

Transcription

V-Spark 4.4.0-1

Medallia Voci

Contents

Transcription	3
Audio	4
Data flow	5
Format	7
Upload	10
UI upload	13
API upload	14
Download	15
Review	15
Delete	16
Audio metadata	17
Configuration.....	18
Format	20
Restrictions	22
Upload	27

Values.....	28
Folders	36
Administration.....	37
Create or modify a folder	39
Delete a folder	48
Configure ASR options	50
Configure callbacks.....	53
Folder-application links.....	60
Folder processing status	61
Callbacks.....	65

Transcription

Audio in V-Spark is transcribed and analyzed using folders. Topics in this section describe how to manage folders, and how to use them to process audio and audio metadata.

Audio

V-Spark processing begins with audio data, but the eventual database record entails much more. For V-Spark, the phrase *audio record* denotes that database record, which includes the initial audio data along with its JSON transcript, metadata, and analytics data. This record may be referenced by two attributes:

- `requestID` – generated when a file or batch of files is first uploaded.
- `transcriptID` – generated when a single audio file is submitted to the ASR engine for transcription.

Records for audio files submitted together share a `requestID`, but individual audio records always have unique `transcriptID`s.

NOTE: Before any files can be transcribed or analyzed, company, organization, and folder entities must be created and configured for processing audio.

Data flow

V-Spark is highly configurable for a wide variety of uses cases. The following generalized workflow does not cover every use case, but it illustrates how audio data moves through the system.

1. Audio and metadata file preparation.
 - Ensure that audio is compatible with V-Spark's built-in audio evaluation tool. V-Spark supports one- or two-channel audio only; audio with more than two channels will be rejected.
 - Ensure metadata is **correctly formatted** and does not violate any **system metadata restrictions**.
2. Audio and metadata upload.
 - Audio may be uploaded with the UI or with the API.
 - Custom metadata may be uploaded with audio data in a zip, or after the audio data in a separate upload.
3. V-Spark queues the audio for transcription.
4. ASR transcription.
 - ASR settings are configured for each folder.
 - Available options depend on the folder's company and organization configuration and licensing.
5. Transcription output queued for analysis by V-Spark.
6. **JSON transcript analytics processing**.
 - **Basic metrics** provided for all transcripts.
 - **Custom analysis** performed by applications depending on configuration.
7. Results available for download.
 - a. If callbacks are enabled, V-Spark sends results to the configured destination.

Other important workflows:

1. Reprocessing audio
2. **Re-scoring transcripts**
3. Deleting audio, transcription, and analytics data
4. **Scheduled and ad-hoc report generation**

Format

V-Spark's audio support is robust. The best way to check audio compatibility and properties is with V-Spark's built-in audio evaluation tool. See [Improving Transcription Accuracy](#) in the [V-Blaze User Guide](#) for more information about how audio's format and other properties impact ASR transcription.

Types

V-Spark converts audio before passing it to the ASR engine for transcription, and so supports a wide variety of audio file formats. That said, audio attributes have a significant impact on the accuracy of ASR transcription, and conversion cannot account for voice data lost due to suboptimal recording and encoding. The level of accuracy for a given transcript affects analytics performance, making audio format and properties key considerations.

The best format for audio submitted for transcription and analysis is lossless G.711 WAV (PCM, uLaw, or aLaw).

Channels

Audio submitted for transcription and analysis must have one or two channels. The number of channels in source audio, along with how those channels are used, affects V-Spark's ability to distinguish between speaker roles. In most cases, these roles are agent and client, and distinguishing between the two is critical for transcript analysis.

Transcription and analysis work best with two-channel (stereo) audio that has each speaker role on a separate channel. Audio with more than one speaker on the same channel may be diarized, a process that separates the audio into two channels and assigns each speaker to a different channel.


IMPORTANT: V-Spark does not support audio with more than 2 channels.

Evaluate

Use V-Spark's built-in audio evaluation tool to verify audio properties, and to configure folder settings to match those properties.

The audio evaluator shows audio properties for the uploaded file, including the number of channels, and whether the audio is supported by V-Spark. Files submitted for evaluation are not saved.

To use the audio evaluator:

1. Click **Will my audio work?** on the V-Spark homepage or in the **General** section of the  **Help** page.
2. Click **Choose File** and select the file to be analyzed.
3. Click **Evaluate**. The file uploads and evaluation results display. The following example shows the results for a compatible file:

Will my audio work?

350+ audio and video formats are supported.

To ensure that your audio is compatible, you may upload a sample for inspection.

Using the "Choose File" button below, select a single audio file and click "Evaluate".

Congratulations!

Your file is supported.

File	Test1.wav
Channels	2 (Stereo)
Sample rate	8,000 KHz
Bit rate	256.00 Kbps
Duration	428 sec
Format	PCM
Supported	Yes

[Test another file](#) or [Upload audio](#).

IMPORTANT: The values displayed for **Supported** and **Channels** must match the configuration of the V-Spark folder that will process the audio the evaluated sample represents. If your audio is not supported, contact the Voci support team.

Upload

Use the UI or the API's `/transcribe` endpoint to upload audio and metadata files to folders configured to process audio with properties that match your upload data.

Audio may be uploaded:

- As individual audio files.
- Individually in a zip file.
- In a batch of other audio files in a zip.
- In a batch of other audio and JSON or XML metadata files in a zip.
- With or without a file extension.

Uploaded files that are not supported are flagged as `UNSUPPORTED` in the folder's processing log.

Reprocessing transcripts

Previously processed JSON transcripts can be re-uploaded to any folder in V-Spark using these methods without the ASR engine needing to re-transcribe the files.

To be reprocessed, the uploaded JSON transcript must have the same filename as the original audio file (except for the extension). For example, to reprocess the transcript for audio uploaded as `File1.wav`, you must upload a JSON transcript with the filename `File1.json`.

Filenames

The names of uploaded audio files and zip archives must adhere to the installation's filename requirements whether they are uploaded through the UI or with the API. By default, these characters are not permitted in uploaded filenames: `#*<>:~/\|{}$!'`"=^`

When uploading a zip file, only the name of the zip file is validated against these requirements; files inside the zip are not checked. This feature

was first implemented with version 4.0.1-3 to help protect against remote code executions.

To disable filename validation, set the `filename_validation` system configuration setting to `off`. To define custom filename character requirements, specify a regular expression for the `filename_validation_pattern` system configuration setting.

Duplicate rejection

Files are not required to have unique names at a system level, but as of V-Spark 4.2.0-1, individual folders may be configured to reject files with duplicate filenames. In either case, filenames should be unique as a best practice. Consider adding the file's timestamp, call ID, or a UUID to create a unique filename. Duplicate filenames make some processing take longer.

NOTE: Independently of the duplicate rejection setting, if two files with identical names are submitted to the same folder at the exact same second, only one of those files will be processed.

When a folder has the deduplication setting enabled, that folder will reject file uploads in the following scenarios:

- A file is uploaded with the same name as a previously uploaded file.
- A zip file contains a file with the same name as a previously uploaded file.
- A zip file contains two or more files with the same name.

Filename-based deduplication may fail if the first file is still in the Job Manager queue waiting to be processed.


The entire zip file is rejected when a duplicate file is detected inside the zip. Duplicate file rejection for zip files nested inside other zip files is not supported. When folder-level deduplication causes a file to be rejected, V-Spark generates a `WARNING`-level message in `server.log` and the **Activity Log**.

A request submitted to the `/transcribe` API endpoint with an invalid filename parameter fails and returns HTTP error code 422.

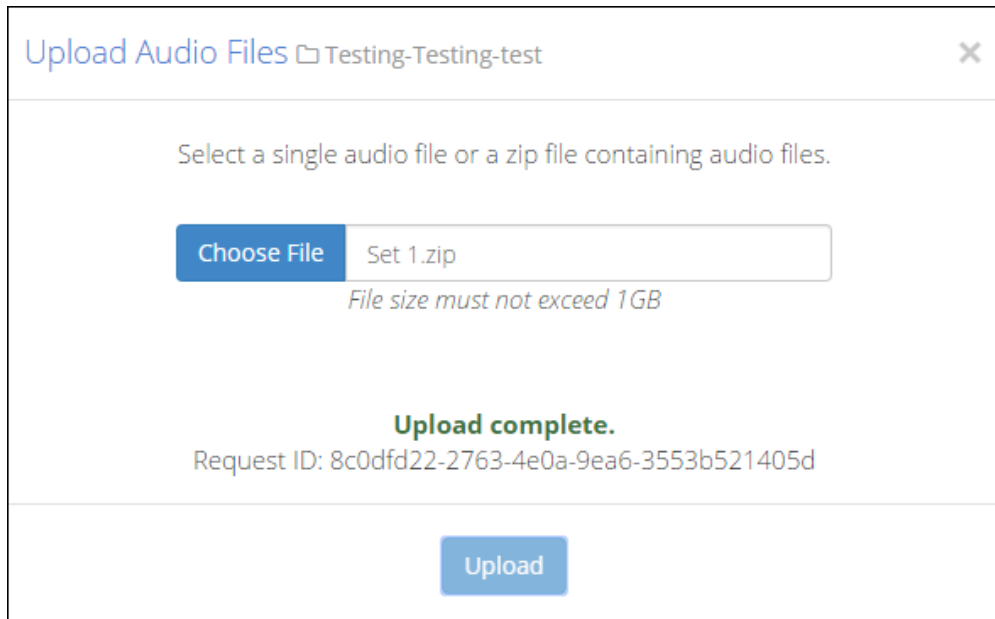
Metadata

To associate custom metadata uploads with the correct audio, the metadata and audio files must have the same name except for the extension.

UI upload

Use the blue  **Upload** icon for a folder with your audio's properties on the **Monitored Folders** to upload audio to V-Spark. File size is limited to 1 GB. Once a file has been uploaded, navigate to the **Dashboard** page and select the upload folder from the breadcrumb menu. After a few minutes the uploaded audio data appears in the folder dashboard.

Upon successful upload, the request ID for the job appears in the **Upload Audio Files** dialog, as shown in the following example:



This request ID will also appear in the JSON transcript, enabling you to track each transcript back to the original upload, view the request ID on the **File details** page, and use the request ID as a search term.

API upload

Audio files can also be uploaded using a POST request made to V-Spark's `/transcribe` API with the audio data to be decoded.

NOTE: By default, the maximum size of a file submitted using the `/transcribe` API is 250 MB. The maximum upload size can be changed using the `transcribe_api_upload_limit` system configuration option, but its value (specified in bytes) may not exceed 10 GB.

The POST must be encoded as a multi-part/form-data request with an audio file and token in their respective fields.

The following example POSTs to the V-Spark host `example.company.com`:

```
curl -F token=00112233445566778899aabbccddeeff \  
-F "file=@path/to/sampleaudio.zip;type= application/zip" \  
-X POST http://example.company.com:3000/transcribe/org_shortname/samplefolder
```

Users who have Amazon Web Services (AWS) Simple Storage Service (S3) configured can also import zipped input directly from S3 via the V-Spark API, as in this example:

```
curl -F token=0123456789abcde0123456789abcde01 \  
-F aws_id=012345678901234567890 \  
-F aws_secret=01234567890123456789012345678901234567890 \  
-F s3key=s3://example.company.com/documentation-TEST.zip \  
-F region=us-east-1 \  
-X POST http://example.company.com:3000/transcribe/Test-Testing/Test01
```

Multi-file uploads

When using the `/transcribe` API, single audio files and JSON transcripts can be submitted individually. Files submitted individually will not

be associated with each other.

Multiple files can be submitted in a single POST request, but they must be in a zip. These zip files can contain both audio data and metadata. Audio files and metadata files submitted as parts of a zip file will remain associated with each other as parts of a single submission.

Download

Bulk download of MP3s, JSON files, and text transcripts are available via the orange download icon shown in Administration. Downloads are sent as one or more zip file(s), with up to 10,000 files per zip file.

NOTE: The maximum size of this zip file is 2 GB. Trying to download files that produce a zip file larger than 2 GB will fail with an error message. To work around this issue, you can download multiple sets of files, using a smaller date range for each or by downloading specific types of files together.

Review

Delete

Delete audio files from V-Spark using the UI or with the API using the `/search` endpoint's `DELETE` method. Once the deletion is confirmed or the API call is made, V-Spark queues the file for deletion.

Deletion entails the removal of the audio file, its transcript and transcription results, and the rest of its system record. Summary charts and tables are not updated when an individual file is removed, but the file will no longer be available in the [📄 Dashboard Files View](#), nor will it appear in search results. A system under heavy load may take several minutes to fully delete the file and its record details, but this situation is unlikely.

NOTE: To delete audio using the UI, users must have write permissions for that audio record's organization.

To delete an individual audio record with the UI:

1. Navigate to the [📄 Dashboard Files View](#).
2. Find the file to be deleted using the **Company** and **Organization** dropdowns or using a search.
3. Click the **✕ Delete** icon for the audio file to be deleted. A confirmation dialog displays.
4. Once confirmed, V-Spark queues the file for deletion.

Audio metadata

Audio files may include standard metadata, such as agent identification and recorded date. Audio files may also include custom metadata; usually, this metadata is defined by the customer using the recording system.

The following is a general workflow for using custom metadata when processing audio files:

1. Establish metadata requirements for your processing flow. Take note of how special fields (`agentid` , `datetime`) are populated during processing and whether any additional custom fields will be required for your use case.
2. Configure a folder's custom metadata fields individually using the UI or in a batch by uploading a file.
3. Prepare the metadata files in JSON or XML format, noting metadata restrictions and formatting and syntax requirements.
4. Upload metadata files either individually or with audio using the V-Spark UI or with the `/transcribe` endpoint.
5. Search or view the custom metadata using the UI or in transcription output.

Configuration

When you create or edit an existing folder, you can configure custom metadata fields for that folder as shown in the following example of the **Update Metadata field names** screen:

Update customMetadata ×

+ ↑


Custom metadata field names:


client name × phone number ×

Back Update

To add custom metadata fields to a folder:

1. Click **Settings > Folders**. A list of all monitored folders displays.

2. To update an existing folder, click the  **Edit** icon on the row with the folder you want to edit. To create a new folder, click the **+ Add** icon.
3. On the second page of **Create/Edit Folder** window, enable **Add/remove custom metadata fields** and click **Next** to display the **Update Metadata field names** dialog.
4. To add a metadata field name, type it into the **Enter metadata field name** field and click the **+ Add** button or press enter.

To upload a file with multiple field names, click the  **Upload** button and add the file from the browser. The file should contain the name of one metadata field in plain text on each line. For example:

```
client-phone  
client-name  
account-number  
agent-name
```

All added field names are listed under **Custom metadata field names**. To remove a field name, click its **✕ Delete** icon.

NOTE:

As a best practice, do not preface custom metadata field names with **=**, **+**, **-**, or **@**. These characters may cause unexpected system behavior:

- Field names prefaced with **+** or **-** are not included in search results.
- Field names beginning with **=**, **+**, **-**, or **@** are prefaced with an apostrophe **'** in CSV exports to prevent spreadsheet applications from interpreting such labels as formulas.

5. Save the list of custom metadata tags by clicking the **Create** or **Update** button. Changes will not be saved if you navigate away from the window before saving.

Format

V-Spark accepts metadata in JSON or XML format. Metadata files must have the same filename as the audio to which they apply, but with `.xml` or `.json` extension in place of the original extension. For example, an audio file named `ABC.wav` could be uploaded with a metadata file named `ABC.json` or `ABC.xml`.

When uploaded with previously processed JSON transcripts, metadata files must use the `.jsonmeta` extension. JSON and XML files must be formatted as shown in the examples below; V-Spark does not do any error correction on metadata files.

NOTE: Metadata timestamps without time zone information are interpreted with the organization's time zone and *not* the system time zone.

The following is an example of the XML format:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
  <metadata>
    <agentid>10005</agentid>
    <datetime>2015-06-02 13:29:20</datetime>
    <agentname>John Smith</agentname>
    <client-phone>123-456-7890</client-phone>
  </metadata>
```

The following is an example of the JSON format:

```
{
  "metadata":
    {
      "agentid": "10005",
      "datetime": "2015-06-02 13:29:20",
```

```
    "agentname": "John Smith",  
    "client-phone": "123-456-7890"  
  }  
}
```

NOTE: A leading `-` in custom metadata is reserved for exclude phrases. As a result, any leading `-` at the beginning of custom metadata values is replaced with `#`. For example, if you have a field name for Call Direction with a metadata value of `-inbound`, the metadata value will be changed to `#inbound`.

Restrictions

All metadata uploaded to V-Spark has the following restrictions:

- All metadata field names are case-insensitive, including reserved and custom field names. For example, AgentID , agentid , and AGENTID are equivalent metadata field names.
- Custom metadata values and field names may be composed of any valid Unicode characters, except for the reserved non-alphanumeric characters : (colon, Unicode 003A) and = (equals sign, Unicode 003D).
- If you upload metadata via XML rather than JSON, your metadata field names cannot contain spaces. Metadata values can contain spaces in both XML and JSON.

NOTE:

As a best practice, do not preface custom metadata field names with = , + , - , or @ . These characters may cause unexpected system behavior:

- Field names prefaced with + or - are not included in search results.
- Field names beginning with = , + , - , or @ are prefaced with an apostrophe ' in CSV exports to prevent spreadsheet applications from interpreting such labels as formulas.

Reserved fields

The following field names are reserved for V-Spark functions. Key values for these fields may be submitted as metadata, but they cannot be configured as custom metadata fields for folders.

- _all
- _id

- agent
- agent id
- agent_v3
- agentid
- all
- app_data
- asr
- audio_duration
- audio_properties
- audiosecs
- call_tag
- caller
- caller_v3
- client
- client_data
- confidence
- datetime
- diarization_score

- doc id
- donedate
- duration
- ended
- file
- filenamelowercase
- gender
- jm_version
- model
- nchannels
- nsubs
- performance_stats
- recvdate
- recvtz
- request id
- requestid
- sentiment
- sentiment_scores

- source
- speakers
- started
- streamtags
- tag
- tid
- transcript id
- utterances

Although reserved field names may not be configured for folders, metadata **values** for those fields may be submitted in uploaded JSON as part of a reprocessing request. With the exception of the `agentid`, `datetime`, and `duration` fields, any submitted metadata values appear in the `client_data` section of the audio record's JSON output whether or not those metadata values were indexed.

WARNING: Key values supplied for `audio_properties`, `diarization_score`, and `jm_version` are overwritten by the system. Any submitted values for these fields may be lost.

Using reserved fields

The key values for some reserved metadata fields may be client-supplied. For call center environments, values for `agentid` and `datetime` should be submitted as a best practice as they are very commonly used. Two additional fields, `audio_duration` and `duration`, may also be submitted, but should not be as a best practice. Overriding or failing to provide values for these fields as a part of processing flow may result in abnormal behavior as noted in the following list:

agentid

- Variable length string limited to 100 characters.
- The only searchable identifier for a single agent; required for applications to create and reference agent scorecards.
- If a value is supplied for `agent_id`, its value is interpreted in the same manner and stored in the `agentid` field.
- If blank ("" or NULL), `agentid` 's value displays as `None` .

datetime

- Represents the audio's recording date and time.
- If this value is not provided with audio data, the system ingestion time becomes the call's date of record.
- String that may be expressed in any format compatible with the [dateutil](#) module for the standard Python module `datetime` .
- In V-Spark examples, expressed as YYYY-MM-DD HH:MM:SS.
- Always assumed to be in the organization's time zone.

audio_duration and duration

- Represents the audio recording's duration in seconds.
- Users may override the system-generated value by providing any plain integer value for either the `audio_duration` or `duration` fields. The `duration` field takes precedence if both fields are provided.
- Overriding this field is not recommended. Doing so is usually not necessary, and updating the field to the actual duration value requires reprocessing the audio.
- To ignore client-supplied `duration` metadata values by default, enable the `ignore_client_duration` system configuration

option.

NOTE: The `agentid`, `audio_duration`, `datetime`, and `duration` fields cannot be configured as custom metadata fields for a folder in any case because they are used by the system.

Upload

Metadata can be uploaded by the same methods as audio files. Metadata values for a particular transcript may also be replaced using the `/metadata` API endpoint.



Processing metadata with audio ensures that, with some restrictions and exceptions, client-submitted metadata fields and their key values will be listed in the `client_data` portion of the audio record's JSON transcript. Client-submitted metadata keys will not be indexed and searchable for that record **unless** those fields are configured as custom metadata fields for the audio record's folder, or unless they are standard system metadata fields needed for processing.

To upload metadata and audio together, place both sets of files in the same zip file and import the zip file using the upload method of your choice. The audio and metadata file must have the same name, not including the file extension. You **must** use zip files to upload metadata and audio together in a clustered V-Spark implementation.

NOTE: Metadata files uploaded with previously processed transcripts must use the extension `.jsonmeta`.

Values


Fields and values for standard and custom metadata for individual transcripts can be viewed in four places:

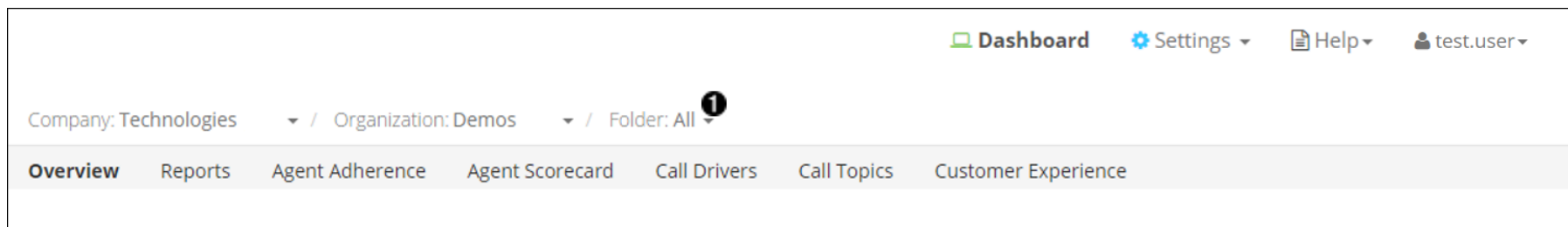
- Search results in the  **Dashboard Files View**.
- CSV reports created using the  **Dashboard Files View**.
- A transcript's **File Details** page.
- A transcript's JSON output.

NOTE: NaN ("not a number"), +Infinity, and -Infinity values for custom metadata fields are stored as strings. Negative infinity values begin with a hyphen (-).

On the **File Details** page, hyphens are displayed as hash symbols (#), so -Infinity values appear as #Infinity.

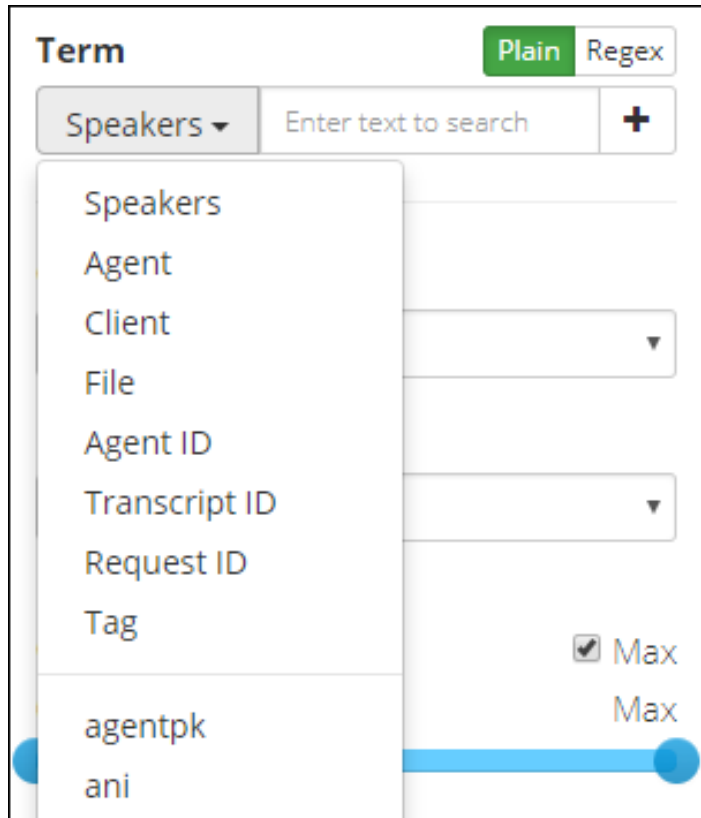
Search results

1. Navigate to the  **Dashboard** and select the folder whose metadata you want to view from the breadcrumb menu shown in the following example:



2. Use the **Term** dropdown to select the metadata field for the search, and enter the field values to search for in the textbox. Custom

metadata field names appear at the end of the **Term** list, as with the `agentpk` and `ani` entries in the following example:



- Files that contain metadata fields and values matching the search display with the metadata field and value listed at the bottom of each search result entry, as with the `field0` entries in the following example:

Term Plain Regex

field0 +

- Speakers
- Agent
- Client
- File
- Agent ID
- Transcript ID
- Request ID
- Tag

Max

field0

field1

field2

field3

field4

field5

Search Terms field0: usa

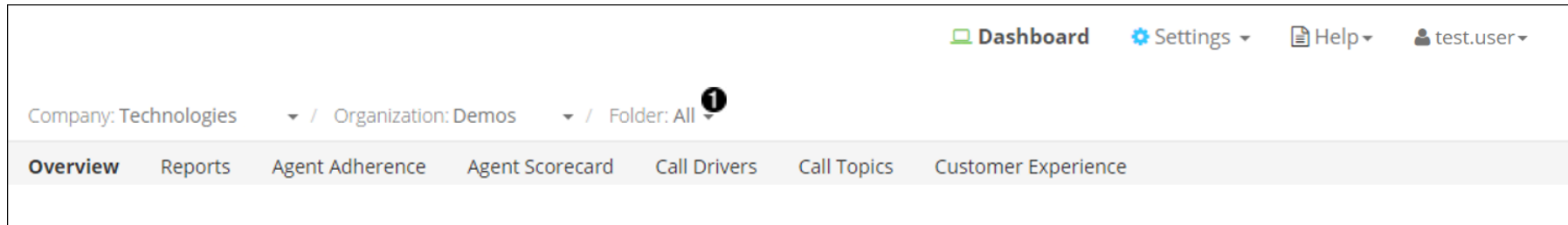
Showing 1 to 10 of 16 entries

File Name	Agent ID	Time (US/Eastern)	Duration
x 759925783659.wav	701	2021-11-22 2:38:28 pm	0:01:18
field0: usa			
x 759925733067.wav		2021-11-18 5:16:34 pm	0:01:07
field0: usa			

CSV reports

1. Navigate to the [Dashboard](#) and select the folder whose metadata you want to view from the breadcrumb menu shown in the

following example:

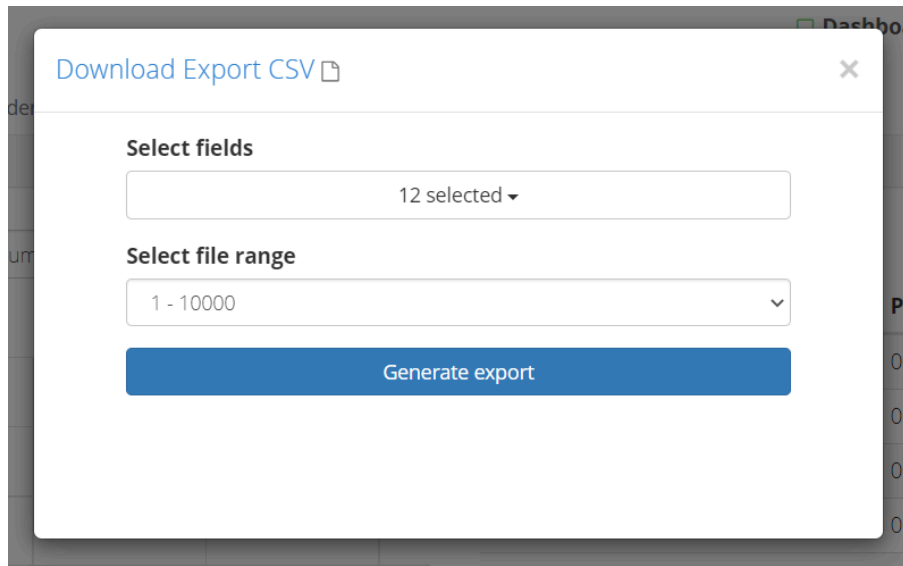


2. Use the **Show/hide columns** dropdown in the **Files View** to enable the metadata fields to display. The dropdown shows all standard and custom metadata fields for the selected folders, as shown in the following example:

The screenshot displays the Voci V-Spark search interface. On the left, there are filters for Term, Gender, Emotion, Call Duration, Call Silence Time, and Client Voice Clarity. The main area shows a table of search results with columns for File Name, Agent ID, Time (US/Eastern), Duration, Request ID, Score, field0, and field1. Two entries are shown, both with a score of 1.0000. An 'Export (CSV)' button is visible above the table. A dialog box titled '(9) Show / Hide columns' is open on the right, showing a list of metadata fields with checkboxes. The 'Standard metadata' section includes Agent ID, Time, Duration, Silence Time, Overtalk, Diarization, Agent Clarity, Client Clarity, Agent Gender, Client Gender, Overall Emotion, Agent Emotion, Client Emotion, and Transcript ID. The 'Custom metadata' section includes field0, field1, and field2. The 'Request ID' and 'Score' fields are checked in the Standard metadata section.

File Name	Agent ID	Time (US/Eastern)	Duration	Request ID	Score	field0	field1
759925733067.wav		2021-11-19 4:27:46 pm	0:01:07	f34d7902-d351-451c-bbd6-2e0c765faca9	1.0000	usa	sela
759925783659.wav		2021-11-19 4:27:46 pm	0:01:18	f34d7902-d351-451c-bbd6-2e0c765faca9	1.0000	usa	swla


- Columns for each enabled field appear in search results and in CSV report exports. To export results for the files shown, click **Export (CSV)**.
- The **Download Export CSV** dialog appears, as shown in the following example:



5. Verify the enabled metadata fields and alter them as needed, select the file range to be included in the report, and click **Generate export**.

File Details page

1. Navigate to the File Details page of a transcript whose metadata you want to view.
2. The **File Details** page lists all file metadata, with custom fields at the end of the list, as with the `agentpk` and `ani` fields in the

Duration	0:16:01
Silence	34.8 %
Overtalk	3.1 %
Agent Clarity	88.4 %
Client Clarity	86.3 %
Agent Gender	male
Client Gender	female
Overall Emotion	Improving
Agent Emotion	Improving
Client Emotion	Improving
<hr/>	
Expand all 	
agentpk	043
ani	0000000419

following example:

JSON transcript

1. Navigate to the File Details page of a transcript whose metadata you want to view.
2. Click the **JSON** link to download a JSON version of the transcript.
3. Custom metadata will appear in the `client_data` section of the JSON transcript. It will resemble the following example:

```
"client_data": {  
  "manager_name": "Melissa Thompson",
```

```
"total_hold_time": "25",  
"skill": "0",  
...  
}
```

Folders

A V-Spark folder is an abstract container that defines a set of ASR, application, and metadata configuration settings for processing and transcribing audio and analyzing transcription results. Folders enable specific ASR configurations for specific sets of files.

A folder is associated with and controlled by a single organization, and contains a subset of that organization's audio files and transcription records. Folders might be used, for example, to store audio from a specific source or time period.

Administration









Click **Settings > Folders** to manage folders with the **Monitored Folders** list, a sortable table of all folders that updates every minute. Use the **Company** and **Organization** breadcrumb dropdowns to view folders associated with those entities. The following example shows the **Monitored Folders** page and lists its functionality:

The screenshot displays the 'Monitored Folders' page. At the top, there are navigation links for 'Dashboard', 'Settings', 'Help', and a user profile 'anewuser'. Below the navigation, there are breadcrumb dropdowns for 'Company: All' and 'Organization: All'. The main content area is titled 'Monitored Folders' and shows a table of folders. The table has columns for 'Name', 'Organization', 'Server', 'Model (Channel #)', '# Speakers', 'Audio Type', and 'Created'. Each row represents a folder, and the 'Server' column contains a URL. The 'Created' column shows the date '2021-04-22'. The table is annotated with numbered callouts (1-10) pointing to various UI elements: 1. Refresh icon, 2. Search bar, 3. Add button, 4. Edit icon, 5. ASR processing status icon, 6. Server field, 7. Play/Pause icon, 8. Dashboard icon, 9. Download icon, 10. Close icon.

Name	Organization	Server	Model (Channel #)	# Speakers	Audio Type	Created
folder1	orgA (co1)	https://asr.office.vocitec.com	eng1:callcenter (all)	1	Mono	2021-04-22
folder2	orgA (co1)	https://asr.office.vocitec.com	eng1:callcenter (all)	1	Mono	2021-04-22
folder3	orgB (co1)	https://asr.office.vocitec.com	eng1:callcenter (all)	1	Mono	2021-04-22
folder4	orgB (co1)	https://asr.office.vocitec.com	eng1:callcenter (all)	1	Mono	2021-04-22
folder5	orgC (co1)	https://asr.office.vocitec.com	eng1:callcenter (all)	1	Mono	2021-04-22

1. **Refresh** the ASR **Server** fields for all folders.
2. **Search** all folders. Search terms are queried against the text of data fields shown on the **Monitored Folders** page.
3. **+** **Add** a new folder.
4. **Edit** the folder's configuration settings or **View** the folder's analytics reports and individual audio records on the **Dashboard**.
5. Shows ASR processing status. A green circle **●** indicates that the folder has successfully processed files. A black circle **●** indicates that

it has been more than one month since the folder last processed files. Hover over the indicator to display the time at which the last file was successfully processed.

6. Shows the folder's ASR server name and authorization status. A green  **Connected** icon indicates an active, authorized connection. A red  **Failed** icon indicates that connection or authorization failed.
7.  **Pause** or  **Resume** file processing for all files in the folder.
8.  **Show processing log** and ASR status for the folder.
9.  **Upload** audio, metadata, or transcript files to the folder, or  **Download** a zip archive of audio, metadata, and transcript files in the folder.
10.  **Delete** the folder.

Create or modify a folder

Folders may be created or modified using the UI, or with the API using the `/config` endpoint. Before using either method, V-Spark must have at least one company and one organization configured for ASR processing.

Dialog 1

1. Click **⚙ Settings > Folders** to navigate to the **Monitored Folders** screen.
2. Click **+** **Add**. The **Create New Folder** dialog displays.
3. Specify the settings displayed on the first page of the dialog, as in the following example:

Create New Folder

**Organization**

co1 ▾ / orgA ▾

Folder NameCan only contain A-Z 0-9 with **no spaces****Servers**

https://asr.office.vocitec.com ▾

of speakers

2 ▾

Audio Type

Stereo ▾

 PCI Redact Text PCI Redact Audio**Next**

IMPORTANT: Settings configured on the first page of the **Create New Folder** dialog can't be changed after the folder is created, except for the ASR server in the **Servers** dropdown. Changing audio settings requires creating a new folder. Check your audio's properties with V-Spark's built-in evaluation tool.

Use the first page of the **Create New Folder** dialog to configure the following options:

- a. **Organization** – Select the folder's company and organization. A folder's company and organization determine the availability of some of its ASR configuration settings.
- b. **Folder Name** – Enter a name for the new folder. Folder names are case sensitive, and must not be longer than 100 characters.
- c. **Servers** – Select the servers to be used for ASR processing. Multiple servers may be selected from the dropdown if available. Server options are based on the folder's company and organization configurations.

NOTE: Configuring a folder to use multiple ASR servers that do not use the same language models may result in an error message indicating that "One or more servers does not have this model anymore." This error will only impact ASR processing if the language model has been removed from all of the folder's servers.

- d. **# of speakers** – Select the number of speakers in the audio, either 1 or 2.
- e. **Audio Type** – Select either mono (single-channel) or stereo (two-channel) audio.

NOTE:

If a folder is configured with the **# of speakers** field set to **2** and the **Audio Type** field set to **Mono**, that folder's processed audio is separated, or diarized, into 2 channels. Each speaker is assigned to a different channel. Speakers in the resultant transcript are classified as either **Agent** or **Client** using side classification.

To prevent diarization of audio with more than two speakers, configure the folder for **Mono** audio with **1** speaker.

- f. **PCI Redact Text** – **Optional**. Select to enable redaction for text transcripts processed by this folder. Enabling text redaction causes the ASR engine to remove payment card information (PCI) and other sensitive numbers from text transcripts.
- g. **PCI Redact Audio** – **Optional**. Select to enable redaction for audio processed by this folder. Enabling audio redaction causes the ASR engine to remove PCI and other sensitive numbers from the processed audio file.

Dialog 2

1. Click **Next**. The dialog displays additional folder configuration options, as in the following example:

Create New Folder

**Link to Applications**

None selected ▾

Please be aware that applications with **custom metadata filters** may result in scores of 0 if linked folders do not share the custom metadata fields used.

Model

Channel 0 / Left

eng1:callcenter ▾

Channel 1 / Right

eng1:callcenter ▾

Which channel is the agent on?

Channel 0 / Left ▾

Advanced settings

- Configure callback delivery method [?](#)
- Add/remove custom metadata fields
- Add/remove ASR options
- Reject duplicate records by filename

Use the second page of the **Create New Folder** dialog to configure the following options:

Link to Applications – Select the applications to be used for analysis on the transcription output for all audio processed by the folder. See Folder-application links for more information.

Model – Select the language models to be used for audio transcription requested by the folder. One model must be selected for each audio channel, and channels may use either identical or distinct language models.

Model selection greatly impacts transcription accuracy because they are tuned to specific audio scenarios and environments, like voicemail or call centers.

Which channel is the agent on? – Select the audio channel for agent speech. This option is used only with stereo audio folder configuration. A value of `0` refers to the left audio channel, and a value of `1` refers to the right audio channel.

IMPORTANT: The agent channel must be set to the correct value for transcription analysis, and it cannot be changed after the folder is created.

Advanced settings – Enable and configure settings for callbacks, custom metadata, ASR options, and duplicate file rejection.

Enabling **Reject duplicate records by filename** prevents the folder from accepting any file with the same name as one that has already been uploaded. The remaining advanced settings must be configured after being enabled.


To configure advanced settings, enable them and click **Next**. An additional **Create New Folder** dialog displays for each enabled setting except for duplicate file rejection. The last page of the configuration dialog has a **Create** button.

Refer to the following topics for more information about configuring other advanced settings:

- Configure callbacks
- Configure custom metadata
- Configure ASR options

2. Click **Create** to close the configuration dialog and create the folder.
3. View the ASR server status field for the newly-created folder on the **Monitored Folders** screen. A green circle ● in the server status field indicates an active and authenticated connection which means the folder is ready to process audio.

Modify

To modify an existing folder, click its  **Edit** icon on the **Accounts** page. The **Update Folder** dialog appears with the same fields as the **Create New Folder** dialog. Some fields may not be modified after a folder is created.

Create a folder with POST and /config

Context

The following is an example workflow for using POST with `/config` to create or modify a folder:

Procedure

- 1 Acquire the authorization token, along with the company and organization short names for the folder to be created or modified. Find this information using the UI on the **Settings > Accounts** screen. The following must be supplied with any POST call to `/config`:

Token

A company API token with write permissions for the target organization.

Company short name

The reference configured for a company when it is first created.

Organization short name

The reference configured for an organization when it is first created.

Company and organization information may also be retrieved using GET with `/config`.

- 2 Prepare a JSON file with the folder's configuration parameters. Using POST to create folders with the `/config` endpoint requires the configuration of many folder properties; as a result, it is strongly recommended to POST a JSON file with the folder's required properties.
- 3 Make a POST request to the `/config` endpoint with the JSON file as a parameter. See the following topics for more information:

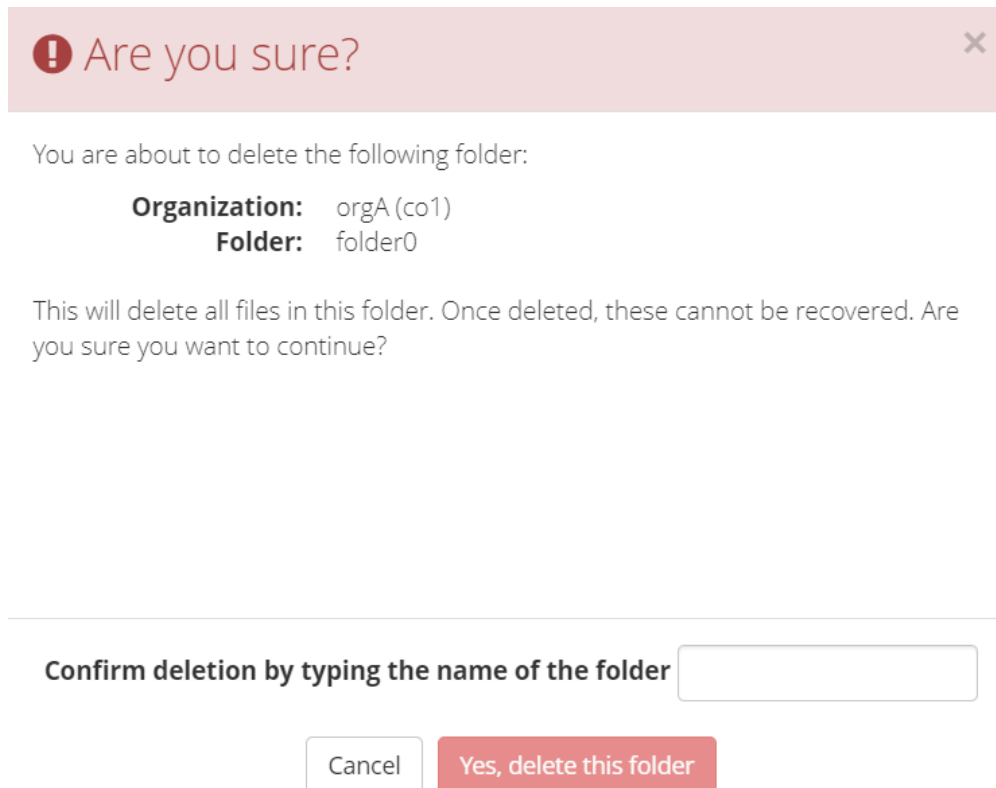
-
- 4 If the folder exists, its configuration is changed to the submitted parameters. If the folder does not exist, it is created with the submitted parameters.
- 5 Verify the folder's configuration using the **Monitored Folders** UI or a GET request to the `/config` endpoint. Refer to the following topics for more information:
 - Administration shows how to view a list of folders on the **Monitored Folders** screen.
 - demonstrates how to retrieve a folder's configuration settings using the API.
 - and demonstrate how to retrieve a folder's ASR processing status using the API.

Delete a folder

Deleting a V-Spark folder queues all the audio, transcription, and analytics data it contains for deletion. This action cannot be undone. Summary charts and reports are not updated when folders and their files are removed.

The following is a general workflow for deleting V-Spark folders:

1. Click **⚙ Settings > Folders** to navigate to the **Monitored Folders** screen.
2. Click **✕ Delete** next to the folder to be deleted. A confirmation dialog appears, as in the following example:



The image shows a confirmation dialog box with a red header bar containing an exclamation mark icon and the text "Are you sure?". Below the header, it says "You are about to delete the following folder:" followed by a list of details: "Organization: orgA (co1)" and "Folder: folder0". A warning message states: "This will delete all files in this folder. Once deleted, these cannot be recovered. Are you sure you want to continue?". At the bottom, there is a text input field with the label "Confirm deletion by typing the name of the folder" and two buttons: "Cancel" and "Yes, delete this folder".

! Are you sure? ✕

You are about to delete the following folder:

Organization: orgA (co1)
Folder: folder0

This will delete all files in this folder. Once deleted, these cannot be recovered. Are you sure you want to continue?

Confirm deletion by typing the name of the folder

3. Verify the dialog shows the company, organization, and name for the correct folder before you confirm deletion.
4. Enter the name of the folder and click **Yes, delete this folder** to queue the folder and its contents for deletion.

Configure ASR options

Context

Use the ASR options dialog screen to pass additional request parameters to the ASR engine that V-Spark uses for transcription. The options that may be specified here depend on the license and configuration of the ASR sources enabled for the folder. Parameters are represented as key-value pairs, such as the **numtrans: false** option in the following example:

Update folder9 ×

subst	:	false	+
-------	---	-------	---

ASR options:

numtrans: false ×

Back Update

NOTE:

Refer to the following topics for more information about parameters used with V-Blaze and V-Cloud ASR, and for examples of configuring specific ASR options for V-Spark folders:

- [V-Blaze transcription parameters](#)
- [V-Cloud transcription parameters](#)
- [Output tuning, substitutions, and redaction](#)

The following is a general workflow for adding ASR tags to V-Spark folders:

Procedure

- 1 Enter the key for the ASR parameter to be included with transcription requests.
- 2 Enter the value. The value for ASR option keys may be a literal JSON string or a redirect to a file on the V-Spark host system. To redirect to a file, enter < and the directory path.
- 3 Click **+** **Add** to add the key-value pair to the ASR options viewer. Pairs that are not displayed here are not saved when the folder is created or updated.
- 4 Click **Create** or **Update**.

Configure callbacks

Context

V-Spark folders use callbacks to send transcription and analysis results to a pre-configured location when processing is complete.

The following is a general workflow for configuring callbacks for V-Spark folders:

Procedure

- 1 Click  **Settings** > **Folders**.
- 2 Use the **Create New Folder** or **Update Folder** dialog to create or modify a folder.
- 3 Enable the **Configure callback delivery method** option on the second page of the dialog.
- 4 Click **Next** to navigate to callback configuration options.
- 5 Configure callbacks for the folder using the following fields:

Automatically send... – Enable any of the following output types:

JSON

A JSON version of the transcript and its associated metadata. JSON output is enabled by default.

MP3

A post-transcription MP3 version of the transcribed audio file. This MP3 contains redacted audio if redaction is enabled for the folder.

Text

A plain text version of the transcript.

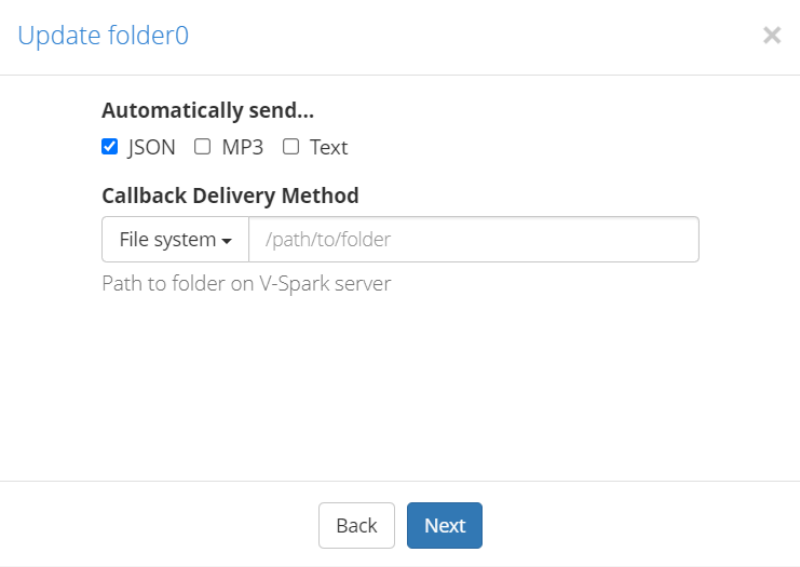
Callback Delivery Method – Use the dropdown to select a delivery method, and enter its path in the text box.

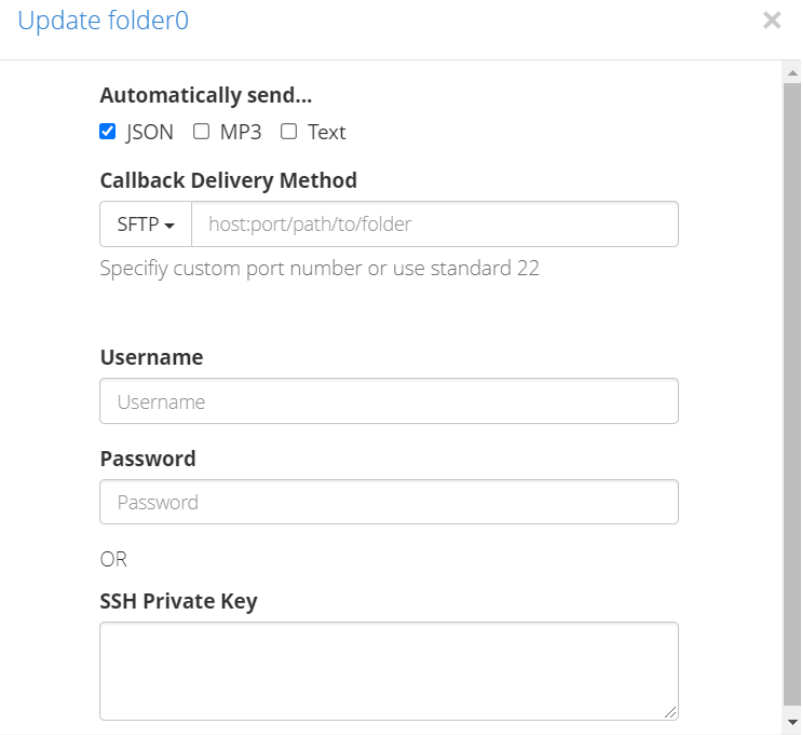
IMPORTANT: Make sure that the machine and port specified in the callback URL are accessible from the V-Spark instance that you are using. In some cases, this requires modifying client-side firewall rules.

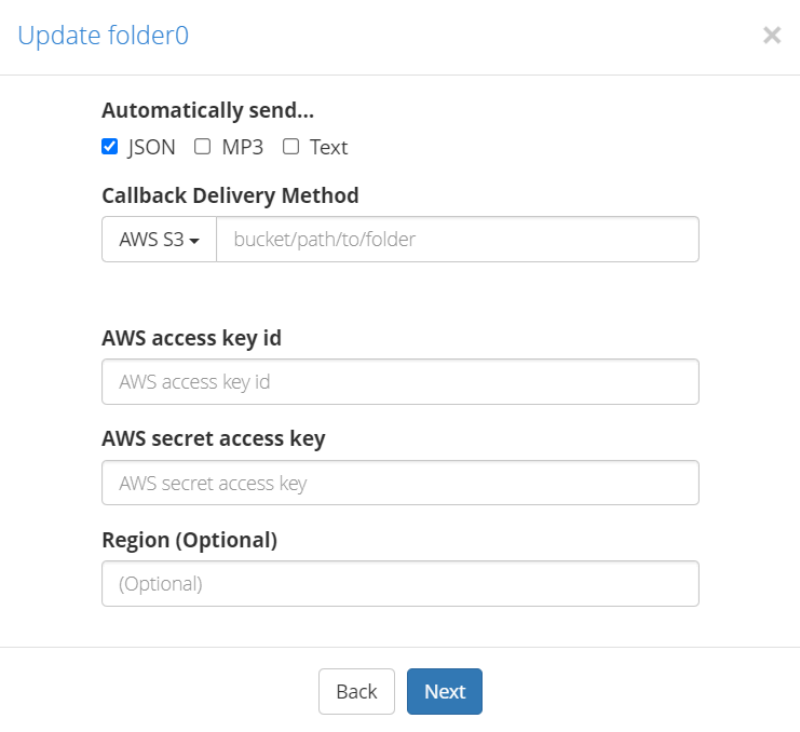
V-Spark supports the following delivery methods:

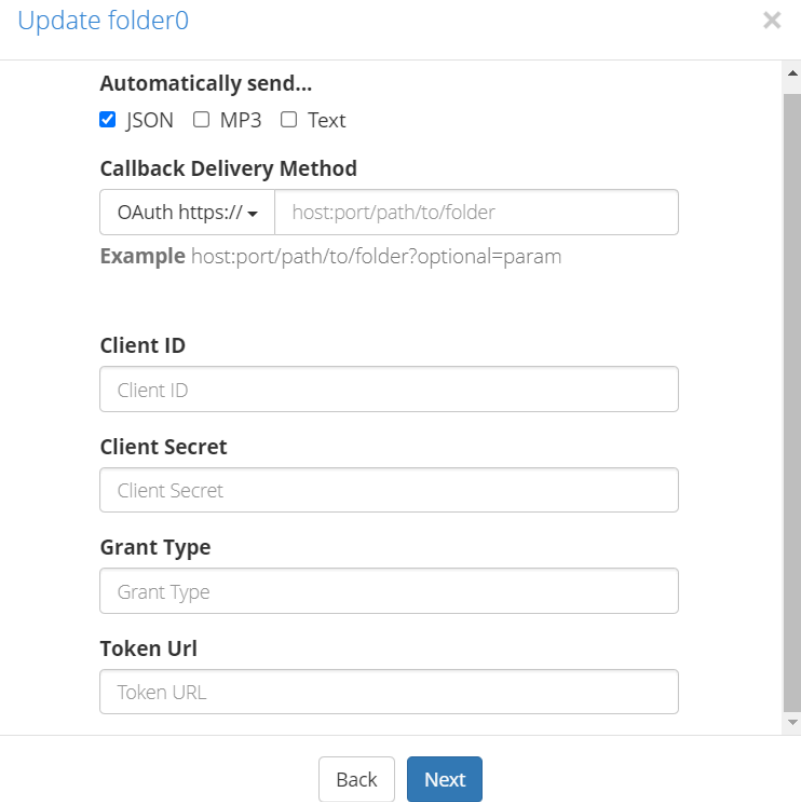
Callback Delivery Types

Delivery Method	Description	Required Parameters	Example
http:// and https://	Standard hypertext transfer protocol. Callbacks are sent to a web server listening on a specified or default port. HTTPS runs over Secure Socket Layer (SSL).	Delivery path	<div style="border: 1px solid #ccc; padding: 5px;"> <p style="margin: 0;">Update folder0 ×</p> <hr/> <p>Automatically send... <input checked="" type="checkbox"/> JSON <input type="checkbox"/> MP3 <input type="checkbox"/> Text</p> <p>Callback Delivery Method</p> <div style="border: 1px solid #ccc; padding: 2px; display: flex; align-items: center;"> http:// ▾ <input style="flex-grow: 1;" type="text" value="host:port/path/to/folder"/> </div> <p>Example username:password@host:port/path/to/folder?optional=param</p> </div> <div style="text-align: right; margin-top: 10px;"> Back Next </div>

Delivery Method	Description	Required Parameters	Example
File system	<p>Callback data is written to a specified location on the host system on which V-Spark is installed.</p> <p>IMPORTANT: When using the file system method, V-Spark's user group must be set to <code>vocisrv</code>, and that user group must have write permissions for the specified directory path.</p>	Local filesystem directory path	

Delivery Method	Description	Required Parameters	Example
SFTP	Callbacks are sent to a server using the Secure File Transfer Protocol (SFTP).	Username and password OR SSH private key	 <p>Update folder0</p> <p>Automatically send... <input checked="" type="checkbox"/> JSON <input type="checkbox"/> MP3 <input type="checkbox"/> Text</p> <p>Callback Delivery Method SFTP host:port/path/to/folder Specify custom port number or use standard 22</p> <p>Username Username</p> <p>Password Password</p> <p>OR</p> <p>SSH Private Key </p> <p>Back Next</p>

Delivery Method	Description	Required Parameters	Example
AWS S3	Callback data is written to the Amazon Web Service (AWS) Simple Storage Service (S3).	AWS access key ID and secret	

Delivery Method	Description	Required Parameters	Example
OAuth https://	As with standard HTTPS, callbacks are sent to a web server listening on a specified or default port, but the callback is authenticated with OAuth credentials instead of a username and password.	Client ID, client secret, grant type, and token URL	

Folder-application links


Linking a folder to an application means that the folder uses the application to analyze the text transcripts of all of that folder's processed audio. Only folders and applications within the same organization can be linked. A single folder may be linked to multiple applications, and a single application may be linked to multiple folders.

A folder-application link may be configured in either the folder's or the application's settings. Use the **Link to Applications** dropdown described in [Create or modify a folder](#), or use the **Link to Folders** dropdown described in [Create or modify an application](#).

Once a folder is linked to an application, new files uploaded to that folder will be scored by that application. Existing files may be reprocessed as described in [Reprocess applications](#).

When a folder-application link is removed, a **Delete** job appears in the **Application Jobs Queue**, and all of the scores for that application and folder pair are scheduled for deletion. See [Monitor jobs](#) for more information.

Folder processing status

Click  **Show process log** on the **Monitored Folders** screen to display the **Folder processing status** window, as shown in the following example:


Folder processing status ✕

ASR processing status [↗](#)

Queued ℹ	Converting ℹ	Transcribing ℹ	Analyzing ℹ	Analysis error ℹ	Last completion ℹ
0	0	0	0	0	2021-07-15T18:15:39.000Z

Processing log

Log Content [↗](#)

2021-07-15 

Show entries

Transcript ID	File name	Request ID	Size	Request start	Analyzed	Status
20	a2.mp3	943ce3cb-e128-4ac1-bbf7-c60fdd9ca809	201.744 KB	2021-07-15T18:15:11.000Z	2021-07-15T18:15:39.000Z	OK
19	a.mp3	2f7b598c-e626-4eb3-888f-bd2b7aeb06d9	201.744 KB	2021-07-15T18:10:09.000Z	2021-07-15T18:10:39.000Z	OK

Showing 1 to 2 of 2 entries

Previous **1** Next

[Close](#)

ASR processing status

The **ASR processing status** section displays the following information:

Queued

The number of requests waiting to be transcribed.

Converting

The number of files being converted to the format required for transcription.

Transcribing

The number of files being transcribed.

Analyzing

The number of files being analyzed.

Analysis error

The number of files that were processed by the ASR engine but could not be ingested for analysis. Use the mouse pointer to hover over the **Analysis error** value to show the date and time of its most recent increment. If this number is nonzero or increasing, contact support for assistance.

Last completion

Displays the date and time at which the system last analyzed files for this folder.

Processing log

The **Processing log** section displays the following filter options and processing information:

Show [number of] entries

Select a value from the dropdown to change the number of entries displayed at one time. Use the paging controls below the table to shift between sets of entries.

Date picker

Limits the results to only entries with particular **Request start** dates. Dates with processing data are highlighted on the calendar. Request dates correspond to when the files were uploaded to the system, and may contain data from previous days.

Transcript ID

The identifier associated with the transcription results for the audio listed in the **File name** column. Transcript ID is useful for finding a specific audio record after its transcript has been analyzed. Records that fail text analysis do not receive a transcript ID.

File name

The name of the uploaded audio file.

Request ID

The identifier associated with the upload that contained the processed file. Multiple audio files uploaded together share a request ID.

Size

The size of the uploaded audio file. Size is displayed only if it can be determined from the data or metadata in the file.

Request start

The date and time that the upload completed and its contents were queued for processing.

Analyzed

The date and time that conversion, transcription, and analysis were all completed and processing finished.

Status

Indicates a file's post-processing status with one of two values:

- **OK** indicates that the file was processed without error.
- **Analyze_ERROR** indicates that file processing failed and shows the error returned.

Callbacks

V-Spark folders can be configured to POST processed JSON transcripts to a callback server. In the simplest case, the callback server is configured to locally save transcripts. V-Spark can also use callbacks to write to local storage, or to send data to other applications for additional analytics or archival. Cloud-based deployments only support HTTP callbacks.

Retries

If V-Spark cannot deliver results to a callback URL, it retries up to 100 times (by default) or for up to 10,000 hours before giving up. To avoid overloading the server and network, the callback worker waits for an increasing amount of time between retries. If the delivery ultimately fails, V-Spark places the results that it was trying to deliver in the `/var/lib/vspark/error/callback/` directory.

The number of times that callbacks are retried is configurable. The location of the error directory is also configurable.

Name collisions

Using callbacks to deliver enriched transcript (and other) files to a callback server can occasionally result in name collisions. A name collision results when files with the same name as the file that is currently being processed already exist on the callback server. This can occur when, for example, two folders have the same callback settings and files with the same name are uploaded to those folders for transcription.

V-Spark's callback mechanism automatically resolves name collisions. Files that are delivered via callbacks automatically insert an incremental version number in file names to avoid name collisions. For example, if a file named `sample.json` already exists in the location where a callback writes files, the JSON file that the callback writes will be named `sample.1.json`. If both a file named `sample.json` and a file named `sample.1.json` already exist in the location where callbacks write files, the JSON file that the callback writes will be named `sample.2.json`. The number will increment until no file with a matching name is found.

IMPORTANT: Callbacks that use the File system, SFTP, and AWS S3 callback delivery methods automatically avoid name collisions. Name collisions in callbacks that use the HTTP and HTTPS callback delivery methods will overwrite existing files.