

SIPob

Installation & Configuration Guide
Version 1.5

Document Revision 3.2

<https://www.kaplansoft.com/>

SIPob is built by Yasin KAPLAN

Read “Readme.txt” for last minute changes and updates, which can be found under the application directory.

Copyright © 2014-2025 KaplanSoft. All Rights Reserved. This document is supplied by KaplanSoft. No part of this document may be reproduced, republished or retransmitted in any form or by any means whatsoever, whether electronically or mechanically, including, but not limited to, by way of photocopying, recording, information recording or through retrieval systems, without the written permission of KaplanSoft. If you would like permission to use any of this material, please contact KaplanSoft.

KaplanSoft reserves the right to revise this document and make changes at any time without prior notice. Specifications contained in this document are subject to change without notice. Please send your comments by email to info@kaplansoft.com.

Microsoft, Win32, Windows 2000, Windows, Windows NT and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

KaplanSoft is the registered trademark of Kaplan Bilisim Teknolojileri Yazılım ve Ticaret Ltd.

ELEVENLABS is a trademark of Elevenlabs Inc.

Table of Contents

Table of Contents	3
Introduction	4
System Requirements	4
Installation	4
Configuration	4
Settings Tab	4
Settings / SIP Accounts Tab	6
Settings / TTS Profiles Tab	7
Settings / ASR Profiles Tab	7
Google API Key	8
Azure API Key	8
ElevenLabs API Key	9
Routing	9
Scenario Editor	10
Scenario Editor / Prompts Tab	11
Scenario Editor / Scenarios Tab	11
Dialer Profiles	15
Command Line Parameters	15
Troubleshooting	16
Echo Test Scenario	17
SIPob Messages	18
SIPob Commercial Mode	18
How to Record a Custom Audio Message	19

Introduction

SIPob is an outbound dialer based on RFC 3261 to make telephone calls to a set of user defined numbers. SIPob executes user defined scenarios when remote party answers. You can select your own audio files to be used in a scenario (*They must be a valid wav file*). SIPob can also read-out texts using TTS (*Text-to-Speech*) engine and perform voice recognition (*Speech to text*) using Microsoft SAPI, Google Cloud SAPI and ElevenLabs SAPI (*TTS Only*).

SIPob supports NAT traversal. You can log session details into a log file and monitor active sessions in real-time.

SIPob also supports UPnP IGD specification. If it is installed behind an UPnP supported Internet gateway device (*e.g., ADSL router*), SIPob automatically detects if it is behind a new NAT gateway and its external IP address. All outgoing requests are manipulated for NAT traversal. You do not need to add manual reverse mappings for SIP or RTP protocols.

SIPob supports UDP, TCP and TLS transports.

System Requirements

1. A Windows system with at least 2 GB of RAM.
2. Microsoft.NET Framework v4.8 (*Min.*)
3. 10 MB of disk space for installation.
4. Administrative privileges.

Installation

Unzip “SIPob.zip” and click the “Setup.exe” that comes with the distribution. Follow the instructions of the setup wizard. Setup will install SIPob and add a shortcut for SIPob to the desktop and the start menu.

Configuration

Run SIPob Manager from Start Menu / Program Files / SIPob. SIPob automatically configures itself at first run.

SIPob also checks if it is installed behind an UPnP supported NAT gateway. If so, SIPob automatically detects the external IP and displays it on the status bar. SIPob also adds a reverse mapping for incoming UDP connections automatically (*Default UDP port 5060*).

Settings Tab

Click the Settings Tab to start configuration. The settings tab has four sub sections.

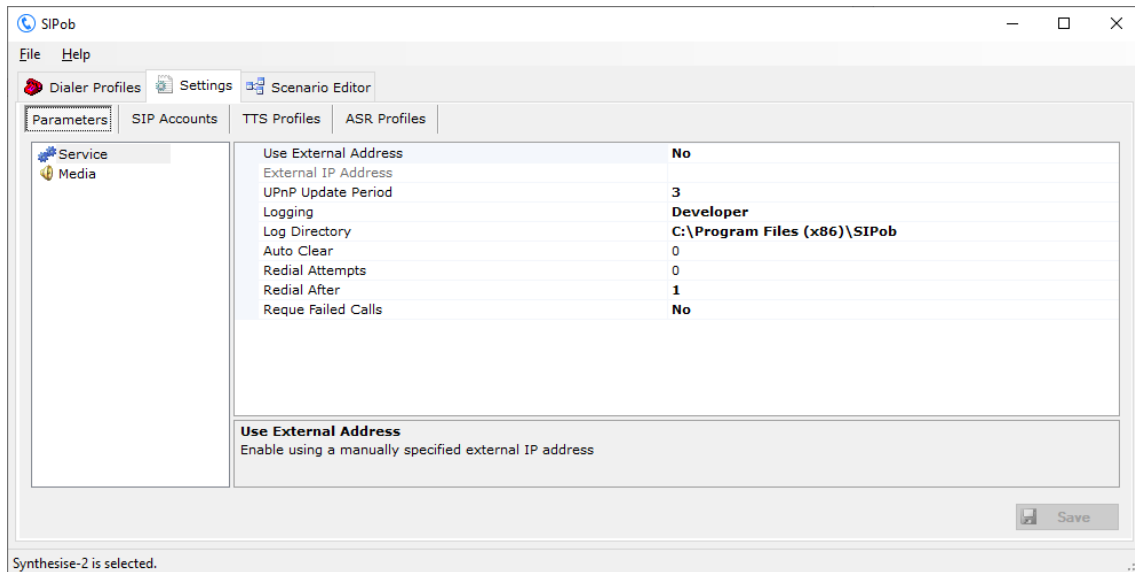


Figure - 1. SIPob Settings / Service Parameters

Enter the following information for the Settings / Parameters:

- **Use External Address:** If SIPob is installed behind a NAT gateway which does not support UPnP, you can set external IP address manually for NAT traversal. If your NAT gateway supports UPnP, set the UPnP Update Period to a value greater than “0”. You can specify a FQDN (*DynDNS address etc.*) as an external address; SIPob will query FQDN every minute for possible IP address changes.
- **UPnP Update Period:** You can specify the period for querying the UPnP Internet Access Gateway. Set to “0” to disable UPnP support.
- **Logging:** Select the logging level of SIPob. Select “None” if you do not want logging, select “Errors” to log errors, and select “Sessions” to log session information and errors. Log files are located under the <Application Directory>\Logs directory by default.
- **TTS Voice:** SIPob can synthesis audio based on the text entered in Audio tab. Select TTS voice profile for audio synthesis.
- **Redial Attempts:** Set how many times SIPob will try to make a call to a failed destination after the first attempt.
- **Redial After:** Specify time in seconds, how much time will SIPob wait between re-dial attempts.
- **Reque Failed Calls:** Set ‘Yes’ if you would like to auto-reque failed calls to be called on next processing of the dialer profile.

Enter the following information for the Settings / Media:

- **DTMF Transport:** TekIVR supports receiving DTMF digits via RFC 2833 and SIP INFO requests, but you must set DTMF delivery method to remote SIP endpoints.
- **Supported Codecs.** You can choose supported codecs; G.711 A-Law, G.711 Mu-Law, G.722 and G.729.
- **RTP Port Range:** Set upper and lower UDP ports for RTP port range.

- **TTS Cache Capacity:** TekIVR allows you to cache synthesized audio when this option is enabled. This reduces the number of requests made to TTS engine for identical text inputs. You can limit the number of the cached items to be kept in memory. Specify a value greater than zero to enable TTS caching.

If you click the [Save] button, the settings will be saved to SIPob.ini.

Settings / SIP Accounts Tab

Click Settings / SIP Accounts settings.

Type a SIP Account name to the bottom left text box and click the add button right to it. Enter the following information for a SIP account.

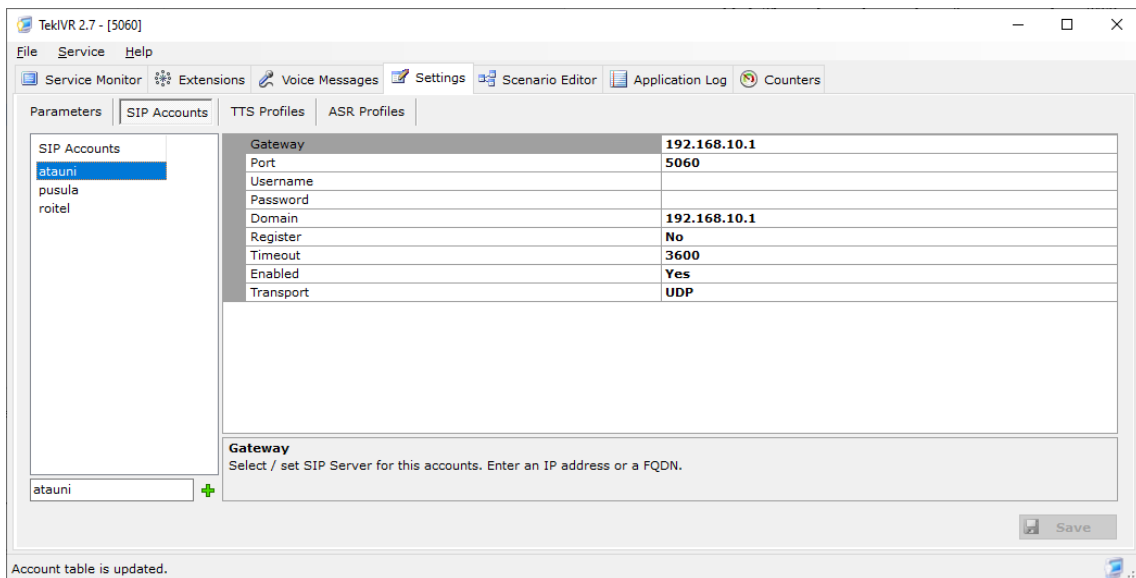


Figure - 2. TekIVR SIP Accounts

- **Gateway:** Enter an IP address or FQDN of your SIP gateway or proxy. Please make sure that this address is resolvable by your SIP client and has a valid entry (*an A record*) in your DNS server if you enter FQDN. You must configure a valid SIP proxy for proper operation.
- **Port:** Enter signaling port for the SIP server. The default value is 5060.
- **Username:** You must specify a SIP endpoint ID for TekConSer. If incoming calls are not destined to the number specified in Account ID parameter, TekConSer will reject the calls. This ID is also used by TekConSer while registering to SIP registrar.
- **Password:** If SIP registrar requires password authentication specify password.
- **Domain:** Enter a FQDN or an IP address of your SIP domain. You must configure a valid SIP domain for proper operation.
- **Register:** SIP endpoint registration is disabled by default. If this option is enabled TekConSer tries to register itself to configure SIP Domain.

- **Timeout:** Set registration timeout for this SIP account. TekConSer will re-register after timeout.
- **Enabled:** Select “Yes” to enable this SIP account.
- **Transport:** Select transport for this SIP server (*TCP or UDP*).

Registered SIP account will be displayed in **green** color. You can delete a SIP account by pressing the delete key on keyboard after selecting.

If you click [Save] button setting will be saved and activated immediately.

Settings / TTS Profiles Tab

SIPob allows you to create multiple TTS profiles to be used in text to speech operations. You can specify a locale, speech engine and voice type for each TTS profile;

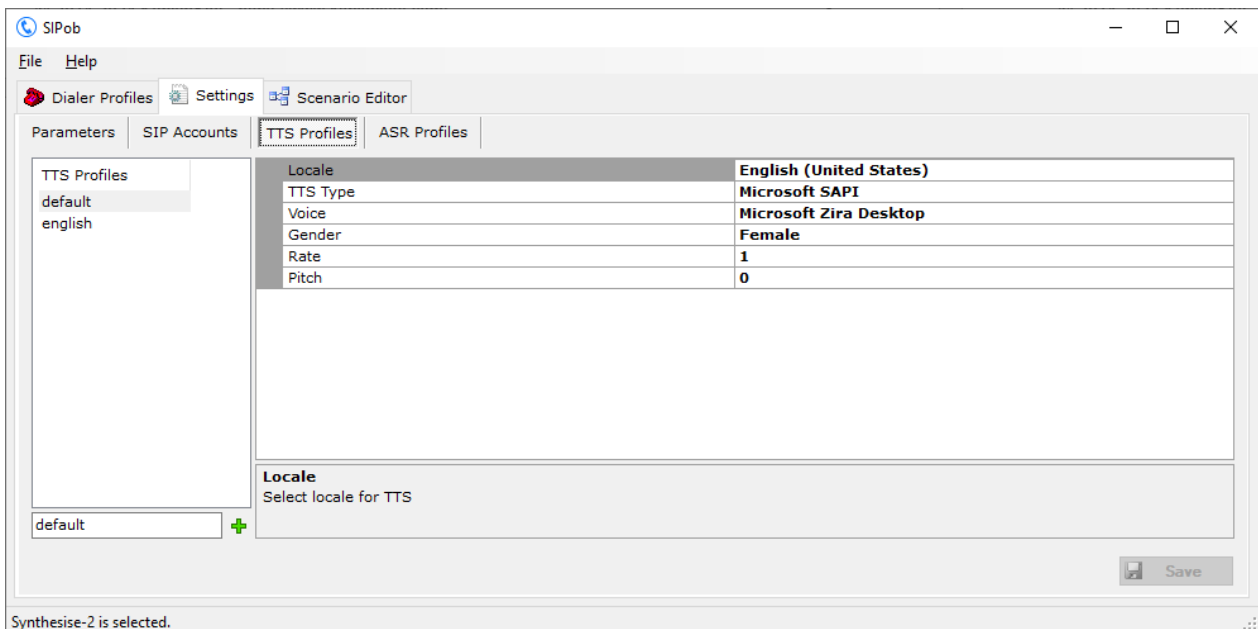
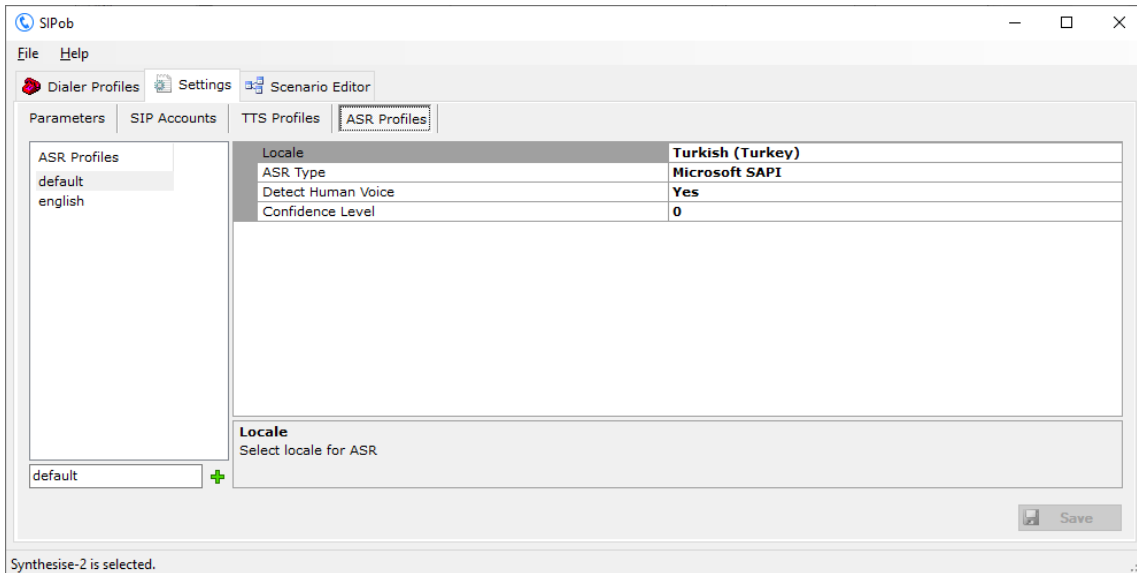


Figure - 4. TekIVR TTS Profiles

You must configure a valid Azure, Google or ElevenLabs API key prior to use Azure, Google Cloud or ElevenLabs SAPI as a TTS option. You cannot list Voice options for Azure and Google Cloud SAPI without having a valid API key.

Settings / ASR Profiles Tab

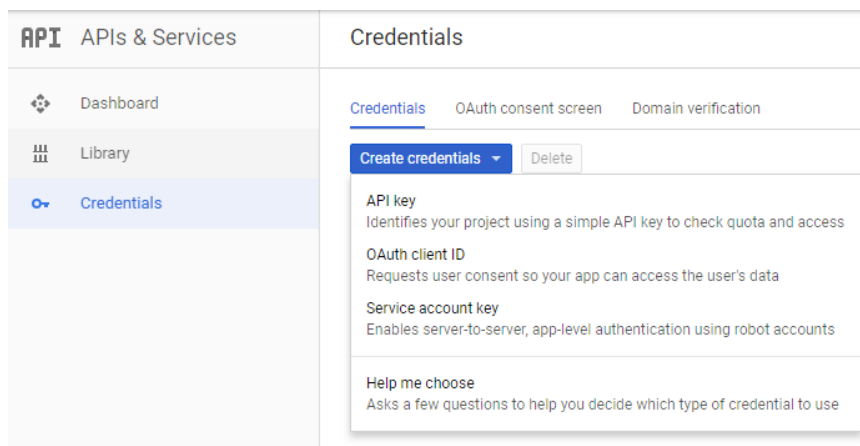
SIPob allows you to create multiple ASR profiles to be used in speech recognition operations. You can specify a locale and speech engine for each ASR profile;



Google API Key

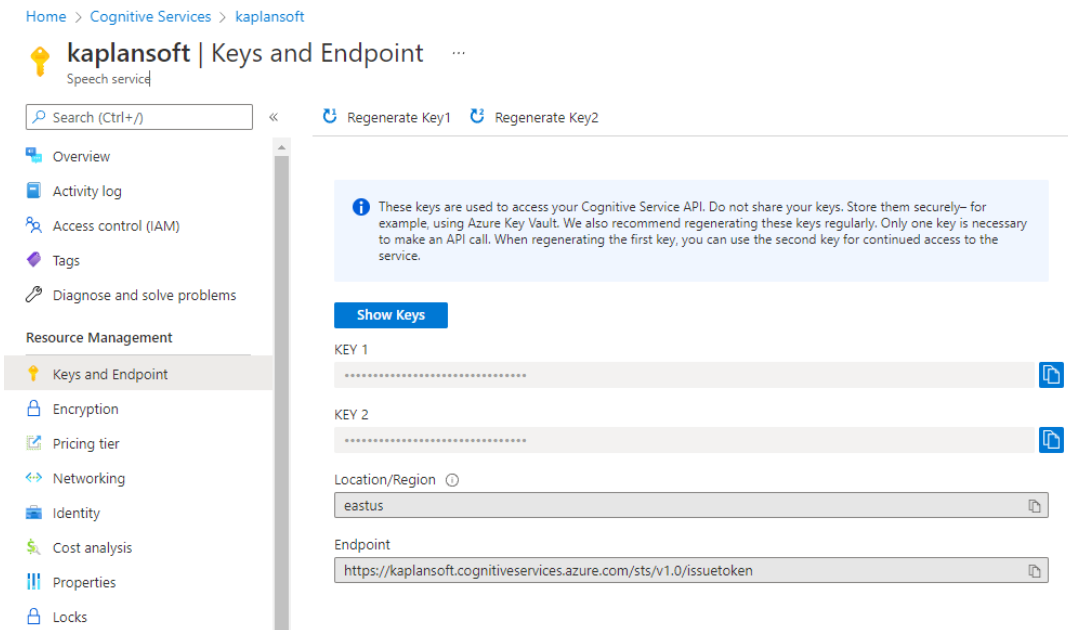
You need to have a Google API key created for Cloud Speech-to-Text API and/or Text-to-Speech API. To create API key, go to Google Cloud Platform and create a project first.

1. Navigate to the APIs & Services -> Credentials panel in Cloud Platform Console.
2. Select Create credentials, then select API key from the dropdown menu.
3. Click the Create button. A dialog box appears which displays your newly created key.



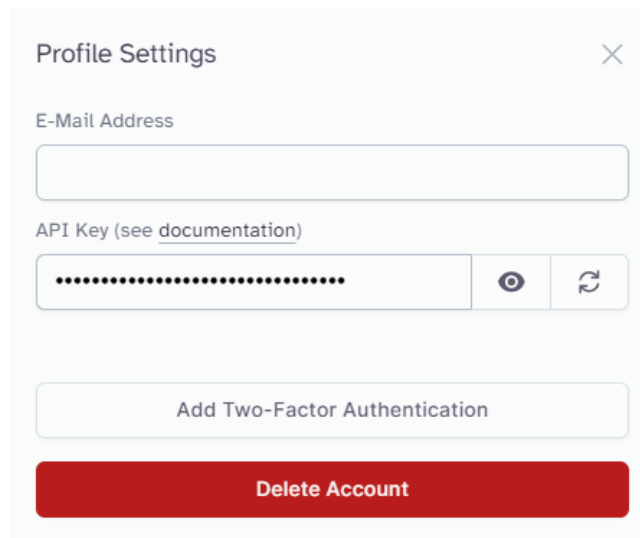
Azure API Key

You need to have an Azure API key created for Cloud Speech-to-Text API and/or Text-to-Speech API. To create an API key, go to Azure Portal, create a service under Cognitive Services. You can find your keys under Keys and Endpoint. You can use the first key. You can set your region while creating the service entry.



ElevenLabs API Key

You need to have an ElevenLabs key created for Text-to-Speech API. Your API key is accessible through your account profile settings.



Routing

You can define static routes to SIP endpoints through the “SIP Destinations” tab. Enter a phone number prefix to bottom leftmost textbox, enter the endpoint IP address to the textbox to the right of the prefix entry. You can also enter the SIP port (*Default 5060*) used by the SIP Endpoint and the Endpoint type (*Default SIP UA*).

You can also have a default route entry as shown in the figure below. SIPob chooses the longest match prefix route. If any match cannot be found, the default route is chosen if it exists. If the next hop configured for a phone prefix requires authentication, you can specify a username and

password for the routing entry. If authentication is not required, you can leave the username and password fields blank.

Enter a prefix and click the “Add Route” button to add a new routing entry. You must edit at least the Gateway entry to be able to commit the changes. You can specify a separate domain name if the domain name is different to the Gateway IP address or the FQDN. If the configured route requires TCP transport, you can set it by the **Transport** parameter. If you set Remove Prefix = Yes, SIPob will remove the defined prefix from the dialed number. You can specify trunk capacity for the defined SIP routing entry. SIPob will make queue calls to this destination when the capacity is full.

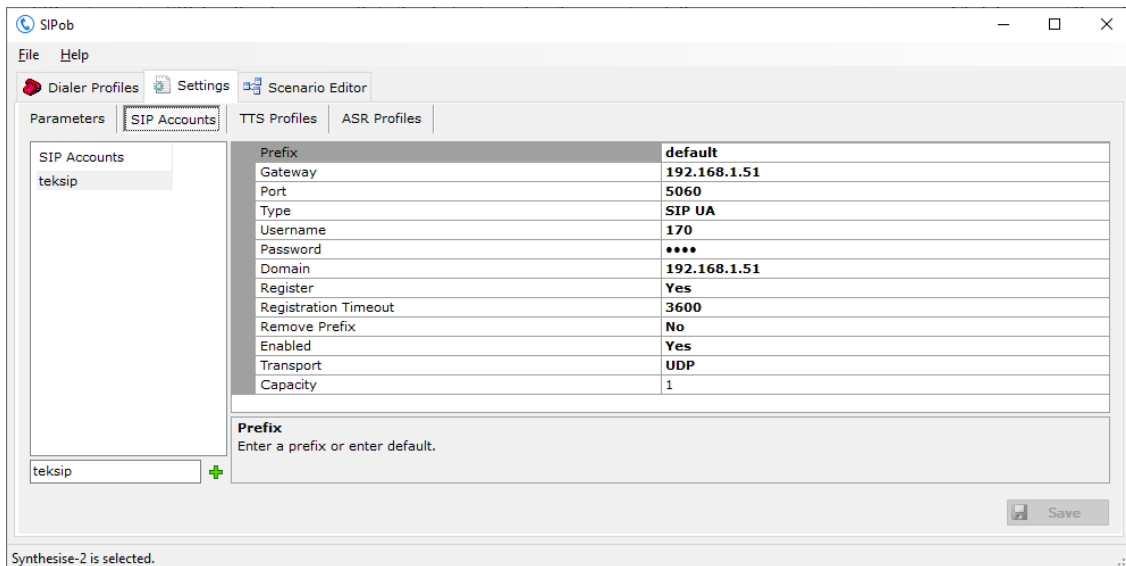


Figure - 2. SIP Destinations Tab

Scenario Editor

You can create your own scenarios using built-in Scenario Editor. Before creating your scenario, you must define audio prompts to be used in your scenario. Prompt and Action definitions for the scenarios are stored in SIPob.db.

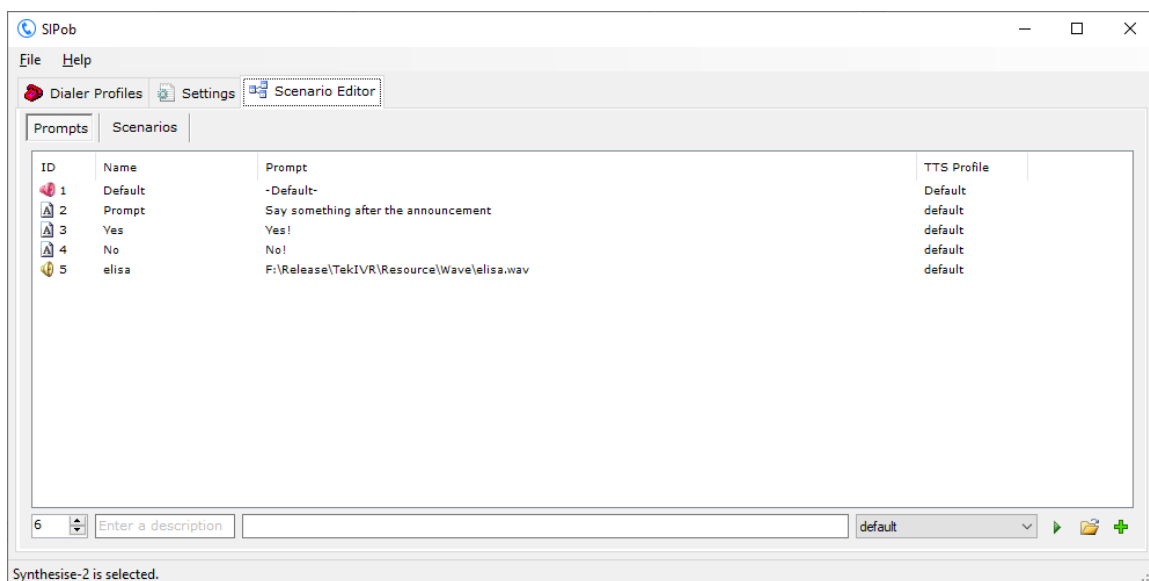


Figure - 3. Scenario Editor / Prompts Tab

Scenario Editor / Prompts Tab

Click Scenario / Prompts Tab to add audio prompts. You can either add wave files or free text to be played out using TTS engine. Each prompt must have a **unique** Prompt ID. You must also have a descriptive name for the prompt. “-Default-” specifies built-in welcome announcement. You can delete a defined prompt pressing delete key on the keyboard. Scenario Editor does not allow deleting a prompt which is used in a “Play” action.

You can enter URLs (*http or https*) which retrieve wave resources from a web server as prompts. (<http://example.com/wavefile.wav> e.g.)

Scenario Editor / Scenarios Tab

Click Scenario / Scenarios Tab to define scenarios and steps in your scenarios. Each action must have a **unique** Action ID in a scenario. You can delete a defined scenario or action pressing delete key on the keyboard after selecting the item to be deleted. You can specify called numbers and time frame when the scenario is invoked by SIPob. You should also specify the maximum number of steps can “d in scenario to prevent infinite loops. Scenario Editor does not allow deleting an action which is used in an action as a *NextAction* and you cannot assign Action’s *NextAction* to itself. SIPob allows some internal and custom variables to be used in action parameters. Variable names are case sensitive.

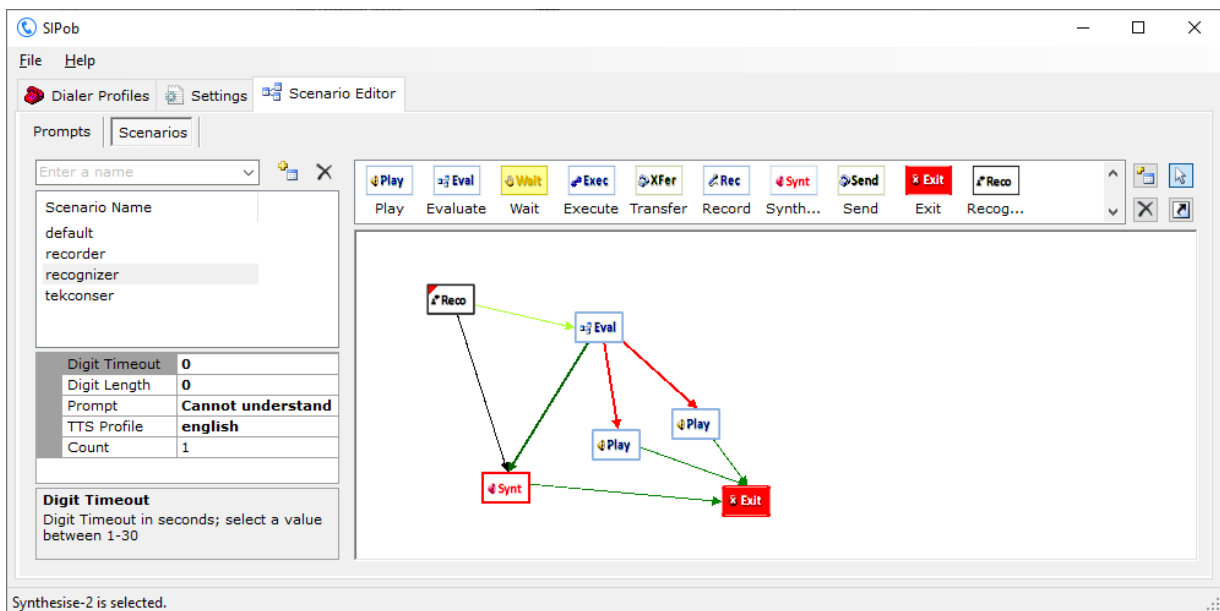


Figure - 4. Scenario Editor / Scenarios Tab

- %recognized% The output of the last Recognize action used in the scenario.
- %pasttext% The output of the previous Recognize action used in the scenario.
- %execout% The output of the last Execute action used in the scenario.
- %pastresponse% The output of the previous Execute action used in the scenario.
- %received_digits% The output of the previous Execute action used in the scenario.
- %called_number% The number which is found in To: header.
- %calling_number% The number which is found in From: header.
- %sipdomain% SIP domain for the calls.

- %date% Current date and time in current locale format.
- %saved% Full path of the latest recorded audio file by the Record action.
- %callid% SIP Call-Id of the call.
- %sessionid% Locally generated unique session id for the call.
- %custom_variable_0% User defined custom variable.
- %custom_variable_1% User defined custom variable.
- %custom_variable_2% User defined custom variable.

SIPob supports the following actions in IVR scenarios

Play. Play action will play out defined in action (*Prompt*) and collects user input in %received_digits% variable. **Variables are case sensitive.** You can set how many times welcome message will be played if no user input (*DTMF digits*) detected (*Count*), how many digits will be collected maximum (*DigitLength*) and how many seconds will be waited after last digit entered if user does not dial # (*DigitTimeout*). You must also define the next step after the action is completed (*NextAction*).

Synthesize. This action is like Play action and has the same parameters. However, Synt synthesises audio prompt dynamically using Microsoft TTS engine by processing text string specified in Prompt parameter. You can use internal variables in prompt text. You can also specify a URL (*http/https*) to fetch text from a web server. If the fetched resource is a wave file, it will be played out directly otherwise if it is a text file it will be used as input for the TTS engine. You cannot use pipe | character in prompt text.

Recognize. SIPob can recognize user input via Google/Azure Cloud Speech API and Microsoft Speech Recognition engine. You can create an ASR profile for each Recognize action. You need to specify a prompt and a mapping file as parameters for this action. Mappings file is a CSV file containing audio input and corresponding values. Recognized input will be stored in %recognized% and if recognized text has a corresponding mappings entry, mapped value will be stored in %received_digits% variable. Variables are case sensitive. Sample entries for Mappings file;

```
"John Doe", "101"
"Jane Doe", "102"
"Yasin KAPLAN", "102"
```

SIPob will execute and process console output if you specify a mapping file with .exe, .bat or .vbs extension. Please note that console output must be identical to a mapping file content as shown above. You can also specify parameters for the executable. Parameters internal variables.

You can also specify an ODBC data source to fetch mapping entries. An example is shown below;

Silence after Speech Duration	5
No Speech Threshold	5
Mappings File	ODBC:Deneme
Parameters	Select Label, Extension from MyTable where Number = '%called_number%'
Prompt	0 Silence
Waiting Prompt	0 Silence

Data tables must have at least two fields; one for item to be recognized and the other for corresponding value (*Number which will be used for call transfer e.g.*). Your query must return these fields in this specified order.

You can have three destination actions for Recognize action; one for successful recognition, second is for failed recognition and third for no user input (*Total silence*). Success action is set in the first link. Create a second link from Recognize action for failure action and a third link for no input action.

SIPob can detect silence. SIPob will start recognition after silence duration after speech. The default silence duration is 2 seconds. You can increase or decrease for better results. You can also set recognition timeout. SIPob will choose failure action if recognition timeout expires.

Record. Record action will record audio from the calling part after playing out a prompt which is configurable. You can restrict the maximum duration of the audio recording by the Duration parameter. Set it to 0 for unlimited duration. Caller can terminate recording by dialing pound key (#). You must also define the next step after the action is completed (*NextAction*).

SIPob will record audio from the caller into SIPob Application Directory\Messages folder if the directory path is set to blank. Audio files named as <Called Number>-<Calling Number>-YearMonthDayHHmmss-ms.wav where the date is file creation date (*2123561212-9990000-20201215130015.wav e.g.*).

Wait. You can force the caller to wait for a caller while playing out a configurable prompt. You can specify the wait time in seconds. Set Detect Human Voice = 'Yes if you would like to detect human voice. Next action will be performed upon human voice detection. You must also define next step after the action is completed (*NextAction*).

Evaluate. Evaluate action evaluates user input and then determines the next action. If SIPob cannot find an action that satisfies conditions in *Options* parameter next action will be the action defined in *DefaultAction* parameter. Conditions must be entered in <Go to Action>;<Case 1>|<Go to Action>;...;<Case n>|<Go to Action> format.

Execute. The Execute action executes the executable defined in *ExecutablePath* parameter. You must enter executable with full path. You can use internal variables in command line parameters. Leave *ExecutableParameters* blank if no parameter will be used. Set WaitforCompletion > 0 in seconds if you wish SIPob to wait completion of the execution of the executable and SIPob will use return value (DOS Errorlevel) of the executable as the next action be executed. You must also define the next step after the action is completed (*NextAction*). If you set WaitforCompletion > 0 SIPob will store executable's console output to %execout% variable and ignores return value (*DOS Errorlevel*) of executable.

You can also specify an HTTP/HTTPS URL in *ExecutablePath* parameter. TekIVR will perform an HTTP GET action if there is not any parameter otherwise TekIVR will perform an HTTP POST action.

Executable Path	http://www.kaplansof.com/MyApp?Number=%called-number%
Executable Parameters	
Wait for completion	60

The Execute action allows you to perform a database query through an ODBC source. You must create an ODBC connection profile (64 Bits) as in system DSNs. Here is an example;

Executable Path	ODBC:MyDataSource
Executable Parameters	Select Label from MyTable where Number = '%called_number%'
Wait for completion	60

You can set `%received_digits%`, `%display_name%`, `%sessionid%`, `%execout%`, `%custom_variable_0%`, `%custom_variable_1%` and `%custom_variable_2%` variable values from the output of the executable. For instance, you can set `%received_digits%` by configuring your executable to return

```
received_digits=desiredvalue
```

TekIVR will parse returned output and set `%received_digits%` to `desiredvalue`. You can set multiple variables

```
received_digits=desiredvalue;custom_variable_0=desiredvalue
```

You can set internal variables `%received_digits%`, `%display_name%`, `%sessionid%`, `%execout%`, `%custom_variable_0%`, `%custom_variable_1%` and `%custom_variable_2%` using Execute action. Enter `set <variable name>` to Executable Path and enter desired value to Executable Parameters:

Executable Path	set execout
Executable Parameters	my value
Wait for completion	10
Encoding	ASCII

Transfer. Transfer action will transfer the call to the extension defined in *Number* parameter. You can have a predefined extension or internal variables in *Number* parameter. Variables are case sensitive. You can specify a prompt to be played out while transferring the active call. You can also specify the next action for failed and successful call transfers.

Send. SIPob can send DTMF digits to the remote party. You can use internal variables and specify inter digit delay in milliseconds.

Dial. SIPob allows you to dial a specified SIP destination and execute a specified child scenario when the remote party answers. Collected digits and executable output are returned to the parent scenario. SIPob can connect the calling party of the parent scenario to the called party of the child scenario when you terminate child scenario with an Exit action with Jump to Scenario = Bridge for an audio conversation session.


Prompt	0 Silence
Timeout	10
Execute Scenario	auxiliary
Dial Number	102



You can also use internal variables for the number parameter.

Exit. Exit action terminates execution of a scenario. You can specify a prompt to be played out while terminating the active call. If any of the action has an undefined action in NextAction parameter SIPob will also terminate execution of the scenario.

Exit action enables you to jump to an existing scenario. You can choose a scenario to be jumped by setting “Jump to Scenario” option.

It is wise to start your scenario with a Play action.

You can add an action to your scenario by dragging it from Actions list on left after clicking  “Add action” button. You can assign an Action’s next action by dragging it to another action after

clicking  “Link action” button. You can change the next action in the same way. You can delete an action or a link after clicking  “Delete object” button and then selecting object.

You can set startup action by double clicking an action in the scenario map.

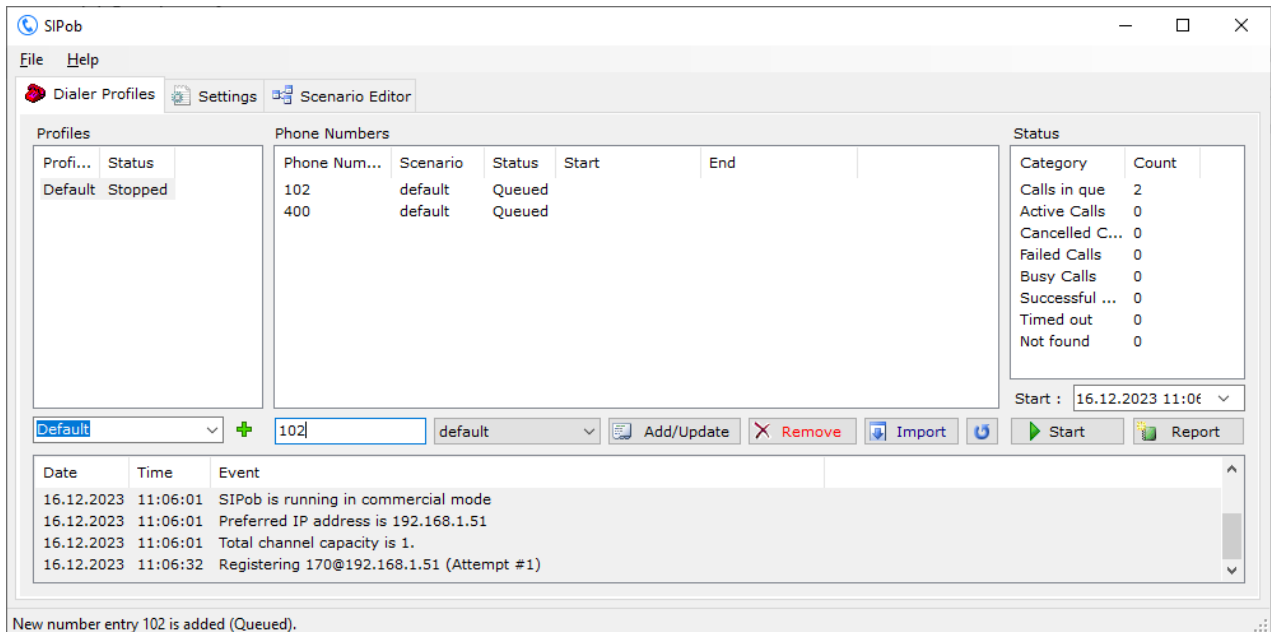


Figure - 5. Dialer Profiles Tab

Dialer Profiles

You can create dialer lists through the Dialer Profiles tab. You can have independent dialer profiles to be executed by user defined date and times. You can either enter phone numbers to a profile manually or you can import a text file which contains phone number entries.

You can start a dialer profile by clicking Play button. SIPob will queue calls if you specify a Start Date / Time greater than current Date / Time. You can monitor call status real time and hang up or force to re-queue a failed call.

If the SIPob dialer profiler cannot be started, please examine the Application Log tab as well as the SIPob log file under <Application Directory>\Logs, ensuring that you have enabled logging in “Settings/Service Parameters”. Log files are also accessible through the file menu.

Command Line Parameters

SIPob can be invoked though the command line with optional parameters.

```
C:\Program Files\SIPob\SIPob.exe <Dialer Profile> "<CSV file for the numbers to be dialed>" "<CSV file for the report>"
```

CSV file for the numbers to be dialed is a plain text file with phone numbers in each line. You can specify scenarios to be executed by concatenating scenario name with comma to the number;

```
5551010,OneScenario
5552020,OtherScenario
```

You must specify at least the Dialer Profile in the command line for automatic execution. SIPob will clear existing number entries in the dialer profile prior to importing the numbers in the CSV file containing numbers to be dialed.

You can also instruct SIPob to generate call reports in CSV format automatically after execution by specifying a file name for the report.

Example;

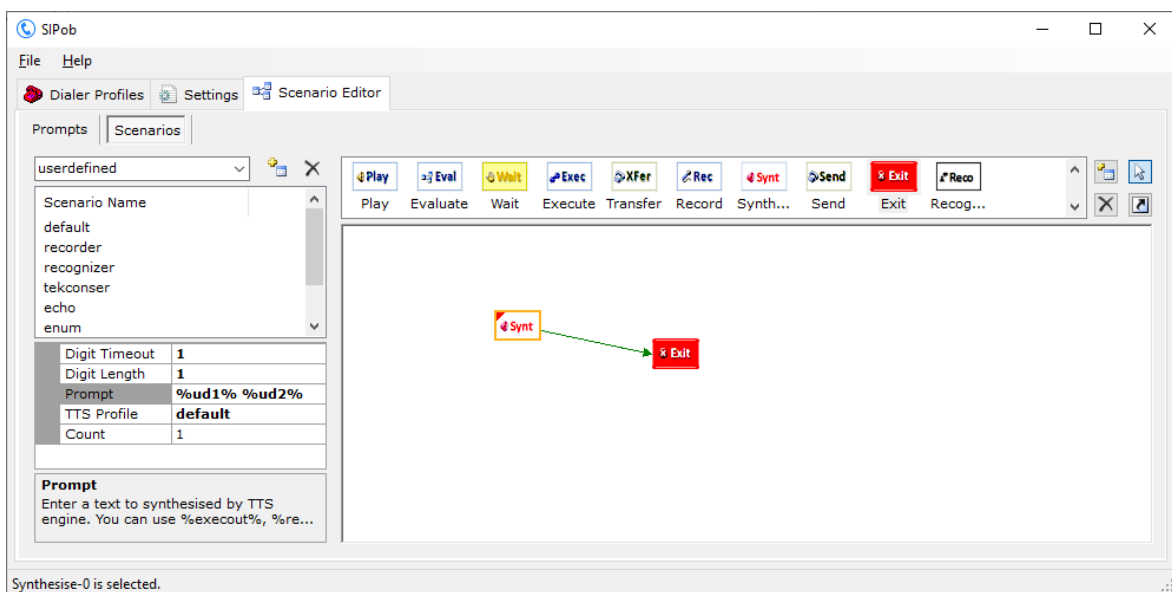
```
C:\Program Files\SIPob\SIPob.exe myDemo "C:\Demo\Numbers.csv"
"C:\Demo\Results.csv"
```

SIPob will automatically exit after the execution.

You can specify your own command line parameters and use them as variables in your scenarios. User defined parameter keys must start with ud prefix. Variables must be surrounded by quotes. Example:

```
C:\Program Files\SIPob\SIPob.exe -ud1 "My variable 1" -ud2 "My other variable"
```

You can use your command line variables in your scenarios by surrounding them with % sign:

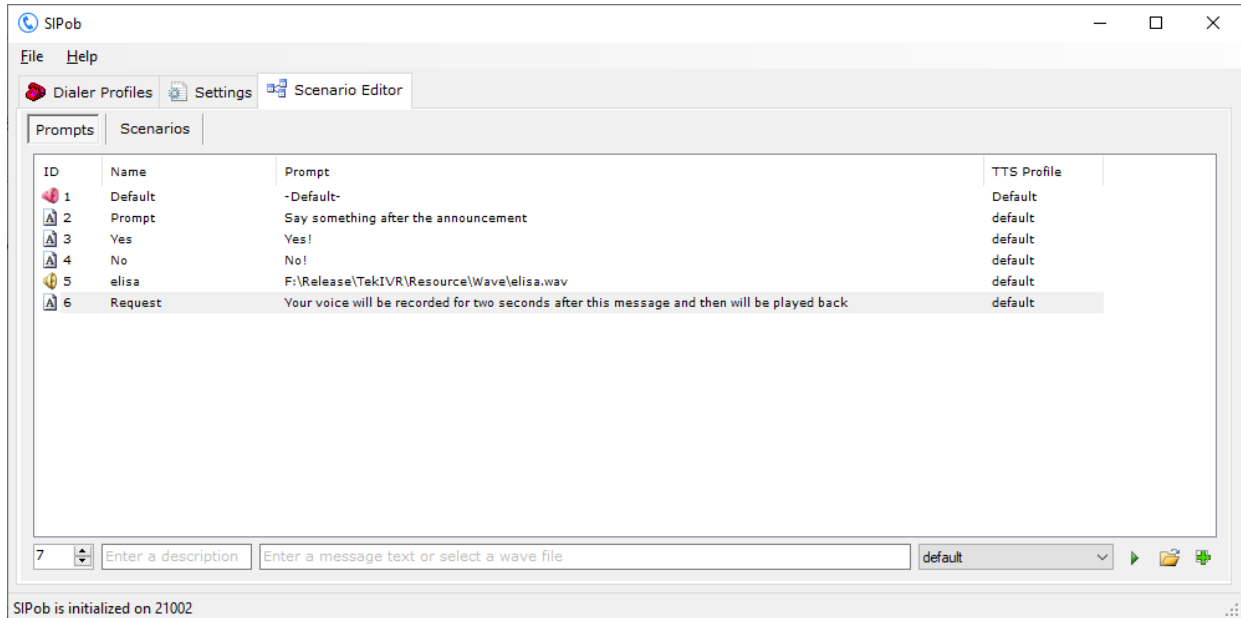


Troubleshooting

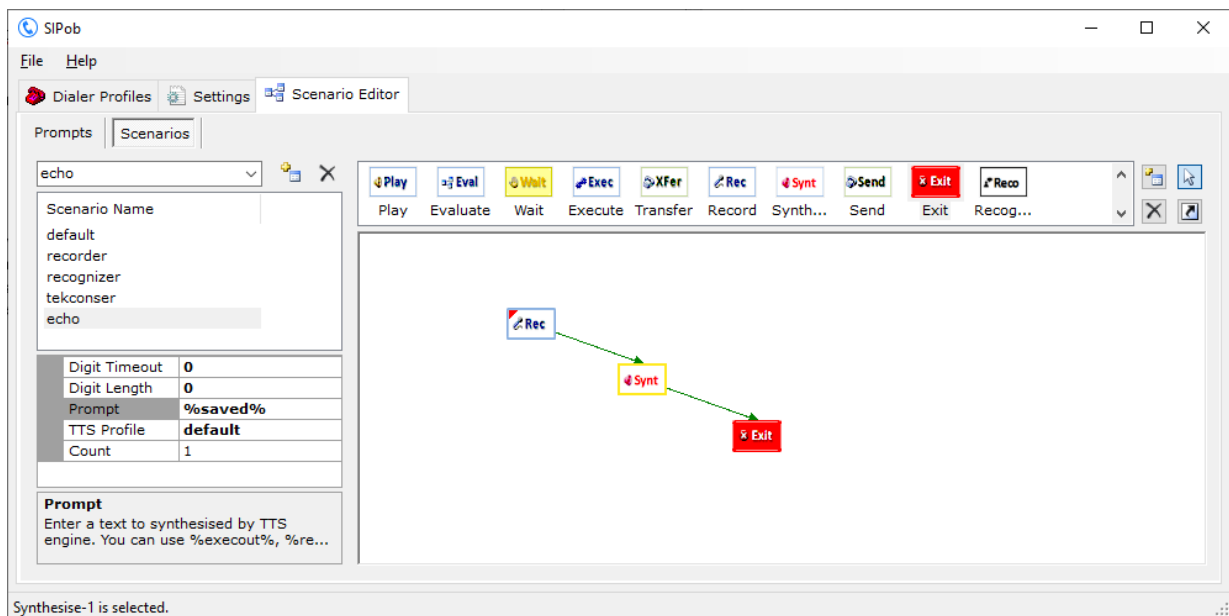
SIPob provides many messages when problems occur. You can see error messages on the SIPob Status bar or in the log file of the SIPob service. You can enable logging in the Settings Tab. There are three levels of logging: None, Errors, and Sessions. If you select “Errors”, SIPob logs just error messages. If you select “Sessions”, both Session and Error messages will be logged. You must save or apply settings changes if you change the logging level setting. Log files are located under the <Application Directory>\Logs directory.

Echo Test Scenario

You can create a special scenario to test if there is a problem with two-way RTP media flow. Create an informative prompt in Prompts tab first:



Add the following actions to the scenario and link them. Set the prompt above as the prompt for action Rec and set %saved% variable as prompt in action Synt. Recommended Record Duration for action Rec is 2 seconds.



SIPob Messages

Unable to initialize UDP/TCP thread [x.x.x.x:5060]

If another application is configured to use the same UDP/TCP port (5060) as SIPob, SIPob cannot initialize the respective thread.

Default route points to this host

You cannot specify a gateway points to SIPob.

New setting(s) applied and activated. Check default route.

There is a problem with the IP address or FQDN of the default route.

Cannot apply changes; enter minimum configuration

There is missing configuration data.

You cannot redirect an endpoint to itself.

You cannot re-direct an endpoint to itself.

Invalid endpoint information or illegal character detected in entries.

Invalid characters found in a SIP username or entry. You can only use numeric characters in SIP username entries. You cannot use a “;” (*Semicolon*) character in password entries.

SIPob Commercial Mode

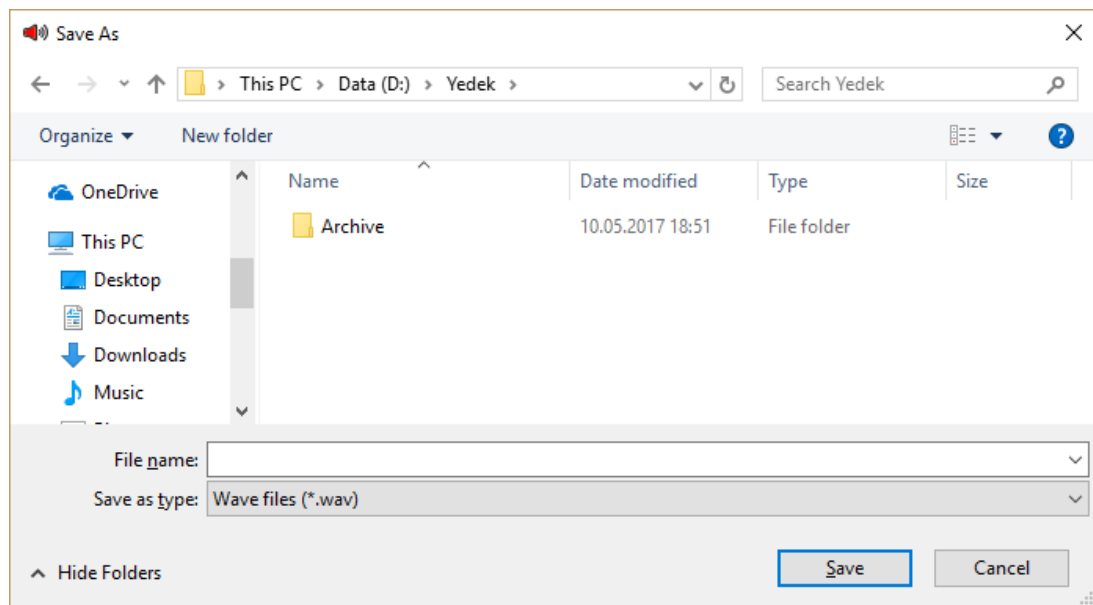
The freeware mode of SIPob does not allow play out for more than 3 seconds.

How to Record a Custom Audio Message

You can use Windows Sound Recorder to record a custom audio prompt. You can use TekRecorder to record audio files compatible with SIPob.



Click the record button to start recording. Click the record button again after finishing. Select "File/Save As" option from File menu.



You can download TekRecorder from KaplanSoft website download section.