

# voiXen Upload API

## Table of contents

|                                   |   |
|-----------------------------------|---|
| Document history                  | 2 |
| Introduction                      | 3 |
| The voiXen upload in short        | 3 |
| The voiXen upload in depth        | 3 |
| What needs to be uploaded?        | 3 |
| Audio files                       | 3 |
| Metadata files                    | 4 |
| The special XML element ExterneID | 6 |
| How is the upload performed?      | 8 |
| When the upload is complete       | 9 |
| Conclusion                        | 9 |

## Document history

| Date       | Version number      | Content, changes  |
|------------|---------------------|---|
| 2015-08-19 | 1.0                 | First version   |
| 2016-01-20 | 1.1                 | New (optional) field "ExterneID"                            |
| 2016-01-27 | 1.2                 | References to fields in the voixen search updated           |
| 2016-02-05 | 1.2.1               | Added notes on file names and audio formats                 |
| 2016-04-11 | 1.2.2               | Added notes on the region                                   |
| 2017-04-05 | 1.2.3               | Tag "Agentendurchwahl" deprecated, new data center included |
| 2018-09-05 | 1.2.4               | Tag "CR_ID" deprecated, additional data center included     |
| 2019-03-08 | 1.3                 | New format, new upload tools inserted                       |
| 2020-06-04 | 1.4 (this document) | Introduction of new tag "NPS"                               |

# Introduction

This document describes how recordings of voice calls can be uploaded to [voixen.com](http://voixen.com) for further analytics. It is intended to be read and understood by administrators and software developers - if you are not one of them, no problem - you might then take the blue pill and just read the content of the friendly grey box below, then put this document aside and continue having a good time with your colleagues, friends, family or your favourite pet. Most probably you will not be able to upload anything to voixen by yourself, but you will definitely know to whom you can delegate the task.

For the admins and devs that luckily got access to this document, you take the red pill and follow us down the rabbit hole.

## The voixen upload in short

voixen uses a file based interface. Your recorder, your ACD or whatever system you use saves individual recordings to a .wav or .mp3 file (just like those you play on your PC, Mac, iPhone) anywhere on your server(s). We will not bore you with details about sampling frequencies, codecs and so on.

To give voixen some information about the recording it adds information from an XML file with some metadata like the date, time, calling agent and whatever you need to find and evaluate the file later. Fairly easy, XML is just a text file with some markers that helps the machine to identify the data in the file.

So we got two files for each recording in the end, the audio data and the meta data. Both together they need to be uploaded to a S3-Bucket. Imagine it like a good old (S)FTP-server, just more modern, more secure and much easier to be handled by cloud-based infrastructures (and often also by your firewall).

The URL and access credentials are provided by us, and your IT team will know how to set up the upload (if they don't, they will, after reading this document completely).

# The voixen upload in depth

## What needs to be uploaded?

The box above already told us (you did read that, too, didn't you?). We need audio data and some accompanying metadata. Let us dive into the nitty gritty details.

### Audio files

voixen accepts audio recordings in either .wav or .mp3 file format. But not any - some specifications have to be met. Let's start with the least technical, but most business relevant spec: Stereo. If you are recording the agent only, mono might be also ok, but if you are recording both agent and customer it is important to have a stereo recording. And stereo does not only mean "two channels", but agent on the one, customer on the other side. It does not matter who is left or right (we can configure that on a per project basis) as long as it is consistent within a project.

Now for the other technical requirements, starting with files in .wav-format. We need:

- Codecs: uncompressed, a-law,  $\mu$ -law, PCM
- Precision: 16-bit
- Sample Rate: 8000 Hz
- Bit Rate: 128k
- Channels: 2 (Stereo) or 1 (Mono)

For .mp3 the requirements are:

- Sample Rate: 8000 Hz
- Bit Rate:  $\geq 48$  k

## Metadata files

A lot of information that is needed for voiXen cannot be encoded in the audio file itself, it is therefore necessary to add one XML file with so-called metadata per call. This file has to be identically named as its audio counterpart, so if you have a recording that is named

`recording_4712_1_3.wav`

the XML file will need to be named

`recording_4712_1_3.xml`

voiXen will only import a recording if both files are present (and valid!) in the incoming bucket (we will get to buckets later). Please make sure that the XML file is correctly encoded in UTF-8 and that the XML is valid. A common mistake we encounter now and then is the missing escaping of certain characters. According to the XML standard the following characters need to be escaped:

| Character | XML notation |
|-----------|--------------|
| <         | &lt;         |
| >         | &gt;         |
| &         | &amp;        |
| “         | &quot;       |
| ’         | &apos;       |

Another common mistake is the wrong order of the XML tags, it is vital to specify them in the same order as in the example below.

Talking about filenames - there are some conventions that you must adhere to in your naming, just because they are part of the S3 protocol specification.

Filenames may contain:

- Small letters [a-z]
- CAPITAL LETTERS [A-Z]
- Digits [0-9]

They must not contain:

- Whitespaces (neither space nor tab)
- Mutated vowels (ä, ö, ü no matter if small or capitalized)
- Special characters [ ,?!\$&:~|%{}[] etc.]

Before we dive into the details, let us start with a simple example of an XML file that is valid for an import into voixen:

```
<?xml version="1.0" encoding="UTF-8"?>
<Recording>
  <Datum>24.10.2019</Datum>
  <Zeit>10:11:42</Zeit>
  <Anmeldename>Max Musteragent</Anmeldename>
  <Dienst>Alukando AG</Dienst>
  <Dauer>242</Dauer>
  <NPS>9</NPS>
  <AnrufendeNummer>+4930123456789</AnrufendeNummer>
  <Dateiname>/media/recordings/test.wav</Dateiname>
  <ExterneId>Basistarif</ExterneId>
</Recording>
```

We will now have a look at the details and meaning of each of the XML elements. Optional elements can be left out and are marked with an asterisk (\*).

| Element Name | Example Value   | Description  |
|--------------|-----------------|--|
| Recording    |                 | The root element.  |
| Datum        | 24.10.2019      | Recording date. Format DD.MM.YYYY. Please pad with leading zeros.                    |
| Zeit         | 10:11:42        | Time of recording. Format: hh:mm:ss 24h format. Please pad with leading zeros.       |
| Anmeldename  | Max Musteragent | Name of the agent. This information can be used within voixen search and dashboards. |

|                 |                            |   |
|-----------------|----------------------------|---|
| Dienst          | Alukando AG                | Project name. This information can be used within voiXen search and dashboards.   |
| Dauer           | 242                        | Duration of the recording in seconds. This information will be recalculated by voiXen during the import. It can be used within voiXen search and dashboards.      |
| NPS *           | 9                          | NPS score of the recording. Allowed values are 1 to 10.   |
| AnrufendeNummer | +4930123456789             | Caller number or number of customer. Prefixes like '+' are allowed. This information can be used within voiXen search and dashboards.                             |
| Dateiname *     | /media/recordings/test.wav | Name and path of the file on your system.   |
| ExterneId *     | see below                  | External IDs are a means of specifying custom metadata fields that can be used for searching and filtering in voiXen. You will find details in the section below. |

As stated above, please make sure to use the correct sequence of XML tags.

### The special XML element ExterneID

The external ID works as an extension of the built-in metadata fields and can be specified either as text or structured data. You can define an arbitrary number of XML sub elements with element names and values, the voiXen UI will present the element name as label and the value as field value to that label. By the way, in the UI we call these attributes.

We will show you three examples, each with the corresponding fragment from the XML file and a screenshot from the resulting voiXen UI.

The most basic one (only some text):

```
<Recording>
  <ExterneId>Basistarif</ExterneId>
</Recording>
```

will be shown as



A little more complex with a sub-element:

```
<Recording>
  <ExterneId>
    <Hotline>Technik</Hotline>
  </ExterneId>
</Recording>
```



You noticed the difference? The attribute is now displayed as key-value-pair.

And last but not least with an additional Link

```
<Recording>
  <ExterneId>
    <Hotline>Technik</Hotline>
    <Region>Süd</Region>
    <Link>https://yourcrm.com/cust123</Link>
  </ExterneId>
</Recording>
```



The link can be clicked, meaning that if you e.g. provided a URL to the CRM activity corresponding to the phone call that has been recorded, it would be shown in a new browser tab.

As you can see this is a very valuable mechanism for the addition of custom metadata. It enables you to collect all kinds of data from your business (business data, contract value, call outcome, CRM information, etc.) that may relate to a phone call and use it to find, group or analyze it according to your stakeholders needs.

## How is the upload performed?

Now we know what files we need to produce, how to name them and we are pretty sure about what content to store in them. But we have not yet discovered how to upload them to voixen. Let us have a look.

Uploading files to voixen is simple, the only thing you need to do is to put them in a S3 bucket using the credentials we provide you at the start of the project. So, if you already know how to upload something to S3, you are fine and may stop reading here. Mission accomplished. If not, here are the details.

Let's start with the necessary credentials. On project start we will provide you with:

- S3 endpoint (either <https://s3.eu-central-1.amazonaws.com> or <https://storage.voixen.com> )  
(if your client needs a region that would be eu-central-1 for AWS )
- S3 bucket name
- S3 access key
- S3 secret access key

We recommend to keep things as easy as possible - at voixen, we use a simple open-source command line tool called minio client (short: mc) to upload files to buckets. Once configured mc can be used on the command line or within a shell script just like a standard copy command offered by your operating system. You can download mc from <https://dl.min.io/client/mc/release/> (please choose the correct version for your operating system).

Let us now assume you received the following credentials:

- S3 endpoint: `https://storage.voixen.com`
- S3 bucket name: `voixen-prod-customer-incoming`
- S3 access key: `AIM325256`
- S3 secret access key: `supersecretaccesskey`

We can now configure and test the client, and even upload our first file like this:

```
# Change to the folder where you downloaded mc.exe and test if mc works
./mc.exe --help

# Configure by registering our endpoint and credentials with an alias "voixen-de"
./mc.exe config host add voixen-de https://storage.voixen.com AIM325256 supersecretaccesskey

# Now upload a first file, works like any copy command
./mc.exe cp sample.wav voixen-de/voixen-prod-customer-incoming/sample.wav

# And list the contents of the buckets to see if everything worked
# (Do not forget the trailing slash!)
./mc.exe ls voixen-de/voixen-prod-customer-incoming/
```

It is as easy as that. Now just write a cron job that calls the copy command for every file in the directory where your audio and metadata files reside and you are done.



If you are a programmer and your requirements cannot be satisfied using a command line script you might also use the S3 protocol programmatically from nearly any available language. There are lots of libraries and bindings available from AWS or the Minio project. Start your journey here:

<https://docs.minio.io> and go to the Minio SDKs to find instructions for your preferred programming language

or do the same at:

<https://aws.amazon.com/developers/getting-started>.

Both sites will provide you with up-to-date libraries, documentation and code examples that will show you how to perform an upload using your favourite programming language and platform. Other possible tools would be S3cmd (<https://s3tools.org/s3cmd>) or Powershell (<https://aws.amazon.com/de/powershell/>).

## When the upload is complete

After uploading a wav and xml pair, the uploaded files will remain in that bucket until they are imported. When they are imported successfully, they will disappear from the bucket and instead appear on [voixen.com](https://voixen.com). Valid files should disappear within 2 to 10 minutes.

If they don't disappear that can mean:

- 1) The import has not yet been enabled for your account. Please contact us, see below.
- 2) The XML is not valid
  - a) Check for syntactic errors?
  - b) Check the spelling of the XML tags?
  - c) Are any required tags missing?
  - d) Did you specify the tags in the wrong order?

## Conclusion

We hope we gave you all information that is necessary to get you started. If that should not be the case or if you have any questions, feel free to contact us. Just send an e-mail to [support@voixen.com](mailto:support@voixen.com) and we will get back to you as soon as possible.