



## **nexiwave Audio Search SaaS Platform, v 2.0**

- *For more information, contact:* [info@nexiwave.com](mailto:info@nexiwave.com)
- *Current version:* <http://nexiwave.com/api/nexiwave.audio.search.SaaS.api.pdf>
- *Current WSDL:* <http://search.nexiwave.com/AudioSearch/services/SearchService2?wsdl>

Table of Contents

- nexiwave Audio Search SaaS Platform, v 2.0..... 1
- 1. Summary ..... 4
  - 1.1 Why Audio Search?..... 4
- 2. General..... 5
  - 2.1 Workflow..... 5
  - 2.2 Additional Requests ..... 5
  - 2.3 Web Service Platform ..... 5
- 3. Common Web Service Data: ..... 6
- 4. Web Service: addByURL..... 7
- 5. Web Service: addLocalFile ..... 8
- 6. Web Service: query ..... 9
- 7. Web Service: remove ..... 10
- 8. Web Service: queryProcessingStatus..... 11
- 9. Web Service: listAllRecordingInfo..... 13
- 10. Advanced Web Services..... 14
  - 10.1 Advanced Web Service: addExt ..... 14
    - 10.1.1 Notes on “httpNotifyBackURL” ..... 16
    - 10.1.2 Notes on “customIndexProperties” ..... 16
    - 10.1.3 Sample Web Service Request and Response:..... 17
  - 10.2 Advanced Web Service: queryExt ..... 17
  - 10.3 Advanced Web Service: estimateTotalNumberOfMatches..... 18
  - 10.4 Advanced Web Service: lookup ..... 19
  - 10.5 Advanced Web Service: listDecodingProfiles ..... 19
  - 10.6 Advanced Web Service: updateRecording..... 20
  - 10.7 Advanced Web Service: purge ..... 21
  - 10.8 Other Advanced Web Service: ..... 22
- 11. Appendix ..... 24
  - 11.1 Audio File Format Recommendations ..... 24
    - 11.1.1 Record per speaker per channel..... 24
    - 11.1.2 Recording frequency..... 24
    - 11.1.3 File Format ..... 24
  - 11.2 Nexiwave Audio Search API v1 .....**Error! Bookmark not defined.**

The nexiwave Audio Search Web Service API Guide is Copyright © 2010 PC-NG Inc. All rights reserved.

This document is for informational use only, it is subject to change without notice, and should not be construed as a commitment by PC-NG Inc. PC-NG Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide. No part of this guide may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Premiere Global. Please note that the content in this guide is protected under copyright law.

Please remember that any existing artwork or images that you may want to include in your project may be protected under copyright law. The unauthorized incorporation of such material into your new work could be a violation of the rights of the copyright owner. Please be sure to obtain any permission required from the copyright owner.

Any references to company names in sample templates are for demonstration purposes only and are not intended to refer to any actual organization.

nexiwave is a service mark of PC-NG Inc. All other trademarks are the property of their respective owners.

# 1. Summary

nexiwave provides a Web Service based Audio Search platform. Our platform features:

1. Advanced audio search, locate exact spot of when those words were spoken in audio, or audio portion of multimedia content
2. Search focused, multi model, multi pass system, high robust search accuracy
3. SaaS platform, private cloud available
4. Contextual machine transcription available

nexiwave's Speech processing engine is built on top of CMU Sphinx Speech Recognition engine, with proprietary extensions with focus on the robustness of speech search accuracy. Therefore, it is also possible to download the raw transcription from nexiwave's speech search engine.

## 1.1 Why Audio Search?

---

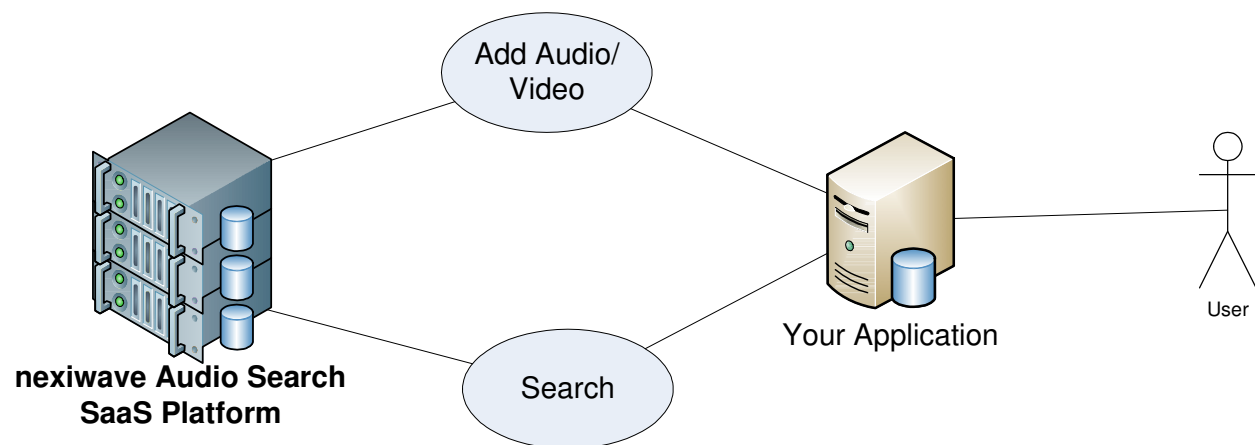
nexiwave's audio search is NOT a simple search in raw transcription. Our search accuracy is aimed to be higher and more robust than raw transcription accuracy.

## 2. General

### 2.1 Workflow

---

To integrate with nexiwave’s audio search SaaS platform, two simple web service calls are provided. Your system will **“add”** a media file to our system. Your system may then issue a **“query”** action which will return a list of recordings that match, the time spots of the matches and the contextual transcription of the matches.



### 2.2 Additional Requests

---

You may **“lookup”** our internal id for the audio you have submitted.

You can also **“remove”** one of your media record from nexiwave system.

You can also **“queryProcessingStatus”** to obtain the current processing status of your audio.

### 2.3 Web Service Platform

---

All available services are exposed via web service (SOAP calls). For the most accurate definition of these web services, please access the Web Service Description Language (WSDL) from our server:

<http://search.nexiwave.com/AudioSearch/services/SearchService2?wsdl>

## 3. Common Web Service Info

### 3.1 Common Data

---

Nexiwave Audio Search web service APIs are stateless API calls. For every request, you must provide this parameter for every request you send to nexiwave Audio Search Platform:

Parameter	Value	Size	Mandatory	Comments
<b>authData</b>	component			
<b>email</b>	String	255	Yes, unless userId is not empty	your user account email address
<b>userId</b>	long		Yes, unless email field is not empty	nexiwave userId.
<b>passwd</b>	string	255	Yes	

### 3.2 RESTful Request

---

Nexiwave uses a modern Web Service engine which fully supports RESTful request. All nexiwave SOAP request may be accessed via REST address. There is no known limit in the length of our supported REST address.

## 4. Web Service: addByURL

Your application invokes this web service call when you would want our system to index one more media file.

Note:

- This is a simplified version of “[addExt](#)” request. Please reference “[addExt](#)” section for more control.

Request:

Parameter	Value	Size	Mandatory	Comments
authData	component		Yes	See “ <a href="#">Common Web Service Data</a> ” section
mediaURL	string	5,000	Yes	The url to download from your environment.
externalKey	String	5,000	no	This external key can be uniquely mapped to our recordingId.

Response:

Parameter	Value	Mandatory	Comments
recordingId	Long		The unique recording id generated from our system

Note:

- mediaURL: an url, we currently support http, ftp and https. For security, please encode your ftp server’s userid/password to the url
- We support virtually all media file types. More specifically, we support: wav, mp3, mp4, flv, wmv

Sample Web Service Request and Response:

Request	<a href="http://nexiwave.com/api/captured.soap.messages/addByURL.request.xml">http://nexiwave.com/api/captured.soap.messages/addByURL.request.xml</a>
Response	<a href="http://nexiwave.com/api/captured.soap.messages/addByURL.response.xml">http://nexiwave.com/api/captured.soap.messages/addByURL.response.xml</a>

## 5. Web Service: addLocalFile

You may also elect to add a local file to our audio indexes. This request is very similar to addRemoteURL, except the audio data is directly attached to the Web Service request (Base64 encoded as required by Web Service protocol).

Note:

- Base64 encoded increases the size of the data by 50%. It is recommended to submit the audio data via the “addRemoteURL” method, which nexiwave can directly download from your server.
- Request:

Parameter	Value	Size	Mandatory	Comments
authData	component		Yes	See “ <a href="#">Common Web Service Data</a> ” section
mediaData	Web service attachment		Yes	Base64 encoded audio data
externalKey	String	5,000	no	This external key can be uniquely mapped to our recordingId.

- 
- Response:

Parameter	Value	Mandatory	Comments
recordingId	Long		The unique recording id generated from our system

Sample Web Service Request and Response:

Request	<a href="http://nexiwave.com/api/captured.soap.messages/addLocalFile.request.xml">http://nexiwave.com/api/captured.soap.messages/addLocalFile.request.xml</a>
Response	<a href="http://nexiwave.com/api/captured.soap.messages/addLocalFile.response.xml">http://nexiwave.com/api/captured.soap.messages/addLocalFile.response.xml</a>

## 6. Web Service: query

You invoke this service when your user performs a search on your system.

Note: this is simplified version of “[queryExt](#)” request. This request only returns the top 200 matches. Please reference “[queryExt](#)” section for more control.

Request:

Parameter	Value	Size	Mandatory	Note
authData	component		Yes	See “ <a href="#">Common Web Service Data</a> ” section
query	String	5,000	Yes	The search term

Response:

Parameter	Value	Mandatory	Note
<b>Recording</b>			
recordingId	Long	Yes	The internal recording id for a match recording
url	url	No (if final audio was saved as indicated in add request)	The secure url you may use to access the audio
<b>Match</b>			
Start	Long	Yes	Match start in seconds
End	Long	Yes	Match end in seconds
context	string	Yes	The contextual transcription generated

Sample Web Service Request and Response:

Request	<a href="http://nexiwave.com/api/captured.soap.messages/query.request.xml">http://nexiwave.com/api/captured.soap.messages/query.request.xml</a>
Response	<a href="http://nexiwave.com/api/captured.soap.messages/query.response.xml">http://nexiwave.com/api/captured.soap.messages/query.response.xml</a>

## 7. Web Service: remove

You may also delete one of your media record from our system.

Note: this is a soft-delete. This recording will be marked as deleted only. To absolutely delete a record, please use the below “purge” method.

Request:

Parameter	Value	Mandatory	Comments
authData	component	Yes	See “ <a href="#">Common Web Service Data</a> ” section
recordingId	Long	Yes	The nexiwave id for this recording

Response:

Parameter	Value	Mandatory	Comments
Result	boolean	Yes	

Sample Web Service Request and Response:

Request	<a href="http://nexiwave.com/api/captured.soap.messages/remove.request.xml">http://nexiwave.com/api/captured.soap.messages/remove.request.xml</a>
Response	<a href="http://nexiwave.com/api/captured.soap.messages/remove.response.xml">http://nexiwave.com/api/captured.soap.messages/remove.response.xml</a>

## 8. Web Service: queryProcessingStatus

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

Request:

Parameter	Value	Size	Mandatory	Comments
authData	component		Yes	See " <a href="#">Common Web Service Data</a> " section
recordingId	Long		Yes	The nexiwave recording ID

Response:

Parameter	Value	Mandatory	Comments
<b>RecordingProcessingStatus</b>			
<b>Status</b>	String	Yes	See below for valid values
<b>extraInfo</b>	(object)		Note: this field may only be valid if the status is "DECODING_IN_PROGRESS"
<b>percentageFinished</b>	float	yes	0-1: percentage of finished
<b>estimatedFinishedTimeInMS</b>	long	yes	Estimated time to finish in milliseconds
<b>otherStats</b>	string	no	Other tech stats

Table 1: Possible Processing Status Values

Value	Meaning	Note
INVALID_AUDIO_SOURCE	Nexiwave had trouble to obtain audio from specified source	
INVALID_RECORDING	The recording exists, but not valid	
RECORD_NOT_EXIST	The recording not exist	
WRITING_RECORD	The recording is still being	

	written to DB	
<b>READY_4_AUDIO_EXTRACTION</b>	The upload request is in queue and audio source has been downloaded and is ready to be extracted.	
<b>EXTRACTING_AUDIO</b>	nexiwave is still extracting audio from the original media file	
<b>NOT_FOUND_IN_QUEUE</b>	Audio is ready, but the recording is not in the processing queue yet	
<b>READY_FOR_DECODING</b>	The recording is ready, but not loaded into queue yet	
<b>QUEUED_FOR_DECODING</b>	Recording is queued for processing, but not started yet	
<b>DECODING_IN_PROGRESS</b>	Decoding has started	extraInfo field contains the current decoding progress: <ul style="list-style-type: none"> <li>• % of finished</li> <li>• estimated finish time</li> <li>• decoding speed</li> </ul>
<b>WORD_SPOTTING_LIST_MATCHED</b>	Some spotting words have been found in the audio (Decoding should have been nearly finished)	extraInfo contains: <ul style="list-style-type: none"> <li>• number of hits (an integer)</li> </ul>
<b>FINISHED</b>	Decoding has finished	
<b>UNKNOWN</b>	The status of this recording is unknown	
<b>FAILED</b>	Nexiwave failed to process this recording	

Sample Web Service Request and Response:

<b>Request</b>	<a href="http://nexiwave.com/api/captured.soap.messages/queryProcessingStatus.request.xml">http://nexiwave.com/api/captured.soap.messages/queryProcessingStatus.request.xml</a>
<b>Response</b>	<a href="http://nexiwave.com/api/captured.soap.messages/queryProcessingStatus.response.xml">http://nexiwave.com/api/captured.soap.messages/queryProcessingStatus.response.xml</a>

## 9. Web Service: listAllRecordingInfo

You may use this method to list all recordings of your account.

Request:

Parameter	Value	Mandatory	Comments
authData	component	Yes	See " <a href="#">Common Web Service Data</a> " section

Response:

Parameter	Value	Mandatory	Comments
<b>Recording</b>			
recordingId	long	Yes	
url	String		
externalKey	String		
uploadTime	Date		
Title	String		
notes	String		
averageAccruacy	Float (0~1.0)		<ul style="list-style-type: none"> <li>0: means no confidence, or confidence not available.</li> <li>1: means extremely good confidence.</li> </ul> <p>A value greater than 0.7 is considered good confidence.</p>

Sample Web Service Request and Response:

Request	<a href="http://nexiwave.com/api/captured.soap.messages/listAllRecordingInfo.request.xml">http://nexiwave.com/api/captured.soap.messages/listAllRecordingInfo.request.xml</a>
Response	<a href="http://nexiwave.com/api/captured.soap.messages/listAllRecordingInfo.response.xml">http://nexiwave.com/api/captured.soap.messages/listAllRecordingInfo.response.xml</a>

## 10. Advanced Web Services

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

### 10.1 Advanced Web Service: addExt

This is an advanced version of the add request. You can supply more parameters for your upload.

Request:

Parameter	Value	Size	Mandatory	Comments
authData	component		Yes	See " <a href="#">Common Web Service Data</a> " section
data	component			
mediaFile2TranscribeURL	Strng	5000	Yes (unless mediaFile Data present)	The url (http/https, please)
mediaFileData	attachment		Yes (unless mediaFile 2TranscribeURL present)	Media data
externalKey	string	5000		The external key. Note: nexiwave simply records this data. No validation will be checked to ensure the uniqueness of this key.
mediaFileKeywords	String	5000		A list of keywords to be used in language model adaptation.
displayTitle	string	5000		An optional field for display in nexiwave Web UI.
notes	String	5000		notes
targetDecodingConfigName	String	100		
httpNotifyBackURL	String	5000		A HTTP(S) GET url that will be called upon each status change of processing of the audio.  Two extra parameters will be added to

				<p>the url: "recordingId" and "status"</p> <p>Please read the below section "More information on "httpNotifyBackURL" for more information.</p>
<b>actualStartTimeInMS</b>	long			<p>The actual start time of the audio, in number of milliseconds after January 1<sup>st</sup>, 1970. (as defined java.util.Date.getTime() method).</p> <p>If this field is not set, the current time at the upload will be used.</p>
<b>customIndexedProperties</b>	Two dimension String array, Key/value pairs			<p>These are the custom indexed properties, key/value pairs, for this particular recording.</p> <p>See below section "<a href="#">More information on customIndexProperties</a>" for more information on this field.</p>
<b>processingPriority</b>	int			<p>Your processing priority. An recording with higher value of this attribute will be processed earlier that one with lower value.</p> <p>Note: this processing priority is only effective within your own audio files.</p>
<b>storeAudio</b>	true/false			<p>A Boolean field to indicate if you'd like nexiwave to store the audio.</p> <p>If the value is false, only the processed index data will be stored for this recording.</p>
<b>wordSpottingList</b>	String	5000		<p>List of words, separated by whitespace.</p> <p>If any of the words were found in the audio, a HTTP notification will be sent back to your server. For more information, see "<a href="#">Note on httpNotifyBackURL</a>".</p>
<b>otherAudioTrunks</b>	component			<p>If you record audio in separate file for each audio leg, you may send the audio leg separately for best accuracy.</p>
<b>mediaFile2TranscribeURL</b>	String			<p>The source URL for this audio leg/trunk</p>
<b>offsetInMS</b>	long			<p>The relative offset of the audio trunk to the start of the whole recording.</p> <p>Note: this field must be positive</p>

				number. The behaviour is undefined for negative numbers.
--	--	--	--	----------------------------------------------------------

Response:

Parameter	Value	Mandatory	Comments
<b>Recording</b>			
<b>recordingId</b>	long	Yes	

### 10.1.1 Notes on “httpNotifyBackURL”

Nexiwave can notify you about the status change of nearly each step of the processing, in an asynchronously manner.

Two parameters will be added to this GET url:

- **recordingId**: this is the nexiwave internal RecordingId assigned to your recording.
- **status**: this is the same status string you can obtain from the “queryProcessingStatus” request. Obviously, some statuses are transitional, which you may only receive in this notification back to you. for a list of possible status valid, please reference table [“Possible Processing Status Values”](#)

Example:

If the supplied notification URL was:

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

and the recordingId was 123456.

Then the GET request that you will receive is:

http://your.domain.com/app/callback?recordingId=12345&status=WRITING\_RECORDING

### 10.1.2 Notes on “customIndexProperties”

This is a powerful attribute that you may associate with each recording that you uploaded.

More importantly, these key/value pairs can be used as field in your search query. For example, if an audio upload request has these customIndexedProperties:

key	value
myAudioId	4321
myCustomerId	1234
mySiteId	789

Your search query can be like this:

*financial forecast +(myAudioid:4321)*

This will search only those recordings that have attribute “myAudioid” with value 4321.

Or, you may use this query:

*financial forecast +(myCustomerId:1234)*

This will only search those recordings that have attribute “myCustomerId” with value 1234.

Note:

- By convention, a key must be unique within one audio upload request.
- The behaviour is undefined if there are two keys with the same name within one audio upload request.

For more information regarding search syntax, please refer the search syntax document:

[nexiwave.Audio.Search-Query.Syntax.pdf](#)

### 10.1.3 Sample Web Service Request and Response:

<b>Request</b>	<a href="http://nexiwave.com/api/captured.soap.messages/addExt.request.xml">http://nexiwave.com/api/captured.soap.messages/addExt.request.xml</a>
<b>Response</b>	<a href="http://nexiwave.com/api/captured.soap.messages/addExt.response.xml">http://nexiwave.com/api/captured.soap.messages/addExt.response.xml</a>

## 10.2 Advanced Web Service: queryExt

---

This is the advanced version of “query”.

The main purpose is to support paged-based results. You may use “estimateTotalNumberOfMatches” to estimate the approximate number of potential matches.

Request:

Parameter	Value	Size	Mandatory	Comments
<b>authData</b>	component		Yes	See “ <a href="#">Common Web Service Data</a> ” section
<b>SearchData</b>	component			
<b>query</b>	string		yes	The actual query
<b>aggressiveFiltering</b>	true/false			Default is “true”

<b>recordsPerPage</b>	Int		Yes	
<b>pageNumber</b>	Int			Default is 0.
<b>context4Display</b>	Int			Default is 2.

Response (same as “query”):

Parameter	Value	Mandatory	Note
<b>Recording</b>			
<b>recordingId</b>	Long	Yes	The internal recording id for a match recording
<b>url</b>	url	No (if final audio was saved as indicated in add request)	The secure url you may use to access the audio
<b>Match</b>	<i>Component array</i>		
<b>Start</b>	Long	Yes	Match start in seconds
<b>End</b>	Long	Yes	Match end in seconds
<b>context</b>	string	Yes	The contextual transcription generated

### 10.3 Advanced Web Service: estimateTotalNumberOfMatches

You may use this service to estimate how many number of matches for a particular query. This is an approximate number.

You should send the exact request as “queryExt” to estimate how many results you may receive.

Request:

Parameter	Value	Size	Mandatory	Comments
<b>authData</b>	component		Yes	See “ <a href="#">Common Web Service Data</a> ” section
<b>SearchData</b>	component			
<b>query</b>	string		yes	The actual query
<b>aggressiveFiltering</b>	true/false			Default is “true”

<b>recordsPerPage</b>	Int		Yes	
<b>pageNumber</b>	Int			Default is 0.
<b>context4Display</b>	Int			Default is 2.

Response:

Parameter	Value	Comments
<b>result</b>	int	The number of potential matches

## 10.4 Advanced Web Service: lookup

---

You may use this service to map your externalKey to nexiwave’s recordingId.

Request:

Parameter	Value	Size	Mandatory	Comments
<b>authData</b>	component		Yes	See “ <a href="#">Common Web Service Data</a> ” section
<b>externalKey</b>	String	5,000	Yes	This external key you used when “add” the media file.

Response:

Parameter	Value	Mandatory	Comments
<b>Result</b>	long	Yes	The matching nexiwave recordingId

Sample Web Service Request and Response:

<b>Request</b>	<a href="http://nexiwave.com/api/captured.soap.messages/lookup.request.xml">http://nexiwave.com/api/captured.soap.messages/lookup.request.xml</a>
<b>Response</b>	<a href="http://nexiwave.com/api/captured.soap.messages/lookup.response.xml">http://nexiwave.com/api/captured.soap.messages/lookup.response.xml</a>

## 10.5 Advanced Web Service: listDecodingProfiles

---

You may use this call to query a list of supported decoding profiles.

Request:

Parameter	Value	Size	Mandatory	Comments
authData	component		Yes	See " <a href="#">Common Web Service Data</a> " section

Response:

Parameter	Value	Mandatory	Comments
DecodingProfile	(array)		
name	string	Yes	Name of the decoding profile, which you can use in addExt request
description	string	no	A brief description about this decoding profile

Sample Web Service Request and Response:

Request	<a href="http://nexiwave.com/api/captured.soap.messages/listDecodingProfiles.request.xml">http://nexiwave.com/api/captured.soap.messages/listDecodingProfiles.request.xml</a>
Response	<a href="http://nexiwave.com/api/captured.soap.messages/listDecodingProfiles.response.xml">http://nexiwave.com/api/captured.soap.messages/listDecodingProfiles.response.xml</a>

## 10.6 Advanced Web Service: updateRecording

You may use this call to update some field of a particular recording.

Request:

Parameter	Value	Size	Mandatory	Comments
authData	component		Yes	See " <a href="#">Common Web Service Data</a> " section
recordingId	Long		Yes	The nexiwave id for this recording
IndexedAudioData	Component			This field is corresponding to the "addExt" field.  Please refer to "addExt" for this field.

<b>externalKey</b>	String			
<b>mediaFileKeywords</b>	String			
<b>displayTitle</b>	String			
<b>notes</b>	String			
<b>actualStartTimeInMS</b>	Long			
<b>customIndexedProperties</b>	Two dimension string array, key/value pairs			

Response:

Parameter	Value	Mandatory	Comments
<b>Result</b>	boolean		

## 10.7 Advanced Web Service: purge

---

Unlike the “remove” call, this call will absolutely delete everything related to this recording from nexiwave system. There will be no way to retrieve it back.

Request:

Parameter	Value	Size	Mandatory	Comments
<b>authData</b>	component		Yes	See “ <a href="#">Common Web Service Data</a> ” section
<b>recordingId</b>	Long		Yes	The nexiwave id for this recording

Response:

Parameter	Value	Mandatory	Comments
<b>Result</b>	boolean		

## 10.8 Advanced Web Service: querySpeakEvents

---

Nexiwave can provide best estimate Speaking Events. Speaking events are ordered in time sequence. Speaking events from same speaker are marked by the automatic speaker name generated by our system.

Request:

Parameter	Value	Size	Mandatory	Comments
authData	component		Yes	See " <a href="#">Common Web Service Data</a> " section
recordingId	Long		Yes	The nexiwave id for this recording

Response:

Parameter	Value	Mandatory	Comments
SpeakEvent	Array	Y	
startTimelnMS	Long	Y	Start time of this event, in milliseconds
lengthInMS	Long	Y	The length of this speak event, in milliseconds
suggestedSpeakerName	String	Y	System generated speaker name. Same speaker will have same name across speaking events

Note:

- Speak event detection uses advanced speaker characteristic detection.
- It might be reasonable to map speaking events to certain punctuation boundary.

## 10.9 Other Advanced Web Service:

---

For performance reasons, nexiwave also offers batched requests for nearly all frequently used API calls.

<b>Batch Request</b>	<b>Singular request</b>	<b>Comments</b>
<b>multiAddExt</b>	addExt	
<b>multiQueryProcessingStatus</b>	queryProcessingStatus	
<b>multiUpdateRecording</b>	updateRecording	

Comparing to the corresponding singular requests, batched requests accepts an array of input data and return an array of result data. Please refer the corresponding singular requests for details of the fields.

# 11. Appendix

## 11.1 Audio File Format Recommendations

---

### 11.1.1 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.

### 11.1.2 Recording frequency

Nexiwave system is prepared to accept any kind of frequency. Your audio file is converted to a nearest frequency that nexiwave can accept.

### 11.1.3 File Format

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

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

- (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 codecs are:

- 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