Nexiwave Speech Recognition SaaS Platform, v 5.0

For more information, contact: info@nexiwave.com

Current version: http://nexiwave.com/static/api/Nexiwave.Speech.Indexing.SaaS.api.pdf

Current WSDL: http://api.nexiwave.com/services/SpeechIndexing?wsdl

1. Summary

Nexiwave provides a Web Service based Speech Recognition platform. The platform features:

- advanced Speech Indexing, locate exact location of the spoken words;
- full text transcription capability for external indexing capability;
- multi-language support (English + French);
- private decoding configuration available;

2. General

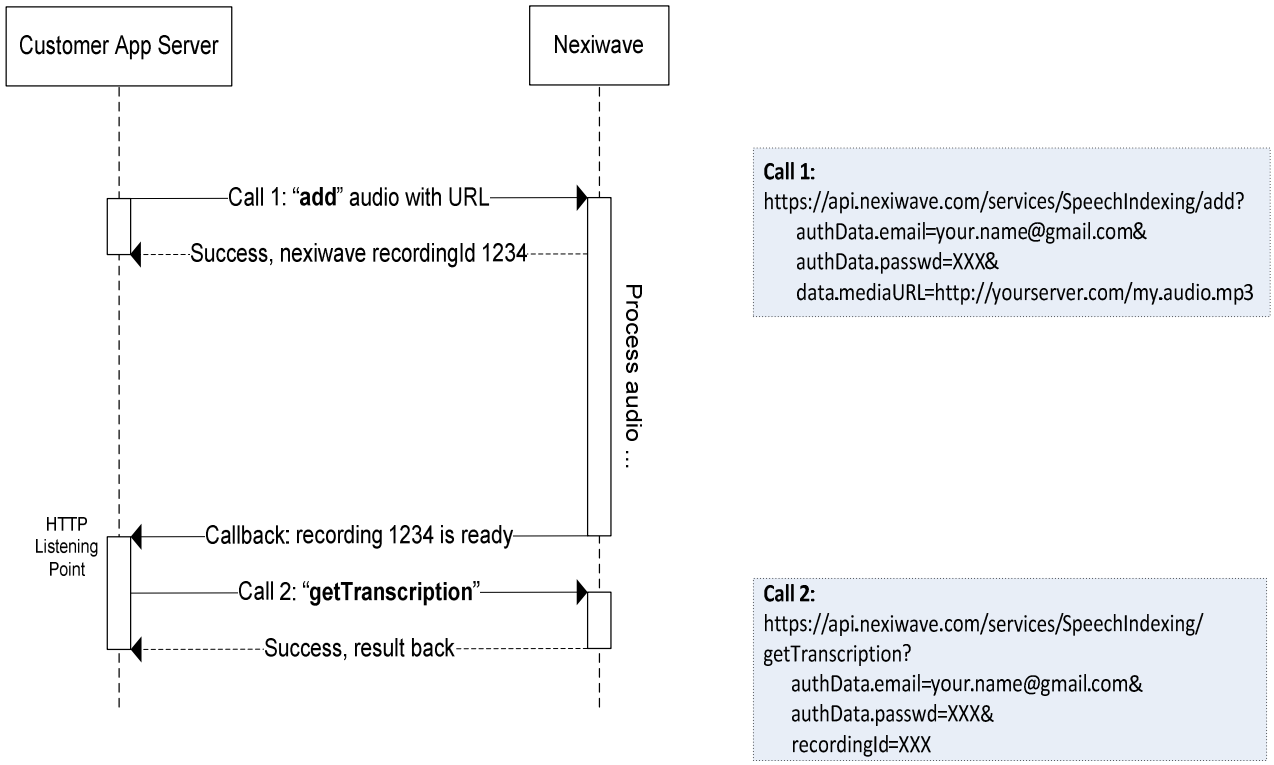
2.1 Function Overview

Nexiwave Speech Recognition Platform API can be roughly separated into these areas:

- **Record Management**: how the audio can be moved into Nexiwave system
- **Obtain Results**: obtain speech-to-text results

2.2 Call Flow

The basic call flow between your application and Nexiwave system are two simple HTTP Get request.



To send an audio to be processed:

https://api.nexiwave.com/services/SpeechIndexing/add?authData.email=XXX&authData.passwd=XXX&data.mediaURL=XXX

To retrieve transcription:

```
https://api.nexiwave.com/services/SpeechIndexing/getTranscription?authData.email=XXX&authData.passwd=XXX&recordingId=XXX
```

2.3 Synchronous vs. Asynchronous Calls

Speech processing can potentially take long time to finish. Depends on your configuration, it may take up to 10 minutes to process one minute of audio.

Nexiwave can process your audio asynchronously. Your application does not need to wait for the processing to finish. Nexiwave will notify your server when the processing is finished and result is ready to be retrieved. See httpNotifyBackURL parameter for details. Alternatively, your application may also poll for the processing status of a particular recording.

When the audio is small, you may also use the synchronized API call. See "Post with synchronous Response" for more details.

3. Common Web Service Info

3.1 Common Data

Nexiwave Speech Recognition web service APIs are all stateless API calls. You must provide these parameters for every request:

Parameter	Value	Size	Mandatory	Comments
authData	component			
email	string	255	yes	Your user account email address
password	string	255	yes	Your password

Example:

https://api.nexiwave.com/services/SpeechIndexing/getTranscription?**authData.email**=XXX&**authData.passwd**=XXX&recordingId=1234

3.2 RESTful API Support

Besides the standard SOAP interface, the same Nexiwave Web Service API can be accessed via simple REST address.

A sample RESTful GET URL:

https://api.nexiwave.com/services/SpeechIndexing/getTranscription?authData.email=XXX&authData.passwd=XXX&recordingId=1234

3.3 XML vs. JSON

The default response content is XML. To receive response as JSON objects, simply add this HTTP parameter to your REST URL:

HTT Parameter	Value
response	application/json

Sample JSON HTTP GET Request:

```
https://api.nexiwave.com/services/SpeechIndexing/add?authData.email=XXX&authData.passwd=XXX&mediaURL=XXX&response=application/json
```

Sample JSON return:

```
{"addResponse":{"return":10805}}
```


4. Record Management Functions

These Nexiwave services provide methods to import speech content into Nexiwave system. Two methods are provided:

- add a URL
- post a file

4.1 Web Service: add

Your application invokes this web service to instruct Nexiwave to index a media file.

Simplified Request Parameters:

(Recommended minimum list of parameter. Please consult Full Parameter List)

Parameter	Value	Size	Mandatory	Comments
authData	component		yes	see Common Web Service section
data	component			
mediaURL	string	5000	yes	an URL (http/https, ftp, please) Mandatory unless mediaFileData present
externalKey	string	5000		the external key Note: nexiwave simply records this data. No validation will be enforced to ensure the uniqueness of this key. If two or more recordings have same externalKey, the definition of finding the recording by this key is undefined. **Note**: Nexiwave recommend to roughly use this field as speaker key, so Nexiwave would know two different audio files belong to the same speaker.

RESTFul Request:

https://api.nexiwave.com/services/SpeechIndexing/add?authData.email=XXX&authData.passwd=XXX&data.mediaURL=XXX&data.externalKey=XXX

Parameter	Value	Comments
recording		
recordingId	long	ID of the recording added to the system

Sample XML response:

http://www.nexiwave.com/static/api/captured.soap.messages/add.response.xml

Sample JSON response:

{"addByURLResponse":{"return":10805}}

4.2 Post Media File

This method is useful if your media file cannot be exposed via a HTTP or FTP server.

Request:

Method	HTTP POST
HTTP Post URL	https://api.nexiwave.com/file/storage/user@email.com/recording/?authData.passwd=XXX
HTTP Body	the media file to be uploaded, the raw form

Response:

Parameter	Value	Mandatory
body	long	the new recordingId

4.3 Post with synchronous response

With JSON support in HTTP POST mode, Nexiwave will attempt to return the transcription results in the same call.

There are some limitations:

- The Nexiwave server will only wait for maximum three minutes for the transcription to finish. If the processing does not finish in time, the developer will need to access the result the old way, using the getTranscription API.
- In the callback scenario, we also optionally provide a pre-authenticated callback URL to retrieve the transcription. The benefit of the callback URL is that it doesn't need the password.

This is a PHP example to perform the processing synchronously:

```
<?php
$user = 'user@mycompanyemail.com';
$passwd = 'my_password';
$file = '/workspace/audio/test.wav';

$url =
sprintf('https://api.nexiwave.com/SpeechIndexing/file/storage/%s/recording/?authData.passwd=%s&syncAttempt=true&&response=application/json', $user, $passwd);

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_HEADER, 0);
curl_setopt($ch, CURLOPT_VERBOSE, 1);

$post = array( "data.mediaFileData"=>sprintf("@%s", $file), );
curl_setopt($ch, CURLOPT_POSTFIELDS, $post);

$result = curl_exec($ch);
$code = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);

switch($code) {
case 200:
  break;
default:
  throw new exception($result);
```



```
}

echo $result;
?>
```

If the processing finished in time, the sample response may look like this:

```
{
    "recordingId":287822,
    "text":"[00:00] HI THIS IS A TEST"
}
```

If the processing did not finish in time, the response may look like this:

```
{
    "recordingId":287822,
    "result_callback":"https://api.nexiwave.com/callback_url_SSSSS"
}
```

4.4 Web Service: remove

You may also delete one of your media record from our system. These actions will be taken:

- the database record, as well as associated audio, transcript and index files, for this recording will be deleted;

- any current jobs will be deleted.

Request:

Parameter	Value	Mandatory	Comments
authData	component	yes	see Common Web Service Data section
recordingId	long	yes	the Nexiwave id for this recording

Sample RESTFul GET Request:

```
https://api.nexiwave.com/services/SpeechIndexing/remove?authData.email=XXX&authData.passwd=XXX&recordingId=XXX
```

Response:

Parameter	Value	Comments

result	boolean	whether the removal was successful

Sample Response:

http://www.nexiwave.com/static/api/captured.soap.messages/remove.response.xml

5. Obtain Results Functions:

5.1 Web Service: getTranscription

To obtain full text transcription of an audio.

Request:

Parameter	Value	Mandatory	Note
authData	component	yes	see Common Web Service section
recordingId	long	yes	the Nexiwave recording ID
alternativeSize	int	no	size of alternative words Negative: return all alternatives. Zero (Default): only the best result is returned. Positive: up to this number of alternative will be returned

Sample RESTFul Request:

https://api.nexiwave.com/services/SpeechIndexing/getTranscription?authData.email=XXX&authData.passwd=XXX&recordingId=XXX

Response:

Parameter	Value	Note
sentenceChunk		a sentence
startInMS	long	start time in milliseconds
endInMS	long	end time in milliseconds
trunkId	long	The ID for the audio channel

wordNode	component	
startInMS	long	word start time in milliseconds
endInMS	long	word end time in milliseconds
wordNodeItem	component	the contextual transcription generated
value	string	the word
confidence	float	the confidence measure, from 0 to 1 (1 as most confident)

Sample XML Response:

http://www.nexiwave.com/static/api/captured.soap.messages/getTranscription.response.xml

Sample JSON response:

http://www.nexiwave.com/static/api/captured.soap.messages/getTranscription.response.json.txt

6. Misc Functions:

6.1 Web Service: queryProcessingStatus

Depends on the length of your multimedia file, Nexiwave indexing can take some time to finish. You may this web service to query the current status of your recording.

Request:

Parameter	Value	Mandatory	Comments
authData	component	yes	see Common Web Service section
recordingId	long	yes	The Nexiwave recording ID

RESTFul GET Request:

https://api.nexiwave.com/services/SpeechIndexing/queryProcessingStatus?authData.email=XXX&authData.passwd=XXX&recordingId=XXX

Response:

Parameter	Value	Comments
recordingProcessingStatus		
status	string	see “Table 1” below for possible values
extraInfo	component	Note: this field may only be valid when the “status” is “DECODING_IN_PROGRESS”
percentageFinished	float	0-100: percentage of finished
estimatedFinishedTimeInMS	long	estimated time to finish in milliseconds
otherStats	string	other tech stats

Sample Response:

http://www.nexiwave.com/static/api/captured.soap.messages/queryProcessingStatus.response.xml

6.1.1. Possible Processing Status

This table below lists all possible statuses.

Table 1: Possible Processing Status Values

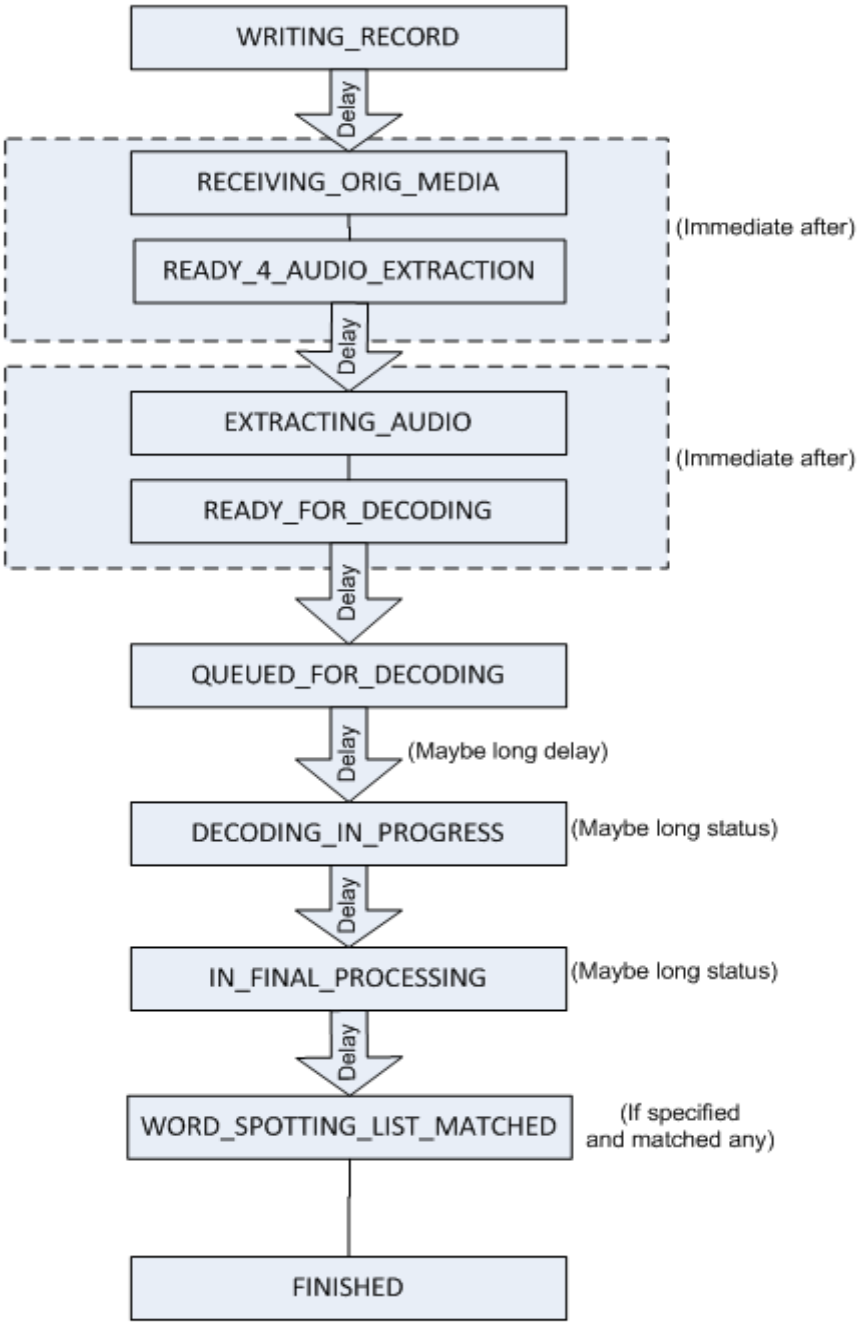
Status Values	Meaning	Note
INVALID_AUDIO_SOURCE	Nexiwave was unable to obtain audio from specified source	
INVALID_RECORDING	The recording downloaded, but the content is not valid media content.	Please refer the support media format section for a list of supported media format.
RECORD_NOT_EXIST	the recording not exist	
WRITING_RECORD	the recording is being written to Database	
RECEIVING_ORIG_MEDIA	starting to receive original media content	
READY_FOR_AUDIO_EXTRACTION	The upload request is in queue, audio source has been obtained. Audio is ready to be extracted.	
EXTRACTING_AUDIO	Nexiwave is still extracting audio from the original media file.	
NOT_FOUND_IN_QUEUE	Audio is ready, but the recording is not in the	Nexiwave prioritize the processing based on three

	processing queue yet.	levels: - paying customer with prioritized processing; - paying customer; - free trial customers.
READY_FOR_DECODING	The recording has been scheduled to be processed and waiting in processing queue.	
QUEUED_FOR_DECODING	Recording is queued for processing, but not started yet.	
DECODING_IN_PROGRESS	decoding has started	ExtraInfo field contains the current decoding progress: - % of finished; - estimated finish time; - decoding speed.
IN_FINAL_PROCESSING	Decoding has finished, in final verification and writing data.	
FINISHED	decoding has finished	
WORD_SPOTTING_LIST_MATCHED	Some spotting words have been found in the audio. (Decoding should have been nearly finished)	ExtraInfo contains number of hits (an integer).
TIME_STAMPING_DONE	Time stamping process has finished for the recording.	
UNKNOWN	The status of this recording is unknown.	
FAILED	Nexiwave failed to process	

	this recording.	

6.1.2. Typical Processing Status Sequence

A most possible sequence of status for success audio import



7. Advanced Functions

Besides the above simplified of the API calls, Nexiwave provides advanced version of API calls. These APIs are meant to expose the full functionalities of Nexiwave underlying system. It is recommended to try out the simplified version first.

7.1 Web Service: add (Full Parameter List)

You can supply more parameters for your upload.

Request:

Parameter	Value	Size	Mandatory	Comments
authData	component		yes	see Common Web Service section
data	component			
mediaURL	string	5000	yes (unless mediaFileData present)	an URL (http/https, ftp, please)
mediaFileData	attachment		yes (unless mediaURL present)	base64 media data in the attachment Please refer SOAP Attachment for how to attach binary file in SOAP call. Note: - this is NOT a recommended way for sending large media content to Nexiwave; - Nexiwave recommends using the mediaURL parameter.

externalKey	string	5000		the external key from your system Note: Nexiwave simply records this data. No validation will be enforced to ensure the uniqueness of this key.
keywords	string	5000		A list of keywords or phrases to be used in language model adaptation. Excellent accuracy improvement is reasonable. Integration Suggestions: - one method of obtaining these keywords and time codes is to use Video OCR service; - you may also extract all words/phrases from an associated PowerPoint presentation.
displayTitle	string	5000		an optional field for display in Nexiwave Web UI
notes	string	5000		any notes
sourceKey	string	128		Your key to help Nexiwave separate content during: - speaker identification process (speaker identification accuracy may deteriorate if one account has too many potential speakers); searches keywords will not be applied to recordings with different sourceKey. For example, if your application has many users of its own, you may use this field to let Nexiwave builds separate Speaker database for each key you used. Ideally, this key should be consistent for your users.

mediaTypeSuffix	string	100		The value is generally the typical file suffix for your media file. For example, “mp3” for mp3 audio files, “wmv” for windows media file. Use this field to help our media decoder to identify the media content type.
decodingProfile	string	100		a case-sensitive valid decoding profile name If no value, or invalid value, is passed in, a Nexiwave configured default decoding profile will be used.
contextualDocURLs	string	5000		URLs for contextual documents separated by semi-colon (;). Document(s) will be downloaded and keywords/phrases extracted and used during indexing process.
contextualDocumentFileData	byte[]			Contextual document that is associated with the media. Supported file types: - all Ms Office files: PowerPoint, Word, etc - pdf - html/xml - txt Words and phrases will be utilized during indexing process.
contextualDocumentFileData2	byte[]			-
contextualDocumentFileData3	byte[]			-

contextualDocumentFileData4	byte[]			-
contextualDocumentFileData5	byte[]			-
httpNotifyBackURL	string	5000		Instead of your code actively polling for processing status, Nexiwave can push the status to your system. A HTTP(S) GET url that will be called by Nexiwave upon each status change event during the processing of your recording. Three extra parameters will be added to the url: "recordingId", "externalKey" and "status" You may also configure the default value for this field in your Nexiwave account "Settings" page, or through the Nexiwave profile API. Please read the below section Notes on httpNotifyBackURL for more information.
actualStartTimeInMS	long			the actual start time of the audio, in number of milliseconds after January 1st, 1970 (as defined java.util.Date.getTime() method). If this field is not set, the current time at the upload will be used.
customIndexedProperties	two dimension String array, key/value pairs			These are the custom indexed properties, key/value pairs, for this particular recording. See below section Notes on customIndexedProperties for more information on this field.
processingPriority	int			your processing priority

				A recording with higher value of this attribute will be processed earlier than the one with lower value. Note: this processing priority is only effective within your account.
storeAudio	boolean			A Boolean field to indicate whether Nexiwave should store the audio after processing. Default is true. Note: - if the value is false, all related audio files will be deleted upon finishing decoding; - other artefacts, such as database record, generated text and timing info, will still remain in Nexiwave system; These data may still be needed for other API calls, such as getTranscription. - a subsequent "remove" API call can be used to thoroughly delete everything related to this recording.
wordSpottingList	string	5000		list of words, separated by whitespace. Upon decoding finish, if any of the words were found in the audio, a HTTP notification will be sent back to your server, with a particular status of WORD_SPOTTING_LIST_MATCHED. For more information, see httpNotifyBackURL.
otherAudioTrunks	component			If you record audio in separate file for each audio leg, you may send the audio leg separately for best accuracy.
mediaFile2TranscribeURL	string			the source URL for this audio leg/trunk

offsetInMS	long			The relative offset of the audio trunk to the start of the whole recording. Note: this field must be positive number. The behaviour is undefined for negative numbers.
hints	component			
minNumberOfSpeakers	int			Minimum number of speakers in the audioDefault is 1
maxNumberOfSpeakers	int			Maximum number of speakers. Zero or negative number means unlimited. Default is zero (unlimited)
speakerNames	string	1000		the speaker names, separated by semi-colon (;)

RESTFul GET Request:

https://api.nexiwave.com/services/SpeechIndexing/add?authData.email=XXX&authData.passwd=XXX&data.mediaURL=XXX&data.targetDecodingConfigName=XXX&....

Response:

Parameter	Value
recording	
recordingId	long

Sample Response:

http://www.nexiwave.com/static/api/captured.soap.messages/addExt.response.xml

7.1.1. Notes on "httpNotifyBackURL"

Using HTTP calls to your server, Nexiwave can notify your application about the status change of nearly each step of the processing, in an asynchronously manner.

Three parameters will be added to this notification GET url:

- **recordingId**: this is the Nexiwave internal recording ID assigned to your recording

- **externalKey**: the externalKey that the particular recording was associated with

- **status**: this is the same status string you can obtain from the "queryProcessingStatus" request.

For a list of possible status valid, please reference table Possible Processing StatusValues .

Example:

If the supplied notification URL was:

```
http://your.domain.com/app/callback
```

Then, the GET request that your server will receive will be:

```
http://your.domain.com/app/callback?recordingId=12345&status=WRITING_RECORDING
```

Your Nexiwave account can also be configured with the default notify back URL through the "Settings" page.

7.1.2. Notes on "customIndexedProperties"

You may associate one or more custom properties to a recording. This is a hash like properties.

Sample REST URL with customIndexedProperties:

```
https://api.nexiwave.com/services/SpeechIndexing/add?authData.email=XXX&authData.passwd=XXX&data.mediaURL=XXX&data.customIndexedProperties[myCustomerId]=1234&data.customIndexedProperties[receivedFromChannel]=XXX&…
```

Nexiwave may use these custom properties to better categorize your audio and process with proper models. Please contact support@nexiwave.com for setup.

8. Appendix

8.1 Supported Input / Output Formats

8.1.1. Input Formats

Nexiwave supports virtually all kind of media format. For best accuracy, nexiwave recommends un-compressed PCM Wave audio.

Nexiwave is capable of extracting audio from these media file formats.

Supported video/audio codec:

- (S)VCD (Super Video CD)
- CDRwin's .bin image file
- MPEG-1/2 (ES/PS/PES/VOB)
- AVI file format
- ASF/WMV/WMA format
- QT/MOV/MP4 format
- RealAudio/RealVideo format
- Ogg/OGM files
- Matroska
- NUT
- VIVO format
- FLI format
- NuppelVideo format
- yuv4mpeg format
- FILM (.cpk) format
- RoQ format
- PVA format

Supported audio codec:

- MPEG layer 1, 2, and 3 (MP3) audio
- AC3/A52, E-AC3, DTS (Dolby Digital) audio (software or SP/DIF)
- AAC (MPEG-4 audio)
- WMA (DivX Audio) v1, v2
- WMA 9 (WMAv3), Voxware audio, ACELP.net etc (using x86 DLLs)
- RealAudio: COOK, SIPRO, ATRAC3 (using Real libraries)
- RealAudio: DNET and older codecs
- QuickTime: Qclp, Q-Design QDMC/QDM2, MACE 3/6 (using QT libraries), ALAC
- Ogg Vorbis audio
- VIVO audio (g723, Vivo Siren) (using x86 DLL)

- alaw/ulaw, (ms)gsm, pcm, *adpcm and other simple old audio formats
- 3gp, amr

8.1.2. Output Formats

Besides machine friendly transcription data, Nexiwave can output machine processing results in these pre-formatted human friendly formats:

File type name	Content type	Description
transcript.lines	text	Transcription format
timedtext.xml	text/xml	timedtext for subtitle
sami.xml	text/xml	SAMI content for subtitle
qt.txt	text	quickTime subtitle
caption.srt	text	subrip SRT subtitle

8.1.3. Output format: Line-by-line format

Line-by-line format looks like this:

```
[00:00] Hi John!
[00:05] How are you?
[00:14] Fine.
[00:24] Is it going to rain today?
```

8.2 Audio File Format Recommendations

8.2.1. Record per speaker per channel

For best accuracy, audio files are processed on a per channel basis. It is recommended to record your audio source file on per speaker per channel basis.

8.2.2. Recording frequency

Nexiwave system is prepared to accept nearly any kind of sample rate. Your audio file is converted to a nearest sample rate that Nexiwave can accept.

Human speaks at less than 11,000hz. Therefore, there is also no need to use too high sample rate.

8.3 Error Handling

List of error codes that Nexiwave uses:

Error Code (all error codes start with "Error")	Description
unable.to.create.user.exists.need.email.verification	unable to create a user The email address has been register with an account already, which is still waiting for user activation.
unable.to.create.user.exists.email.verified	unable to create a user The email address has been register with an active account already.
registration.invalidEMail	invalid email used for registration
registration.invalidPassword	invalid password used for registration
readonly.account	while invoking a non-read-only action, this account is read-only
profile.changepassword.newpassword.empty	new password is empty
profile.password.minimum.length	new password is too short It must have the length of six or more.
profile.password.must.has.digit	the new password must have at least one digit
profile.password.must.has.letter	the new password must have at least one letter

nothing.to.upload	while add request, nothing to import
speaker.ID.not.exist	While add request, speaker ID is specified, but not found in Nexiwave DB.
invalid.recordingId	While a recording access call, recordingId is not valid.
empty.query	Requested a search, the query is empty.
security.invalid.access.to.recording	security error The caller does not have access to the recording.
security.not.authorized.feature	security error The caller is not authorized to use the feature.
security.invalid.userId.and.password	security error Unable to authenticate with userId and password supplied.
security.email.not.verified	security error UserId and password matches, but email has not been verified. The user should check verification email that was sent by Nexiwave.
security.account.locked.out	security error UserId and password matches, but the account has been locked out due to too many failed attempts. Please contact support@nexiwave.com to unlock the account.
security.invalid.request	security error The request is not valid.

Sample error message:


```
<soapenv:Fault>
        <soapenv:Code>
                <soapenv:Value>
                        axis2ns4:Error.security.invalid.userId.and.password
                </soapenv:Value>
        </soapenv:Code>
        <soapenv:Reason>
                <soapenv:Text xml:lang="en-US">
                Unable to authenticate with supplied userid/passwd for user Auth Request for
user            test@nexiwave.com, group 0, passwd: (length=0). Result : FAILURE. Please
        contact support@nexiwave.com
                </soapenv:Text>
        </soapenv:Reason>
        <soapenv:Detail/>
</soapenv:Fault>
```

(End of document)

