure Video Indexer Cognitive Insights widget and use a different video player

If you use a video player other than Azure Media Player, you must manually manipulate the video player 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 the event comes from the videoindexer domain.
    if ((origin === "https://www.videoindexer.ai") && evt.data.time !== undefined){

      // Call your player's "jumpTo" implementation.
      playerInstance.currentTime = evt.data.time;

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

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

})()

</script>
```

For more information, see the [Azure Media Player + VI Insights demo](#).

Adding subtitles

If you embed Azure Video Indexer insights with your own [Azure Media Player](#), you can use the `GetVttUrl` method to get closed captions (subtitles). You can also call a JavaScript method from the Azure Video Indexer AMP plug-in `getSubtitlesUrl` (as shown earlier).

Customizing embeddable widgets

Cognitive Insights widget

You can choose the types of insights that you want. To do this, specify them as a value to the following URL parameter that's added to the embed code that you get (from the API or from the web app):

`&widgets=<list of wanted widgets>`.

The possible values are: `people` , `animatedCharacters` , `keywords` , `labels` , `sentiments` , `emotions` , `topics` , `keyframes` , `transcript` , `ocr` , `speakers` , `scenes` , and `namedEntities` .

For example, if you want to embed a widget that contains only people and keywords insights, the iframe embed

URL will look like this:

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

The title of the iframe window can also be customized by providing `&title=<YourTitle>` to the iframe URL. (It customizes 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 Azure 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, Azure Video Indexer player has autogenerated closed captions that are based on the transcript of the video. The transcript is 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 Code>` to the embed player URL. 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=en-us
```

Autoplay

By default, the player will start playing the video. You can choose not to by passing `&autoplay=false` to the preceding embed URL.

Code samples

See the [code samples](#) repo that contains samples for Azure Video Indexer API and widgets:

FILE/FOLDER	DESCRIPTION
<code>azure-media-player</code>	Load an Azure Video Indexer video in a custom Azure Media Player.
<code>azure-media-player-vi-insights</code>	Embed VI Insights with a custom Azure Media Player.
<code>control-vi-embedded-player</code>	Embed VI Player and control it from outside.
<code>custom-index-location</code>	Embed VI Insights from a custom external location (can be customer a blob).
<code>embed-both-insights</code>	Basic usage of VI Insights both player and insights.
<code>embed-insights-with-AMP</code>	Embed VI Insights widget with a custom Azure Media Player.
<code>customize-the-widgets</code>	Embed VI widgets with customized options.
<code>embed-both-widgets</code>	Embed VI Player and Insights and communicate between them.
<code>url-generator</code>	Generates widgets custom embed URL based on user-specified options.
<code>html5-player</code>	Embed VI Insights with a default HTML5 video player.

Supported browsers

For more information, see [supported browsers](#).

Embed and customize Azure Video Indexer widgets in your app using npm package

Using our [@azure/video-analyzer-for-media-widgets](#) NPM package, you can add the insights widgets to your app and customize it according to your needs.

Instead of adding an iframe element to embed the insights widget, with this new package you can easily embed & communicate between our widgets. Customizing your widget is only supported in this package - all in one place.

For more information, see our official [GitHub](#).

Next steps

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

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

Logic Apps connector with ARM-based AVI accounts

9/22/2022 • 9 minutes to read • [Edit Online](#)

Azure Video Indexer (AVI) REST API supports both server-to-server and client-to-server communication. The API enables you to integrate video and audio insights into your application logic. To make the integration easier, we support Logic Apps and Power Automate connectors that are compatible with the Azure Video Indexer API.

You can use the connectors to set up custom workflows to effectively index and extract insights from a large amount of video and audio files, without writing a single line of code. Furthermore, using the connectors for the integration gives you better visibility on the health of your workflow and an easy way to debug it.

TIP

If you are using a classic AVI account, see [Logic Apps connector with classic-based AVI accounts](#).

Get started with the Azure Video Indexer connectors

To help you get started quickly with the Azure Video Indexer connectors, the example in this article creates Logic App flows. The Logic App and Power Automate capabilities and their editors are almost identical, thus the diagrams and explanations are applicable to both. The example in this article is based on the ARM AVI account. If you're working with a classic account, see [Logic App connectors with classic-based AVI accounts](#).

The "upload and index your video automatically" scenario covered in this article is composed of two different flows that work together. The "two flow" approach is used to support async upload and indexing of larger files effectively.

- The first flow is triggered when a blob is added or modified in an Azure Storage account. It uploads the new file to Azure Video Indexer with a callback URL to send a notification once the indexing operation completes.
- The second flow is triggered based on the callback URL and saves the extracted insights back to a JSON file in Azure Storage.

The logic apps that you create in this article, contain one flow per app. The second section ("Create a second flow - JSON extraction") explains how to connect the two. The second flow stands alone and is triggered by the first one (the section with the callback URL).

Prerequisites

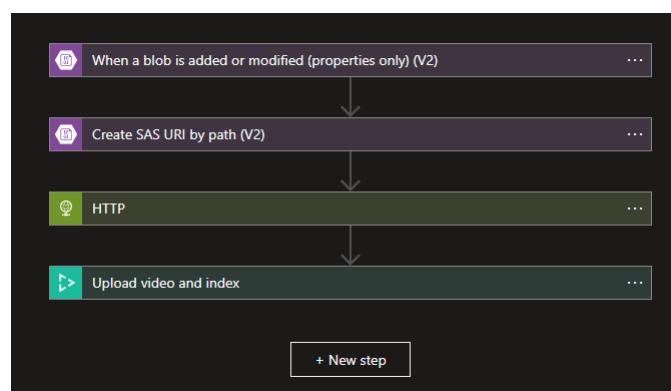
- If you don't have an [Azure subscription](#), create an [Azure free account](#) before you begin.
- Create an ARM-based [Azure Video Indexer account](#).
- Create an Azure Storage account. Keep note of the access key for your Storage account.

Create two containers: one to store the media files, second to store the insights generated by Azure Video Indexer. In this article, the containers are `videos` and `insights`.

Set up the first flow - file upload

In this section you'll, you create the following flow. The first flow is triggered when a blob is added or modified in an Azure Storage account. It uploads the new file to Azure Video Indexer with a callback URL to send a notification once the indexing operation completes.

The following image shows the first flow:



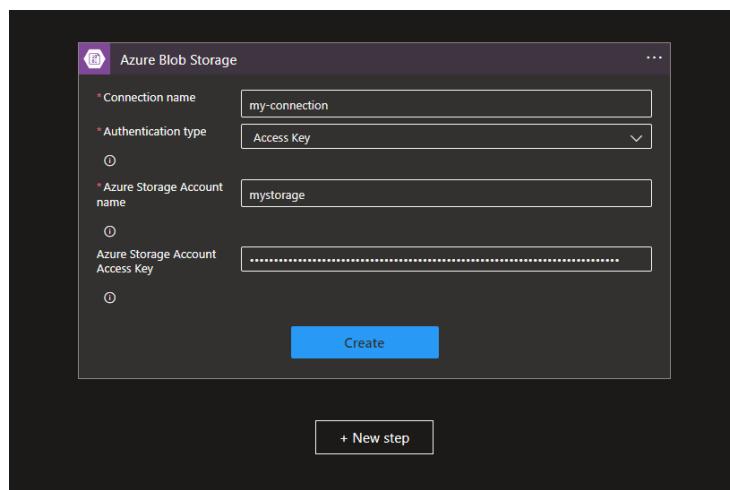
1. Create the [Logic App](#). We create a Logic App in the same region as the Azure Video Indexer region (recommended but not required). We call the logic app `UploadIndexVideosApp`.
 - a. Select **Consumption** for **Plan type**.
 - b. Press **Review + Create** -> **Create**.
 - c. Once the Logic App deployment is complete, in the Azure portal, go to the newly created Logic App.
 - d. Under the **Settings** section, on the left side's panel, select the **Identity** tab.
 - e. Under **System assigned**, change the **Status** from **Off** to **On** (the step is important for later on in this tutorial).
 - f. Press **Save** (on the top of the page).
 - g. Select the **Logic app designer** tab, in the pane on the left.

- h. Pick a **Blank Logic App** flow.
- i. Search for "blob".
- j. In the **All** tab, choose the **Azure Blob Storage** component.
- k. Under **Triggers**, select the **When a blob is added or modified (properties only) (V2)** trigger.
2. Set the storage connection.

After creating a **When a blob is added or modified (properties only) (V2)** trigger, the connection needs to be set to the following values:

KEY	VALUE
Connection name	<Name your connection>.
Authentication type	Access Key
Azure Storage Account name	<Storage account name where media files are going to be stored>.
Azure Storage Account Access Key	To get access key of your storage account: in the Azure portal -> my-storage -> under Security + networking -> Access keys -> copy one of the keys.

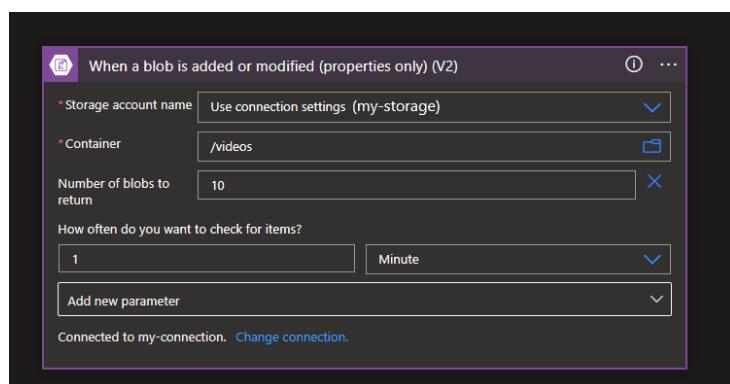
Select **Create**.



After setting the connection to the storage, it's required to specify the blob storage container that is being monitored for changes.

KEY	VALUE
Storage account name	Storage account name where media files are stored
Container	/videos

Select **Save** -> **+ New step**



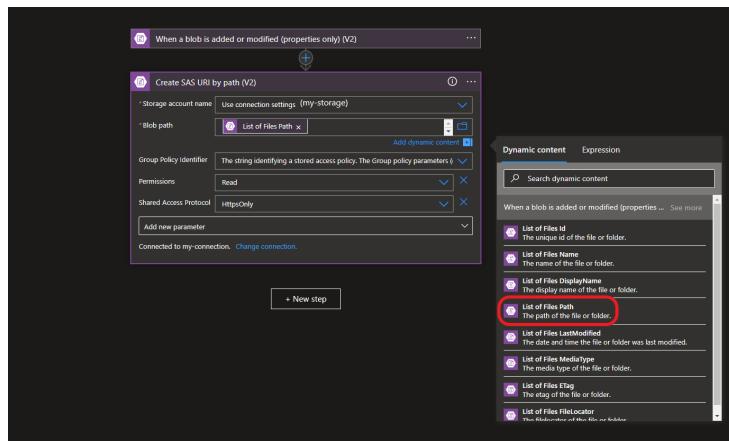
3. Create SAS URI by path action.

- a. Select the **Action** tab.
- b. Search for and select **Create SAS URI by path (V2)**.

KEY	VALUE
Storage account name	<The storage account name where media files are stored>.
Blob path	Under Dynamic content , select List of Files Path
Group Policy Identifier	Leave the default value.

KEY	VALUE
Permissions	Read
Shared Access protocol (appears after pressing Add new parameter)	HttpsOnly

Select **Save** (at the top of the page).



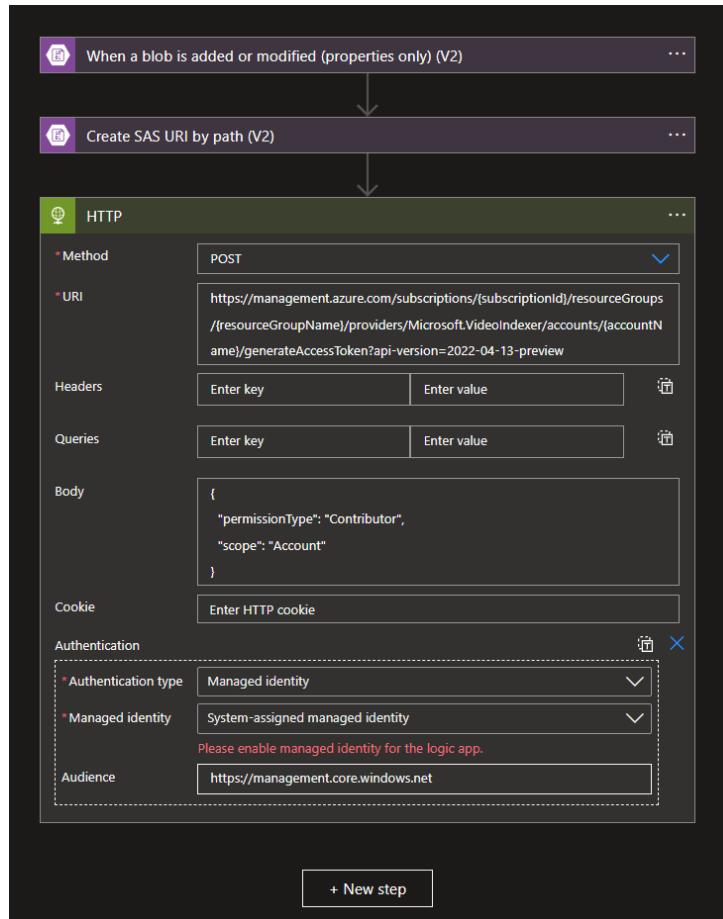
Select **+ New Step**.

4. Generate an access token.

NOTE
For details about the ARM API and the request/response examples, see [Generate an Azure Video Indexer access token](#).
Press **Try it** to get the correct values for your account.

Search and create an HTTP action.

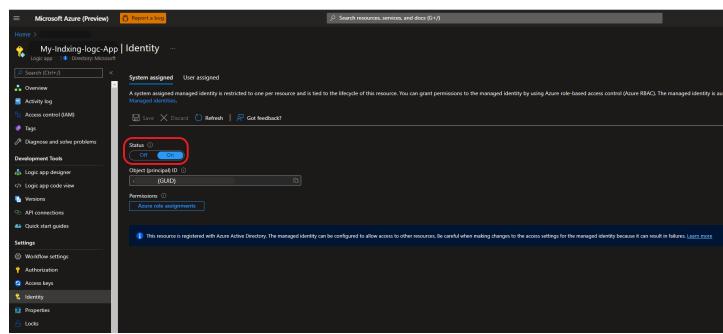
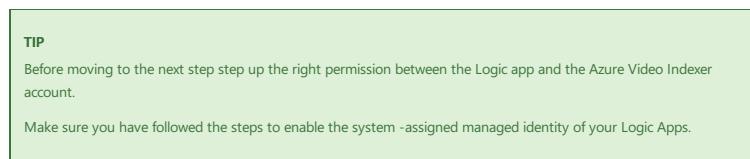
KEY	VALUE
Method	POST
URI	<code>https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers</code> <code>api-version={API-version}</code>
Body	<code>{ "permissionType": "Contributor", "scope": "Account" }</code>
Add new parameter	Authentication



After the **Authentication** parameter is added, fill the required parameters according to the table below:

KEY	VALUE
Authentication type	Managed identity
Managed identity	System-assigned managed identity
Audience	https://management.core.windows.net

Select **Save**.



- Set up system assigned managed identity for permission on Azure Video Indexer resource.

In the Azure portal, go to your Azure Video Indexer resource/account.

- On the left side blade, and select **Access control**.
- Select **Add** -> **Add role assignment** -> **Contributor** -> **Next** -> **User, group, or service principal** -> **+Select members**.
- Under **Members**, search for the Logic Apps name you created (in this case, `UploadIndexVideosApp`).
- Press **Select**.
- Press **Review + assign**.

- Back in your Logic App, create an **Upload video and index** action.

- Select **Video Indexer(V2)**.

b. From Video Indexer(V2) chose **Upload Video and index**.

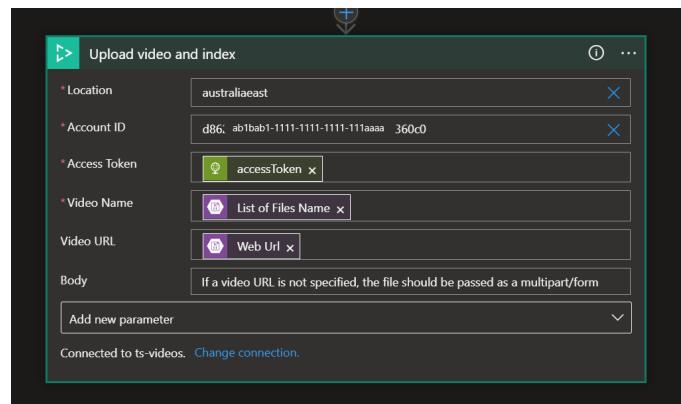
c. Set the connection to the Video Indexer account.

KEY	VALUE
Connection name	<Enter a name for the connection>, in this case <code>aviconnection</code> .
API key	This is your personal API key, which is available under Profile in the developer portal

Select **Create**.

d. Fill **Upload video and index** action parameters.

TIP	
If the AVI Account ID cannot be found and isn't in the drop-down, use the custom value.	
KEY	VALUE
Location	Location of the associated the Azure Video Indexer account.
Account ID	Account ID of the associated Azure Video Indexer account. You can find the Account ID in the Overview page of your account, in the Azure portal. Or, the Account settings tab, left of the Azure Video Indexer website .
Access Token	Use the <code>body('HTTP')['accessToken']</code> expression to extract the access token in the right format from the previous HTTP call.
Video Name	Select List of Files Name from the dynamic content of When a blob is added or modified action.
Video URL	Select Web Url from the dynamic content of Create SAS URI by path action.
Body	Can be left as default.

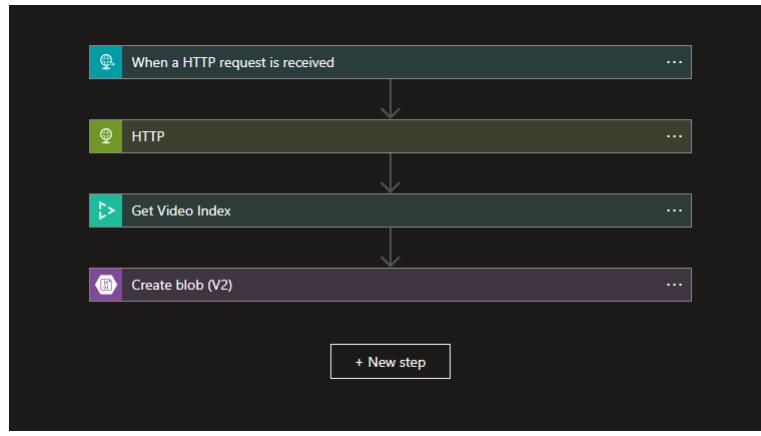


Select **Save**.

The completion of the uploading and indexing from the first flow will send an HTTP request with the correct callback URL to trigger the second flow. Then, it will retrieve the insights generated by Azure Video Indexer. In this example, it will store the output of your indexing job in your Azure Storage. However, it's up to you what you do with the output.

Create a second flow - JSON extraction

Create the second flow, Logic Apps of type consumption. Thesecond flowis triggered based on the callback URL and saves the extracted insights back to a JSON file in Azure Storage.



1. Set up the trigger

Search for the **When an HTTP request is received**.

The screenshot shows the Logic Apps Designer interface with the 'HTTP' category selected under 'Triggers'. The 'When a HTTP request is received' trigger is highlighted with a red box.

For the trigger, we'll see an HTTP POST URL field. The URL won't be generated until after you save your flow; however, you'll need the URL eventually.

TIP

We will come back to the URL created in this step.

2. Generate an access token.

Follow all the steps from:

- Generate an access token we did for the first flow.
- Select Save -> + New step.

3. Get Video Indexer insights.

- Search for "Video Indexer".
- From **Video Indexer(V2)** chose **Get Video Index** action.

Set the connection name:

KEY	VALUE
Connection name	<A name for connection>. For example, <code>aviconnection</code> .
API key	This is your personal API key, which is available under Profile at the developer portal . For more information, see Subscribe to the API .

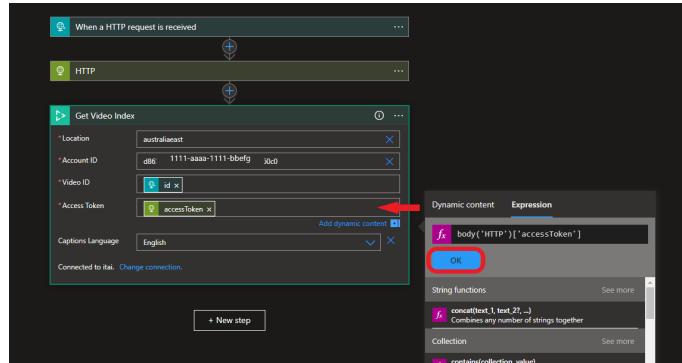
- Select **Create**.

c. Fill out the required parameters according to the table:

KEY	VALUE
Location	The Location of the Azure Video Indexer account.
Account ID	The Video Indexer account ID can be copied from the resource/account Overview page in the Azure portal.
Video ID*	For Video ID, add dynamic content of type Expression and put in the following expression: <code>triggerOutputs()['queries']['id']</code> .
Access Token	From the dynamic content, under the Parse JSON section select the accessToken that is the output of the parse JSON action.

*This expression tells the connector to get the Video ID from the output of your trigger. In this case,

the output of your trigger will be the output of **Upload video** and **index** in your first trigger.



Select **Save** -> **+ New step**.

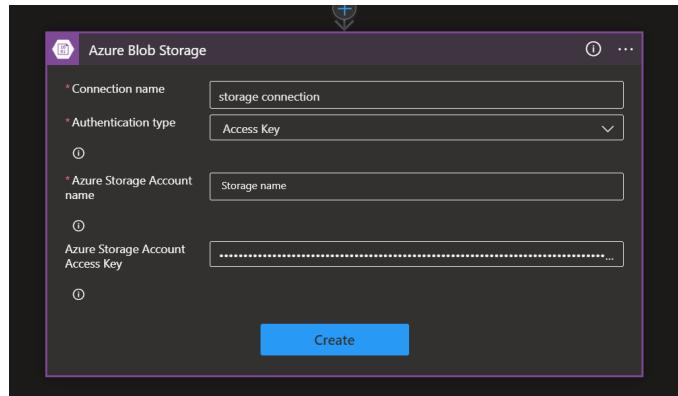
4. Create a blob and store the insights JSON.

a. Search for "Azure blob", from the group of actions.

b. Select **Create blob(V2)**.

c. Set the connection to the blob storage that will store the JSON insights files.

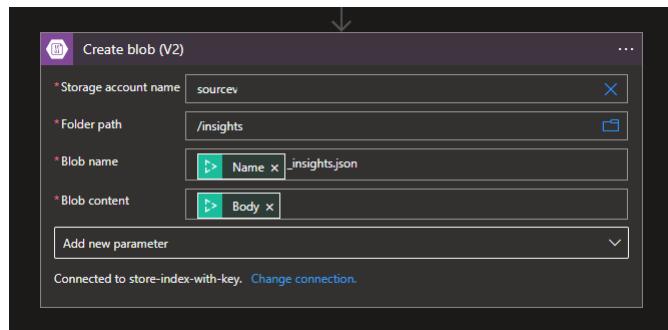
KEY	VALUE
Connection name	<Enter a connection name>.
Authentication type	Access Key
Azure Storage Account name	<* The storage account name where insights will be stored*>.
Azure Storage Account Access key	Go to Azure portal-> my-storage-> under Security + networking ->Access keys -> copy one of the keys.



d. Select **Create**.

e. Set the folder in which insights will be stored.

KEY	VALUE
Storage account name	<Enter the storage account name that would contain the JSON output (in this tutorial is the same as the source video).>
Folder path	From the dropdown, select the <code>/insights</code>
Blob name	From the dynamic content, under the Get Video Index section select Name and add <code>_insights.json</code> , insights file name will be the video name + <code>_insights.json</code>
Blob content	From the dynamic content, under the Get Video Index section, select the Body .

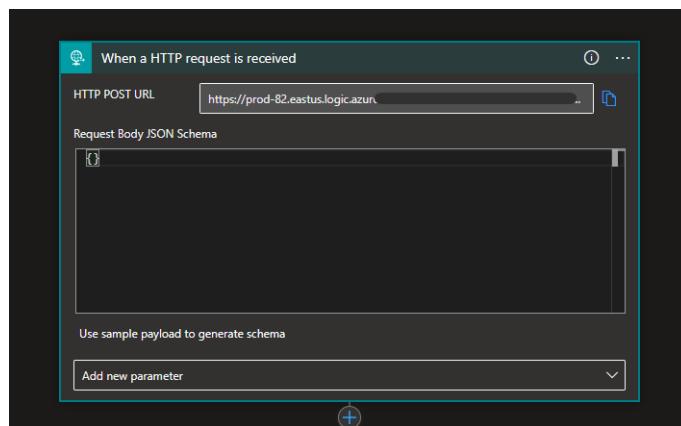


f. Select **Save flow**.

5. Update the callback URL to get notified when an index job is finished.

Once the flow is saved, an HTTP POST URL is created in the trigger.

a. Copy the URL from the trigger.



b. Go back to the first flow and paste the URL in the **Upload video and index** action for the **Callback URL parameter**.

Make sure both flows are saved.

Next steps

Try out your newly created Logic App or Power Automate solution by adding a video to your Azure blobs container, and go back a few minutes later to see that the insights appear in the destination folder.

Use Azure Video Indexer with Logic App and Power Automate

9/22/2022 • 6 minutes to read • [Edit Online](#)

Azure Video Indexer [REST API](#) supports both server-to-server and client-to-server communication and enables Azure Video Indexer users to integrate video and audio insights easily into their application logic, unlocking new experiences and monetization opportunities.

To make the integration even easier, we support [Logic Apps](#) and [Power Automate](#) connectors that are compatible with our API. You can use the connectors to set up custom workflows to effectively index and extract insights from a large amount of video and audio files, without writing a single line of code. Furthermore, using the connectors for your integration gives you better visibility on the health of your workflow and an easy way to debug it.

To help you get started quickly with the Azure Video Indexer connectors, we will do a walkthrough of an example Logic App and Power Automate solution you can set up. This tutorial shows how to set up flows using Logic Apps. However, the editors and capabilities are almost identical in both solutions, thus the diagrams and explanations are applicable to both Logic Apps and Power Automate.

The "upload and index your video automatically" scenario covered in this tutorial is comprised of two different flows that work together.

- The first flow is triggered when a blob is added or modified in an Azure Storage account. It uploads the new file to Azure Video Indexer with a callback URL to send a notification once the indexing operation completes.
- The second flow is triggered based on the callback URL and saves the extracted insights back to a JSON file in Azure Storage. This two flow approach is used to support async upload and indexing of larger files effectively.

This tutorial is using Logic App to show how to:

- Set up the file upload flow
- Set up the JSON extraction flow

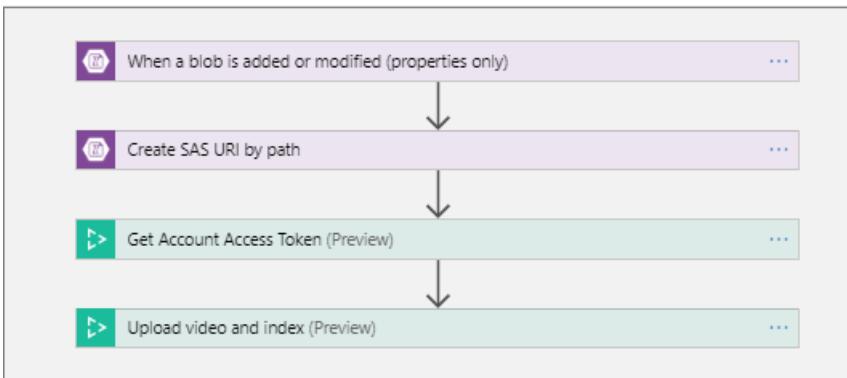
If you don't have an [Azure subscription](#), create an [Azure free account](#) before you begin.

Prerequisites

- To begin with, you will need an Azure Video Indexer account along with [access to the APIs via API key](#).
- You will also need an Azure Storage account. Keep note of the access key for your Storage account. Create two containers – one to store videos in and one to store insights generated by Azure Video Indexer in.
- Next, you will need to open two separate flows on either Logic Apps or Power Automate (depending on which you are using).

Set up the first flow - file upload

The first flow is triggered whenever a blob is added in your Azure Storage container. Once triggered, it will create a SAS URI that you can use to upload and index the video in Azure Video Indexer. In this section you will create the following flow.



To set up the first flow, you will need to provide your Azure Video Indexer API Key and Azure Storage credentials.

Azure Blob Storage

- *Connection Name
- *Azure Storage Accou...
- *Azure Storage Accou...

Create **Cancel**

[Browse current subscription](#)

Video Indexer (V2)

- *Connection Name
- *API Key

Create **Cancel**

TIP

If you previously connected an Azure Storage account or Azure Video Indexer account to a Logic App, your connection details are stored and you will be connected automatically.

You can edit the connection by clicking on **Change connection** at the bottom of an Azure Storage (the storage window) or Azure Video Indexer (the player window) action.

Once you can connect to your Azure Storage and Azure Video Indexer accounts, find and choose the "When a blob is added or modified" trigger in **Logic Apps Designer**.

Select the container that you will place your video files in.

When a blob is added or modified (properties only)

*Container: /logic-app

Number of blobs to return from the trigger: 10

How often do you want to check for items?

*Interval: 3

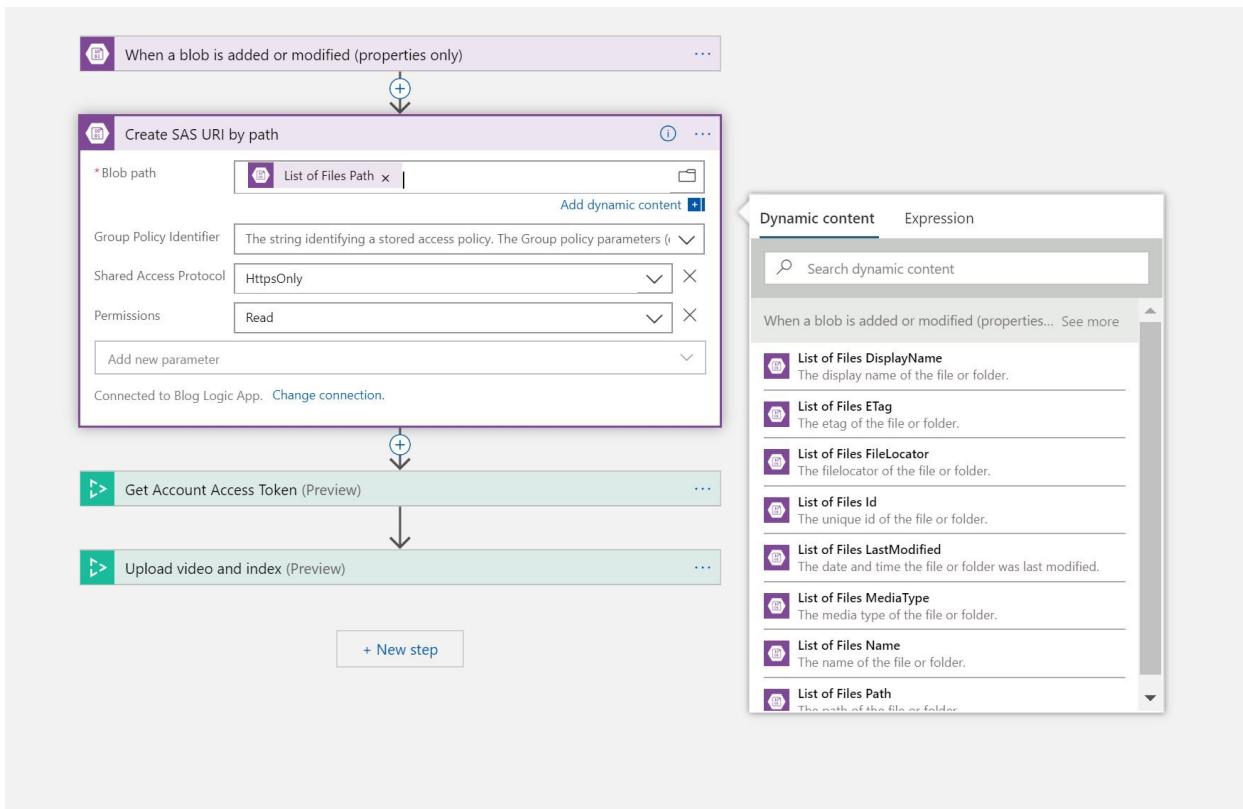
*Frequency: Minute

Add new parameter

Connected to Blog Logic App. [Change connection](#).

Next, find and select the "Create SAS URI by path" action. In the dialog for the action, select List of Files Path from the Dynamic content options.

Also, add a new "Shared Access Protocol" parameter. Choose HttpsOnly for the value of the parameter.



Fill out [your account location](#) and [account ID](#) to get the Azure Video Indexer account token.

The 'Get Account Access Token (Preview)' step configuration includes:

- Location: Indicates the Azure region to which the call should be routed.
- Account ID: Account ID is a globally unique identifier (GUID) for the account.
- Allow Edit: Yes

Connected to Blog Logic App. [Change connection.](#)

For "Upload video and index", fill out the required parameters and Video URL. Select "Add new parameter" and select Callback URL.

The 'Upload video and index (Preview)' step configuration includes:

- Location: Indicates the Azure region to which the call should be routed.
- Account ID: Account ID is a globally unique identifier (GUID) for the account.
- Access Token: Access Token (dropdown menu showing 'List of Files Name')
- Video Name: List of Files Name
- Video URL: Web Url
- Body: If a video URL is not specified, the file should be passed as a multipart/form
- Callback URL: A URL to notify when indexing is complete. (This field is empty)

Add new parameter

Connected to Blog Logic App. [Change connection.](#)

You will leave the callback URL empty for now. You'll add it only after finishing the second flow where the callback URL is created.

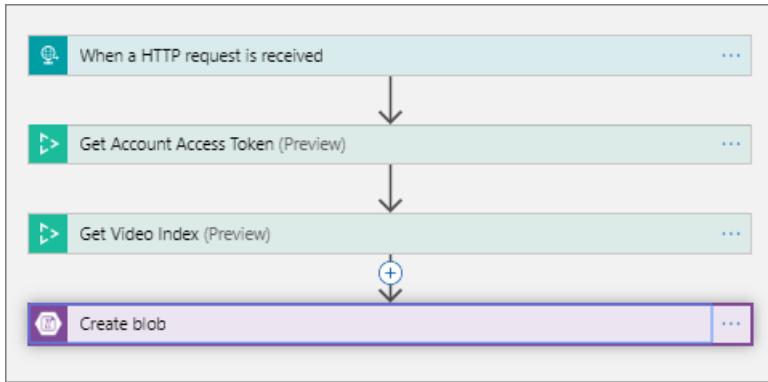
You can use the default value for the other parameters or set them according to your needs.

Click **Save**, and let's move on to configure the second flow, to extract the insights once the upload and indexing is completed.

Set up the second flow - JSON extraction

The completion of the uploading and indexing from the first flow will send an HTTP request with the correct callback URL to trigger the second flow. Then, it will retrieve the insights generated by Azure Video Indexer. In this example, it will store the output of your indexing job in your Azure Storage. However, it is up to you what you can do with the output.

Create the second flow separate from the first one.



To set up this flow, you will need to provide your Azure Video Indexer API Key and Azure Storage credentials again. You will need to update the same parameters as you did for the first flow.

For your trigger, you will see an HTTP POST URL field. The URL won't be generated until after you save your flow; however, you will need the URL eventually. We will come back to this.

Fill out [your account location](#) and [account ID](#) to get the Azure Video Indexer account token.

Go to the "Get Video Index" action and fill out the required parameters. For Video ID, put in the following expression: `triggerOutputs()['queries']['id']`

The screenshot shows a Logic App flow with the following steps:

- When a HTTP request is received
- Get Account Access Token (Preview)
- Get Video Index (Preview)
 - * Location: Indicates the Azure region to which the call should be routed.
 - * Account ID: Account ID is a globally unique identifier (GUID) for the account.
 - * Video ID: The video ID.
 - * Access Token: Access Token (dynamic content)
 - Captions Language: English
- Create blob

A dynamic content dialog is open over the "Access Token" input field, showing the expression `triggerOutputs()['queries']['id']`. The dialog also displays a list of available functions categorized into String functions, Collection, and Logical functions.

String functions
<code>fx concat(text_1, text_2, ...)</code> Combines any number of strings together

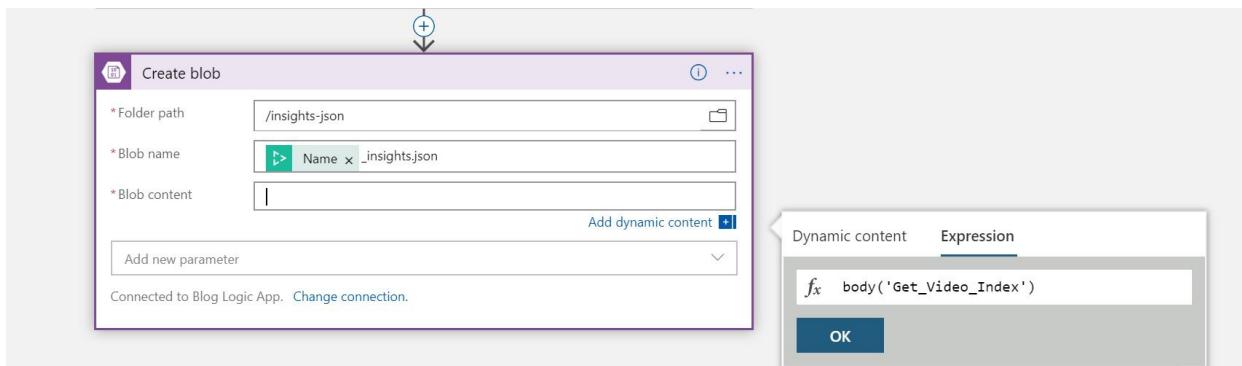
Collection
<code>fx contains(collection, value)</code> Returns true if a dictionary contains a key, if an array co...
<code>fx length(collection)</code> Returns the number of elements in an array or string

Logical functions
<code>fx if(expression, valueIfTrue, valueIfFalse)</code> Returns a specified value based on whether the expressi...
<code>fx equals(object1, object2)</code> Returns true if two values are equal
<code>fx and(expression1, expression2)</code> Returns true if both parameters are true

This expression tells the connector to get the Video ID from the output of your trigger. In this case, the output of

your trigger will be the output of "Upload video and index" in your first trigger.

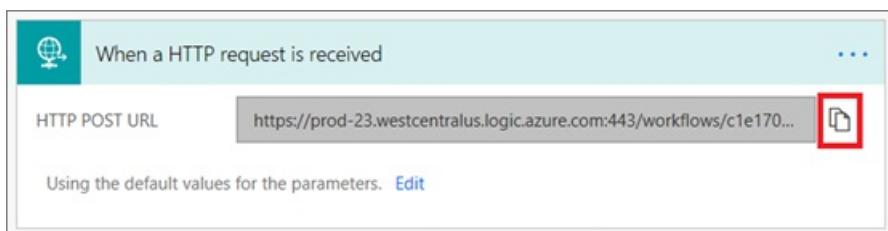
Go to the "Create blob" action and select the path to the folder in which you will save the insights to. Set the name of the blob you are creating. For Blob content, put in the following expression: body('Get_Video_Index')



This expression takes the output of the "Get Video Index" action from this flow.

Click **Save flow**.

Once the flow is saved, an HTTP POST URL is created in the trigger. Copy the URL from the trigger.



Now, go back to the first flow and paste the URL in the "Upload video and index" action for the Callback URL parameter.

Make sure both flows are saved, and you're good to go!

Try out your newly created Logic App or Power Automate solution by adding a video to your Azure blobs container, and go back a few minutes later to see that the insights appear in the destination folder.

Generate captions

See the following blog for the steps that show [how to generate captions with Azure Video Indexer and Logic Apps](#).

The article also shows how to index a video automatically by copying it to OneDrive and how to store the captions generated by Azure Video Indexer in OneDrive.

Clean up resources

After you are done with this tutorial, feel free to keep this Logic App or Power Automate solution up and running if you need. However, if you do not want to keep this running and do not want to be billed, Turn Off both of your flows if you're using Power Automate. Disable both of the flows if you're using Logic Apps.

Next steps

This tutorial showed just one Azure Video Indexer connectors example. You can use the Azure Video Indexer connectors for any API call provided by Azure Video Indexer. For example: upload and retrieve insights, translate the results, get embeddable widgets and even customize your models. Additionally, you can choose to trigger those actions based on different sources like updates to file repositories or emails sent. You can then choose to have the results update to our relevant infrastructure or application or generate any number of action items.

[Use the Azure Video Indexer API](#)

For additional resources, refer to [Azure Video Indexer](#)

Live stream analysis with Azure Video Indexer

9/22/2022 • 2 minutes to read • [Edit Online](#)

Azure Video Indexer is an Azure service designed to extract deep insights from video and audio files offline. This is to analyze a given media file already created in advance. However, for some use cases it's important to get the media insights from a live feed as quick as possible to unlock operational and other use cases pressed in time. For example, such rich metadata on a live stream could be used by content producers to automate TV production.

A solution described in this article, allows customers to use Azure Video Indexer in near real-time resolutions on live feeds. The delay in indexing can be as low as four minutes using this solution, depending on the chunks of data being indexed, the input resolution, the type of content and the compute powered used for this process.

Live stream analysis using Video Indexer



I'm here with Pete Carroll at the Seahawks training facility.



Pete Carroll Satya Nadella

Topics

Health and Wellbeing/Emotions Pete Carroll Football Coach

Labels

person outdoor grass sport athletic game

Keywords

hawks

Figure 1 – Sample player displaying the Azure Video Indexer metadata on the live stream

The [stream analysis solution](#) at hand, uses Azure Functions and two Logic Apps to process a live program from a live channel in Azure Media Services with Azure Video Indexer and displays the result with Azure Media Player showing the near real-time resulted stream.

In high level, it is comprised of two main steps. The first step runs every 60 seconds, and takes a subclip of the last 60 seconds played, creates an asset from it and indexes it via Azure Video Indexer. Then the second step is called once indexing is complete. The insights captured are processed, sent to Azure Cosmos DB, and the subclip indexed is deleted.

The sample player plays the live stream and gets the insights from Azure Cosmos DB, using a dedicated Azure Function. It displays the metadata and thumbnails in sync with the live video.

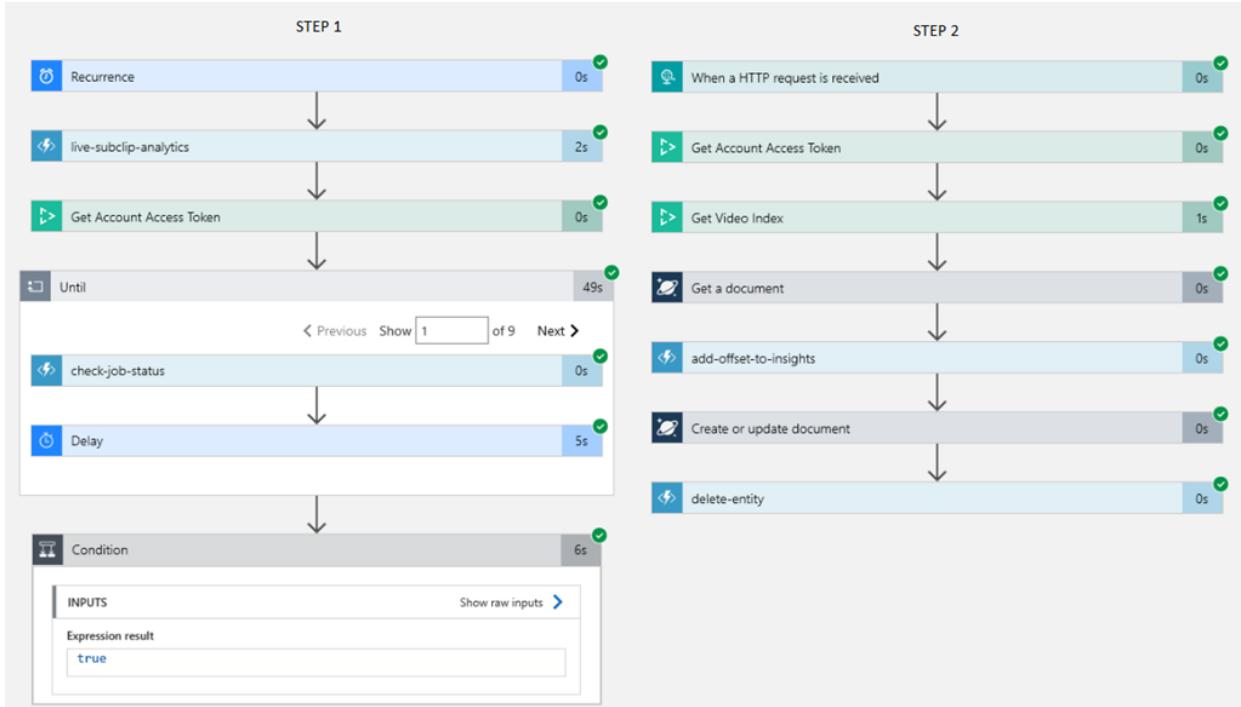


Figure 2 – The two logic apps processing the live stream every minute in the cloud.

Step-by-step guide

The full code and a step-by-step guide to deploy the results can be found in [GitHub project for Live media analytics with Azure Video Indexer](#).

Next steps

[Azure Video Indexer overview](#)

Monitoring Azure Video Indexer

9/22/2022 • 3 minutes to read • [Edit Online](#)

When you have critical applications and business processes relying on Azure resources, you want to monitor those resources for their availability, performance, and operation.

This article describes the monitoring data generated by Azure Video Indexer. Azure Video Indexer uses [Azure Monitor](#). If you are unfamiliar with the features of Azure Monitor common to all Azure services that use it, read [Monitoring Azure resources with Azure Monitor](#).

Some services in Azure have a special focused pre-built monitoring dashboard in the Azure portal that provides a starting point for monitoring your service. These special dashboards are called "insights".

Monitoring data

Azure Video Indexer collects the same kinds of monitoring data as other Azure resources that are described in [Monitoring data from Azure resources](#).

See [Monitoring Azure Video Indexer data reference](#) for detailed information on the metrics and logs metrics created by Azure Video Indexer.

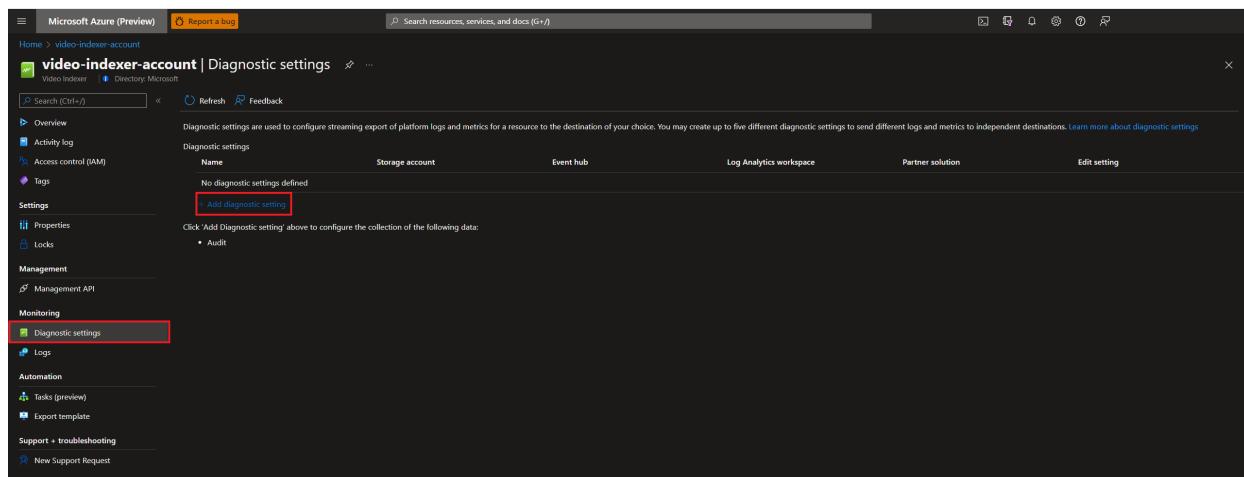
Collection and routing

Activity log are collected and stored automatically, but can be routed to other locations by using a diagnostic setting.

Resource Logs are not collected and stored until you create a diagnostic setting and route them to one or more locations.

See [Create diagnostic setting to collect platform logs and metrics in Azure](#) for the detailed process for creating a diagnostic setting using the Azure portal, CLI, or PowerShell. When you create a diagnostic setting, you specify which categories of logs to collect. The categories for *Azure Video Indexer* are listed in [Azure Video Indexer monitoring data reference](#).

CATEGORY	DESCRIPTION
Audit	Read/Write operations



The screenshot shows the Azure portal interface for managing diagnostic settings. On the left, there's a navigation sidebar with options like Overview, Activity log, Access control (IAM), Tags, Properties, Locks, Management API, Monitoring (with Diagnostic settings selected), Automation, Tasks (preview), Export template, Support + troubleshooting, and New Support Request. The main content area is titled 'video-indexer-account | Diagnostic settings'. It shows a table with one row for 'Audit'. The 'Audit' row has columns for Name (containing 'Audit'), Storage account (empty), Event hub (empty), Log Analytics workspace (empty), and Partner solution (empty). Below the table, there's a note: 'Diagnostic settings are used to configure streaming export of platform logs and metrics for a resource to the destination of your choice. You may create up to five different diagnostic settings to send different logs and metrics to independent destinations.' A red box highlights the 'Add diagnostic setting' button at the bottom of the table.

A diagnostic setting specifies a list of categories of platform logs and/or metrics that you want to collect from a resource, and one or more destinations that you would stream them to. Normal usage charges for the destination will occur. [Learn more about the different log categories and contents of those logs](#)

Diagnostic setting name * My-new-diagnostic-settings

Logs

Category groups ⓘ

allLogs

Categories

Audit

Destination details

Send to Log Analytics workspace

Subscription

My-subscription

Log Analytics workspace

Target-workspace (eastus)

Archive to a storage account

Stream to an event hub

Send to partner solution

The metrics and logs you can collect are discussed in the following sections.

Analyzing metrics

Currently Azure Video Indexer does not support monitoring of metrics.

Analyzing logs

Data in Azure Monitor Logs is stored in tables where each table has its own set of unique properties.

All resource logs in Azure Monitor have the same fields followed by service-specific fields. The common schema is outlined in [Azure Monitor resource log schema](#). The schema for Azure Video Indexer resource logs is found in the [Azure Video Indexer Data Reference](#).

The [Activity log](#) is a type of platform sign-in Azure that provides insight into subscription-level events. You can view it independently or route it to Azure Monitor Logs, where you can do much more complex queries using Log Analytics.

For a list of the types of resource logs collected for Azure Video Indexer, see [Monitoring Azure Video Indexer data reference](#)

For a list of the tables used by Azure Monitor Logs and queryable by Log Analytics, see [Monitoring Azure Video Indexer data reference](#)

Sample Kusto queries

IMPORTANT

When you select **Logs** from the Azure Video Indexer account menu, Log Analytics is opened with the query scope set to the current Azure Video Indexer account. This means that log queries will only include data from that resource. If you want to run a query that includes data from other Azure Video Indexer account or data from other Azure services, select **Logs** from the **Azure Monitor** menu. See [Log query scope and time range in Azure Monitor Log Analytics](#) for details.

Following are queries that you can use to help you monitor your Azure Video Indexer account.

```
// Project failures summarized by operationName and Upn, aggregated in 30m windows.  
VIAudit  
| where Status == "Failure"  
| summarize count() by OperationName, bin(TimeGenerated, 30m), Upn  
| render timechart
```

```
// Project failures with detailed error message.  
VIAudit  
| where Status == "Failure"  
| parse Description with "ErrorType: " ErrorType ". Message: " ErrorMessage ". Trace" *  
| project TimeGenerated, OperationName, ErrorMessage, ErrorType, CorrelationId, _ResourceId
```

Alerts

Azure Monitor alerts proactively notify you when important conditions are found in your monitoring data. They allow you to identify and address issues in your system before your customers notice them. You can set alerts on [metrics](#), [logs](#), and the [activity log](#). Different types of alerts have benefits and drawbacks.

The following table lists common and recommended alert rules for Azure Video Indexer.

ALERT TYPE	CONDITION	DESCRIPTION
Log Alert	Failed operation	Send an alert when an upload failed

```
//All failed uploads, aggregated in one hour window.  
VIAudit  
| where OperationName == "Upload-Video" and Status == "Failure"  
| summarize count() by bin(TimeGenerated, 1h)
```

Next steps

- See [Monitoring Azure Video Indexer data reference](#) for a reference of the metrics, logs, and other important values created by Azure Video Indexer account.
- See [Monitoring Azure resources with Azure Monitor](#) for details on monitoring Azure resources.

NSG service tags for Azure Video Indexer

9/22/2022 • 2 minutes to read • [Edit Online](#)

Azure Video Indexer is a service hosted on Azure. In some architecture cases the service needs to interact with other services in order to index video files (that is, a Storage Account) or when a customer orchestrates indexing jobs against our API endpoint using their own service hosted on Azure (i.e AKS, Web Apps, Logic Apps, Functions). Customers who would like to limit access to their resources on a network level can use [Network Security Groups with Service Tags](#). A service tag represents a group of IP address prefixes from a given Azure service, in this case Azure Video Indexer. Microsoft manages the address prefixes grouped by the service tag and automatically updates the service tag as addresses change in our backend, minimizing the complexity of frequent updates to network security rules by the customer.

Get started with service tags

Currently we support the global service tag option for using service tags in your network security groups:

Use a single global AzureVideoAnalyzerForMedia service tag: This option opens your virtual network to all IP addresses that the Azure Video Indexer service uses across all regions we offer our service. This method will allow for all IP addresses owned and used by Azure Video Indexer to reach your network resources behind the NSG.

NOTE

Currently we do not support IPs allocated to our services in the Switzerland North Region. These will be added soon. If your account is located in this region you cannot use Service Tags in your NSG today since these IPs are not in the Service Tag list and will be rejected by the NSG rule.

Use a single global Azure Video Indexer service tag

The easiest way to begin using service tags with your Azure Video Indexer account is to add the global tag `AzureVideoAnalyzerForMedia` to an NSG rule.

1. From the [Azure portal](#), select your network security group.
2. Under **Settings**, select **Inbound security rules**, and then select **+ Add**.
3. From the **Source** drop-down list, select **Service Tag**.
4. From the **Source service tag** drop-down list, select `AzureVideoAnalyzerForMedia`.

 Add inbound security rule

prxprod-nsg

Source ⓘ

Service Tag

Source service tag * ⓘ

AzureVideoAnalyzerForMedia

Source port ranges * ⓘ

*

Destination ⓘ

Any

Service ⓘ

Custom

Destination port ranges * ⓘ

443

Protocol

Any

TCP

UDP

ICMP

Action

Allow

Deny

Priority *

350

Name *

AzureVideoAnalyzerForMedia

Description

This tag contains the IP addresses of Azure Video Indexer services for all regions where available. The tag will ensure that your resource can communicate with the Azure Video Indexer services no matter where it's created.

Using Azure CLI

You can also use Azure CLI to create a new or update an existing NSG rule and add the **AzureVideoAnalyzerForMedia** service tag using the `--source-address-prefixes`. For a full list of CLI commands and parameters see [az network nsg](#)

Example of a security rule using service tags. For more details, visit <https://aka.ms/servicetags>

```
az network nsg rule create -g MyResourceGroup --nsg-name MyNsg -n MyNsgRuleWithTags --priority 400 --source-address-prefixes AzureVideoAnalyzerForMedia --destination-address-prefixes '*' --destination-port-ranges '*' --direction Inbound --access Allow --protocol Tcp --description "Allow from VideoAnalyzerForMedia"
```

Next steps

[Disaster recovery](#)

Azure Video Indexer failover and disaster recovery

9/22/2022 • 2 minutes to read • [Edit Online](#)

Azure Video Indexer doesn't provide instant failover of the service if there's a regional datacenter outage or failure. This article explains how to configure your environment for a failover to ensure optimal availability for apps and minimized recovery time if a disaster occurs.

We recommend that you configure business continuity disaster recovery (BCDR) across regional pairs to benefit from Azure's isolation and availability policies. For more information, see [Azure paired regions](#).

Prerequisites

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

Fail over to a secondary account

To implement BCDR, you need to have two Azure Video Indexer accounts to handle redundancy.

1. Create two Azure Video Indexer accounts connected to Azure (see [Create an Azure Video Indexer account](#)).
Create one account for your primary region and the other to the paired Azure region.
2. If there's a failure in your primary region, switch to indexing using the secondary account.

TIP

You can automate BCDR by setting up activity log alerts for service health notifications as per [Create activity log alerts on service notifications](#).

For information about using multiple tenants, see [Manage multiple tenants](#). To implement BCDR, choose one of these two options: [Azure Video Indexer account per tenant](#) or [Azure subscription per tenant](#).

Next steps

[Manage an Azure Video Indexer account connected to Azure](#).

Repair the connection to Azure, examine errors/warnings

9/22/2022 • 3 minutes to read • [Edit Online](#)

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

NOTE

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

Prerequisites

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

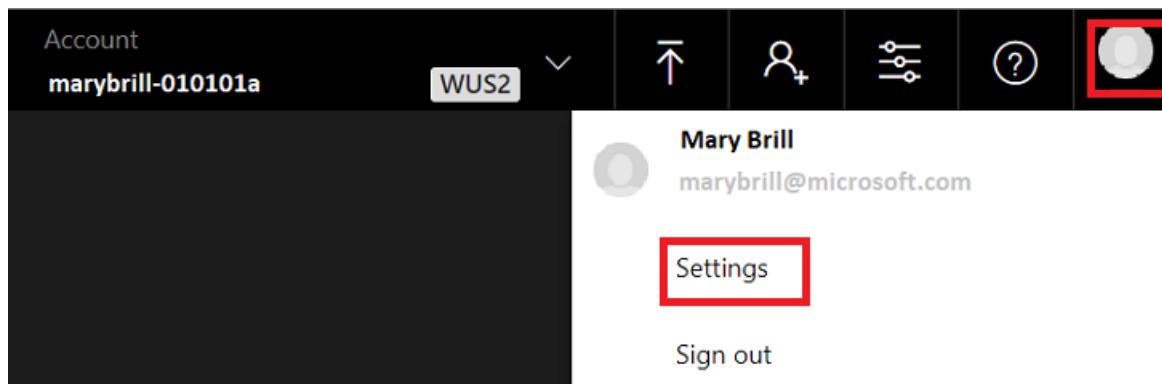
Make sure to follow [Prerequisites](#) and review [Considerations](#) in the article.

Examine account settings

This section examines settings of your Azure 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 things:

- 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'll 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.

Repair the connection to Azure

In the [Update connection to Azure Media Services](#) dialog of your [Azure Video Indexer](#) page, you're asked

to provide values for the following settings:

SETTING	DESCRIPTION
Azure subscription ID	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.
Application ID	<p>The Azure AD application ID (with permissions for the specified Media Services account) that you created for this Azure Video Indexer account.</p> <p>To get the app ID, navigate to Azure portal. Under the Media Services account, choose your account and go to API Access. Select Connect to Media Services API with service principal -> Azure AD App. Copy the relevant parameters.</p>
Application key	<p>The Azure AD application key associated with your Media Services account that you specified above.</p> <p>To get the app key, navigate to Azure portal. Under the Media Services account, choose your account and go to API Access. Select Connect to Media Services API with service principal -> Manage application -> Certificates & secrets. Copy the relevant parameters.</p>

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 Event Grid resource provider using the Azure portal. In the [Azure portal](#), go to **Subscriptions** > [subscription] > **ResourceProviders** > **Microsoft.EventGrid**. If not in the **Registered** state, select **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 can't watch videos from this Media Services account or in Azure 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 or Azure Video Indexer accounts that are connected to Azure by following the instructions in: [Use APIs](#).

Use the same Azure AD user you used when connecting to Azure.

Switch between multiple tenants

9/22/2022 • 2 minutes to read • [Edit Online](#)

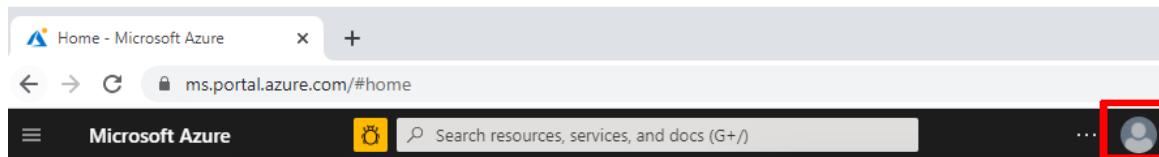
This article shows how to switch between multiple tenants on the Azure Video Indexer website. When you create an Azure Resource Manager (ARM)-based account, the new account may not show up on the Azure Video Indexer website. So you need to make sure to sign in with the correct domain.

The article shows how to sign in with the correct domain name into the Azure Video Indexer website:

1. Sign into the [Azure portal](#) with the same subscription where your Video Indexer ARM account was created.
2. Get the domain name of the current Azure subscription tenant.
3. Sign in with the correct domain name on the [Azure Video Indexer](#) website.

Get the domain name from the Azure portal

1. In the [Azure portal](#), sign in with the same subscription tenant in which your Azure Video Indexer Azure Resource Manager (ARM) account was created.
2. Hover over your account name (in the right-top corner).



3. Get the domain name of the current Azure subscription, you'll need it for the last step of the following section.

If you want to see domains for all of your directories and switch between them, see [Switch and manage directories with the Azure portal](#).

Sign in with the correct domain name on the AVI website

1. Go to the [Azure Video Indexer](#) website.
2. Press **Sign out** after pressing the button in the top-right corner.
3. On the AVI website, press **Sign in** and choose the Azure AD account.

Choose an account to continue:

 AAD account

 Personal Microsoft account

 Google

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

[I was using Linkedin or Facebook, where did it go?](#)

4. Press **Use another account**.



Pick an account

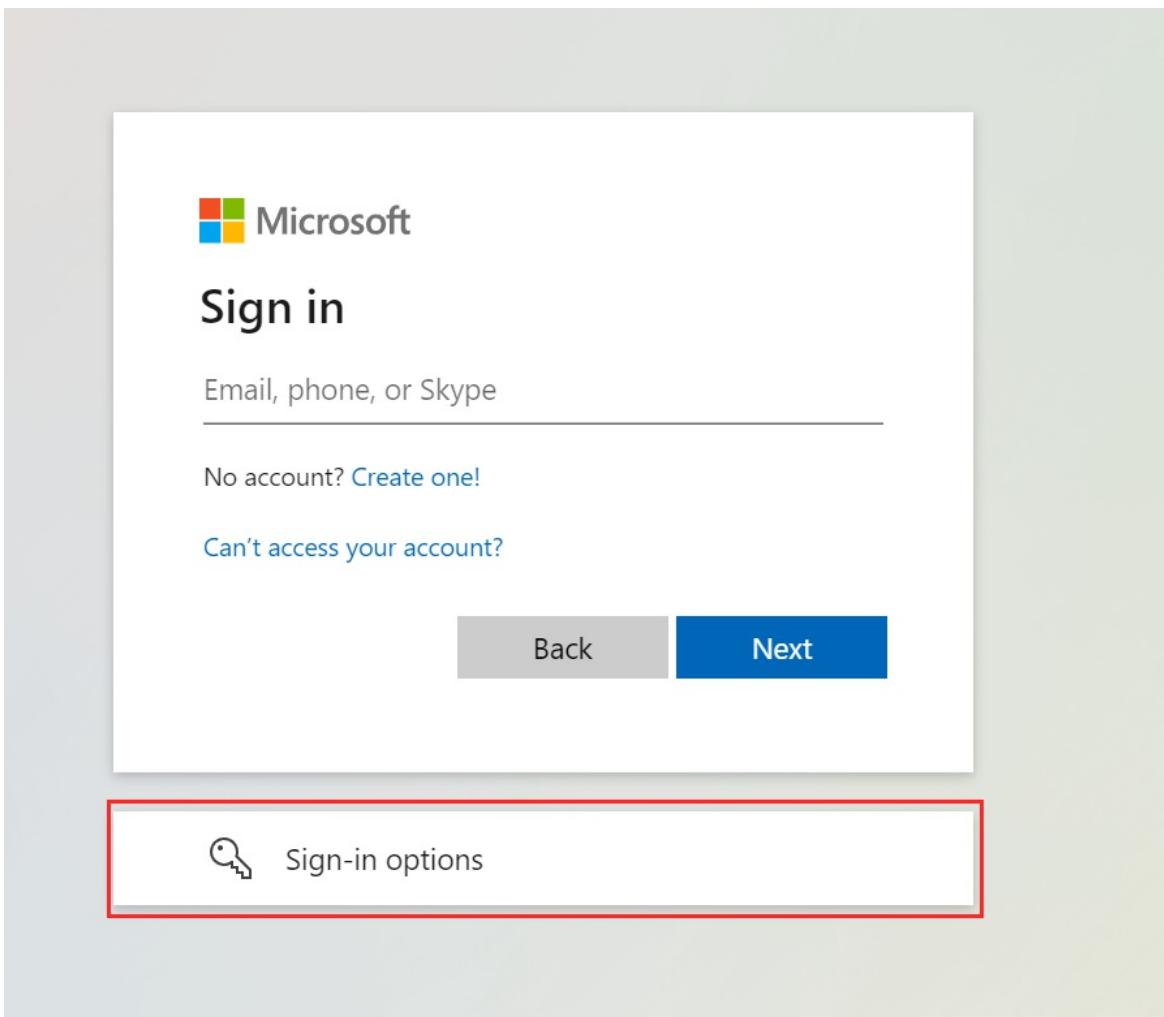


User
user@contoso.com
Connected to Windows



[Use another account](#)

5. Choose **Sign-in with other options**.



6. Press **Sign in to an organization**.



Sign-in options



Sign in with Windows Hello or a security key



Choose this only if you have enabled Windows Hello or a security key for your account.



Sign in with GitHub

Personal accounts only



Sign in to an organization

Search for a company or an organization you're working with.

Back

7. Enter the domain name you copied in the [Get the domain name from the Azure portal](#) section.



Find your organization

Enter the domain name of the organization you'd like to sign in to.

Domain name

Back

Next

Next steps

[FAQ](#)

Monitor Azure Video Indexer data reference

9/22/2022 • 2 minutes to read • [Edit Online](#)

See [Monitoring Azure Video Indexer](#) for details on collecting and analyzing monitoring data for Azure Video Indexer.

Metrics

Azure Video Indexer currently does not support any monitoring on metrics.

For more information, see a list of [all platform metrics supported in Azure Monitor](#).

Metric dimensions

Azure Video Indexer currently does not support any monitoring on metrics.

Resource logs

This section lists the types of resource logs you can collect for Azure Video Indexer.

For reference, see a list of [all resource logs category types supported in Azure Monitor](#).

Azure Video Indexer

Resource Provider and Type: [Microsoft.VideoIndexer/accounts](#)

CATEGORY	DISPLAY NAME	ADDITIONAL INFORMATION
VIAudit	Azure Video Indexer Audit Logs	Logs are produced from both the Video Indexer portal and the REST API.

Azure Monitor Logs tables

This section refers to all of the Azure Monitor Logs Kusto tables relevant to Azure Video Indexer and available for query by Log Analytics.

RESOURCE TYPE	NOTES
Azure Video Indexer	

Azure Video Indexer

TABLE	DESCRIPTION	ADDITIONAL INFORMATION
VIAudit	Events produced using Azure Video Indexer portal or REST API .	

For a reference of all Azure Monitor Logs / Log Analytics tables, see the [Azure Monitor Log Table Reference](#).

Activity log

The following table lists the operations related to Azure Video Indexer that may be created in the Activity log.

OPERATION	DESCRIPTION
Generate_AccessToken	
Accounts_Update	
Write tags	
Create or update resource diagnostic setting	
Delete resource diagnostic setting	

For more information on the schema of Activity Log entries, see [Activity Log schema](#).

Schemas

The following schemas are in use by Azure Video Indexer

```
{
  "time": "2022-03-22T10:59:39.5596929Z",
  "resourceId": "/SUBSCRIPTIONS/602a61eb-c111-43c0-8323-74825230a47d/RESOURCEGROUPS/VI-
  RESOURCEGROUP/PROVIDERS/MICROSOFT.VIDEOINDEXER/ACCOUNTS/VIDEOINDEXERACCOUNT",
  "operationName": "Get-Video-Thumbnail",
  "category": "Audit",
  "location": "westus2",
  "durationMs": "192",
  "resultSignature": "200",
  "resultType": "Success",
  "resultDescription": "Get Video Thumbnail",
  "correlationId": "33473fc3-bcbc-4d47-84cc-9fba2f3e9faa",
  "callerIpAddress": "46.*****",
  "operationVersion": "Operations",
  "identity": {
    "externalUserId": "4704F34286364F2*****",
    "upn": "alias@outlook.com",
    "claims": { "permission": "Reader", "scope": "Account" }
  },
  "properties": {
    "accountName": "videoIndexerAccoount",
    "accountId": "8878b584-d8a0-4752-908c-00d6e5597f55",
    "videoId": "1e2ddfd77"
  }
}
```

Next steps

- See [Monitoring Azure Video Indexer](#) for a description of monitoring Azure Video Indexer.
- See [Monitoring Azure resources with Azure Monitor](#) for details on monitoring Azure resources.

Azure regions in which Azure Video Indexer exists

9/22/2022 • 2 minutes to read • [Edit Online](#)

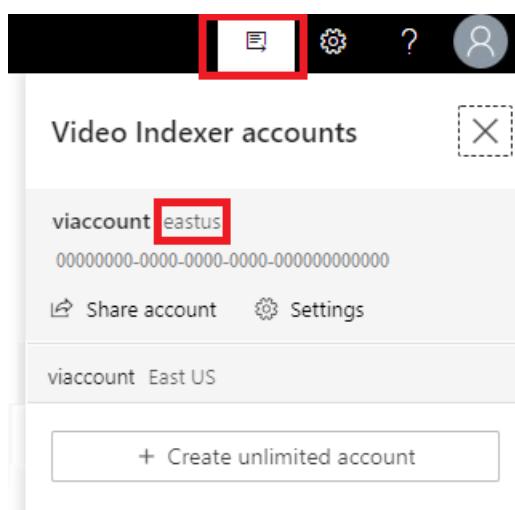
Azure Video Indexer APIs contain a **location** parameter that you should set to the Azure region to which the call should be routed. This must be an [Azure region in which Azure Video Indexer is available](#).

Locations

The `location` parameter must be given the Azure region code name as its value. If you are using Azure Video Indexer in preview mode, you should put `"trial"` as the value. `trial` is the default value for the `location` parameter. 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 use the Azure portal or run a [Azure CLI](#) command.

Azure portal

1. Sign in on the [Azure Video Indexer](#) website.
2. Select **User accounts** from the top-right corner of the page.
3. Find the location of your account in the top-right corner.



CLI command

```
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/00000000-0000-0000-0000-000000000000/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](#)

Azure Video Indexer release notes

9/22/2022 • 25 minutes to read • [Edit Online](#)

Get notified about when to revisit this page for updates by copying and pasting this URL:

`https://learn.microsoft.com/api/search/rss?search=%22Azure+Media+Services+Video+Indexer+release+notes%22&locale=en-us`

into your RSS feed reader.

To stay up-to-date with the most recent Azure Video Indexer developments, this article provides you with information about:

- [Important notice](#) about planned changes
- The latest releases
- Known issues
- Bug fixes
- Deprecated functionality

Upcoming critical changes

IMPORTANT

This section describes a critical upcoming change for the [Upload-Video](#) API.

Upload-Video API

In the past, the [Upload-Video](#) API was tolerant to calls to upload a video from a URL where an empty multipart form body was provided in the C# code, such as:

```
var content = new MultipartFormDataContent();
var uploadRequestResult = await client.PostAsync($"  

{apiUrl}/{accountInfo.Location}/Accounts/{accountInfo.Id}/Videos?{queryParams}", content);
```

In the coming weeks, our service will fail requests of this type.

In order to upload a video from a URL, change your code to send null in the request body:

```
var uploadRequestResult = await client.PostAsync($"  

{apiUrl}/{accountInfo.Location}/Accounts/{accountInfo.Id}/Videos?{queryParams}", null);
```

September 2022

General availability of Azure Resource Management (ARM)

With the ARM-based [paid \(unlimited\)](#) account you are able to use:

- [Azure role-based access control \(RBAC\)](#).
- Managed Identity to better secure the communication between your Azure Media Services and Azure Video Indexer account, Network Service Tags, and native integration with Azure Monitor to monitor your account (audit and indexing logs).
- Scale and automate your [deployment with ARM-template, bicep or terraform](#).

- Create logic apps connector for ARM-based accounts.

To create an ARM-based account, see [create an account](#).

New source languages support for STT, translation, and search

Now supporting source languages for STT (speech-to-text), translation, and search in Ukraine and Vietnamese. It means transcription, translation, and search features are also supported for these languages in Azure Video Indexer web applications, widgets and APIs.

For more information, see [supported languages](#).

Expanded the supported languages in LID and MLID through the API

We expanded the list of the languages to be supported in LID (language identification) and MLID (multi language Identification) using APIs.

For more information, see [supported languages](#).

Configure confidence level in a person model with an API

Use the [Patch person model](#) API to configure the confidence level for face recognition within a person model.

August 2022

Update topic inferencing model

Azure Video Indexer topic inferencing model was updated and now we extract more than 6.5 million topics (for example, covering topics such as Covid virus). To benefit from recent model updates you need to re-index your video files.

Topic inferencing model is now available on Azure Government

You can now leverage topic inferencing model in your Azure Video Indexer paid account on [Azure Government](#) in Virginia and Arizona regions. With this release we completed the AI parity between Azure global and Azure Government. To benefit from the model updates you need to re-index your video files.

Session length is now 30 days in the Azure Video Indexer website

The [Azure Video Indexer website](#) session length was extended to 30 days. You can preserve your session without having to re-login every 1 hour.

July 2022

The featured clothing insight (preview)

The featured clothing insight enables more targeted ads placement.

The insight provides information of key items worn by individuals within a video and the timestamp in which the clothing appears. This allows high-quality in-video contextual advertising, where relevant clothing ads are matched with the specific time within the video in which they are viewed.

To view the featured clothing of an observed person, you have to index the video using Azure Video Indexer advanced video settings. For details on how featured clothing images are ranked and how to view this insight, see [featured clothing](#).

June 2022

Create Video Indexer blade improvements in Azure portal

Azure Video Indexer now supports the creation of new resource using system-assigned managed identity or system and user assigned managed identity for the same resource.

You can also change the primary managed identity using the **Identity** tab in the [Azure portal](#).

Limited access of celebrity recognition and face identification features

As part of Microsoft's commitment to responsible AI, we are designing and releasing Azure Video Indexer – identification and celebrity recognition features. These features are designed to protect the rights of individuals and society and fostering transparent human-computer interaction. Thus, there is a limited access and use of Azure Video Indexer – identification and celebrity recognition features.

Identification and celebrity recognition features require registration and are only available to Microsoft managed customers and partners. Customers who wish to use this feature are required to apply and submit an [intake form](#). For more information, read [Azure Video Indexer limited access](#).

Also, see the following: the [announcement blog post](#) and [investment and safeguard for facial recognition](#).

May 2022

Line breaking in transcripts

Improved line break logic to better split transcript into sentences. New editing capabilities are now available through the Azure Video Indexer portal, such as adding a new line and editing the line's timestamp. For more information, see [Insert or remove transcript lines](#).

Azure Monitor integration

Azure Video Indexer now supports Diagnostics settings for Audit events. Logs of Audit events can now be exported through diagnostics settings to Azure Log Analytics, Storage, Event Hubs, or a third-party solution.

The additions enable easier access to analyze the data, monitor resource operation, and create automatically flows to act on an event. For more information, see [Monitor Azure Video Indexer](#).

Video Insights improvements

Object Character Reader (OCR) is improved by 60%. Face Detection is improved by 20%. Label accuracy is improved by 30% over a wide variety of videos. These improvements are available immediately in all regions and do not require any changes by the customer.

Service tag

Azure Video Indexer is now part of [Network Service Tags](#). Video Indexer often needs to access other Azure resources (for example, Storage). If you secure your inbound traffic to your resources with a Network Security Group you can now select Video Indexer as part of the built-in Service Tags. This will simplify security management as we populate the Service Tag with our public IPs.

Celebrity recognition toggle

You can now enable or disable the celebrity recognition model on the account level (on classic account only). To turn on or off the model, go to the **Model customization** > toggle on/off the model. Once you disable the model, Video Indexer insights will not include the output of celebrity model and will not run the celebrity model pipeline.

Content model customization

People Language Brands Animated characters (preview)

Click on a person to manage their model and add images to enhance the model.
Uploaded photos should include only one face, and the file format can be .jpg or .png.
You can add up to 50 person models on each account.

[Learn more](#)

Include celebrity recognition when indexing uploaded media. [Learn more](#)

Azure Video Indexer repository name

As of May 1st, our new updated repository of Azure Video Indexer widget was renamed. Use <https://www.npmjs.com/package/@azure/video-indexer-widgets> instead

April 2022

Renamed Azure Video Analyzer for Media back to Azure Video Indexer

As of today, Azure Video analyzer for Media product name is **Azure Video Indexer** and all product related assets (web portal, marketing materials). It is a backward compatible change that has no implication on APIs and links. **Azure Video Indexer's new logo:**



March 2022

Closed Captioning files now support including speakers' attributes

Azure Video Indexer enables you to include speakers' characteristic based on a closed captioning file that you choose to download. To include the speakers' attributes, select Downloads -> Closed Captions -> choose the closed captioning downloadable file format (SRT, VTT, TTML, TXT, or CSV) and check **Include speakers** checkbox.

Improvements to the widget offering

The following improvements were made:

- Azure Video Indexer widgets support more than 1 locale in a widget's parameter.
- The Insights widgets support initial search parameters and multiple sorting options.
- The Insights widgets also include a confirmation step before deleting a face to avoid mistakes.
- The widget customization now supports width as strings (for example 100%, 100vw).

February 2022

Public preview of Azure Video Indexer account management based on ARM in Government cloud

Azure Video Indexer website is now supporting account management based on ARM in public preview (see, [November 2021 release note](#)).

Leverage open-source code to create ARM based account

Added new code samples including HTTP calls to use Azure Video Indexer create, read, update and delete (CRUD) ARM API for solution developers. See [this sample](#).

January 2022

Improved audio effects detection

The audio effects detection capability was improved to have a better detection rate over the following classes:

- Crowd reactions (cheering, clapping, and boozing),
- Gunshot or explosion,
- Laughter

For more information, see [Audio effects detection](#).

New source languages support for STT, translation, and search on the website

Azure Video Indexer introduces source languages support for STT (speech-to-text), translation, and search in Hebrew (he-IL), Portuguese (pt-PT), and Persian (fa-IR) on the [Azure Video Indexer](#) website. It means transcription, translation, and search features are also supported for these languages in Azure Video Indexer web applications and widgets.

December 2021

The projects feature is now GA

The projects feature is now GA and ready for productive use. There is no pricing impact related to the "Preview to GA" transition. See [Add video clips to your projects](#).

New source languages support for STT, translation, and search on API level

Azure Video Indexer introduces source languages support for STT (speech-to-text), translation, and search in Hebrew (he-IL), Portuguese (pt-PT), and Persian (fa-IR) on the API level.

Matched person detection capability

When indexing a video with Azure Video Indexer advanced video settings, you can view the new matched person detection capability. If there are people observed in your media file, you can now view the specific person who matched each of them through the media player.

November 2021

Public preview of Azure Video Indexer account management based on ARM

Azure Video Indexer introduces a public preview of Azure Resource Manager (ARM) based account management. You can leverage ARM-based Azure Video Indexer APIs to create, edit, and delete an account from the [Azure portal](#).

NOTE

The Government cloud includes support for CRUD ARM based accounts from Azure Video Indexer API and from the Azure portal.

There is currently no support from the Azure Video Indexer [website](#).

For more information go to [create an Azure Video Indexer account](#).

People's clothing detection

When indexing a video with Azure Video Indexer advanced video settings, you can view the new people's clothing detection capability. If there are people detected in your media file, you can now view the clothing type they are wearing through the media player.

Face bounding box (preview)

You can now turn on a bounding box for detected faces during indexing of the media file. The face bounding box feature is available when indexing your file by choosing the **standard**, **basic**, or **advanced** indexing presets.

You can enable the bounding boxes through the player.

October 2021

Embed widgets in your app using Azure Video Indexer package

Use the new Azure Video Indexer (AVAM) `@azure/video-analyzer-for-media-widgets` npm package to add `insights` widgets to your app and customize it according to your needs.

The new AVAM package enables you to easily embed and communicate between our widgets and your app, instead of adding an `iframe` element to embed the insights widget. Learn more in [Embed and customize Azure Video Indexer widgets in your app](#).

August 2021

Re-index video or audio files

There is now an option to re-index video or audio files that have failed during the indexing process.

Improve accessibility support

Fixed bugs related to CSS, theming and accessibility:

- high contrast
- account settings and insights views in the [portal](#).

July 2021

Automatic Scaling of Media Reserved Units

Starting August 1st 2021, Azure Video Indexer enabled [Media Reserved Units \(MRUs\)](#) auto scaling by [Azure Media Services](#), as a result you do not need to manage them through Azure Video Indexer. That will allow price optimization, for example price reduction in many cases, based on your business needs as it is being auto scaled.

June 2021

Azure Video Indexer deployed in six new regions

You can now create an Azure Video Indexer paid account in France Central, Central US, Brazil South, West Central US, Korea Central, and Japan West regions.

May 2021

New source languages support for speech-to-text (STT), translation, and search

Azure Video Indexer now supports STT, translation, and search in Chinese (Cantonese) ('zh-HK'), Dutch (Netherlands) ('nl-NL'), Czech ('cs-CZ'), Polish ('pl-PL'), Swedish (Sweden) ('sv-SE'), Norwegian('nb-NO'), Finnish('fi-FI'), Canadian French ('fr-CA'), Thai('th-TH'), Arabic: (United Arab Emirates) ('ar-AE', 'ar-EG'), (Iraq) ('ar-IQ'), (Jordan) ('ar-JO'), (Kuwait) ('ar-KW'), (Lebanon) ('ar-LB'), (Oman) ('ar-OM'), (Qatar) ('ar-QA'), (Palestinian Authority) ('ar-PS'), (Syria) ('ar-SY'), and Turkish('tr-TR').

These languages are available in both API and Azure Video Indexer website. Select the language from the combobox under **Video source language**.

New theme for Azure Video Indexer

New theme is available: 'Azure' along with the 'light' and 'dark' themes. To select a theme, click on the gear icon in the top-right corner of the website, find themes under **User settings**.

New open-source code you can leverage

Three new GitHub projects are available at our [GitHub repository](#):

- Code to help you leverage the newly added [widget customization](#).
- Solution to help you add [custom search](#) to your video libraries.
- Solution to help you add [de-duplication](#) to your video libraries.

New option to toggle bounding boxes (for observed people) on the player

When indexing a video through our advanced video settings, you can view our new observed people capabilities. If there are people detected in your media file, you can enable a bounding box on the detected person through the media player.

April 2021

The Video Indexer service was renamed to Azure Video Indexer.

Improved upload experience in the portal

Azure Video Indexer has a new upload experience in the [portal](#). To upload your media file, press the **Upload** button from the **Media files** tab.

New developer portal in available in gov-cloud

[Azure Video Indexer Developer Portal](#) is now also available in Azure for US Government.

Observed people tracing (preview)

Azure Video Indexer now detects observed people in videos and provides information such as the location of the person in the video frame and the exact timestamp (start, end) when a person appears. The API returns the bounding box coordinates (in pixels) for each person instance detected, including its confidence.

For example, if a video contains a person, the detect operation will list the person appearances together with their coordinates in the video frames. You can use this functionality to determine the person path in a video. It also lets you determine whether there are multiple instances of the same person in a video.

The newly added observed people tracing feature is available when indexing your file by choosing the **Advanced option -> Advanced video or Advanced video + audio** preset (under Video + audio indexing). Standard and basic indexing presets will not include this new advanced model.

When you choose to see Insights of your video on the Azure Video Indexer website, the Observed People Tracing will show up on the page with all detected people thumbnails. You can choose a thumbnail of a person and see where the person appears in the video player.

The feature is also available in the JSON file generated by Azure Video Indexer. For more information, see [Trace observed people in a video](#).

Detected acoustic events with Audio Effects Detection (preview)

You can now see the detected acoustic events in the closed captions file. The file can be downloaded from the Azure Video Indexer portal and is available as an artifact in the GetArtifact API.

Audio Effects Detection (preview) component detects various acoustics events and classifies them into different acoustic categories (such as Gunshot, Screaming, Crowd Reaction and more). For more information, see [Audio effects detection](#).

March 2021

Audio analysis

Audio analysis is available now in additional new bundle of audio features at different price point. The new

Basic Audio analysis preset provides a low-cost option to only extract speech transcription, translation and format output captions and subtitles. The **Basic Audio** preset will produce two separate meters on your bill, including a line for transcription and a separate line for caption and subtitle formatting. More information on the pricing, see the [Media Services pricing](#) page.

The newly added bundle is available when indexing or re-indexing your file by choosing the **Advanced option** -> **Basic Audio** preset (under the **Video + audio indexing** drop-down box).

New developer portal

Azure Video Indexer has a new [Developer Portal](#), try out the new Azure Video Indexer APIs and find all the relevant resources in one place: [GitHub repository](#), [Stack overflow](#), [Azure Video Indexer tech community](#) with relevant blog posts, [Azure Video Indexer FAQs](#), [User Voice](#) to provide your feedback and suggest features, and '[CodePen](#)' link with widgets code samples.

Advanced customization capabilities for insight widget

SDK is now available to embed Azure Video Indexer's insights widget in your own service and customize its style and data. The SDK supports the standard Azure Video Indexer insights widget and a fully customizable insights widget. Code sample is available in [Azure Video Indexer GitHub repository](#). With this advanced customization capabilities, solution developer can apply custom styling and bring customer's own AI data and present that in the insight widget (with or without Azure Video Indexer insights).

Azure Video Indexer deployed in the US North Central, US West and Canada Central

You can now create an Azure Video Indexer paid account in the US North Central, US West and Canada Central regions

New source languages support for speech-to-text (STT), translation and search

Azure Video Indexer now supports STT, translation and search in Danish ('da-DK'), Norwegian('nb-NO'), Swedish('sv-SE'), Finnish('fi-FI'), Canadian French ('fr-CA'), Thai('th-TH'), Arabic ('ar-BH', 'ar-EG', 'ar-IQ', 'ar-JO', 'ar-KW', 'ar-LB', 'ar-OM', 'ar-QA', 'ar-S', and 'ar-SY'), and Turkish('tr-TR'). Those languages are available in both API and Azure Video Indexer website.

Search by Topic in Azure Video Indexer Website

You can now use the search feature, at the top of the [Azure Video Indexer website](#) page, to search for videos with specific topics.

February 2021

Multiple account owners

Account owner role was added to Azure Video Indexer. You can add, change, and remove users; change their role. For details on how to share an account, see [Invite users](#).

Audio event detection (public preview)

NOTE

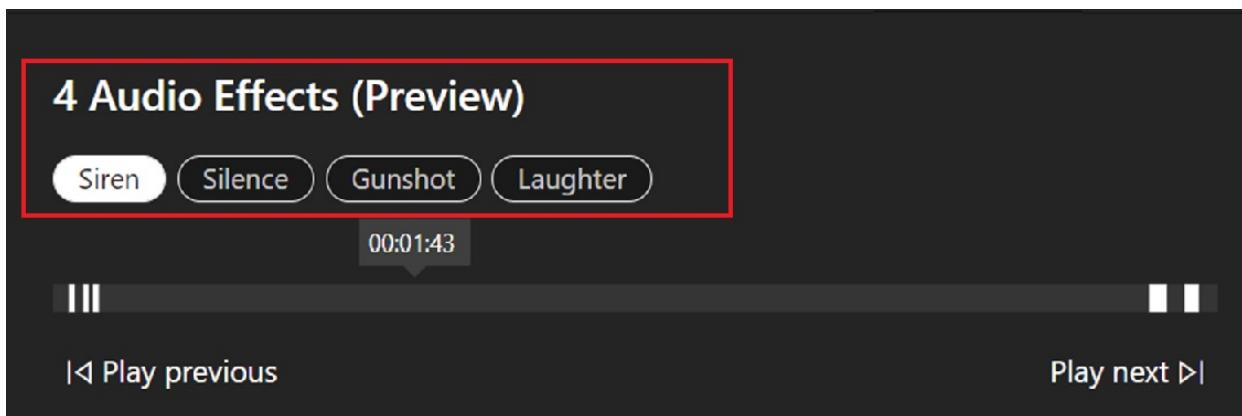
This feature is only available in trial accounts.

Azure Video Indexer now detects the following audio effects in the non-speech segments of the content: gunshot, glass shatter, alarm, siren, explosion, dog bark, screaming, laughter, crowd reactions (cheering, clapping, and booing) and Silence.

The newly added audio affects feature is available when indexing your file by choosing the **Advanced option** -> **Advanced audio** preset (under Video + audio indexing). Standard indexing will only include **silence** and **crowd reaction**.

The clapping event type that was included in the previous audio effects model, is now extracted a part of the crowd reaction event type.

When you choose to see **Insights** of your video on the [Azure Video Indexer](#) website, the Audio Effects show up on the page.



Named entities enhancement

The extracted list of people and location was extended and updated in general.

In addition, the model now includes people and locations in-context which are not famous, like a 'Sam' or 'Home' in the video.

January 2021

Azure Video Indexer is deployed on US Government cloud

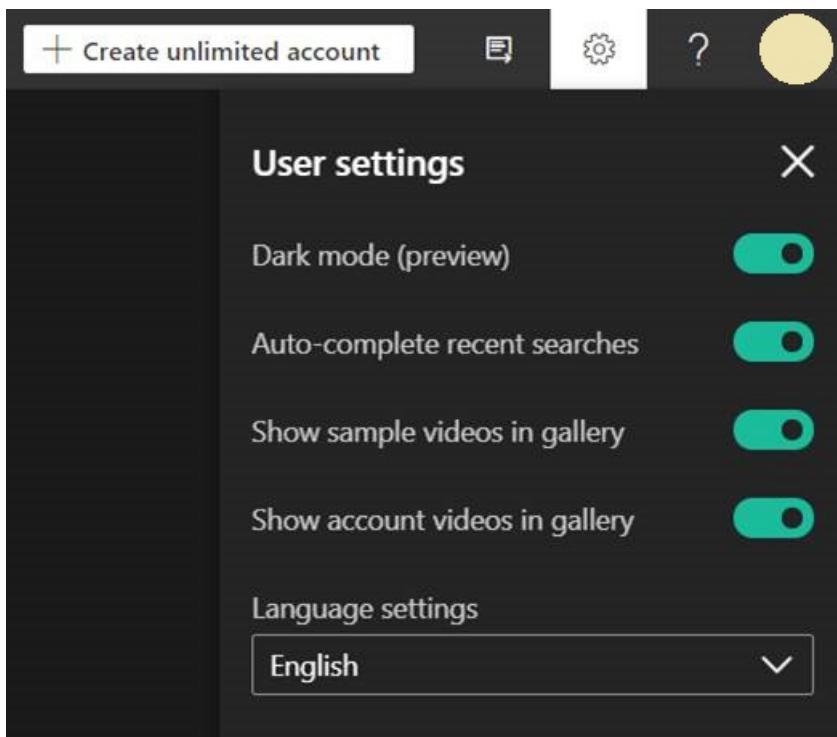
You can now create an Azure Video Indexer paid account on US government cloud in Virginia and Arizona regions. Azure Video Indexer trial offering isn't available in the mentioned region. For more information go to [Azure Video Indexer Documentation](#).

Azure Video Indexer deployed in the India Central region

You can now create an Azure Video Indexer paid account in the India Central region.

New Dark Mode for the Azure Video Indexer website experience

The Azure Video Indexer website experience is now available in dark mode. To enable the dark mode open the settings panel and toggle on the **Dark Mode** option.



December 2020

Azure Video Indexer deployed in the Switzerland West and Switzerland North

You can now create an Azure Video Indexer paid account in the Switzerland West and Switzerland North regions.

October 2020

Animated character identification improvements

Azure Video Indexer supports detection, grouping, and recognition of characters in animated content via integration with Cognitive Services custom vision. We added a major improvement to this AI algorithm in the detection and characters recognition, as a result insight accuracy and identified characters are significantly improved.

Planned Azure Video Indexer website authentication changes

Starting March 1st 2021, you no longer will be able to sign up and sign in to the [Azure Video Indexer website developer portal](#) using Facebook or LinkedIn.

You will be able to sign up and sign in using one of these providers: Azure AD, Microsoft, and Google.

NOTE

The Azure Video Indexer accounts connected to LinkedIn and Facebook will not be accessible after March 1st 2021.

You should [invite](#) an Azure AD, Microsoft, or Google email you own to the Azure Video Indexer account so you will still have access. You can add an additional owner of supported providers, as described in [invite](#).

Alternatively, you can create a paid account and migrate the data.

August 2020

Mobile design for the Azure Video Indexer website

The Azure Video Indexer website experience is now supporting mobile devices. The user experience is responsive to adapt to your mobile screen size (excluding customization UIs).

Accessibility improvements and bug fixes

As part of WCAG (Web Content Accessibility guidelines), the Azure Video Indexer website experience is aligned with grade C, as part of Microsoft Accessibility standards. Several bugs and improvements related to keyboard navigation, programmatic access, and screen reader were solved.

July 2020

GA for multi-language identification

Multi-language identification is moved from preview to GA and ready for productive use.

There is no pricing impact related to the "Preview to GA" transition.

Azure Video Indexer website improvements

Adjustments in the video gallery

New search bar for deep insights search with additional filtering capabilities was added. Search results were also enhanced.

New list view with ability to sort and manage video archive with multiple files.

New panel for easy selection and configuration

Side panel for easy selection and user configuration was added, allowing simple and quick account creation and sharing as well as setting configuration.

Side panel is also used for user preferences and help.

June 2020

Search by topics

You can now use the search API to search for videos with specific topics (API only).

Topics is added as part of the `textScope` (optional parameter). See [API](#) for details.

Labels enhancement

The label tagger was upgraded and now includes more visual labels that can be identified.

May 2020

Azure Video Indexer deployed in the East US

You can now create an Azure Video Indexer paid account in the East US region.

Azure Video Indexer URL

Azure Video Indexer regional endpoints were all unified to start only with www. No action item is required.

From now on, you reach www.videoindexer.ai whether it is for embedding widgets or logging into Azure Video Indexer web applications.

Also `wus.videoindexer.ai` would be redirected to `www`. More information is available in [Embed Azure Video Indexer widgets in your apps](#).

April 2020

New widget parameters capabilities

The **Insights** widget includes new parameters: `language` and `control`.

The **Player** widget has a new `locale` parameter. Both `locale` and `language` parameters control the player's language.

For more information, see the [widget types](#) section.

New player skin

A new player skin launched with updated design.

Prepare for upcoming changes

- Today, the following APIs return an account object:

- [Create-Paid-Account](#)
- [Get-Account](#)
- [Get-Accounts-Authorization](#)
- [Get-Accounts-With-Token](#)

The Account object has a `Ur1` field pointing to the location of the [Azure Video Indexer website](#). For paid accounts the `Ur1` field is currently pointing to an internal URL instead of the public website. In the coming weeks we will change it and return the [Azure Video Indexer website](#) URL for all accounts (trial and paid).

Do not use the internal URLs, you should be using the [Azure Video Indexer public APIs](#).

- If you are embedding Azure Video Indexer URLs in your applications and the URLs are not pointing to the [Azure Video Indexer website](#) or the Azure Video Indexer API endpoint (`https://api.videoindexer.ai`) but rather to a regional endpoint (for example, `https://wus2.videoindexer.ai`), regenerate the URLs.

You can do it by either:

- Replacing the URL with a URL pointing to the Azure Video Indexer widget APIs (for example, the [insights widget](#))
- Using the Azure Video Indexer website to generate a new embedded URL:

Press **Play** to get to your video's page -> click the **</> Embed** button -> copy the URL into your application:

The regional URLs are not supported and will be blocked in the coming weeks.

January 2020

Custom language support for additional languages

Azure Video Indexer now supports custom language models for `ar-SY`, `en-UK`, and `en-AU` (API only).

Delete account timeframe action update

Delete account action now deletes the account within 90 days instead of 48 hours.

New Azure Video Indexer GitHub repository

A new Azure Video Indexer GitHub with different projects, getting started guides and code samples is now available: <https://github.com/Azure-Samples/media-services-video-indexer>

Swagger update

Azure Video Indexer unified **authentications** and **operations** into a single [Azure Video Indexer OpenAPI Specification \(swagger\)](#). Developers can find the APIs in [Azure Video Indexer Developer Portal](#).

December 2019

Update transcript with the new API

Update a specific section in the transcript using the [Update-Video-Index](#) API.

Fix account configuration from the Azure Video Indexer portal

You can now update Media Services connection configuration in order to self-help with issues like:

- incorrect Azure Media Services resource
- password changes
- Media Services resources were moved between subscriptions

To fix the account configuration, in the Azure Video Indexer portal navigate to Settings > Account tab (as owner).

Configure the custom vision account

Configure the custom vision account on paid accounts using the Azure Video Indexer portal (previously, this was only supported by API). To do that, sign in to the Azure Video Indexer portal, choose Model Customization > Animated characters > Configure.

Scenes, shots and keyframes – now in one insight pane

Scenes, shots, and keyframes are now merged into one insight for easier consumption and navigation. When you select the desired scene you can see what shots and keyframes it consists of.

Notification about a long video name

When a video name is longer than 80 characters, Azure Video Indexer shows a descriptive error on upload.

Streaming endpoint is disabled notification

When streaming endpoint is disabled, Azure Video Indexer will show a descriptive error on the player page.

Error handling improvement

Status code 409 will now be returned from [Re-Index Video](#) and [Update Video Index](#) APIs in case a video is actively indexed, to prevent overriding the current re-index changes by accident.

November 2019

- Korean custom language models support

Azure Video Indexer now supports custom language models in Korean (`ko-KR`) in both the API and portal.

- New languages supported for speech-to-text (STT)

Azure Video Indexer APIs now support STT in Arabic Levantine (ar-SY), English UK dialect (en-GB), and English Australian dialect (en-AU).

For video upload, we replaced zh-HANS to zh-CN, both are supported but zh-CN is recommended and more accurate.

October 2019

- Search for animated characters in the gallery

When indexing animated characters, you can now search for them in the account's video galley. For more information, see [Animated characters recognition](#).

September 2019

Multiple advancements announced at IBC 2019:

- Animated character recognition (public preview)

Ability to detect group ad recognize characters in animated content, via integration with custom vision.

For more information, see [Animated character detection](#).

- Multi-language identification (public preview)

Detect segments in multiple languages in the audio track and create a multilingual transcript based on them. Initial support: English, Spanish, German and French. For more information, see [Automatically identify and transcribe multi-language content](#).

- Named entity extraction for People and Location

Extracts brands, locations, and people from speech and visual text via natural language processing (NLP).

- Editorial shot type classification

Tagging of shots with editorial types such as close up, medium shot, two shot, indoor, outdoor etc. For more information, see [Editorial shot type detection](#).

- Topic inferencing enhancement - now covering level 2

The topic inferencing model now supports deeper granularity of the IPTC taxonomy. Read full details at [Azure Media Services new AI-powered innovation](#).

August 2019 updates

Azure Video Indexer deployed in UK South

You can now create an Azure Video Indexer paid account in the UK south region.

New Editorial Shot Type insights available

New tags added to video shots provides editorial "shot types" to identify them with common editorial phrases used in the content creation workflow such as: extreme closeup, closeup, wide, medium, two shot, outdoor, indoor, left face and right face (Available in the JSON).

New People and Locations entities extraction available

Azure Video Indexer identifies named locations and people via natural language processing (NLP) from the video's OCR and transcription. Azure Video Indexer uses machine learning algorithm to recognize when specific locations (for example, the Eiffel Tower) or people (for example, John Doe) are being called out in a video.

Keyframes extraction in native resolution

Keyframes extracted by Azure Video Indexer are available in the original resolution of the video.

GA for training custom face models from images

Training faces from images moved from Preview mode to GA (available via API and in the portal).

NOTE

There is no pricing impact related to the "Preview to GA" transition.

Hide gallery toggle option

User can choose to hide the gallery tab from the portal (similar to hiding the samples tab).

Maximum URL size increased

Support for URL query string of 4096 (instead of 2048) on indexing a video.

Support for multi-lingual projects

Projects can now be created based on videos indexed in different languages (API only).

July 2019

Editor as a widget

The Azure Video Indexer AI-editor is now available as a widget to be embedded in customer applications.

Update custom language model from closed caption file from the portal

Customers can provide VTT, SRT, and TTML file formats as input for language models in the customization page of the portal.

June 2019

Azure Video Indexer deployed to Japan East

You can now create an Azure 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 Azure Video Indexer](#).

New download transcript formats – TXT and CSV

In addition to the closed captioning format already supported (SRT, VTT, and TTML), Azure Video Indexer now supports downloading the transcript in TXT and CSV formats.

Next steps

[Overview](#)