Powered by Universal Speech Solutions LLC

UniMRCP

Yandex SR Plugin

Usage Guide

Revision: 5

Created: December 22, 2018

Last updated: February 26, 2021

Author: Arsen Chaloyan

Table of Contents

1 Overview 4

1.1 Installation 4
1.2 Applicable Versions 4

2 Supported Features 5

2.1 MRCP Methods 5
2.2 MRCP Events 5
2.3 MRCP Header Fields 5
2.4 Grammars 6
2.5 Results 6

3 Configuration Format 7

3.1 Document 7
3.2 Streaming Recognition 8
3.3 Speech Contexts 10
3.4 Speech Context 11
3.5 Phrase 12
3.6 Speech and DTMF Input Detector 13
3.7 Utterance Manager 14
3.8 RDR Manager 15
3.9 Monitoring Agent 16
3.10 Usage Change Handler 17
3.11 Usage Refresh Handler 18
3.12 License Server 18

4 Configuration Steps 21

4.1 Using Default Configuration 21
4.2 Specifying Folder ID 21
4.3 Specifying Recognition Language 21
4.4 Specifying Sampling Rate 21
4.5 Specifying Speech Input Parameters 22
4.6 Specifying DTMF Input Parameters 22
4.7 Specifying No-Input and Recognition Timeouts 23
4.8 Specifying Speech Recognition Mode 23
4.9 Specifying Vendor-Specific Parameters 23
4.10 Maintaining Utterances 25
4.11 Maintaining Recognition Details Records 26

5 Recognition Grammars and Results 27

5.1 Using Built-in Speech Contexts 27
5.2 Using Dynamic Speech Contexts 28

5.3 Using Built-in DTMF Grammars........28
5.4 Retrieving Results........28

6 Monitoring Usage Details........29

6.1 Log Usage........29
6.2 Update Usage........29
6.3 Dump Channels........30

7 Usage Examples........31

7.1 Speech Recognition without Speech Context........31
7.2 Speech Recognition with Built-in Speech Context........32
7.3 Speech Recognition with Dynamic Speech Context........33
7.4 DTMF Recognition with Built-in Grammar........35
7.5 Speech and DTMF Recognition........36

8 Sequence Diagrams........38

8.1 MRCPv1........38
8.2 MRCPv2........38

9 Security Considerations........40

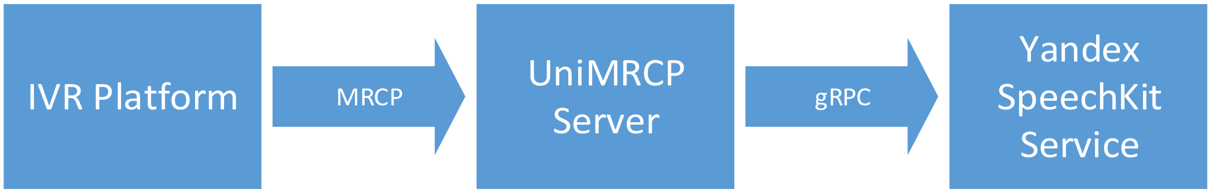
9.1 Network Connection........40
9.2 Network Port........40

10 References........41

10.1 Yandex SpeechKit........41
10.2 Specifications........41

1 Overview

This guide describes how to configure and use the Yandex Speech Recognition (SR) plugin to the UniMRCP server. The document is intended for users having a certain knowledge of Yandex SpeechKit Speech-to-Text API and UniMRCP.



1.1 Installation

For installation instructions, use one of the guides below.

- RPM Package Installation (Red Hat / Cent OS)
- Deb Package Installation (Debian / Ubuntu)

1.2 Applicable Versions

Instructions provided in this guide are applicable to the following versions.

UniMRCP 1.6.0 and above

UniMRCP Yandex SR Plugin 1.0.0 and above

2 Supported Features

This is a brief check list of the features currently supported by the UniMRCP server running with the Yandex SR plugin.

2.1 MRCP Methods

- ✓ DEFINE-GRAMMAR
- ✓ RECOGNIZE
- ✓ START-INPUT-TIMERS
- ✓ STOP
- ✓ SET-PARAMS
- ✓ GET-PARAMS

2.2 MRCP Events

- ✓ RECOGNITION-COMPLETE
- ✓ START-OF-INPUT

2.3 MRCP Header Fields

- ✓ Input-Type
- ✓ No-Input-Timeout
- ✓ Recognition-Timeout
- ✓ Speech-Complete-Timeout
- ✓ Speech-Incomplete-Timeout
- ✓ Waveform-URI
- ✓ Media-Type
- ✓ Completion-Cause
- ✓ Confidence-Threshold
- ✓ Start-Input-Timers
- ✓ DTMF-Interdigit-Timeout
- ✓ DTMF-Term-Timeout
- ✓ DTMF-Term-Char
- ✓ Save-Waveform
- ✓ Speech-Language
- ✓ Cancel-If-Queue

- ✓ Sensitivity-Level

2.4 Grammars

- ✓ Built-in and dynamic speech contexts
- ✓ Built-in/embedded DTMF grammar
- ✓ SRGS XML (limited support)

2.5 Results

- ✓ NLSML

3 Configuration Format

The configuration file of the Yandex SR plugin is located in */opt/unimrcp/conf/umsyandexsr.xml*. The configuration file is written in XML.

3.1 Document

The root element of the XML document must be <*umsyandexsr*>.

Attributes

Name	Unit	Description
license-file	File path	Specifies the license file. File name may include patterns containing '*' sign. If multiple files match the pattern, the most recent one gets used.
subscription-key-file	File path	Specifies the Yandex SpeechKit subscription key file to use. File name may include patterns containing '*' sign. If multiple files match the pattern, the most recent one gets used.

Parent

None.

Children

Name	Unit	Description
<streaming-recognition>	String	Specifies parameters of streaming recognition employed via gRPC.
<speech-contexts>	String	Contains a list of speech contexts.
<speech-dtmf-input-detector>	String	Specifies parameters of the speech and DTMF input detector.
<utterance-manager>	String	Specifies parameters of the utterance manager.
<rdr-manager>	String	Specifies parameters of the Recognition Details Record (RDR) manager.
<monitoring-agent>	String	Specifies parameters of the monitoring manager.

<license-server>	String	Specifies parameters used to connect to the license server. The use of the license server is optional.

Example

This is an example of a bare document.

```
<umsyandexsr license-file="umsyandexsr_*.lic"
             subscription-key-file ="yandex.subscription.key">
</ umsyandexsr>
```

3.2 Streaming Recognition

This element specifies parameters of streaming recognition.

Attributes

Name	Unit	Description
folder-id	String	Specifies the Yandex SpeechKit folder identifier. Required.
language	String	Specifies the default language to use, if not set by the client.
model	String	Specifies the language model to use. Can be overridden by client. Available since YSR 1.4.0.
raw-results	Boolean	Specifies whether to provide raw ("nine applies") or normalized ("9 apples") results. Can be overridden by client. Available since YSR 1.4.0.
data-logging	Boolean	Specifies whether to enable data logging at the service side. Can be overridden by client. Available since YSR 1.5.0.
interim-results	Boolean	Specifies whether to request interim results or not.
start-of-input	String	Specifies the source of start of input event sent to the client (use "service-originated" for an event originated based on a first-received interim result and "internal" for an event determined by plugin).
max-alternatives	Integer	Specifies the maximum number of speech recognition result alternatives to be returned. Can be overridden by

		client by means of the header field *N-Best-List-Length*.
alternatives-below-threshold	Boolean	Specifies whether to return speech recognition result alternatives with the confidence score below the confidence threshold.
confidence-format	String	Specifies the format of the confidence score to be returned (use "auto" for a format based on protocol version, "mrcpv2" for a float value in the range of 0..1, "mrcpv1" for an integer value in the range of 0..100).
tag-format	String	Specifies the format of the instance to be returned (use "extended" to have the recorded utterance name concatenated to the result with a sign '|'). Can be overridden by client. Available since YSR 1.5.0.
single-utterance	Boolean	Specifies whether to detect a single spoken utterance or perform continuous recognition.
results-indent	Integer	Specifies the indentation used to compose NLSML results.
skip-unsupported-grammars	Boolean	Specifies whether to skip or raise an error while referencing a malformed or not supported grammar.
transcription-grammar	String	Specifies the name of the built-in speech transcription grammar. The grammar can be referenced as *builtin:speech/transcribe* or *builtin:grammar/transcribe*, where *transcribe* is the default value of this parameter.
http-proxy	String	Specifies the URI of HTTP proxy, if used. Available since YSR 1.2.0.
profanity-filter	Boolean	Specifies whether to filter out words containing profanity. Can be overridden by client. Available since YSR 1.6.0.
match-srgs	Boolean	Specifies whether to match transcribed text specified SRGS grammar. Can be overridden by the 'scope' attribute in the grammar. Available since YSR 1.6.0.
stream-creation-timeout	Time interval [msec]	Specifies how long to wait for gRPC stream creation. If timeout is set 0, no timer is used. Otherwise, if timeout is elapsed, gRPC stream creation is cancelled. Available since YSR 1.6.0.
inter-result-timeout	Time interval [msec]	Specifies a timeout between interim results containing transcribed speech. If the timeout is elapsed, input is considered complete. The timeout defaults to 0 (disabled). Available since YSR 1.6.0.
grpc-log-redirection	Boolean	Specifies whether to enable gRPC log redirection.

		Available since YSR 1.6.0.
grpc-log-verbosity	String	Specifies gRPC logging verbosity. One of DEBUG, INFO, ERROR. See GRPC_VERBOSITY for more info. Available since YSR 1.6.0.
grpc-log-trace	String	Specifies a comma separated list of tracers producing gRPC logs. Use 'all' to turn all tracers on. See GRPC_TRACE for more info. Available since YSR 1.6.0.

Parent

<umsyandexsr>

Children

None.

Example

This is an example of streaming recognition element.

```
<streaming-recognition
   folder-id="abcd1234edfg"
   interim-results="true"
   start-of-input="service-originated"
   language="en-US"
   max-alternatives="1"
   alternatives-below-threshold="false"
   confidence-format="auto"
   single-utterance="true"
   results-indent="2"
   skip-unsupported-grammars="true"
   transcription-grammar="transcribe"
/>
```

3.3 Speech Contexts

This element specifies a list of speech contexts.

Attributes

None.

Parent

<umsyandexsr>

Children

<speech-context>

Example

The example below defines two speech contexts *booking* and *directory*.

```
<speech-contexts>
  <speech-context id="booking" enable="true">
    <phrase>I would like to book a flight from New York to Rome with a ticket eligible for free cancellation</phrase>
    <phrase>I would like to book a one-way flight from New York to Rome</phrase>
  </speech-context>

  <speech-context id="directory" enable="true">
    <phrase>call Steve</phrase>
    <phrase>call John</phrase>
    <phrase>dial 5</phrase>
    <phrase>dial 6</phrase>
  </speech-context>
</speech-contexts>
```

3.4 Speech Context

This element specifies a speech context.

Attributes

Name	Unit	Description
id	String	Specifies a unique string identifier of the speech context to be referenced by the MRCP client.
enable	Boolean	Specifies whether the speech context is enabled or disabled.
speech-complete	Boolean	Specifies whether to complete input as soon as an interim result matches one of the specified phrases.
language	String	The language the phrases are defined for.

Parent

<speech-contexts>

Children

<phrase>

Example

This is an example of speech context element.

```
<speech-context id="directory" enable="true">
  <phrase>call Steve</phrase>
  <phrase>call John</phrase>
  <phrase>dial 5</phrase>
  <phrase>dial 6</phrase>
</speech-context>
```

3.5 Phrase

This element specifies a phrase in the speech context.

Attributes

Name	Unit	Description
tag	String	Specifies an optional arbitrary string identifier to be returned as an instance in the NLSML result, if the transcription result matches the phrase.

Parent

<speech-context>

Children

None.

This is an example of a speech context with phrases having tags specified.

```
<speech-context id="boolean" speech-complete="true" enable="true">
  <phrase tag="true">yes</phrase>
  <phrase tag="true">sure</phrase>
  <phrase tag="true">correct</phrase>
  <phrase tag="false">no</phrase>
  <phrase tag="false">not sure</phrase>
  <phrase tag="false">incorrect </phrase>
</speech-context>
```

3.6 Speech and DTMF Input Detector

This element specifies parameters of the speech and DTMF input detector.

Attributes

Name	Unit	Description
vad-mode	Integer	Specifies an operating mode of VAD in the range of [0 ... 3]. Default is 1.
speech-start-timeout	Time interval [msec]	Specifies how long to wait in transition mode before triggering a start of speech input event.
speech-complete-timeout	Time interval [msec]	Specifies how long to wait in transition mode before triggering an end of speech input event. The complete timeout is used when there is an interim result available.
speech-incomplete-timeout	Time interval [msec]	Specifies how long to wait in transition mode before triggering an end of speech input event. The incomplete timeout is used as long as there is no interim result available. Afterwards, the complete timeout is used.
noinput-timeout	Time interval [msec]	Specifies how long to wait before triggering a no-input event.
input-timeout	Time interval [msec]	Specifies how long to wait for input to complete.
dtmf-interdigit-timeout	Time interval [msec]	Specifies a DTMF inter-digit timeout.
dtmf-term-timeout	Time interval [msec]	Specifies a DTMF input termination timeout.
dtmf-term-char	Character	Specifies a DTMF input termination character.
speech-leading-silence	Time interval [msec]	Specifies desired silence interval preceding spoken input.
speech-trailing-silence	Time interval [msec]	Specifies desired silence interval following spoken input.
speech-output-period	Time interval [msec]	Specifies an interval used to send speech frames to the recognizer.

Parent

<umsyandexsr>

Children

None.

Example

The example below defines a typical speech and DTMF input detector having the default parameters set.

```
<speech-dtmf-input-detector
  vad-mode="2"
  speech-start-timeout="300"
  speech-complete-timeout="1000"
  speech-incomplete-timeout="3000"
  noinput-timeout="5000"
  input-timeout="10000"
  dtmf-interdigit-timeout="5000"
  dtmf-term-timeout="10000"
  dtmf-term-char=""
  speech-leading-silence="300"
  speech-trailing-silence="300"
  speech-output-period="200"
/>
```

3.7 Utterance Manager

This element specifies parameters of the utterance manager.

Attributes

Name	Unit	Description
save-waveforms	Boolean	Specifies whether to save waveforms or not.
purge-existing	Boolean	Specifies whether to delete existing records on start-up.
max-file-age	Time interval [min]	Specifies a time interval in minutes after expiration of which a waveform is deleted. Set 0 for infinite.
max-file-count	Integer	Specifies the max number of waveforms to store. If reached, the oldest waveform is

		deleted. Set 0 for infinite.
waveform-base-uri	String	Specifies the base URI used to compose an absolute waveform URI.
waveform-folder	Dir path	Specifies a folder the waveforms should be stored in.
file-prefix	String	Specifies a prefix used to compose the name of the file to be stored. Defaults to 'umsyandexsr-', if not specified.
use-logging-tag	Boolean	Specifies whether to use the MRCP header field Logging-Tag, if present, to compose the name of the file to be stored. Available since YSR 1.6.0.

Parent

<umsyandexsr>

Children

None.

Example

The example below defines a typical utterance manager having the default parameters set.

```
<utterance-manager
  save-waveforms="false"
  purge-existing="false"
  max-file-age="60"
  max-file-count="100"
  waveform-base-uri="http://localhost/utterances/"
  waveform-folder=""
/>
```

3.8 RDR Manager

This element specifies parameters of the Recognition Details Record (RDR) manager.

Attributes

Name	Unit	Description
save-records	Boolean	Specifies whether to save recognition details

		records or not.
purge-existing	Boolean	Specifies whether to delete existing records on start-up.
max-file-age	Time interval [min]	Specifies a time interval in minutes after expiration of which a record is deleted. Set 0 for infinite.
max-file-count	Integer	Specifies the max number of records to store. If reached, the oldest record is deleted. Set 0 for infinite.
record-folder	Dir path	Specifies a folder to store recognition details records in. Defaults to ${UniMRCPInstallDir}/var.
file-prefix	String	Specifies a prefix used to compose the name of the file to be stored. Defaults to 'umsyandexsr-', if not specified.
use-logging-tag	Boolean	Specifies whether to use the MRCP header field Logging-Tag, if present, to compose the name of the file to be stored. Available since YSR 1.6.0.

Parent

<umsyandexsr>

Children

None.

Example

The example below defines a typical utterance manager having the default parameters set.

```
<rdr-manager
  save-records="false"
  purge-existing="false"
  max-file-age="60"
  max-file-count="100"
  waveform-folder=""
/>
```

3.9 Monitoring Agent

This element specifies parameters of the monitoring agent.

Attributes

Name	Unit	Description
refresh-period	Time interval [sec]	Specifies a time interval in seconds used to periodically refresh usage details. See <usage-refresh-handler>.

Parent

<umsyandexsr>

Children

<usage-change-handler>
<usage-refresh-handler>

Example

The example below defines a monitoring agent with usage change and refresh handlers.

```
<monitoring-agent refresh-period="60">

  <usage-change-handler>
    <log-usage enable="true" priority="NOTICE"/>
  </usage-change-handler>

  <usage-refresh-handler>
    <dump-channels enable="true" status-file="umsyandexsr-channels.status"/>
  </usage-refresh-handler >

</monitoring-agent>
```

3.10 Usage Change Handler

This element specifies an event handler called on every usage change.

Attributes

None.

Parent

<monitoring-agent>

Children

<log-usage>

<update-usage>
<dump-channels>

Example

This is an example of the usage change event handler.

```
<usage-change-handler>
  <log-usage enable="true" priority="NOTICE"/>
  <update-usage enable="false" status-file="umsyandexsr-usage.status"/>
  <dump-channels enable="false" status-file="umsyandexsr-channels.status"/>
</usage-change-handler>
```

3.11 Usage Refresh Handler

This element specifies an event handler called periodically to update usage details.

Attributes

None.

Parent

<monitoring-agent>

Children

<log-usage>
<update-usage>
<dump-channels>

Example

This is an example of the usage change event handler.

```
<usage-refresh-handler>
  <log-usage enable="true" priority="NOTICE"/>
  <update-usage enable="false" status-file="umsyandexsr-usage.status"/>
  <dump-channels enable="false" status-file="umsyandexsr-channels.status"/>
</usage-refresh-handler>
```

3.12 License Server

This element specifies parameters used to connect to the license server.

Attributes

Name	Unit	Description
enable	Boolean	Specifies whether the use of license server is enabled or not. If enabled, the license-file attribute is not honored.
server-address	String	Specifies the IP address or host name of the license server.
certificate-file	File path	Specifies the client certificate used to connect to the license server. File name may include patterns containing a '*' sign. If multiple files match the pattern, the most recent one gets used.
ca-file	File path	Specifies the certificate authority used to validate the license server.
channel-count	Integer	Specifies the number of channels to check out from the license server. If not specified or set to 0, either all available channels or a pool of channels will be checked based on the configuration of the license server.
http-proxy-address	String	Specifies the IP address or host name of the HTTP proxy server, if used. Available since YSR 1.6.0.
http-proxy-port	Integer	Specifies the port number of the HTTP proxy server, if used. Available since YSR 1.6.0.

Parent

<umsyandexsr>

Children

None.

Example

The example below defines a typical configuration which can be used to connect to a license server located, for example, at 10.0.0.1.

```
<license-server
  enable="true"
  server-address="10.0.0.1"
  certificate-file="unilic_client_*.crt"
```

```
        ca-file="unilic_ca.crt"
      />
```

For further reference to the license server, visit

```
    http://unimrcp.org/licserver
```

4 Configuration Steps

This section outlines common configuration steps.

4.1 Using Default Configuration

The default configuration should be sufficient for the general use.

4.2 Specifying Folder ID

The Yandex SpeechKit folder identifier must be set in the configuration file *umsyandexsr.xml*.

4.3 Specifying Recognition Language

Recognition language can be specified by the client per MRCP session by means of the header field *Speech-Language* set in a *SET-PARAMS* or *RECOGNIZE* request. Otherwise, the parameter *language* set in the configuration file *umsyandexsr.xml* is used. The parameter defaults to *en-US*.

Since YSR 1.6.0, the recognition language can also be set by the attribute *xml:lang* specified in the SRGS grammar.

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar mode="voice" root="transcribe" version="1.0"
        xml:lang="en-US"
        xmlns="http://www.w3.org/2001/06/grammar">
  <meta name="scope" content="builtin"/>
  <rule id="transcribe"><one-of/></rule>
</grammar>
```

Since YSR 1.6.0, the recognition language can also be set by the optional parameter *language* passed to a built-in grammar.

```
builtin:speech/transcribe?language=en-US
```

4.4 Specifying Sampling Rate

Sampling rate is determined based on the SDP negotiation. Refer to the configuration guide of the UniMRCP server on how to specify supported encodings and sampling rates to be used in communication between the client and server.

The native sampling rate with the linear16 audio encoding is used in gRPC streaming to the Yandex Cloud SpeechKit service.

4.5 Specifying Speech Input Parameters

While the default parameters specified for the speech input detector are sufficient for the general use, various parameters can be adjusted to better suit a particular requirement.

- speech-start-timeout

This parameter is used to trigger a start of speech input. The shorter is the timeout, the sooner a *START-OF-INPUT* event is delivered to the client. However, a short timeout may also lead to a false positive.

- speech-complete-timeout

This parameter is used to trigger an end of speech input. The shorter is the timeout, the shorter is the response time. However, a short timeout may also lead to a false positive.
Note that both events, an expiration of the speech complete timeout and an *END-OF-SINGLE-UTTERANCE* response delivered from the Yandex Cloud SpeechKit service, are monitored to trigger an end of speech input, on whichever comes first basis. In order to rely solely on an event delivered from the speech service, the parameter *speech-complete-timeout* needs to be set to a higher value.

- vad-mode

This parameter is used to specify an operating mode of the Voice Activity Detector (VAD) within an integer range of [0 … 3]. A higher mode is more aggressive and, as a result, is more restrictive in reporting speech. The parameter can be overridden per MRCP session by setting the header field *Sensitivity-Level* in a *SET-PARAMS* or *RECOGNIZE* request. The following table shows how the *Sensitivity-Level* is mapped to the *vad-mode*.

Sensitivity-Level	Vad-Mode
[0.00 … 0.25)	0
[0.25 … 0.50)	1
[0.50 … 0.75)	2
[0.75 … 1.00]	3

4.6 Specifying DTMF Input Parameters

While the default parameters specified for the DTMF input detector are sufficient for the general use, various parameters can be adjusted to better suit a particular requirement.

- dtmf-interdigit-timeout

This parameter is used to set an inter-digit timeout on DTMF input. The parameter can be overridden per MRCP session by setting the header field *DTMF-Interdigit-Timeout* in a *SET-PARAMS* or *RECOGNIZE* request.

- dtmf-term-timeout

This parameter is used to set a termination timeout on DTMF input and is in effect when dtmf-term-char is set and there is a match for an input grammar. The parameter can be overridden per MRCP session by setting the header field *DTMF-Term-Timeout* in a *SET-PARAMS* or *RECOGNIZE* request.

- dtmf-term-char

This parameter is used to set a character terminating DTMF input. The parameter can be overridden per MRCP session by setting the header field *DTMF-Term-Char* in a *SET-PARAMS* or *RECOGNIZE* request.

4.7 Specifying No-Input and Recognition Timeouts

- noinput-timeout

This parameter is used to trigger a no-input event. The parameter can be overridden per MRCP session by setting the header field *No-Input-Timeout* in a *SET-PARAMS* or *RECOGNIZE* request.

- input-timeout

This parameter is used to limit input (recognition) time. The parameter can be overridden per MRCP session by setting the header field *Recognition-Timeout* in a *SET-PARAMS* or *RECOGNIZE* request.

4.8 Specifying Speech Recognition Mode

Single Utterance Mode

By default, if the configuration parameter *single-utterance* is set to true, recognition is performed in the single utterance mode and is terminated upon an expiration of the speech complete timeout or an *END-OF-SINGLE-UTTERANCE* response delivered from the service.

Continuous Recognition Mode

In the continuous speech recognition mode, when the configuration parameter *single-utterance* is set to false, recognition is terminated upon an expiration of the speech complete timeout, which is recommended to be set in the range of 1500 msec to 3000 msec. The service may return multiple results (sub utterances), which are concatenated and sent back to the MRCP client in a single RECOGNITION-COMPLETE event.

The parameter *single-utterance* can be overridden per MRCP session by setting the header field *Vendor-Specific-Parameters* in a *SET-PARAMS* or *RECOGNIZE* request, where the parameter name is *single-utterance* and acceptable values are *true* and *false*.

4.9 Specifying Vendor-Specific Parameters

The following parameters can optionally be specified by the MRCP client in *SET-PARAMS, DEFINE-GRAMMAR* and *RECOGNIZE* requests via the MRCP header field *Vendor-Specific-Parameters*.

Name	Unit	Description

folder-id	String	Specifies the Yandex SpeechKit folder identifier. Available since YSR 1.4.0.
model	String	Specifies the language model to use. Available since YSR 1.4.0.
raw-results	Boolean	Specifies whether to provide raw ("nine applies") or normalized ("9 apples") results. Available since YSR 1.4.0.
start-of-input	String	Specifies the source of start of input event sent to the client (use "service-originated" for an event originated based on a first-received interim result and "internal" for an event determined by plugin). Available since YSR 1.4.0.
alternatives-below-threshold	Boolean	Specifies whether to return speech recognition result alternatives with the confidence score below the confidence threshold. Available since YSR 1.4.0.
single-utterance	Boolean	Specifies whether to detect a single spoken utterance or perform continuous recognition. Available since YSR 1.4.0.
speech-start-timeout	Time interval [msec]	Specifies how long to wait in transition mode before triggering a start of speech input event. Available since YSR 1.4.0.
tag-format	String	Specifies the format of the instance to be returned (use "extended" to have the recorded utterance name concatenated to the result with a sign '|'). Available since YSR 1.5.0.
data-logging	Boolean	Specifies whether to enable data logging at the service side. Available since YSR 1.5.0.
profanity-filter	Boolean	Specifies whether to attempt to filter out profanities. Available since YSR 1.6.0.
interim-result-timeout	Time interval [msec]	Specifies a timeout between interim results containing transcribed speech. If the timeout is elapsed, input is considered complete. Available since YSR 1. 6.0.
logging-tag	String	Specifies the logging tag. Available since YSR 1.6.0.

All the vendor-specific parameters can also be specified at the grammar-level via a built-in or SRGS XML grammar.

The following example demonstrates the use of a built-in grammar with the vendor-specific parameters *model* and *raw-results* set to *general* and *true* correspondingly.

```
builtin:speech/transcribe?model=general;raw-results=true
```

The following example demonstrates the use of an SRGS XML grammar with the vendor-specific parameters *model* and *raw-results* set to *general* and *true* correspondingly.

```
<grammar mode="voice" root="transcribe" version="1.0" xml:lang="en-US"
xmlns="http://www.w3.org/2001/06/grammar">
   <meta name="scope" content="builtin"/>
   <meta name="model" content="general"/>
   <meta name="raw-results" content="true"/>
   <rule id="transcribe">
      <one-of ><item>blank</item></one-of>
   </rule>
</grammar>
```

4.10Maintaining Utterances

Saving of utterances is not required for regular operation and is disabled by default. However, enabling this functionality allows to save utterances sent to the service and later listen to them offline.

The relevant settings can be specified via the element *utterance-manager*.

- save-waveforms

  Utterances can optionally be recorded and stored if the configuration parameter *save-waveforms* is set to true. The parameter can be overridden per MRCP session by setting the header field *Save-Waveforms* in a *SET-PARAMS* or *RECOGNIZE* request.

- purge-existing

  This parameter specifies whether to delete existing waveforms on start-up.

- max-file-age

  This parameter specifies a time interval in minutes after expiration of which a waveform is deleted. If set to 0, there is no expiration time specified.

- max-file-count

  This parameter specifies the maximum number of waveforms to store. If the specified number is reached, the oldest waveform is deleted. If set to 0, there is no limit specified.

- waveform-base-uri

  This parameter specifies the base URI used to compose an absolute waveform URI returned in the header field *Waveform-Uri* in response to a RECOGNIZE request.

- waveform-folder

  This parameter specifies a path to the directory used to store waveforms in. The directory defaults to

${UniMRCPInstallDir}/var.

4.11 Maintaining Recognition Details Records

Producing of recognition details records (RDR) is not required for regular operation and is disabled by default. However, enabling this functionality allows to store details of each recognition attempt in a separate file and analyze them later offline. The RDRs ate stored in the JSON format.

The relevant settings can be specified via the element *rdr-manager*.

- save-records

This parameter specifies whether to save recognition details records or not.

- purge-existing

This parameter specifies whether to delete existing records on start-up.

- max-file-age

This parameter specifies a time interval in minutes after expiration of which a record is deleted. If set to 0, there is no expiration time specified.

- max-file-count

This parameter specifies the maximum number of records to store. If the specified number is reached, the oldest record is deleted. If set to 0, there is no limit specified.

- record-folder

This parameter specifies a path to the directory used to store records in. The directory defaults to *${UniMRCPInstallDir}/var*.

5 Recognition Grammars and Results

5.1 Using Built-in Speech Contexts

Pre-set built-in speech contexts can be referenced by the MRCP client in a RECOGNIZE request as follows:

```
builtin:speech/$id
```

Where *$id* is a unique string identifier of built-in speech context.

Speech contexts are defined in the configuration file *umsyandexsr.xml*. A speech context is assigned a unique string identifier and holds a list of phrases.

Below is a definition of a sample speech context *directory*:

```
<speech-context id="directory">
  <phrase>call Steve</phrase>
  <phrase>call John</phrase>
  <phrase>dial 5</phrase>
  <phrase>dial 6</phrase>
</speech-context>
```

Which can be referenced in a RECOGNIZE request as follows:

```
builtin:speech/directory
```

The prefixes *builtin:speech* and *builtin:grammar* can be used interchangeably as follows:

```
builtin:grammar/directory
```

For generic speech transcription, having no speech contexts defined, a pre-set identifier *transcribe* must be used.

```
builtin:speech/transcribe
```

The name of the identifier *transcribe* can be changed from the configuration file *umsyandexsr.xml*.

5.2 Using Dynamic Speech Contexts

The MRCP client can also dynamically specify a speech context either

- in a DEFINE-GRAMMAR request by further referencing the defined speech context in a RECOGNIZE request using the session URI scheme
- or inline in a RECOGNIZE request

While composing a DEFINE-GRAMMAR or RECOGNIZE request containing speech context definition, the following should be considered.

- The value of the header field *Content-Id* must be used as a unique string identifier of the speech context being defined.
- The value of the header field *Content-Type* must be set to *application/xml*.
- The message body must contain a definition of the speech context, composed based on the XML format of the element <speech-context>, specified in the configuration file *umsyandexsr.xml*. Note that the unique identifier of the speech context is set based on the header field *Content-Id*, as opposed to the attribute *Id* when loading from configuration.

5.3 Using Built-in DTMF Grammars

Pre-set built-in DTMF grammars can be referenced by the MRCP client in a RECOGNIZE request as follows:

```
builtin:dtmf/$id
```

Where *$id* is a unique string identifier of the built-in DTMF grammar. For example:

```
builtin:dtmf/digits
```

Note that only a DTMF grammar identifier *digits* is currently supported.

5.4 Retrieving Results

Results received from the Yandex Cloud SpeechKit service are transformed to the NLSML format with no semantic interpretation performed and sent to the MRCP client in a *RECOGNITION-COMPLETE* event.

6 Monitoring Usage Details

The number of in-use and total licensed channels can be monitored in several alternate ways. There is a set of actions which can take place on certain events. The behavior is configurable via the element *monitoring-agent*, which contains two event handlers: *usage-change-handler* and *usage-refresh-handler*.

While the *usage-change-handler* is invoked on every acquisition and release of a licensed channel, the *usage-refresh-handler* is invoked periodically on expiration of a timeout specified by the attribute *refresh-period*.

The following actions can be specified for either of the two handlers.

6.1 Log Usage

The action *log-usage* logs the following data in the order specified.

- The number of currently in-use channels.
- The maximum number of channels used concurrently.
- The total number of licensed channels.

The following is a sample log statement, indicating 0 in-use, 0 max-used and 2 total channels.

```
[NOTICE] YandexSR Usage: 0/0/2
```

6.2 Update Usage

The action *update-usage* writes the following data to a status file *umsyandexsr-usage.status*, located by default in the directory *${UniMRCPInstallDir}/var/status*.

- The number of currently in-use channels.
- The maximum number of channels used concurrently.
- The total number of licensed channels.
- The current status of the license permit.
- The license server alarm. Set to *on*, if the license server is not available for more than one hour; otherwise, set to *off*. This parameter is maintained only if the license server is used. Available since YSR 1.2.0.

The following is a sample content of the status file.

```
in-use channels: 0
```

```
        max used channels: 0
        total channels: 2
        license permit: true
        licserver alarm: off
```

6.3 Dump Channels

The action *dump-channels* writes the identifiers of in-use channels to a status file *umsyandexsr-channels.status*, located by default in the directory *${UniMRCPInstallDir}/var/status*.

7 Usage Examples

7.1 Speech Recognition without Speech Context

This example demonstrates how to perform speech recognition by using a RECOGNIZE request, having no speech contexts defined.

C->S:

```
MRCP/2.0 336 RECOGNIZE 1
Channel-Identifier: 6e1a2e4e54ae11e7@speechrecog
Content-Id: request1@form-level
Content-Type: text/uri-list
Cancel-If-Queue: false
No-Input-Timeout: 5000
Recognition-Timeout: 10000
Start-Input-Timers: true
Confidence-Threshold: 0.87
Save-Waveform: true
Content-Length: 25

builtin:speech/transcribe
```

S->C:

```
MRCP/2.0 83 1 200 IN-PROGRESS
Channel-Identifier: 6e1a2e4e54ae11e7@speechrecog
```

S->C:

```
MRCP/2.0 115 START-OF-INPUT 1 IN-PROGRESS
Channel-Identifier: 6e1a2e4e54ae11e7@speechrecog
Input-Type: speech
```

S->C:

```
MRCP/2.0 498 RECOGNITION-COMPLETE 1 COMPLETE
Channel-Identifier: 6e1a2e4e54ae11e7@speechrecog
Completion-Cause: 000 success
Waveform-Uri: <http://localhost/utterances/utter-6e1a2e4e54ae11e7-1.wav>;size=20480;duration=1280
```

```
Content-Type: application/x-nlsml
Content-Length: 214

<?xml version="1.0"?>
<result>
  <interpretation grammar="builtin:speech/transcribe" confidence="1">
    <instance>Dial 5</instance>
    <input mode="speech">Dial 5</input>
  </interpretation>
</result>
```

7.2 Speech Recognition with Built-in Speech Context

This example demonstrates how to perform speech recognition by using a RECOGNIZE request to reference a pre-set built-in speech context *directory* on the server.

C->S:

```
MRCP/2.0 335 RECOGNIZE 1
Channel-Identifier: 3ea18b9854af11e7@speechrecog
Content-Id: request1@form-level
Content-Type: text/uri-list
Cancel-If-Queue: false
No-Input-Timeout: 5000
Recognition-Timeout: 10000
Start-Input-Timers: true
Confidence-Threshold: 0.87
Save-Waveform: true
Content-Length: 24

builtin:speech/directory
```

S->C:

```
MRCP/2.0 83 1 200 IN-PROGRESS
Channel-Identifier: 3ea18b9854af11e7@speechrecog
```

S->C:

```
MRCP/2.0 115 START-OF-INPUT 1 IN-PROGRESS
Channel-Identifier: 3ea18b9854af11e7@speechrecog
Input-Type: speech
```

S->C:

```
MRCP/2.0 497 RECOGNITION-COMPLETE 1 COMPLETE
Channel-Identifier: 3ea18b9854af11e7@speechrecog
Completion-Cause: 000 success
Waveform-Uri: <http://localhost/utterances/utter-3ea18b9854af11e7-
1.wav>;size=20480;duration=1280
Content-Type: application/x-nlsml
Content-Length: 213

<?xml version="1.0"?>
<result>
  <interpretation grammar="builtin:speech/directory" confidence="1">
    <instance>call Steve</instance>
    <input mode="speech">call Steve</input>
  </interpretation>
</result>
```

7.3 Speech Recognition with Dynamic Speech Context

This examples demonstrates how to perform speech recognition, by using a DEFINE-GRAMMAR request to specify a speech context and further reference the defined speech context in a RECOGNIZE request.

C->S:

```
MRCP/2.0 314 DEFINE-GRAMMAR 1
Channel-Identifier: 25902c3a54b011e7@speechrecog
Content-Type: application/xml
Content-Id: request1@form-level
Content-Length: 146

<speech-context>
  <phrase>call Steve</phrase>
  <phrase>call John</phrase>
  <phrase>dial 5</phrase>
  <phrase>dial 6</phrase>
</speech-context>
```

S->C:

```
MRCP/2.0 112 1 200 COMPLETE
Channel-Identifier: 25902c3a54b011e7@speechrecog
Completion-Cause: 000 success
```

C->S:

```
MRCP/2.0 305 RECOGNIZE 2
Channel-Identifier: 25902c3a54b011e7@speechrecog
Content-Type: text/uri-list
Cancel-If-Queue: false
No-Input-Timeout: 5000
Recognition-Timeout: 10000
Start-Input-Timers: true
Confidence-Threshold: 0.87
Save-Waveform: true
Content-Length: 27

session:request1@form-level
```

S->C:

```
MRCP/2.0 83 2 200 IN-PROGRESS
Channel-Identifier: 25902c3a54b011e7@speechrecog
```

S->C:

```
MRCP/2.0 115 START-OF-INPUT 2 IN-PROGRESS
Channel-Identifier: 25902c3a54b011e7@speechrecog
Input-Type: speech
```

S->C:

```
MRCP/2.0 500 RECOGNITION-COMPLETE 2 COMPLETE
Channel-Identifier: 25902c3a54b011e7@speechrecog
Completion-Cause: 000 success
Waveform-Uri: <http://localhost/utterances/utter-25902c3a54b011e7-2.wav>;size=20480;duration=1280
Content-Type: application/x-nlsml
Content-Length: 216

<?xml version="1.0"?>
<result>
  <interpretation grammar="session:request1@form-level" confidence="1">
    <instance>call Steve</instance>
    <input mode="speech">call Steve</input>
  </interpretation>
</result>
```

7.4 DTMF Recognition with Built-in Grammar

This example demonstrates how to reference a built-in DTMF grammar in a RECOGNIZE request.

C->S:

```
MRCP/2.0 266 RECOGNIZE 1
Channel-Identifier: d26bef74091a174c@speechrecog
Content-Type: text/uri-list
Cancel-If-Queue: false
Start-Input-Timers: true
Confidence-Threshold: 0.7
Speech-Language: en-US
Dtmf-Term-Char: #
Content-Length: 19

builtin:dtmf/digits
```

S->C:

```
MRCP/2.0 83 1 200 IN-PROGRESS
Channel-Identifier: d26bef74091a174c@speechrecog
```

S->C:

```
MRCP/2.0 113 START-OF-INPUT 1 IN-PROGRESS
Channel-Identifier: d26bef74091a174c@speechrecog
Input-Type: dtmf
```

S->C:

```
MRCP/2.0 382 RECOGNITION-COMPLETE 1 COMPLETE
Channel-Identifier: d26bef74091a174c@speechrecog
Completion-Cause: 000 success
Content-Type: application/x-nlsml
Content-Length: 197

<?xml version="1.0"?>
```

```
<result>
  <interpretation grammar="builtin:dtmf/digits" confidence="1.00">
    <input mode="dtmf">1 2 3 4</input>
    <instance>1234</instance>
  </interpretation>
</result>
```

7.5 Speech and DTMF Recognition

This example demonstrates how to perform recognition by activating both speech and DTMF grammars. In this example, the user is expected to input a 4-digit pin.

C->S:

```
MRCP/2.0 275 RECOGNIZE 1
Channel-Identifier: 6ae0f23e1b1e3d42@speechrecog
Content-Type: text/uri-list
Cancel-If-Queue: false
Start-Input-Timers: true
Confidence-Threshold: 0.7
Speech-Language: en-US
Content-Length: 47

builtin:dtmf/digits?length=4
builtin:speech/pin
```

S->C:

```
MRCP/2.0 83 2 200 IN-PROGRESS
Channel-Identifier: 6ae0f23e1b1e3d42@speechrecog
```

S->C:

```
MRCP/2.0 115 START-OF-INPUT 2 IN-PROGRESS
Channel-Identifier: 6ae0f23e1b1e3d42@speechrecog
Input-Type: speech
```

S->C:

```
MRCP/2.0 399 RECOGNITION-COMPLETE 2 COMPLETE
Channel-Identifier: 6ae0f23e1b1e3d42@speechrecog
```

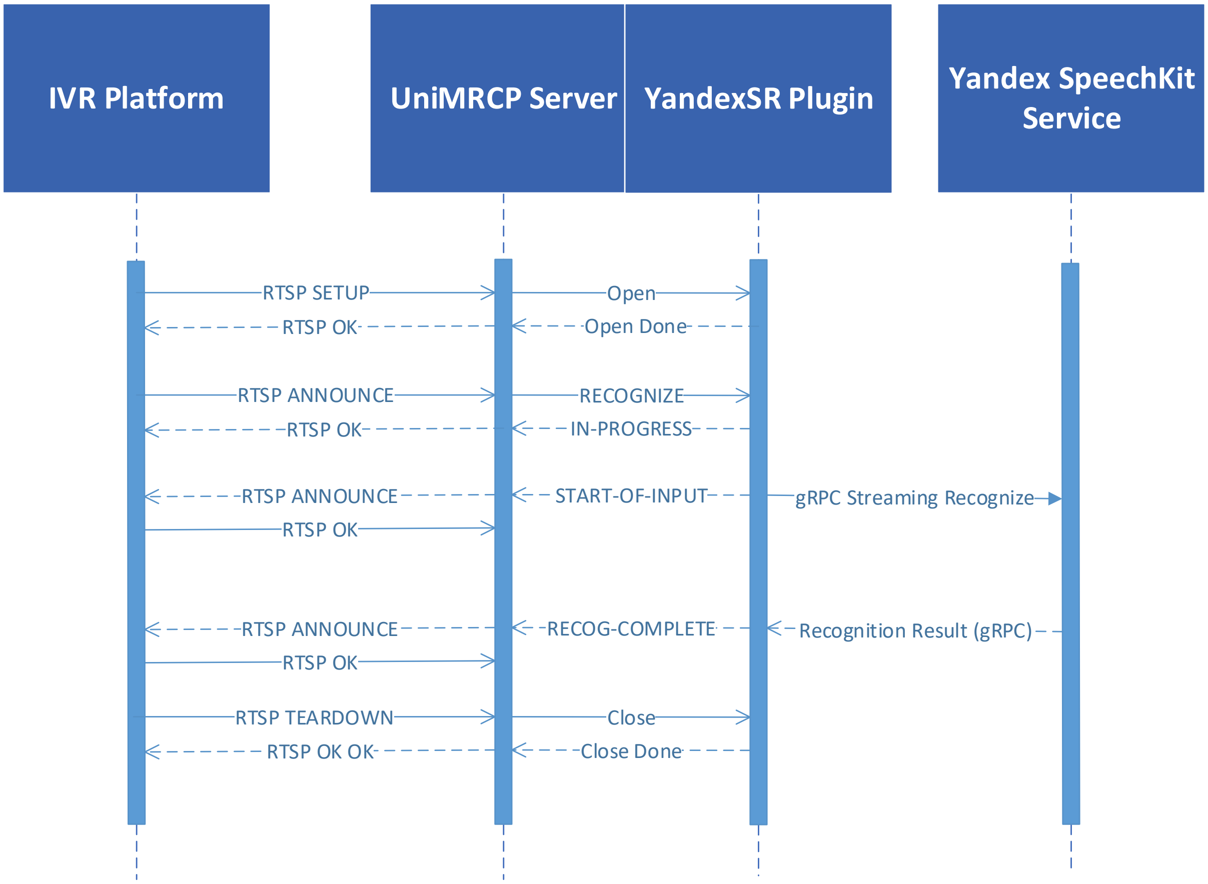
```
Completion-Cause: 000 success
Content-Type: application/x-nlsml
Content-Length: 214

<?xml version="1.0"?>
<result>
  <interpretation grammar=" builtin:speech/pin" confidence="1.00">
    <instance>one two three four</instance>
    <input mode="speech">one two three four</input>
  </interpretation>
</result>
```

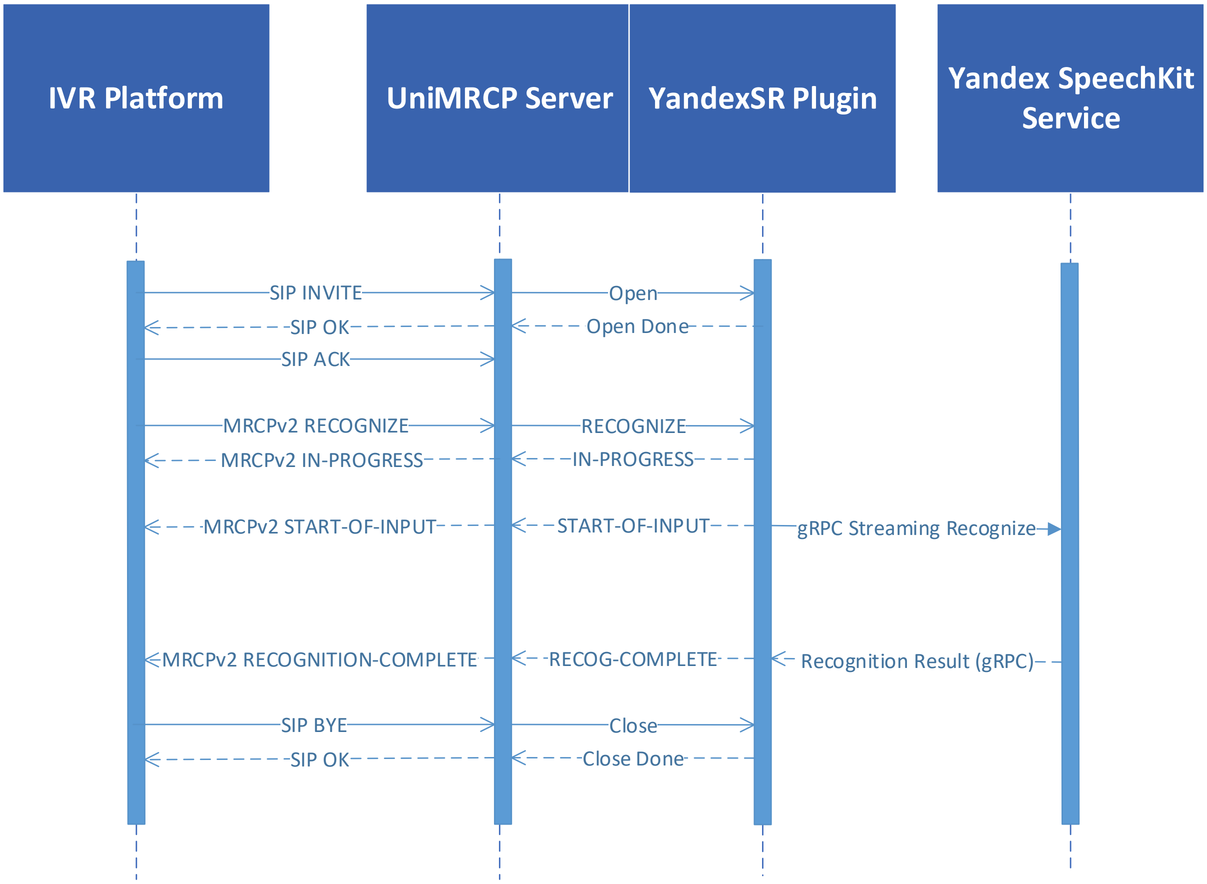
8 Sequence Diagrams

The following sequence diagrams outline common interactions between all the main components involved in a typical recognition session performed over MRCPv1 and MRCPv2 respectively.

8.1 MRCPv1



8.2 MRCPv2


IVR Platform
UniMRCP Server
YandexSR Plugin
Yandex SpeechKit Service
SIP INVITE
SIP OK
SIP ACK
MRCPv2 RECOGNIZE
MRCPv2 IN-PROGRESS
MRCPv2 START-OF-INPUT
MRCPv2 RECOGNITION-COMPLETE
SIP BYE
SIP OK
Open
Open Done
RECOGNIZE
IN-PROGRESS
START-OF-INPUT
RECOG-COMPLETE
Close
Close Done
gRPC Streaming Recognize
Recognition Result (gRPC)

9 Security Considerations

9.1 Network Connection

All the data transmitted to and received from the Yandex Cloud SpeechKit API is carried over a secure TLS v1.2 connection via the gRPC streaming.

9.2 Network Port

The standard TLS port 443 is used for the gRPC streaming.

10 References

10.1 Yandex SpeechKit

- Speech to Text API
- Authentication

10.2 Specifications

- Speech Recognizer Resource
- NLSML Results