

# ***SipCLI SEditor***

**User Guide  
Version 2.9**

## Document Revision 2.6

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

SEditor is built by Yasin KAPLAN

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

*Copyright © 2010-2024 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](mailto:info@kaplansoft.com).*

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

*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.*

## **Table of Contents**

Table of Contents .....	3
Introduction .....	4
System Requirements .....	4
Installation .....	4
Scenario Editor .....	4
Prompts Tab .....	4
Scenario Designer Tab .....	5
Settings Tab .....	7
Echo Test Scenario .....	9
How to Record a Custom Welcome Message .....	10

## **Introduction**

SipCLI is a command line SIP user agent runs under Windows (*XP/Vista/7/8/10 & 2003-2016 Server*) which enables making SIP (*Based on RFC 3261*) based calls. SipCLI supports automated tasks defined in scenario scripts. SEditor is a GUI for creating scenario scripts. Visit <https://www.kaplansoft.com/sipcli> regularly for updates.

## **System Requirements**

SEditor requires Microsoft .NET Framework 4.0 installed with the latest patches. Pentium i3 class CPU with 3 GB of RAM is ideal for most configurations.

## **Installation**

Unzip “sipcli.zip” and run Seditor.exe which comes with the distribution. No installation required.

## **Scenario Editor**

You can create your own scenario using SipCLI Scenario Editor. Before creating your scenario, you must define audio prompts to be used in your scenario. Prompt and Action definitions for the scenario are stored in text files.

## **Prompts Tab**

Click Scenario / Prompts Tab to add audio prompts. You can either add wave files in 16 bit per sample, 1 channel and 8000 Hz sampling frequency format or free text to be played out using TTS engine. Each prompt must have a **unique** Prompt ID. 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.)

You can open an existing scenario file from File Menu / Open menu option.

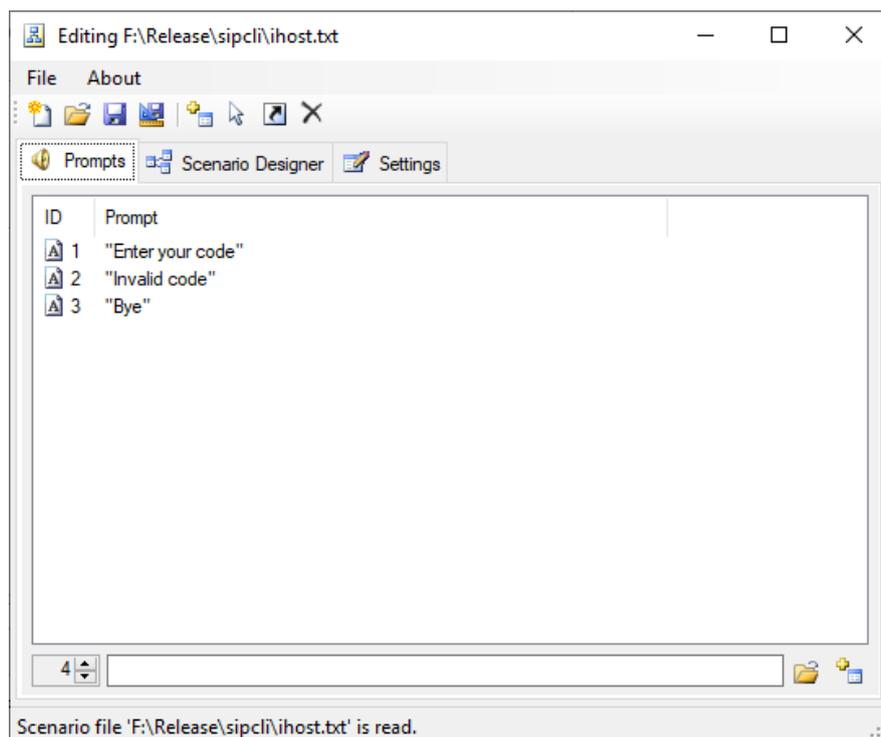


Figure - 1. Prompts Tab

## Scenario Designer Tab

Click Scenario / Scenario Designer Tab to define steps in your scenario. Each action will have a **unique** Action ID. Scenario Designer does not allow deleting an action which is used in an action as a *NextAction* and you cannot assign Action's *NextAction* to itself. SipCLI supports following actions in scenarios:

**Play.** Play action will play out defined in action (*Prompt*) and collects user input in `%Received_Digits%` variable. 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 second 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*).

**Synthesise.** This action is like Play action and has the same parameters. However, Synt synthesises audio prompt dynamically using Microsoft TTS engine by processing test string specified in Prompt parameter. You can use internal variables in prompt text like `%called_number%`, `%calling_number%`, `%received_digits%`, `%date%`, `%execout%` and `%sipdomain%`. 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 text file it will be used as input for the TTS engine. You cannot use pipe | character in prompt text.

**Record.** Record action will record audio from 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*).

SipCLI 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.).

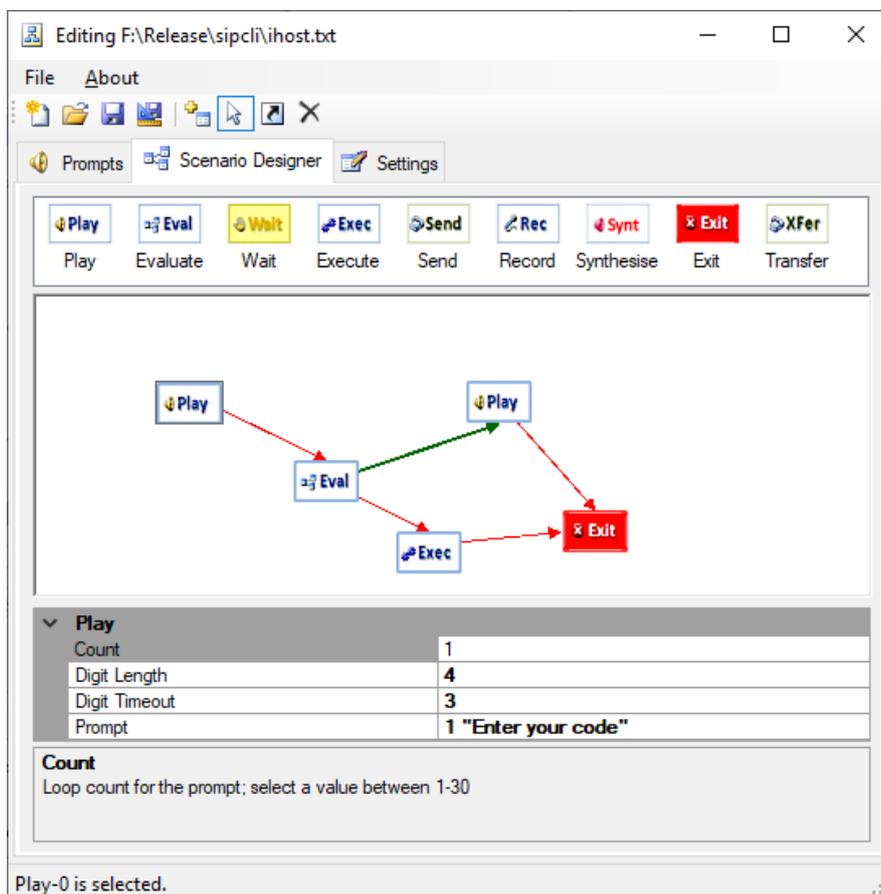


Figure - 2. Scenario Editor Tab

**Evaluate.** Evaluate action evaluates user input and then determines the next action. If SipCLI cannot find an action 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.** Execute action executes the executable defined in *ExecutablePath* parameter. You must enter executable with full path. Valid predefined options for the command line parameters (*ExecutableParameters*) are %called\_number%, %calling\_number%, %received\_digits%, %saved\_audio\_file% and %sipdomain%. Predefined parameters are case sensitive. Leave *ExecutableParameters* blank if no parameter will be used. Set *WaitforCompletion* > 0 in seconds if you wish SipCLI to wait completion of the execution of the executable. You must also define the next step after the action is completed (*NextAction*). If you set *WaitforCompletion* > 0 SipCLI 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	<code>http://www.kaplansof.com/MyApp?Number=%called-number%</code>
Executable Parameters	
Wait for completion	<b>60</b>

Execute action allows you 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	<b>ODBC:MyDataSource</b>
Executable Parameters	<b>Select Label from MyTable where Number = '%called_number%'</b>
Wait for completion	<b>60</b>

**Transfer.** Transfer action will transfer the call to the extension defined in *Number* parameter. You can have a predefined extension or `%called_number%`, `%calling_number%`, `%received_digits%` (*User input*), `%date%`, `%execout%` and `%sipdomain%` 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.** Send action will send digits to remote party defined in *SendDigits* parameter. You can specify inter digit delay in milliseconds.

**Wait.** You can 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 the next step after the action is completed (*NextAction*).

**Exit.** Exit action terminates execution of scenario. You can specify a prompt to be played out while exiting. You can also specify a value which is returned as DOS *errorlevel*. This can be an integer constant value or `%received_digits%` variable.

If any of the actions in a scenario has an undefined action in *NextAction* parameter SipCLI will also terminate execution of scenario.

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.

## Settings Tab

Settings tab organized in three sub tabs, Application, Media and SIP Proxy. Scenario Editor reads SipCLI.ini under SipCLI application directory and allows you edit configuration parameters. Changes are auto saved when you exit Scenario Editor.

SIP account password saved in encrypted form in SipCLI.ini when you edit SipCLI.ini using Scenario Editor.

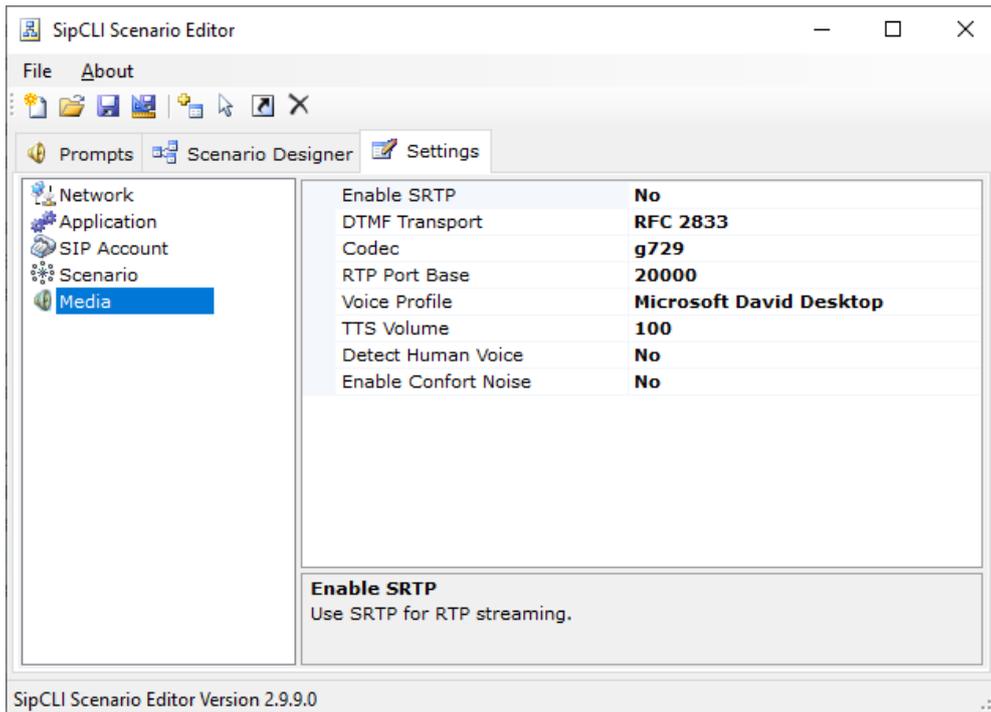
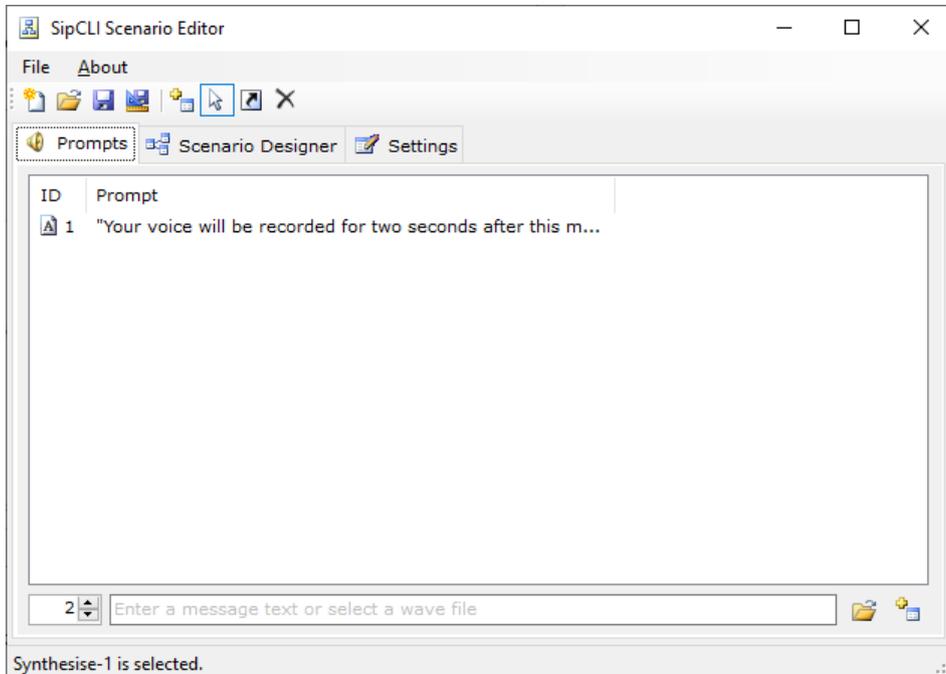


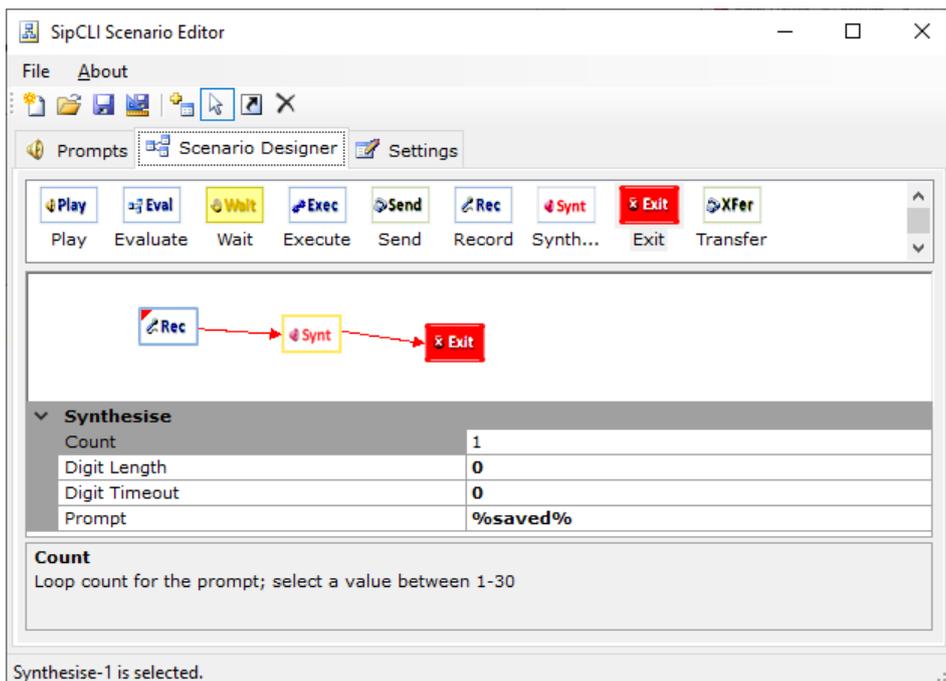
Figure - 3. Settings Tab

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



## How to Record a Custom Welcome Message

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

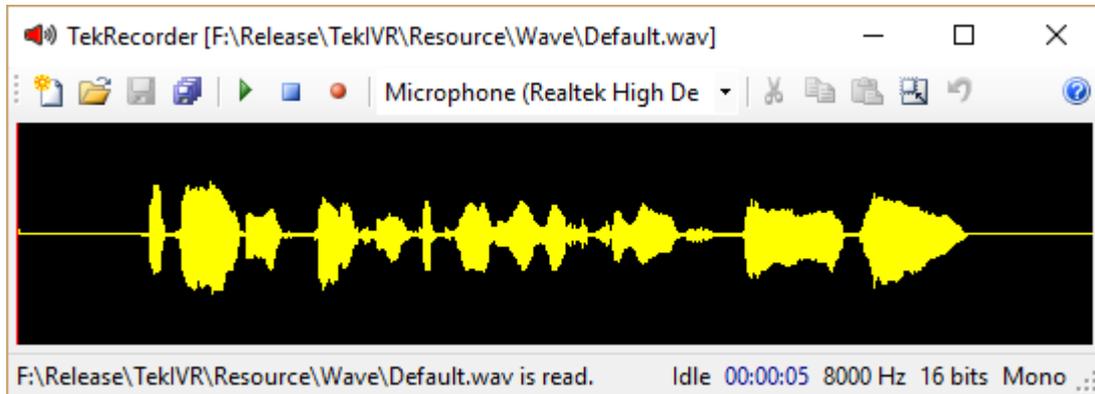


Figure - 3. TekRecorder

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

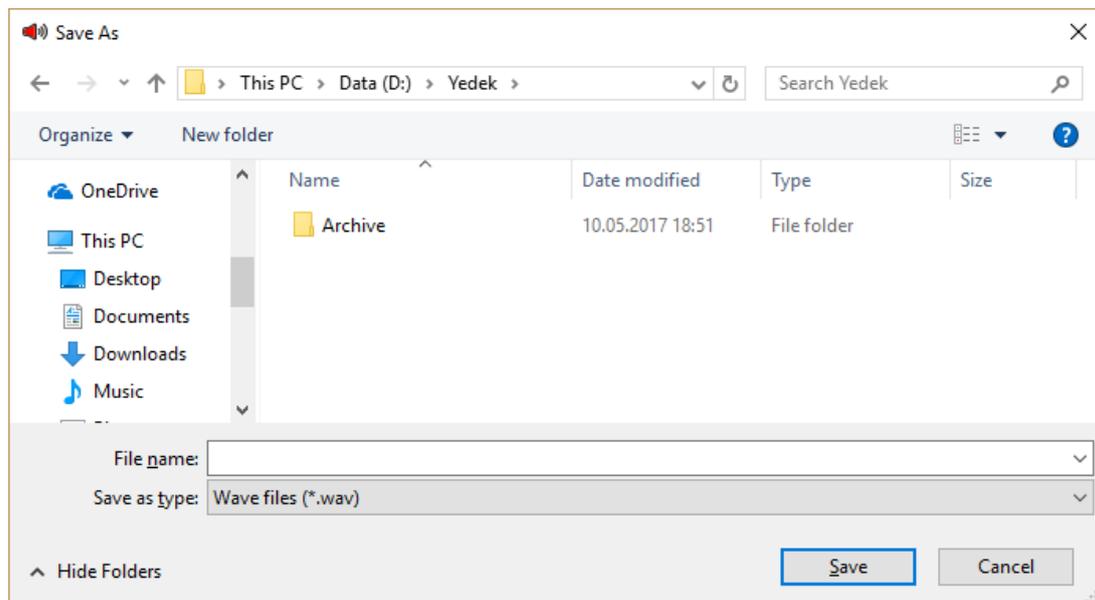


Figure - 3. Save a wave file

Audio file will be saved in “8000 Hz; 16 Bit; Mono” format. You can use TekRecorder to convert existing wave files to 16 bit, 8 KHz mono format. You can download TekRecorder from KaplanSoft website download section.