

3.42.10 AIC. - Artificial Intelligence Command

Accessing Artificial Intelligence Services

The commands provided here offer you a gateway to the cutting-edge Artificial Intelligence services provided by industry leaders such as **OpenAI** and **GPT4All** and **Stability AI**.

Also you can use AI-Systems like Stable Diffusion local on your Computer.

In addition, if you're interested in top-tier speech synthesis, **Elevenlabs'** services are also readily accessible through these commands.



If it looks like magic, its possibly AI. The AIC.-Command is the tool of choice to combine the Ingredients.

[Please read details about how to prepare for using these services.](#)⁴⁷⁰

The AIC.-Command is a collection of Subcommands that will connect you SPR-Script with the world of Artificial Intelligence.

You can start with manual testing your Prompt and API-Key

You can test your idea and verify your API-Key here, online using:

<https://bettergpt.chat/>

There you have all options that the SPR also offers in the Script.

If you prefer a local installation of "Better GPT" (which i can recommend) then download and install:

Local Install BetterGPT

win-better-chatgpt-1.0.4-x64.exe	83.2 MB	Jun 11
win-better-chatgpt-1.0.4-x64.exe.blockmap	90.1 KB	Jun 11
Source code (zip)		Jun 11
Source code (tar.gz)		Jun 11

This way you have the client local, and you can export your prompts into a ".csv" and use them later within your Script.

You can combine all sorts of AI using the SPR.

The following Sample-Code will take your Theme, ask Open AI to improve it. And then generate the Picture with **SDL**.

You need to have your [OpenAI API-Key](#)^[470] saved locally in the Scriptfolder using the [Save Key](#)^[762] Command.

```
VAR.$$THE="neural network beautiful golden girl Galaxy robot digital photorealistic"
VAR.$$PAO=Please make me a prompt below 231 characters, for stable diffusion use:
' Use Image Register 0
VIN.$$IMR=0

' Set OpenAI API-Key from the saved File
AIC.SetKey|File
' Set Model-Temperature
AIC.Set_Temperature|0

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000 Tokens)
' The more Tokens you use the more you need to pay. But the longer Input and Output
AIC.SetMax_Token|300

FOR.$$STP|1|25
  AIC.Ask_Chat|$$PAO|$$RET
  DBP. Got: <<$$RET >>
  $$FIR=?path\Sample Pics\Testresult_?.png
  FIL.gen|$$FIR|1|0|$$FIL
  SDL.gtf|$$RET|$$FIL
  ANA.Load|$$IMR|$$FIL
  ' This will print a Text inside the Picture
  $$TXT=SPR/SDO. generated: $$STP Steps
  $$COA=&HFFFFFF
  $$COB==&H0
  $$BGC=-2
  $$XPA=30
  $$YPA=450
  ANA.PrintAt|$$IMR|$$TXT|$$COA|$$BGC|$$XPA|$$YPA|24
  ANA.PrintAt|$$IMR|$$TXT|$$COB|$$BGC|($$XPA+1)|($$YPA+2)|24
  ' Now we load the picture
  DBP.Loaded in IR:$$IMR -> $$FIL
  ANA.Save|$$IMR|$$FIL
  ANA.Show|$$IMR!
```


NEX.
ENR.



One of the SDL-generated Pictures.

3.42.10.1 ! Open AI - DALL.E

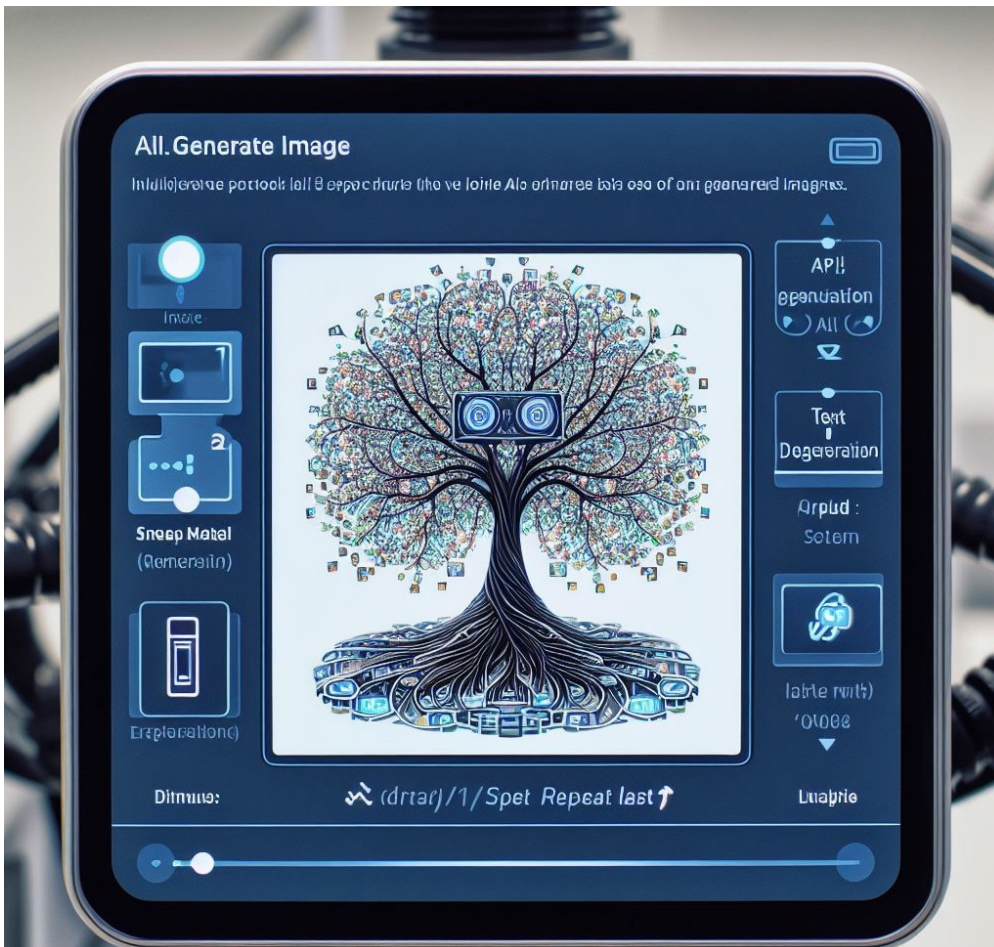
3.42.10.1.1 Generate Image

[AIC.Generate Image](#) [Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Generate Image

Use the Open AI-API Server to generate Images from a Text-Prompt



AI generated Pictures look often photorealistic, yet the Text generation is not always perfect.

Pic_0001.png	26.06.2023 19:00	PNG-Datei	193 KB
Pic_0002.png	26.06.2023 19:00	PNG-Datei	193 KB
Pic_0003.png	26.06.2023 19:00	PNG-Datei	193 KB

Intention

The `AIC.Generate Image` command in the Smart-Package Robot (SPR) allows users to generate images through the OpenAI Image Generation Endpoint. The command requires three parameters to generate images based on a text prompt.

Unlike other AI Systems, the OPEN AI - Text to Image System can work with very long an detailed prompts.

Syntax:

```
AIC.Generate Image|<Prompt>|<Filename>|<ImageSize>
```

Parameters:

1. `<Prompt>`: A text string that describes the image you want to generate. For example, "A sunset over the mountains."
2. `<Filename and path>`: The name for the file where the generated image will be saved. This can contain '****' which will be replaced by a unique image number. For example, "image_****.png" might become "image_0001.png".
3. `<ImageSize>`: The resolution of the image to be generated. It must be in the format `WIDTHxHEIGHT` and can range from `256x256` to `1024x1024`. For example, "512x512".

Example Usage:

```
AIC.Generate Image|A beautiful landscape with a flowing river and trees|landscape
```

Costs:

The costs for using the OpenAI Image Generation Endpoint through the SPR command are based on the resolution of the output image. As of the time of writing, the costs are as follows:

- 256x256 resolution: \$0.016 per image
- 512x512 resolution: \$0.018 per image
- 1024x1024 resolution: \$0.020 per image

Please note that if you are a new user of OpenAI's API, you can benefit from a free trial that allows you to use \$18 of free credits within your first three months. Keep in mind that pricing may change, so it's advisable to consult the official OpenAI documentation for the most current information.

Ethical Considerations:

OpenAI has policies in place to ensure ethical use of its services. Prompts that are assumed to be illegal, harmful, or offensive may be rejected. It is essential to ensure that the content you generate complies with OpenAI's policies and ethical guidelines or else your Prompts will be rejected.

Note:

The OpenAI Images API is in public beta, which means that it might evolve and change significantly. It is also subject to rate limits of ten images per minute and twenty-five images per five minutes. It's advisable to keep this in mind if you are planning to use this command in a production environment.

Please consult the official OpenAI documentation for the most up-to-date information on the API parameters and pricing.

```
' Set OpenAI API-Key from the saved File
AIC.SetKey|File
```

```
' This will make the AI to generate 3 Pictures.
AIC.SetNumber|3
```

```
' Picture-Size can be 256 or 512 or 1024
AIC.Set Image Size|256
```

```
$$TXT=Chessboard with a nice golden Cowgirls on it.
AIC.Generate Image|$$TXT
```

```
' Uncomment these lines below to see the Original Output o
' in case of errors
'AIC.gro|$$ROW
'MBX.$$ROW
ENR.
```



Here you can see the other 2 pictures that have been created using the Dalle2-AI from Open AI.

Important:

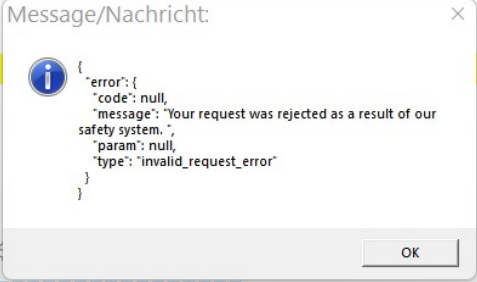
Be aware that the AI will refuse to generate several types of pictures for ethical reasons.

```
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

$$TXT=Chessboard with some nice golden girls on it.
AIC.Generate Image|$$TXT

AIC.gro|$$ROW
MBX.$$ROW
ENR.

CLP.$$RET
DBP.Temp.: $$
DRD
```



In case you do not get a picture out of your prompt, you can use AIC.gro to read the original output of the Open AI Server.
In this case the word "Girl" in the prompt made the server to refuse to generate a picture.

Syntax

AIC.Generate Image|P1 [|P2] [|P3]
AIC.GIM|P1 [|P2] [|P3]

Parameter Explanation

P1 - Prompt/ Text that describes the Picture that you want to generate.

P2 - opt. File path and File name where you want the Picture to be saved. Please include "****" where the Picture number should be inserted.

P3 - opt. Picture size: This can only be one of these: "256x256", "512x512" or "1024x1024". If you omit this, "1024x1024" is chosen.

Example

```
*****
' EXAMPLE 1: AIC.-Commands
*****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' This will make the AI to generate 3 Pictures
AIC.SetNumber|3
' Can be 256 or 512 or 1024
AIC.Set Image Size|256
$$TXT=Chessboard with a nice golden Cowgirls on a horse with 4 legs and an galac
AIC.Generate Image|$$TXT
```



```
' Use this to see the Original Output of the API
' in case of errors
'AIC.gro|$$ROW
'MBX.$$ROW
ENR.
```

Remarks

We can not take responsibility that the generated Images are free of Copyrights.
Generating Pictures with AI is currently a new process where the copyright situation is not perfectly clear.
Therefore take this into account for whatever you use these pictures.

Limitations:

At the time of the writing of this manual, **DALLE 2** is the only official available Endpoint from OpenAI.
There are however rumors that **DALLE 3** will soon be available. Possibly this newer model can be used if you chose another endpoint using the
AIC.Set Image Generation Endpoint - Command.
If there are more changes there will be an updated for the new model.

See also:

-

3.42.10.1.2 Set Image Generation Endpoint

[AIC.Set Image Generation Endpoint / AIC.sge](#) [Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set Image Generation Endpoint / AIC.sge

Configures the image generation endpoint.



Intention

The `AIC.Set Image Generation Endpoint` command sets the endpoint URL for image generation.

If **P1** is omitted, the endpoint is reset to its default value.

Syntax

```
AIC.Set Image Generation  
Endpoint [ | P1 ]
```

AIC . sge [| P1]

Parameter Explanation

P1: Optional. The endpoint URL for image generation. If omitted, the endpoint is reset to the default value.

Example

```
!*****  
! AIC.-Sample  
!*****  
$$ENP=https://api.openai.com/v1/images/generations  
AIC.Set Image Generation Endpoint|$$ENP
```

Remarks

The default endpoint URL is:
"https://api.openai.com/v1/images/generations".

At the time of writing this manual, DALLE 3 was announced but the Endpoint URL is not yet available.

Limitations:

-

See also:

-

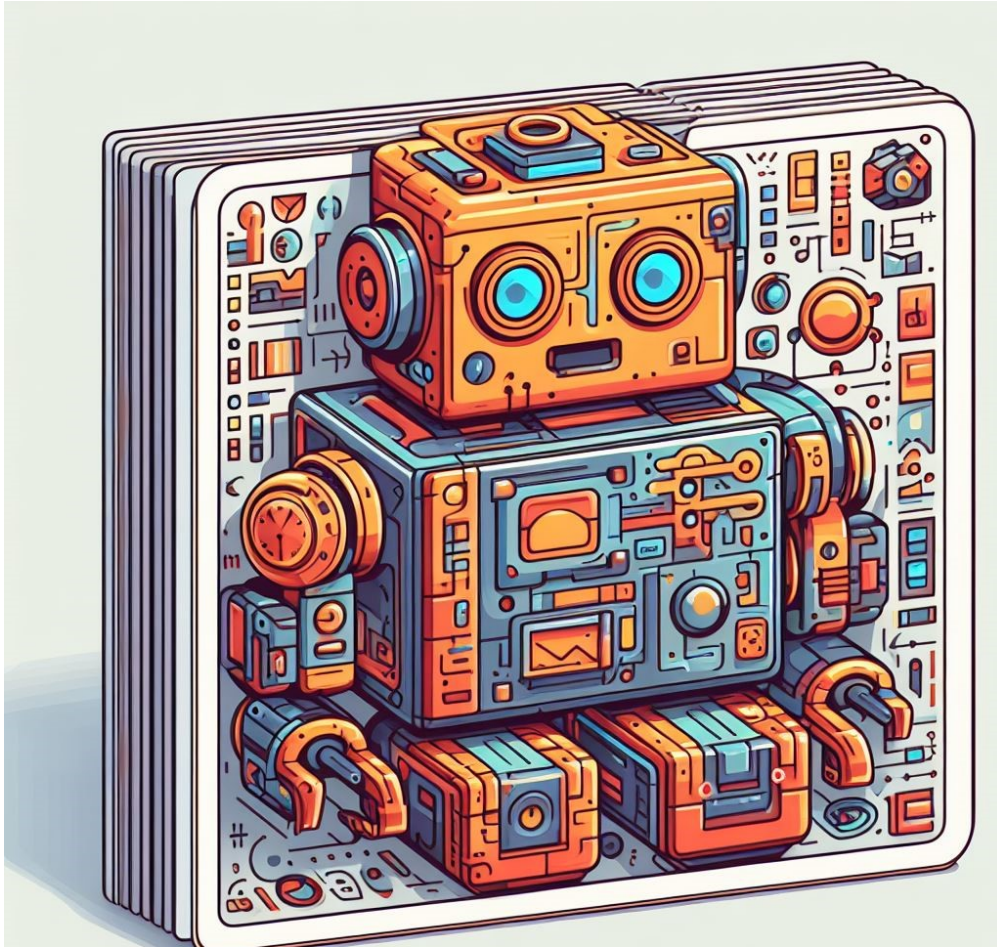
3.42.10.1.3 Set Image Size

[AIC.Set Image Size](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set Image Size

Use the Open AI-API Server to Set Image Sizes from a Text-Prompt



Intention

The `AIC.Set Image Size` command in the Smart-Package Robot (SPR) allows users to set the desired image size for images that will be generated using the OpenAI Image Generation endpoint. This command should be used prior to issuing any commands that generate images, to ensure that the output images are of the desired resolution.

Syntax:

```
AIC.Set Image Size|<ImageSize>
```

Parameters:

<ImageSize>: The resolution of the image to be generated. It must be in the format `WIDTHxHEIGHT` and can range from `256x256` to `1024x1024`. For example, "512x512".

Example Usage:

```
AIC.Set Image Size|1024x1024
```

Costs:

The costs for using the OpenAI Image Generation Endpoint through the SPR command are based on the resolution of the output image.

As of the time of writing, the costs are as follows:

- 256x256 resolution: \$0.016 per image
- 512x512 resolution: \$0.018 per image
- 1024x1024 resolution: \$0.020 per image

Please note that if you are a new user of OpenAI's API, you can benefit from a free trial that allows you to use \$18 of free credits within your first three months. Keep in mind that pricing may change, so it's advisable to consult the official OpenAI documentation for the most current information.

```
' Set OpenAI API-Key from the saved File
AIC.SetKey|File
```

```
' This will make the AI to generate 3 Pictures.
AIC.SetNumber|3
```

```
' Picture-Size can be 256 or 512 or 1024
AIC.Set Image Size|256
```

```
$$TXT=Chessboard with a nice golden Cowgirls on it.
AIC.Set Image Size|$$TXT
```

```
' Uncomment these lines below to see the Original Output o
' in case of errors
'AIC.gro|$$ROW
'MBX.$$ROW
ENR.
```

Syntax

AIC.Set Image Size|P1

AIC.SIS | P1

Parameter Explanation

P1 - opt. Picturesize: This can only be one of these: "256x256", "512x512" or "1024x1024". If you omit this, "256x256" is chosen.

Example

```
'*****
' EXAMPLE 1: AIC.-Commands
'*****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' This will make the AI to generate 3 Pictures
AIC.SetNumber|3
' Can be 256 or 512 or 1024
AIC.Set Image Size|256
$$TXT=Chessboard with a nice golden Cowgirls on a horse with 4 legs and an galac
AIC.Set Image Size|$$TXT

' Use this to see the Original Output of the API
' in case of errors
'AIC.gro|$$ROW
'MBX.$$ROW
ENR.
```

Remarks

-

Limitations:

-

See also:

- [DMP. Dump System Values](#)^[1823]

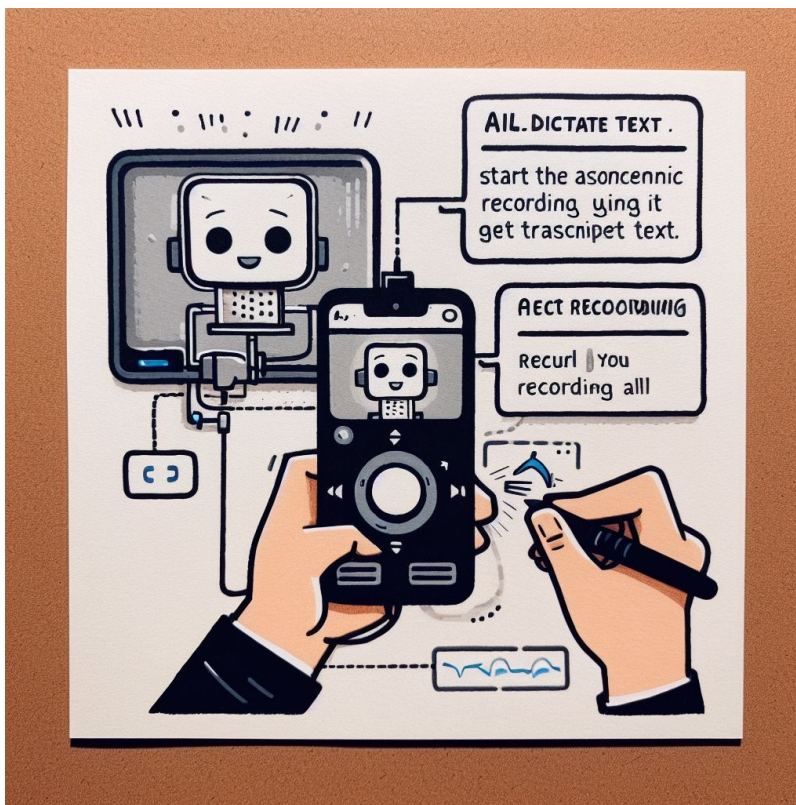
3.42.10.2 ! Open AI - Whisper

What is Whisper?

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

What is Whisper and how does Whisper Work?



Whisper is the leading Speech to Text Technology from Open AI.

Whisper uses machine learning algorithms trained on a vast dataset of audio and corresponding text, to implement the leading **Speech to Text** (STT-) Technology. When you send an audio clip to the Whisper Cloud, it processes the audio and returns the transcribed text in real-time.

Here's a simplified workflow:

1. **Capture Audio**: Record the audio you want to transcribe.
2. **AIW.Ask_Whisper**: This command will send the audio file to the **Whisper Cloud**.
3. **Get Response**: You will receive the transcribed / translated text directly in a SPR-Variable.
4. **Display or Use Text**: Use the transcribed text in your project as needed.
5. **Process with GPT-3.5 or GPT-4**: Use the transcribed text in your project and have it automatically corrected and processed, using GPT-3.5 or GPT-4 as needed.

Smart Package Robot Commands for Whisper API

To make it easier for SPR users, we have integrated specific commands that interact with the Whisper API.

Why is Whisper Good?

No Training Required

Unlike some **STT** systems that require you to "train" the model to understand your voice, Whisper works "out of the box." It's designed to understand a wide range of accents and dialects.

Multilingual Support

Whisper can transcribe nearly 100 languages, making it incredibly versatile for international projects.

High Accuracy

Whisper's advanced algorithms ensure that the transcriptions are highly accurate, even in noisy environments.

Real-Time Transcription

The API is designed for low-latency, real-time transcription, which is crucial for interactive applications.

Cloud-Based or Local

Whisper offers both cloud-based and local solutions. The cloud-based API is cost-effective and doesn't require any installation, while the local version is ideal for those who need to keep their data in-house.

Easy Integration with GPT-3.5

Whisper can be easily combined with OpenAI's GPT-3.5 to not just transcribe the text but also to understand and act upon it, making your SPR projects smarter and more interactive.

Conclusion

Whisper's advanced Speech-to-Text capabilities make it an invaluable tool for Smart Package Robot users. Its ease of use, high accuracy, and real-time processing capabilities set it apart from other STT technologies. By integrating Whisper into your projects, you open up a whole new realm of possibilities for voice-activated automation and data collection.

Schema of the Whisper AI from Open AI, see.

<https://raw.githubusercontent.com/openai/whisper/main/approach.png>

from:

<https://github.com/openai/whisper#available-models-and-languages>
or the [Whisper Whitepaper](#)

3.42.10.2.1 ! Open AI - Whisper "Speech to Text"

[Whisper "Speech 2 Text"](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)



Whisper AI - "Speech to Text"

Whisper API for SPR: Speech-to-Text Made Easy

Introduction

Whisper is an advanced Speech-to-Text (STT) system developed by [OpenAI](#). It's designed to convert spoken language into written text with high accuracy and minimal latency.

Whisper is particularly useful for Smart Package Robot users who want to integrate voice commands or transcriptions into their projects. In this section, we'll explore how Whisper works and why it's a valuable addition to your Scripts.

API-Key needed

To use Whisper you will need an [OpenAI API key](#).

This is the same key that you can also use for the other OpenAI services like ChatGPT and DALLE 2 image generation.

We have in this manual a chapter on how you obtain this key and the details how you use the key.

From all services [WHISPER is the cheapest](#) with a price of currently just \$0.006 / minute (rounded to the nearest second).

Which equals about 6 Cent per 10 Minutes of transcription.

Combining AI's with the SPR

Imagine a world where your voice can be transformed into text and then spoken back to you in a completely different voice, all within seconds. Welcome to the incredible synergy of ElevenLabs' Speech Synthesis and OpenAI's Whisper Speech-to-Text! 🗣️ 🎧

How Does It Work? 🤖

- You Speak: Simply say something out loud.
- Whisper Listens: OpenAI's Whisper technology converts your spoken words into text.
- 11 Labs Speaks: This text is then sent to 11 Labs, which synthesizes it into speech using a different voice.

Prompts and Quality Improvement

You can use a prompt to enhance the quality of the transcripts generated by the Whisper API. The model will attempt to match the style of the prompt, making it more likely to use capitalization and punctuation if the prompt does. However, the current prompting system is more limited than other language models and only provides limited control over the generated audio. Here are some ways prompting can assist:

- **Specific Words/Acronyms:** Prompts can correct specific words or acronyms that the model often misrecognizes. For instance, the prompt "The transcript is about OpenAI which makes technology like DALL-E, GPT-3, and ChatGPT..." can improve the transcription of words like DALL-E and GPT-3.
- **Preserving Context:** To maintain the context of a segmented file, prompt the model with the transcript of the preceding segment. This makes the transcript more accurate, as the model will use the relevant information from the previous audio.
- **Punctuation:** The model might sometimes skip punctuation. This can be rectified by using a prompt that includes punctuation, e.g., "Hello, welcome to my lecture."
- **Filler Words:** The model may omit common filler words. To retain these in your transcript, use a prompt containing them, e.g., "Umm, let me think like, hmm... Okay, here's what I'm, like, thinking."
- **Writing Styles:** Some languages, like Chinese, have different writing styles (simplified or traditional). Use a prompt in your preferred style to guide the model.

Here is a sample Prompt that will tell Whisper how several acronyms are written:

```
' Using Parameters could look like this, and will help the AI to transcribe this
$$PRO=ZyntriQix, Digique Plus, CynapseFive, VortiQore V8, EchoNix Array,
$$PRO+OrbitalLink Seven, DigiFractal Matrix, PULSE, RAPT, B.R.I.C.K., Q.U.A.R.T
AIC.Set Whisper Default|$$PRO
```

The Best Part? 🪄

You can achieve this magical experience with just 5 lines of code (See below)! Yes, you read that right. Six lines are all it takes to create this voice transformation loop. 🪄

```
AIC.Set Key|file

AIC.Set Whisper default
AIC.Dictate Text|$$RET
```

```
' Send Whisper Output to Elevenlabs to speak
AIS.Say Text|$$RES
ENR.
```

And what if you just add one more Line calling ChatGPT with the result from Whisper??

```
AIS.Set Key|file
AIC.Set Key|file

VAF.$$FIL=?exeloc\Test.mp3

AIC.crb
AIC.rsb.|$$FIL
AIC.dtx|$$FIL|$$RES

$$INS=Please take the following text and convert it into a rhyme of Goethe, retu
$$INS+$crlf+$$RES
AIC.Ask_Chat|$$INS|$$OUT

DBP.$$OUT
' Now we speak the Text
AIS.Say Text|$$OUT
ENR.
```

See also:

-

3.42.10.2.2 AIC.-Recording

AIC. - Recording

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

Recording Commands

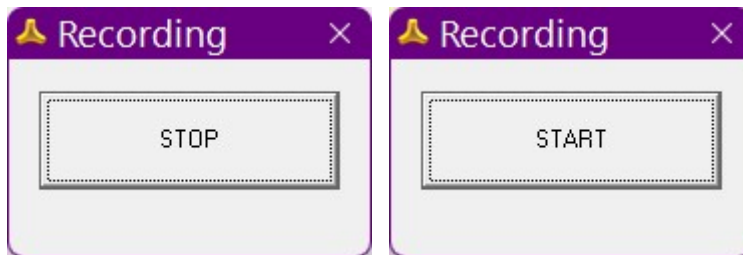
If Result available

Intention

To be able to use the speech-to-text engine Whisper, we thought it's necessary that you can also do the recording just from the Smart Package Robot. For this we have implemented basic recording features. While the Smart Package Robot has already some sound mixing commands, in another topic we have here only the recording and you have a very simple recording control using the recording buttons. Means you can start the recording and when you are done you press the button and the robot will then work on the resulting text.

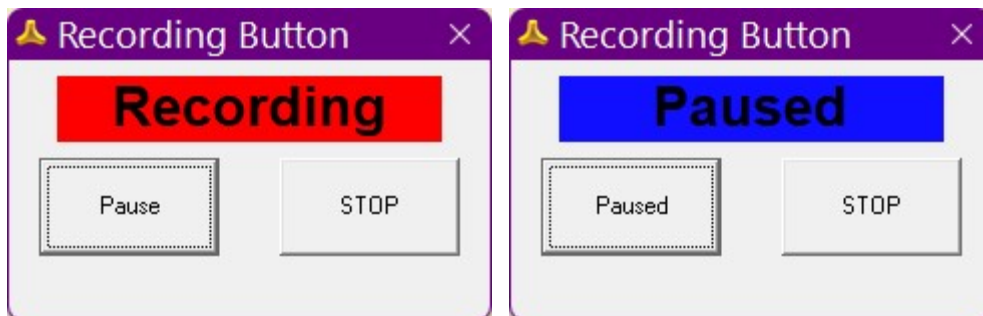
Using the two Commands: `AIC.Dictate Text` and `AIC.Dictate Letter` you can immediately start dictating your Text and have it transcribed.

Here you can see the available Recording Buttons.



AIC.Create Rec Button|1|0|0

AIC.Create Rec Button|0|0



AIC.crb|1|0|1

AIC.crb|0|0|1

Example

```
'*****  
' AIC.-Sample  
'*****  
AIC.Set Key|file  
  
AIC.Dictate Letter|$$OUT  
  
' Here we get the result  
DBP.$$OUT  
ENR.
```

Remarks

-

Limitations:

-

See also:

- [1.6.1. Program Flow Control](#)¹³⁴
- [! Smart Package Robot 's Parallel Robot Operations](#)³⁷⁴⁵
- [1.5. Features and Hints](#)⁹³

3.42.10.2.2.1 Create Rec Button

AIC.Create Rec Button

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Create Rec Button

Creates a Record Button with dynamic text.

Intention

The `AIC.Create Rec Button` command creates a button that can be used to start or stop a recording.

The button's text and behavior can be customized using the **P1** and **P2** parameters.

The concept of the "Record" button is to have a background-running button whose status—whether it's pressed or not—can be accessed from your Script.

You can initiate the recording through a separate command, and then clicking this button will terminate the recording by internally altering the recording state.

Essentially, you can start recording an MP3 file with one command and stop it by simply clicking the button.

You can customize the button's behavior using the available parameters. For instance, you can set the button to disappear immediately upon clicking.

Alternatively, you can keep the button visible at all times, allowing you to start and stop various tasks, such as recording, with a simple click.

Generally the Record-Button has 2 States, "START" and "STOP"-State.

While Recording the State of the Record-Button is "1" and the Button shows "STOP".

If you press the Button, the State switches from "1" to "0" and from "0" to "1".

```
' You can get the current Status of the Record-Button
```

```
AIC.Get Rec State|$$STA
```

```
' You can change the state of the Recording
```

```
AIC.Set Rec State|$$STA
```

```
' Close the Recording Button with
```

```
AIC.End Rec Button
```

If you do not want to press the button and instead react "**on Mouseover**" you can use:

```
' Get 0/1 if Mouse is over the Record-Button
```

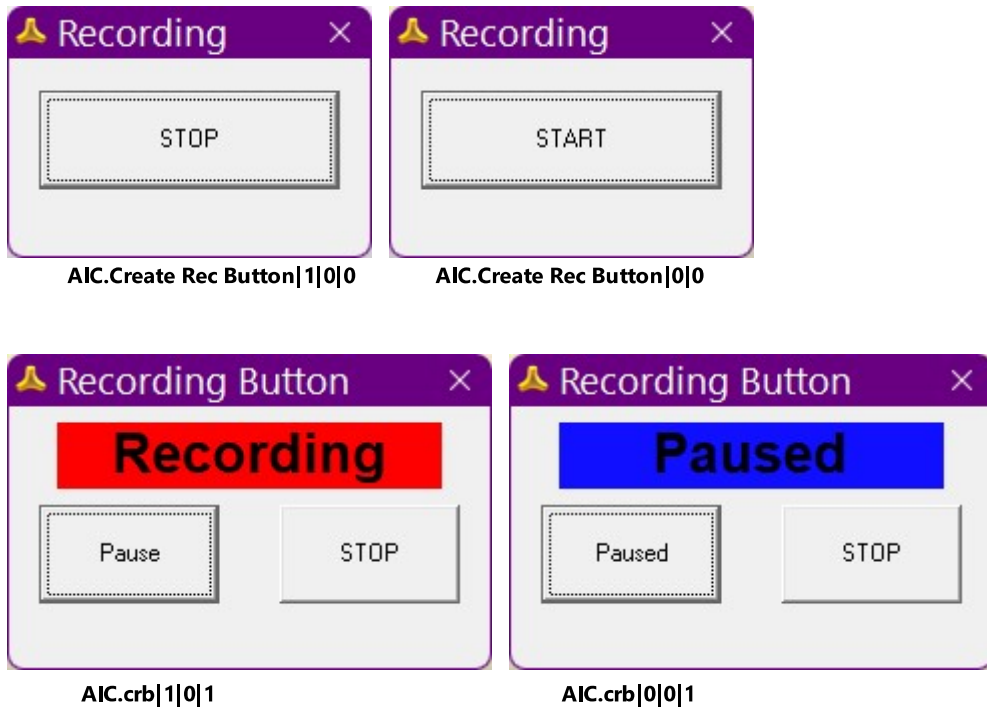
```
AIC.Get MouseOver State|$$STA
```

There are some more Commands that are used with this "Record-Button".

```
' Change Text on the Button
```

```
AIC.Set RecordButton Text|$$TXT
```

Here you can see the



Syntax

AIC.Create Rec Button[|P1][|P2][|P3]
AIC.Crb[|P1][|P2][|P3]

Parameter Explanation

- **P1:** Determines the initial text on the button. If set to 0, the text will be "STOP". If set to 1, the text will be "START".
- **P2:** Determines the button's behavior when clicked. If set to 1, the button will end its function when clicked. If set to 0, the button will toggle its text between "START" and "STOP" when clicked. In that case
- **P3:** 0 - Only a STOP / START - Button
 1 - Has 2 Buttons

Example

```

| *****
| AIC.-Sample
| *****
    
```

```
' We create a Recording Button, to be able to stop the recording
AIC.Create Rec Button|1|0

' This Loop is endless unless you move the Mouse over the Record Button
DOL.1
  ' Get Mouseover State
  AIC.gms|$$RET
  PRT.$$RET

OOP.($$RET=1)

' Close the Recording Button with
AIC.End Rec Button
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.10.2.2.2 End Rec Button

AIC.End Rec Button

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.End Rec Button

Close the Recording Button Form



You can close the Recording Button Form using this command.

Intention

This Command will close the Recording Button-Form and stop the Recording if needed.

Syntax

AIC.End Rec Button

Parameter Explanation

- no Parameters.

Example

```

'*****
' IRS.-Sample
'*****
AIC.crb|0|0|1
DOL.

'AIC.Get Rec State|$$RET
AIC.grt|$$RET
DBP.$$RET

```

```
PAU.1  
AIC.grs|$$RET  
OOP.($$RET>0)  
AIC.End Rec Button  
MBX.!  
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.10.2.2.3 Get MouseOver State

[AIC.Get MouseOver State](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Get MouseOver State

Checks if the mouse is over the "Recording Button".



Intention

The `AIC.Get MouseOver State` command checks whether the mouse cursor is currently hovering over the "Recording-Button".
In case you are using the Recording Form with 2 Buttons, only the `STOP`-Button is active for Mouseover.

The result is returned as a 0 or 1, indicating the absence or presence of the mouse over the button, respectively.
The concept behind the "`AIC.get mouseover state`" command is to enable interaction without requiring a mouse click. Simply by hovering your mouse over the button, you can either start or stop the recording.

Syntax

```
AIC.Get MouseOver State [ | P1 ]
```

```
AIC.gms [ | P1 ]
```

Parameter Explanation

P1: Optional. Variable to store the result. If the mouse is over the button, the variable will be set to 1; otherwise, it will be set to 0.
If omitted, the result is pushed onto the TOS (Top Of Stack).

Example

```
'*****  
' AIC.-Sample  
'*****  
AIC.Set Key|file  
  
' We create a Recording Button, to be able to stop the recording  
AIC.Create Rec Button|1|0  
  
' Close Recording button and end recording on Mouseover  
DOL.1  
  AIC.gms|$$RET  
  PRT.$$RET  
OOP.($$RET=1)  
  
' We end the recording (if its running)  
AIC.Set Rec State|0  
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.10.2.2.4 Get Rec State

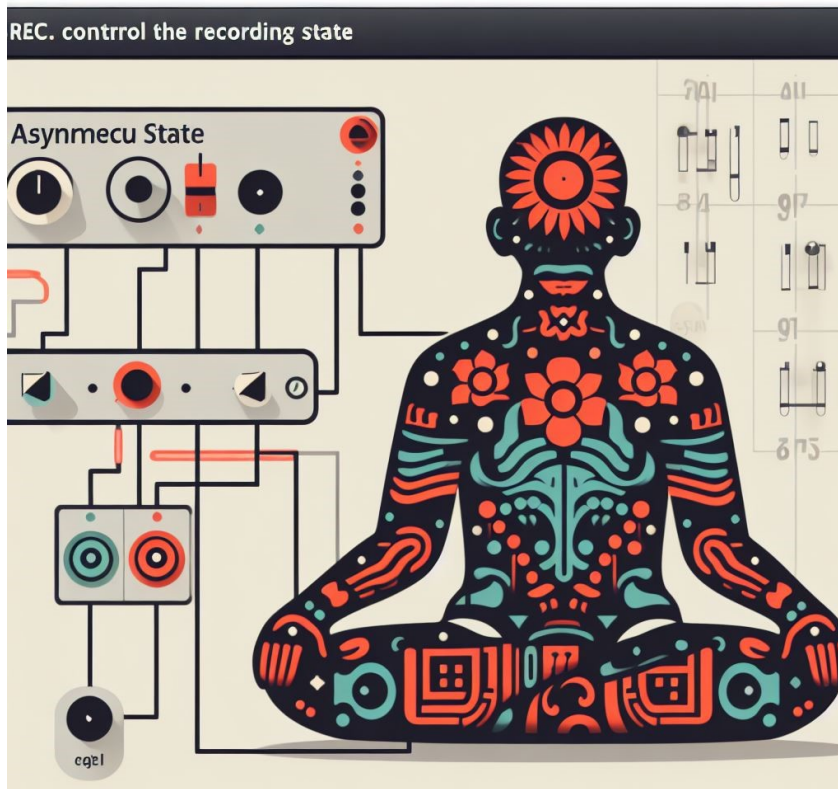
AIC.Get Rec State / AIC.grs

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Get Rec State / AIC.grs

Returns recording state.

**Intention**

The `AIC.Get Rec State` command returns the current recording state of the system in the variable specified in **P1**.

The recording state is an internal register that can be read or altered, typically to stop the recording.

Here is a sample Script that shows how you can use the command together with Asynchronous Recording.

```
AIC.Set Key|file

' We create a Recording Button, to be able to stop the recording
AIC.Create Rec Button|0|0|1

VAF.$$FIL=?exeloc\Test.mp3
' We start the recording
AIC.Record Asynchron|$$FIL

DOL.
```



```

AIC.grs|$$RET
PRT.$$RET
OOP. ($$RET>0)

```

```
ENR.
```

Syntax

AIC.Get Rec State|P1

AIC.Grs|P1

Parameter Explanation

P1: Optional. Variable to store the recording state. If omitted, the result is placed on the Top of Stack (TOS).

The following Rec-States will be returned:

- 0 - Buttons are in recording State (not Stopped, not Paused)
- 1 - Recording is PAUSED
- 2 - Recording is Stopped/Ended
- 3 - Stopped and Paused

Example

```

'*****
' IRS.-Sample
'*****
AIC.Set Key|file

' We create a Recording Button, to be able to stop the recording
AIC.Create Rec Button|1|0

VAF.$$FIL=?exeloc\Test.mp3
' We start the recording
AIC.Record Asynchron|$$FIL

DOL.
  AIC.grs|$$RET
  PRT.$$RET
  OOP. ($$RET>0)

ENR.

```

Remarks

-

Limitations:

-

See also:

-

3.42.10.2.2.5 Get Rec State Text

AIC.Get Rec State Text / AIC.grt

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Get Rec State Text / AIC.grt

Returns recording state.



Intention

The `AIC.Get Rec State Text` command returns the current recording state of the system in the variable specified in **P1**.

The recording state is an internal register that can be read or altered, typically to stop the recording.

This command has only 3 States:

"Recording" - if either `PAUSE` or `STOP` Button is activated

"Stopped" - if `STOP` - Button is activated (in this case `PAUSE` Button is ignored)

"Paused" - if `PAUSE` Button has been pressed

Here is a sample Script that shows how you can use the command together with Asynchronous Recording.

```
AIC.crb|0|0|1
DOL.
  AIC.grt|$$RET
  DBP.$$RET
  PAU.1
OOP.
' Here we get the result
PRT.$$RET
ENR.
```

Syntax

```
AIC.Get Rec State Text|P1
AIC.Grt|P1
```

Parameter Explanation

P1: Optional. Variable to store the recording state. If omitted, the result is placed on the Top of Stack (TOS).

The following Rec-States will be returned:

This command has only 3 States:

"Recording" - if either PAUSE or STOP Button is activated

"Stopped" - if STOP - Button is activated (in this case PAUSE Button is ignored)

"Paused" - if PAUSE Button has been pressed

Example

```

' *****
' IRS.-Sample
' *****
AIC.crb|0|0|1
DOL.

'AIC.Get Rec State|$$RET
AIC.grt|$$RET
DBP.$$RET
PAU.1
OOP.($$RET>0)
' Here we get the result
DBP.$$RET
ENR.

```

Remarks

-

Limitations:

-

See also:

•

3.42.10.2.2.6 Record Asynchron

AIC.Record Asynchron

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Record Asynchron

Initiates asynchronous recording.



You can use the "Recording Button" to start or End the recording.

Intention

The `AIC.Record Asynchron` command starts an asynchronous audio recording (Background recording while the Script continues running) and saves it to the specified file-name in **P1**. If **P1** is omitted, the system will generate a file-name automatically.

In reality, this command initiates the audio recording while allowing the script to continue running in parallel.

You have multiple options for stopping the audio recording:

- you can either click the recording button or
- simply set the recording state to 0.

of course the Recording will also end when the Script ends.

```
' We create a Recording Button, to be able to stop the recording
AIC.Create Rec Button|0|0
```

```
VAF.$$FIL=?exeloc\Test.mp3
```

```
' We start the recording
AIC.Record Asynchron|$$FIL
```

```
DOL.
  AIC.grs|$$RET
  PRT.$$RET
  OOP.($$RET>0)
ENR.
```

Syntax

AIC.Record Asynchron [|P1] [|P2]

AIC.Rea [|P1] [|P2]

Parameter Explanation

- **P1**: Optional. The filename to which the audio will be recorded. If omitted, the system will generate a filename.
- **P2**: Optional. Variable to store the filename used for recording. Useful when **P1** is omitted and the system generates a filename. If omitted, the result is placed on the Top of Stack (TOS).

Example

```

!*****
' AIC.-Sample
!*****
' We create a Recording Button, to be able to stop the recording
AIC.Create Rec Button|0|0

VAF.$$FIL=?exeloc\Test.mp3
' We start the recording
AIC.Record Asynchron|.$$FIL

DOL.
  AIC.grs|$$RET
  PRT.$$RET
OOP.($$RET>0)
ENR.

```

Remarks

-

Limitations:

-

See also:

-

3.42.10.2.2.7 Record Asynchron Button

AIC.Record Asynchron Button

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Record Asynchron Button

Initiates asynchronous recording and starts the "Recording Button".



You can use the "Recording Button" to start or End the recording.

Intention

The `AIC.Record Asynchron Button` command starts an asynchronous audio recording (Background recording while the Script continues running), and also brings up the "Recording Button".

Finally it will save the mp3-result to the specified file-name in **P1**. If **P1** is omitted, the system will generate a file-name automatically.

In reality, this command initiates the audio recording while allowing the script to continue running in parallel.

You have multiple options for stopping the audio recording:

- you can either click the recording button or
- simply set the recording state to 0.

of course the Recording will also end when the Script ends.

This command is configured to NOT close the Recording Button once it was pressed.

```
' We start the recording
AIC.Record Asynchron Button||$$FIL

AIC.wbs|2|$$RET

MBX.File was saved: $$FIL Button: $$RET
ENR.
```

Syntax

AIC.Record Asynchron [| P1] [| P2]

AIC.Rea [| P1] [| P2]

Parameter Explanation

- **P1:** Optional. The filename to which the audio will be recorded. If omitted, the system will generate a filename.
- **P2:** Optional. Variable to store the filename used for recording. Useful when P1 is omitted and the system generates a filename. If omitted, the result is placed on the Top of Stack (TOS).

Example

```

!*****
' AIC.-Sample
!*****

' We start the recording
AIC.Record Asynchron Button||$$FIL

' Wait for Button pressed
AIC.wbs|2|$$RET

MBX.File was saved: $$FIL Button: $$RET
ENR.

ENR.

```

Remarks

-

Limitations:

-

See also:

-

3.42.10.2.2.8 Record Synchron

AIC.Record Synchron[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Record Synchron

Initiates synchronous recording.



You can use the "Recording Button" to End the recording.

Intention

The `AIC.Record Synchron` command starts an synchronous audio recording (Background recording while the Script waits), It will not brings up the "Recording Button" that you need to end the recording. Therefore you should do that yourself, else you can not stop the recording.

Finally it will save the mp3-result to the specified file-name in **P1**. If **P1** is omitted, the system will generate a file-name automatically. In reality, this command initiates the audio recording while halting the script.

To stop the recording, press the **STOP** Button. Of course the Recording will also end when the Script ends.

Syntax

AIC.Record Synchron [| P1] [| P2]
AIC.Rec [| P1] [| P2]

Parameter Explanation

- **P1**: Optional. The file name to which the audio will be recorded. If omitted, the system will generate a file name.
- **P2**: Optional. Variable to store the file name used for recording. Useful when P1 is omitted and the system generates a file name. If omitted, the result is placed on the Top of Stack (TOS).

Example

```
*****  
' IRS.-Sample  
*****  
' We make the button manually  
AIC.crb|0|0  
' We start the recording  
AIC.Record Synchron||$$FIL  
  
MBX.File was saved: $$FIL  
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.10.2.2.9 Record Synchron Button

[AIC.Record Synchron Button](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Record Synchron Button

Initiates synchronous recording and starts the "Recording Button".



You can use the "Recording Button" to End the recording.

Intention

The `AIC.Record Synchron Button` command starts an synchronous audio recording (Background recording while the Script waits), and also brings up the "Recording Button" that you need to end the recording.

Finally it will save the mp3-result to the specified file-name in **P1**. If **P1** is omitted, the system will generate a file-name automatically.

In reality, this command initiates the audio recording while halting the script.

To stop the recording, press the STOP Button.
Of course the Recording will also end when the Script ends.

This command is configured to close the Recording Button once it was pressed.

Syntax

```
AIC.Record Synchron Button[|P1]
[|P2]
AIC.Rsb[|P1][|P2]
```

Parameter Explanation

- **P1**: Optional. The file name to which the audio will be recorded. If omitted, the system will generate a file name.

- **P2:** Optional. Variable to store the file name used for recording. Useful when P1 is omitted and the system generates a file name. If omitted, the result is placed on the Top of Stack (TOS).

Example

```
' *****  
' IRS.-Sample  
' *****  
' We start the recording  
AIC.Record Synchron Button||$$FIL  
  
MBX.File was saved: $$FIL  
ENR.
```

Remarks

-

Limitations:

-

See also:

- [1.6.1. Program Flow Control](#)¹³⁴
- [! Smart Package Robot 's Parallel Robot Operations](#)³⁷⁴⁵
- [1.5. Features and Hints](#)⁹³

3.42.10.2.2.10 Set Rec State

[AIC.SetRec State](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.SetRec State

Sets the status of the recording process like if you press the STOP-Button.



Intention

The `AIC.SetRec State` command is designed to manually control the recording state.

This is useful for asynchronous recording or for implementing custom logic that requires changing the recording state.

The command will set the recording state based on the value provided in **P1**.

If **P1** is omitted, the command will use "0".

This is useful for scenarios where you want to programmatically control the recording state.

Syntax

AIC.SetRec State [|P1] AIC.Srs [|P1]

Parameter Explanation

P1: Optional. A number that represents the desired recording state.

- **0** - Let recording run (default)
- **1** - Stop recording
- **2** - Invert the current state (if recording, then stop; if stopped, then record)

Example

```
*****  
' AIC.-Sample  
*****  
' This will stop the recording  
AIC.SetRec State|1
```

Remarks

-

Limitations:

-

See also:

-

3.42.10.2.2.11 Set RecordButton Text

[AIC.Set Record Button Text](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set Record Button Text

Sets or resets the text to be displayed on the Record Button.



You can use this command to change the Text for the Recording STOP Button, temporarily.

Intention

The `AIC.Set Record Button Text` command allows you to customize the text displayed on the Record Button.

This is useful for localization or for providing additional context to the user or if you want to use the Button for something else.

The command will update the text on the Record Button based on the value provided in **P1**.

If **P1** is omitted or empty, the button will display its default text. This is useful to restore the original text.

Of course you can also use all Standard Robot commands like "`SDT`." to change all Texts on Buttons and Forms.

Syntax

```
AIC.Set Record Button Text[|P1]  
AIC.Sbt[|P1]
```

Parameter Explanation

P1: Optional. A string that represents the text to be displayed on the Record Button.

If this parameter is empty or omitted, the default text will be displayed on the button.

Example

```
'*****  
' AIC.-Sample  
'*****  
  ' This will set the button text to "Start"  
AIC.Set Record Button Text|Start
```

Remarks

-

Limitations:

-

See also:

-

3.42.10.2.2.12 Wait for Button State

AIC.Wait for Button State

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Wait for Button State

Waits for a specific button state in the recording process.



Intention

The `AIC.Wait for Button State` command is designed to pause the execution of the script until the recording button reaches a specified state.

This is useful for synchronizing the script's actions with the recording process.

The command will halt the script's execution until the button reaches the state specified in P1.

If **P1** is zero any Press on the **STOP** or **PAUSE** Button will make the command leave the waiting Loop.

This is particularly useful in scenarios where you want to ensure that the recording has reached a certain state before proceeding with other actions in the script.

Syntax

```
AIC.Wait for Button State[P1] [|
P2]
AIC.Wbs [P1] [| P2]
```

Parameter Explanation

- **P1**: Required. A number from **0** to **3** that represents the desired button state.
 - **0** - Recording
 - **1** - Paused
 - **2** - Stopped
 - **3** - Paused and Stopped

- **P2:** Optional. A variable to store the button state. If omitted, the result is placed on the Top of Stack (TOS).

Example

```
'*****  
' AIC.-Sample  
'*****  
' We start the recording  
AIC.Record Asynchron Button||$$FIL  
  
' We will wait until any Press on a Button  
AIC.wbs||$$RET  
MBX.File was saved: $$FIL Button: $$RET  
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.10.2.2.13 Wait No Pause

[AIC.Wait No Pause](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Wait No Pause

Wait until the Pause Button is not more in "Paused" State

Intention

This command halts the Script until the Pause-Button will not be in "Paused"-Mode.

```
AIC.crb|0|0|1
' We start the recording
AIC.Record Asynchron||$$FIL
```

```
AIC.wbs|1|$$RET
AIC.Wait No Pause
```

```
MBX.File was saved: $$FIL Button: $$RET
ENR.
```

Syntax

AIC.Wait No Pause

Parameter Explanation

- No Parameters

Example

```

' *****
' IRS.-Sample
' *****
AIC.crb|0|0|1
' We start the recording
AIC.Record Asynchron||$$FIL

AIC.wbs|1|$$RET
AIC.Wait No Pause

MBX.File was saved: $$FIL Button: $$RET
ENR.
```


Remarks

-

Limitations:

-

See also:

-

3.42.10.2.2.14 Wait Rec Button Close

[AIC.Wait Rec Button Close](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Wait Rec Button Close

If Result available

Intention

The `AIC.Wait Rec Button Close` command pauses the script execution until the user closes the Record Button.

This is useful in scenarios where you want to ensure that the user has finished interacting with the Record Button before proceeding with other tasks. Especially in asynchronous recording.

```
' We start the recording
AIC.Record Asynchron Button||$$FIL
AIC.wrc
MBX.File was saved: $$FIL
ENR.
```

Syntax

AIC.Wait Rec Button Close AIC.Wrc

Parameter Explanation

- *no Parameters*

Example

```
*****
' AIC.-Sample
*****
' We start the recording
AIC.Record Asynchron Button||$$FIL
AIC.wrc
MBX.File was saved: $$FIL
ENR.
```

Remarks

-

Limitations:

-

See also:

-

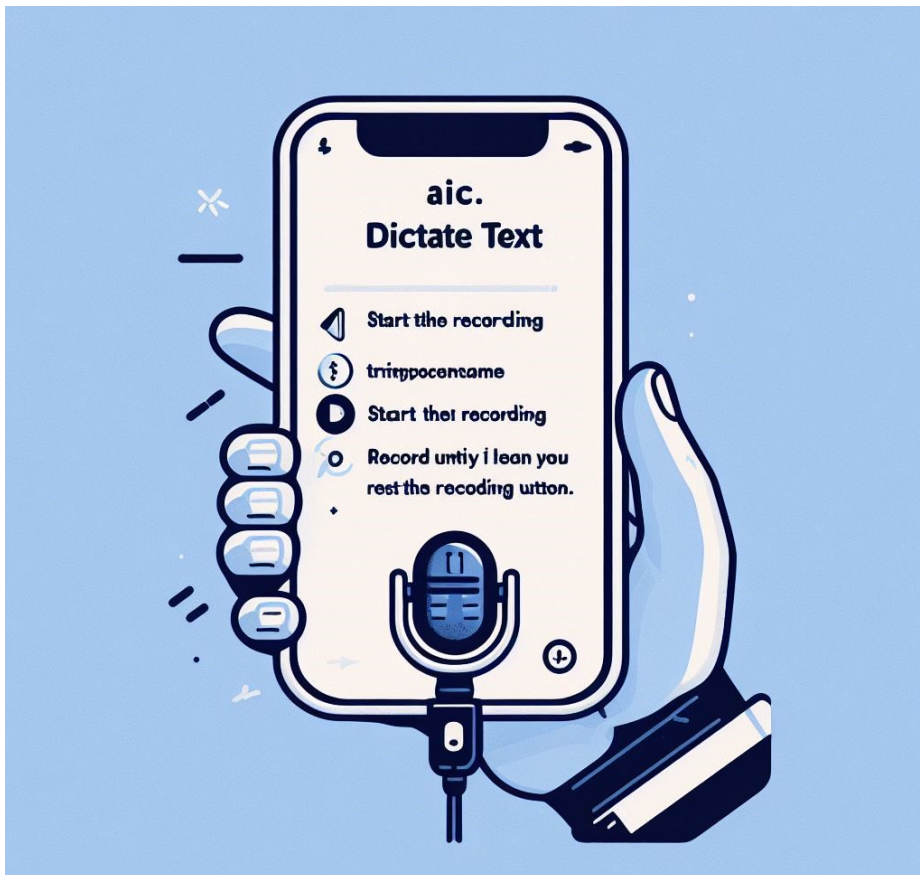
3.42.10.2.3 Dictate Letter

[AIC.Dictate Letter](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Dictate Letter

Initiates voice recording and transcribes the audio.



Whisper is the "Speech to Text" Option from Open AI.

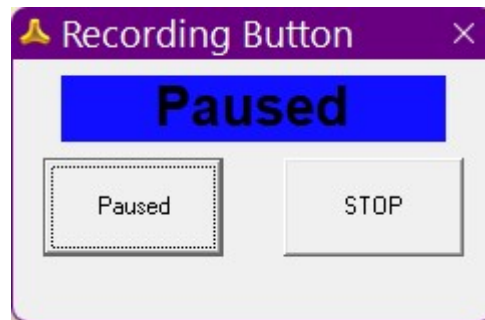
Intention

The Dictate Letter command is the easiest command to Dictate a longer Letter and get it transcribed into text.

The `AIC.Dictate Letter` command starts the recording process, showing you the Recording Form below.



This is shown while the Recording is running.



This is shown while the recording is paused.

The `AIC.Dictate Letter` command is the "One stop get all" command and will do this:

- start the Synchronous recording using a unique, temporary file name
- start the Recording Button
- record until you press the "PAUSE" or the "STOP" - Button
- If you want to interrupt the recording, just press "PAUSE".
- If you want to end the "Dictation" press "STOP".

Internally pressing "PAUSE" will also start the transcription process.

This is done to prevent the recorded files from getting too long.

Depending on the length of the recorded Speech this may take some seconds.

The resulting text is kept internally and will altogether be returned when you press the STOP-Button.

Pressing "Pause" will internally start the Text transcription.

The transcribed Text is not shown until you press the "STOP"-Button.

Recording will Start immediately when you see the Recording Button.

It will last until you press "STOP" or "PAUSE".

If you press "PAUSE" you can rest and continue the recording when you are ready.

If you press "STOP" then the command will close the Recording Button and deliver you the Text.

Important:

WHISPER may need some time to transcribe longer Texts, therefore the longer your Speech, the longer you will need to wait to get the final result.

Also Whisper has a limitation of 24 MB for upload. In our Tests a 30 Seconds Recording will take up to 385 kb.

Therefore this command has a limit of about 30 Minutes Recording Time, after that the resulting MP3-File would be larger than 25 MB and can not be transcribed using WHISPER.

Using the PAUSE-Button will eliminate this Limit, as each "Segment" is transcribed by itself.

Means there is no real world limit, you should be able to dictate a full book using this command.

Just press Pause at least all 30 Minutes.

```
'*****
' AIC.-Sample
'*****
AIC.Set Key|file
```

AIC.Dictate Letter|\$\$OUT

```
' Here we get the result
DBP.$$OUT
ENR.
```

Once the recording is stopped (typically using the PAUSE or the STOP Button), the dictated text is transcribed.

If you used the PAUSE Button the Text will be kept internally until its complete. And returned once you press STOP.

You can Dictate Letter in a lot of languages (see below). While you can use the The AIC.Set Language for Whisper command, to set the Input Language, in most case that is not needed, as Whisper will identify the languages automatically.

Important:

Whisper will take into account the following Settings:

- response_format
- language
- temperature
- prompt

If you have Set the

Supported languages:

Afrikaans, Arabic, Armenian, Azerbaijani, Belarusian, Bosnian, Bulgarian, Catalan, Chinese, Croatian, Czech, Danish, Dutch, English, Estonian, Finnish, French, Galician, German, Greek, Hebrew, Hindi, Hungarian, Icelandic, Indonesian, Italian, Japanese, Kannada, Kazakh, Korean, Latvian, Lithuanian, Macedonian, Malay, Marathi, Maori, Nepali, Norwegian, Persian, Polish, Portuguese, Romanian, Russian, Serbian, Slovak, Slovenian, Spanish, Swahili, Swedish, Tagalog, Tamil, Thai, Turkish, Ukrainian, Urdu, Vietnamese, And Welsh.

You can use the Command: AIC.Set Format for Whisper to decide which Output-Format you want. The default format is "text".

Supported Output Formats:

Number	Format	Description
0	text	Plain text

1	vtt	WebVTT (Web Video Text Tracks)
2	srt	SubRip Text (Subtitles)
3	raw	Raw data
4	json	JSON format
5	verbose_json	Verbose JSON format

Syntax

AIC.Dictate Letter|P1

Parameter Explanation

P1: Optional. The variable where the transcribed text will be stored. If omitted, the result is placed on the TOS.

Example

```

' *****
' AIC.-Sample
' *****
AIC.Set Key|file
AIS.Set Key|file

AIC.Dictate Letter|$$OUT

' Here we get the result
DBP.$$OUT

' We choose a speaker
AIS.Set Voice|7

' And we speak it using Elevenlabs
AIS.Say Text|$$OUT
ENR.

```

Remarks

Ensure that the environment is quiet enough for clear recording and accurate transcription. The Record Button will be displayed during the recording process, and pressing it will stop the recording.

It's important to keep in mind that Whisper only considers **the first 244 tokens of the prompt**.

Limitations:

As we explored in the prompting section, one of the most common challenges faced when using Whisper is the model often does not recognize uncommon words or acronyms.

To address this, we have highlighted different techniques which improve the reliability of Whisper in these cases:

1. The first method involves using the optional prompt parameter to pass a dictionary of the correct spellings.

Since it wasn't trained using instruction-following techniques, Whisper operates more like a base GPT model.

It's important to keep in mind that Whisper only considers the first 244 tokens of the prompt.

2. The second method involves a post-processing step using GPT-4 or GPT-3.5-Turbo.

We start by providing instructions for GPT-4 through the `system_prompt` variable.

Similar to what we did with the prompt parameter earlier, we can define our company and product names.

Sample: "You are a helpful assistant for the company ZyntriQix. Your task is to correct any spelling discrepancies in the transcribed text. Make sure that the names of the following products are spelled correctly: ZyntriQix, DigiQue Plus, CynapseFive, VortiQore V8, EchoNix Array, Orbitallink Seven, DigiFractal Matrix, PULSE, RAPT, B.R.I.C.K., Q.U.A.R.T.Z., F.L.I.N.T. Only add necessary punctuation such as periods, commas, and capitalization, and use only the context provided."

See also:

-

3.42.10.2.4 Dictate Text

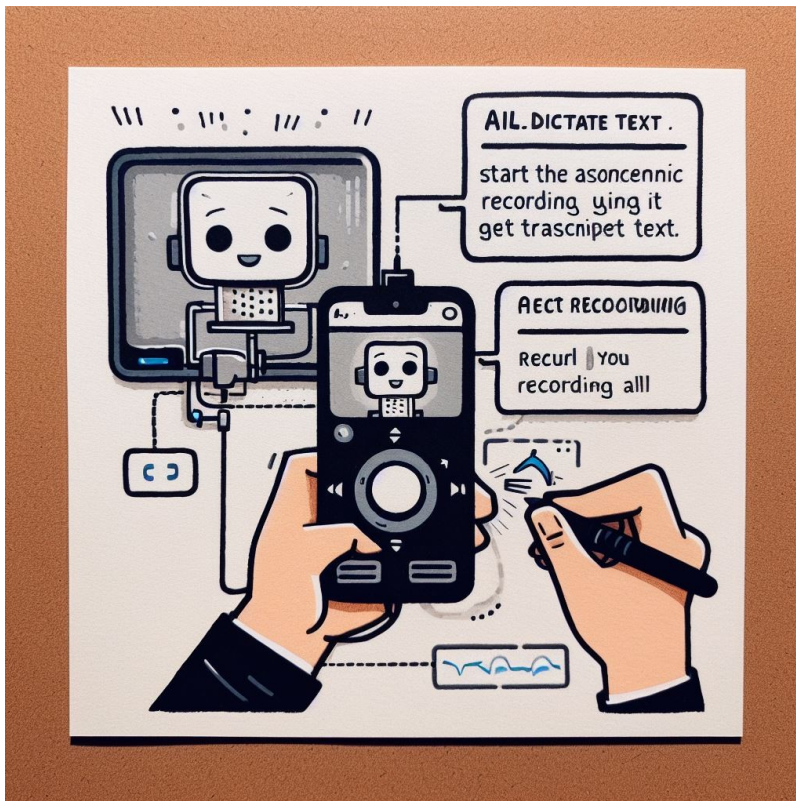
AIC.Dictate Text

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Dictate Text

Initiates voice recording and transcribes the audio.



Whisper is the "Speech to Text" Option from Open AI.

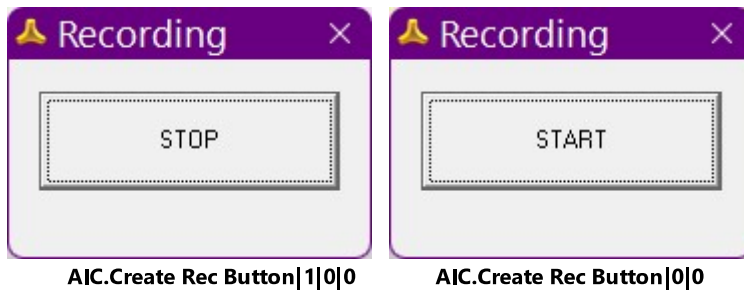
Intention

The dictate text command is the easiest command to dictate text and get it transcribed into text.

The `AIC.Dictate Text` command starts the recording process, allowing the user to dictate text.

The `AIC.Dictate Text` command is the "One stop get all" command and will do this:

- start the asynchronous recording using a unique, temporary filename
- start the Recording Button
- record until you press the recording Button



Recording will Start immediately when you see the Recording Button.

It will last until you press "STOP".

Then it will close the Recording Button and deliver you the Text.

Important:

WHISPER may need some time to transcribe longer Texts, therefore the longer your Speech, the longer you will need to wait to get the final result.

Also Whisper has a limitation of 24 MB for upload. In our Tests a 30 Seconds Recording will take up to 385 kb.

Therefore this command has a limit of about 30 Minutes Recording Time, after that the resulting MP3-File would be larger than 25 MB and can not be transcribed using WHISPER.

If you plan to make longer Recordings, use the

`AIC.Dictate Letter` - Command and make Pause between the Recordings, then there is virtually no limit.

```

!*****
! AIC.-Sample
!*****
AIC.Set Key|file

AIC.Dictate Letter|$$OUT

! Here we get the result
DBP. $$OUT
ENR.

```

Once the recording is stopped (typically using the Record Button), the dictated text is transcribed,

and returned in the specified variable or on the Top of Stack (TOS) if no variable is provided.

You can dictate text in a lot of languages (see below).

While you can use the

The `AIC.Set Language for Whisper` command, to set the Input Language, in most cases that is not needed, as Whisper will identify the languages automatically.

Important:

Whisper will take into account the following Settings:

- response_format
- language
- temperature
- prompt

If you have Set the

Supported languages:

Afrikaans, Arabic, Armenian, Azerbaijani, Belarusian, Bosnian, Bulgarian, Catalan, Chinese, Croatian, Czech, Danish, Dutch, English, Estonian, Finnish, French, Galician, German, Greek, Hebrew, Hindi, Hungarian, Icelandic, Indonesian, Italian, Japanese, Kannada, Kazakh, Korean, Latvian, Lithuanian, Macedonian, Malay, Marathi, Maori, Nepali, Norwegian, Persian, Polish, Portuguese, Romanian, Russian, Serbian, Slovak, Slovenian, Spanish, Swahili, Swedish, Tagalog, Tamil, Thai, Turkish, Ukrainian, Urdu, Vietnamese, And Welsh.

You can use the Command: `AIC.Set Format for Whisper` to decide which Output-Format you want.

The default format is "text".

Supported Output Formats:

Number	Format	Description
0	text	Plain text
1	vtt	WebVTT (Web Video Text Tracks)
2	srt	SubRip Text (Subtitles)
3	raw	Raw data
4	json	JSON format
5	verbose_json	Verbose JSON format

Syntax**AIC.Dictate Text | P1****Parameter Explanation**

P1: Optional. The variable where the transcribed text will be stored. If omitted, the result is placed on the TOS.

Example

```

'*****
' AIC.-Sample
'*****
' This Sample will transcribe what you say and send it to
ElevenLabs.io to say it loud.
AIC.Set Key|file
AIS.Set Key|file

AIC.Dictate Text|$$OUT

' Here we get the result
DBP.$$OUT

'We choose a speaker
AIS.Set Voice|7

' And we speak it using Elevenlabs
AIS.Say Text|$$OUT
ENR.

```

Remarks

Ensure that the environment is quiet enough for clear recording and accurate transcription. The Record Button will be displayed during the recording process, and pressing it will stop the recording.

It's important to keep in mind that Whisper only considers **the first 244 tokens of the prompt**.

Limitations:

As we explored in the prompting section, one of the most common challenges faced when using Whisper is the model often does not recognize uncommon words or acronyms.

To address this, we have highlighted different techniques which improve the reliability of Whisper in these cases:

1. The first method involves using the optional prompt parameter to pass a dictionary of the correct spellings.

Since it wasn't trained using instruction-following techniques, Whisper operates more like a base GPT model.

It's important to keep in mind that Whisper only considers the first 244 tokens of the prompt.

2. The second method involves a post-processing step using GPT-4 or GPT-3.5-Turbo. We start by providing instructions for GPT-4 through the system_prompt variable.

Similar to what we did with the prompt parameter earlier, we can define our company and product names.

Sample: "You are a helpful assistant for the company ZyntriQix. Your task is to correct any spelling discrepancies in the transcribed text. Make sure that the names of the following products are spelled correctly: ZyntriQix, DigiQue Plus, CynapseFive, VortiQore V8, EchoNix Array, OrbitalLink Seven, DigiFractal Matrix, PULSE, RAPT, B.R.I.C.K., Q.U.A.R.T.Z., F.L.I.N.T. Only add necessary punctuation such as periods, commas, and capitalization, and use only the context provided."

See also:

-

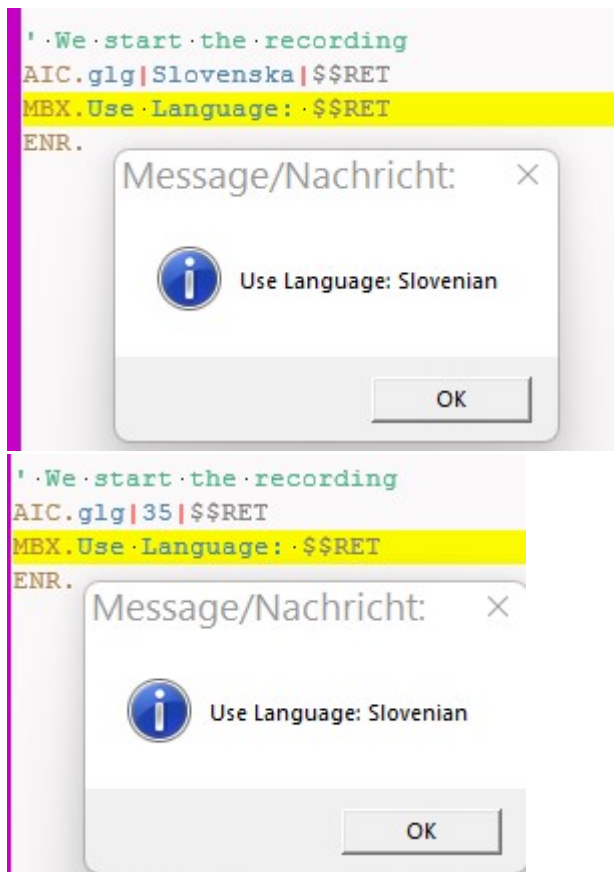
3.42.10.2.5 Get Language GPT

[AIC.Get Language GPT](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Get Language GPT

Retrieves the language setting for GPT based on a given number or language name.



This command helps you to find the right language to use for ChatGPT.

Intention

The `AIC.Get Language GPT` command is used to fetch the language name, supported by the GPT model. This can be useful for choosing the language in which the GPT model should be operating.

Syntax

AIC.Get Language GPT | P1 [| P2]

Parameter Explanation

- **P1:** This can be either a number corresponding to a language or the name of the language itself.
For example, **1** might correspond to English, or you could directly use "English".
- **P2 (Optional):** This is the variable where the result will be stored. If omitted, the result is pushed onto the Top Of Stack (TOS).

No.	Language	No.	Language
0	English	45	Zulu
1	Spanish	46	Afrikaans
2	French	47	Pashto
3	German	48	Farsi (Persian)
4	Italian	49	Bengali
5	Dutch	50	Punjabi
6	Portuguese	51	Gujarati
7	Russian	52	Marathi
8	Chinese (Simplified)	53	Tamil
9	Chinese (Traditional)	54	Telugu
10	Japanese	55	Kannada
11	Korean	56	Malayalam
12	Arabic	57	Nepali
13	Hebrew	58	Sinhalese
14	Hindi	59	Burmese
15	Urdu	60	Khmer
16	Swedish	61	Lao
17	Norwegian	62	Maldivian (Dhivehi)
18	Danish	63	Armenian
19	Finnish	64	Azerbaijani
20	Polish	65	Georgian
21	Czech	66	Kazakh
22	Hungarian	67	Kyrgyz
23	Romanian	68	Tajik
24	Greek	69	Turkmen

25	Turkish	70	Uzbek
26	Thai	71	Belarusian
27	Indonesian	72	Albanian
28	Malay	73	Bosnian
29	Filipino (Tagalog)	74	Macedonian
30	Vietnamese	75	Maltese
31	Ukrainian	76	Basque
32	Bulgarian	77	Corsican
33	Croatian	78	Frisian
34	Serbian	79	Galician
35	Slovenian	80	Breton
36	Slovak	81	Occitan
37	Icelandic	82	Sardinian
38	Estonian	83	Sicilian
39	Lithuanian	84	Romansh
40	Latvian	85	Sami
41	Irish	86	Faroese
42	Welsh	87	Greenlandic
43	Scottish Gaelic	88	Inuktitut
44	Swahili		

Example

```

'*****
' AIC.-Sample
'*****
' We start the recording
AIC.glg|Slovenska|$$RET
MBX.Use Language: $$RET
ENR.
    
```

Remarks

-

Limitations:

-

See also:

-

3.42.10.2.6 Set Format for Whisper

[AIC.Set Format](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set Format

Sets the output format for Open AI - WHISPER



Intention

The `AIC.Set Format` command is used to specify the format in which WHISPER will output the results.

This is useful for tailoring the output to specific needs, such as plain text for readability or JSON for data manipulation.

Syntax

AIC.Set Format|P1

Parameter Explanation

- **P1**: This can be either a number corresponding to a format or the name of the format itself. The available options are:

Supported Output Formats:

Number	Format	Description
0	text	Plain text
1	vtt	WebVTT (Web Video Text Tracks)
2	srt	SubRip Text (Subtitles)
3	raw	Raw data
4	json	JSON format
5	verbose_json	Verbose JSON format

Example

```
!*****  
' AIC.-Sample  
!*****  
' Sets the output format to WebVTT  
AIC.Set Format|1
```

Remarks

-

Limitations:

-

See also:

-

3.42.10.2.7 Set Language for Whisper

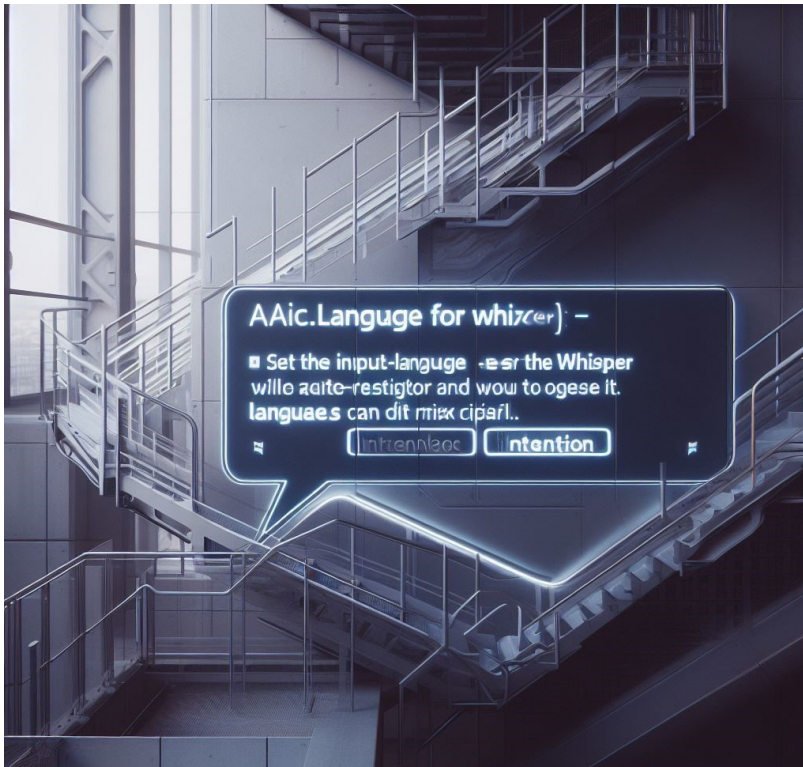
AIC.Set Language for Whisper

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set Language for Whisper

Set the language for the Whisper Input system.



Intention

Regularly you do NOT need to set the Input-language as Whisper will recognize it anyway.

In cases where languages are quite similar or where you want to mix languages, you can use this command to direct Whisper.

The `AIC.Set Language for Whisper` command is used to set the language for the Whisper Input system.

The command takes a string parameter and matches it to the best-fitting language available for Whisper Input. There is no return value for this command.

As Parameter **P1** you can either just use a number, or the text. If you write something else, this command will choose the best fit for your text.

Here is a List of supported languages at current date.

Number	Language
--------	----------

0	Afrikaans
1	Arabic
2	Armenian
3	Azerbaijani
4	Belarusian
5	Bosnian
6	Bulgarian
7	Catalan
8	Chinese
9	Croatian
10	Czech
11	Danish
12	Dutch
13	English
14	Estonian
15	Finnish
16	French
17	Galician
18	German
19	Greek
20	Hebrew
21	Hindi
22	Hungarian
23	Icelandic
24	Indonesian
25	Italian
26	Japanese
27	Kannada
28	Kazakh
29	Korean
30	Latvian
31	Lithuanian
32	Macedonian
33	Malay
34	Marathi
35	Maori
36	Nepali
37	Norwegian
38	Persian
39	Polish

40	Portuguese
41	Romanian
42	Russian
43	Serbian
44	Slovak
45	Slovenian
46	Spanish
47	Swahili
48	Swedish
49	Tagalog
50	Tamil
51	Thai
52	Turkish
53	Ukrainian
54	Urdu
55	Vietnamese
56	Welsh

Syntax

AIC.Set Language for Whisper|P1

Parameter Explanation

P1 - (Language, Text) Variable containing the language string or the number to be matched to the best-fitting language available for Whisper Input.

Example

```
'*****
' AIC.Set Language for Whisper
'*****
' Language for Whisper Input is set to English
VAR.$$LAN=English
AIC.Set Language for Whisper|$$LAN

' Language for Whisper Input is set to English
VAR.$$LAN=13
AIC.Set Language for Whisper|$$LAN

ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.10.2.8 Set Language Free

AIC.Set Language Free

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set Language Free

Sets the input language for WHISPER directly without changes



Intention

The `AIC.Set Language Free` command is used to specify the language that WHISPER will use for Automatic Speech Recognition (ASR).

This command allows for greater flexibility by accepting a text string that corresponds to a language.

Unlike with `AIC.Set Language for Whisper`, this command will not try to evaluate the input and directly use it.

Allowed values as of today:

af, am, ar, as, az, ba, be, bg, bn, bo, br, bs, ca, cs, cy, da, de, el, en, es, gu, ha, haw, he, hi, hr, ht, hu, hy, id, is, it, ja, jw, ka, kk, km, kn, ko, la, mk, ml, mn, mr, ms, mt, my, ne, nl, nn, no, oc, pa, pl, ps, pt, ro, ru, sa, sd, su, sv, sw, ta, te, tg, th, tk, tl, tr, tt, uk, ur, uz, vi, yi, yo, zh,

Afrikaans, Albanian, Amharic, Arabic, Armenian, Assamese, Azerbaijani, Bashkir, Bosnian, Breton, Bulgarian, Burmese, Castilian, Catalan, Chinese, Croatian, Czech, Estonian, Faroese, Finnish, Flemish, French, Galician, Georgian, German, Greek, Hausa, Hawaiian, Hebrew, Hindi, Hungarian, Icelandic, Indonesian, Italian, Japanese, Khmer, Korean, Lao, Latin, Latvian, Letzeburgesch, Lingala, Lithuanian, Luxembourgish, Malay, Malayalam, Maltese, Maori, Marathi, Moldavian, Moldovan, Mongolian, Myanmar, Occitan, Panjabi, Pashto, Persian, Polish, Portuguese, Punjabi, Pushto, Romanian, Sindhi, Sinhala, Sinhalese, Slovak, Slovenian, Somali, Spanish, Sundanese, Swahili, Tatar, Telugu, Thai, Tibetan, Turkish, Turkmen, Ukrainian, Urdu, Uzbek, Valencian

Syntax

AIC.Set Language Free|P1

Parameter Explanation

P1: A text string that should correspond to a language. This sets the language that WHISPER will use for ASR.

For example, "English", "Spanish", "French", etc.

Example

```

'*****
' AIC.-Sample
'*****
' Sets the input language to English
AIC.Set Language Free|English

```

Remarks

-

Limitations:

-

See also:

-

3.42.10.2.9 Set Whisper Default

[AIC.Set Whisper Default / AIC.swd](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set Whisper Default / AIC.swd

Resets or sets default parameters for the speech-to-text process.



Intention

The `AIC.Set Whisper Default` command is designed to initialize or reset important parameters.

This ensures that any content in the internal registers does not negatively influence the outcome of the speech-to-text process.

By setting these parameters to their default values, the system can achieve optimal performance and accuracy.

In detail the command will do this:

- Clear or set the **PosPrompt** Register
- Clear or set the **Language** Register
- Clear or set the **Temperature** Register
- Set the **Output Format** to "text"
- Set the **Model** to the default Model

Especially the **PosPrompt** and the **Temperature** Register are used for other A.I.-Models also therefore it makes sense to apply this command before using Whisper.

If you have Specific Words/Acronyms, the Setting P1 to a **Prompt**, can correct specific words or acronyms that the model often misrecognizes.

For instance, if you use this as **P1**: "The transcript is about OpenAI which makes technology like DALL-E, GPT-3, and ChatGPT..."

this will improve the transcription of words like DALL-E and GPT-3.

Prompts and Quality Improvement

You can use a prompt to enhance the quality of the transcripts generated by the Whisper API. The model will attempt to match the style of the prompt, making it more likely to use capitalization and punctuation if the prompt does. However, the current prompting system is more limited than other language models and only provides limited control over the generated audio. Here are some ways prompting can assist:

- **Specific Words/Acronyms:** Prompts can correct specific words or acronyms that the model often misrecognizes. For instance, the prompt "The transcript is about OpenAI which makes technology like DALL-E, GPT-3, and ChatGPT..." can improve the transcription of words like DALL-E and GPT-3.
- **Preserving Context:** To maintain the context of a segmented file, prompt the model with the transcript of the preceding segment. This makes the transcript more accurate, as the model will use the relevant information from the previous audio.
- **Punctuation:** The model might sometimes skip punctuation. This can be rectified by using a prompt that includes punctuation, e.g., "Hello, welcome to my lecture."
- **Filler Words:** The model may omit common filler words. To retain these in your transcript, use a prompt containing them, e.g., "Umm, let me think like, hmm... Okay, here's what I'm, like, thinking."
- **Writing Styles:** Some languages, like Chinese, have different writing styles (simplified or traditional). Use a prompt in your preferred style to guide the model.

Syntax

```
AIC.Set Whisper Default[|P1][|P2][|P3]
AIC.Swd[|P1][|P2][|P3]
```

Parameter Explanation

- **P1 - Optional.** Positive **Prompt Text**. If omitted, the default value is an empty string ("").
- **P2 - Optional.** The **Language (see below)**. If omitted, the default value is an empty string ("").
- **P3 - Optional.** A float parameter representing the **Temperature**. It should be a value between **0.1** and **1**. If omitted, the default value is **0**.

Here is a List of supported languages at current date.

Number	Language
0	Afrikaans
1	Arabic
2	Armenian
3	Azerbaijani
4	Belarusian
5	Bosnian
6	Bulgarian
7	Catalan
8	Chinese
9	Croatian
10	Czech
11	Danish
12	Dutch
13	English
14	Estonian
15	Finnish
16	French
17	Galician
18	German
19	Greek
20	Hebrew
21	Hindi
22	Hungarian
23	Icelandic
24	Indonesian
25	Italian
26	Japanese
27	Kannada
28	Kazakh
29	Korean
30	Latvian
31	Lithuanian
32	Macedonian
33	Malay
34	Marathi
35	Maori
36	Nepali

37	Norwegian
38	Persian
39	Polish
40	Portuguese
41	Romanian
42	Russian
43	Serbian
44	Slovak
45	Slovenian
46	Spanish
47	Swahili
48	Swedish
49	Tagalog
50	Tamil
51	Thai
52	Turkish
53	Ukrainian
54	Urdu
55	Vietnamese
56	Welsh

Example

```

'*****
' AIC.-Sample
'*****
' Most often you can just use the command "plain" without parameters
AIC.Set Whisper Default

' Using Parameters could look like this, and will help the AI to transcribe this
$$PRO=ZyntriQix, Digique Plus, CynapseFive, VortiQore V8, EchoNix Array,
$$PRO+OrbitalLink Seven, DigiFractal Matrix, PULSE, RAPT, B.R.I.C.K., Q.U.A.R.T
AIC.Set Whisper Default|$$PRO
AIC.Set Whisper Default|$$PRO|German
    
```

Remarks

-

Limitations:

-

See also:

-

3.42.10.2.10 Translate to Any

[AIC.Translate to Any](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Translate to Any

Translates the given text to a specified target language using ChatGPT.



Intention

The `AIC.Translate to Any` command is designed to translate a given text in **P1** into a specified target language.

The translation is performed by ChatGPT and the result is stored in an optional variable **P3** or on the Top of Stack (TOS).

You can choose all Languages supported by `AIC.Get Language GPT`.

This function is called internally to choose the language using **P2**.

As Whisper itself can only translate into english, we transcribe into english and then translate using ChatGPT.

Syntax

```
AIC.Translate to Any|P1|P2[|P3]
```

```
AIC.Tta|P1|P2[|P3]
```


Parameter Explanation

- **P1:** The mp3-file with the Audio to be transcribed and translated.
- **P2:** The target language to which the text should be translated.
- **P3:** Optional variable to store the translated text. If omitted, the result is placed on the Top of Stack (TOS).

Example

```
'*****  
' AIC.-Sample  
'*****  
' Translates the mp3 to Spanish and stores it in $$TRA  
AIC.Set Key|file  
$$FIL=?exeloc\Hallo.mp3  
AIC.Translate to Any|$$FIL|Spanish|$$TRA  
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.10.2.11 Translate to English

[AIC.Translate to English](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Translate to English

Translates the given text to a specified target language using ChatGPT.



Intention

The `AIC.Translate to English` command is designed to translate a given text in **P1** into english language.

The translation is performed directly by WHISPER and the result is stored in an optional variable **P2** or on the Top of Stack (TOS).

Syntax

```

AIC.Translate to English|P1 [|
P2]
AIC.Tte|P1 [|P2]

```

Parameter Explanation

- **P1:** The mp3-file with the Audio to be transcribed and translated.
- **P2:** Optional variable to store the translated text. If omitted, the result is placed on the Top of Stack (TOS).

Example

```
'*****  
' AIC.-Sample  
'*****  
' Translates the mp3 to English and stores it in $$TRA  
AIC.Set Key|file  
$$FIL=?exeloc\Hallo.mp3  
AIC.Translate to English|$$FIL|$$TRA  
ENR.
```

Remarks

-

Limitations:

-

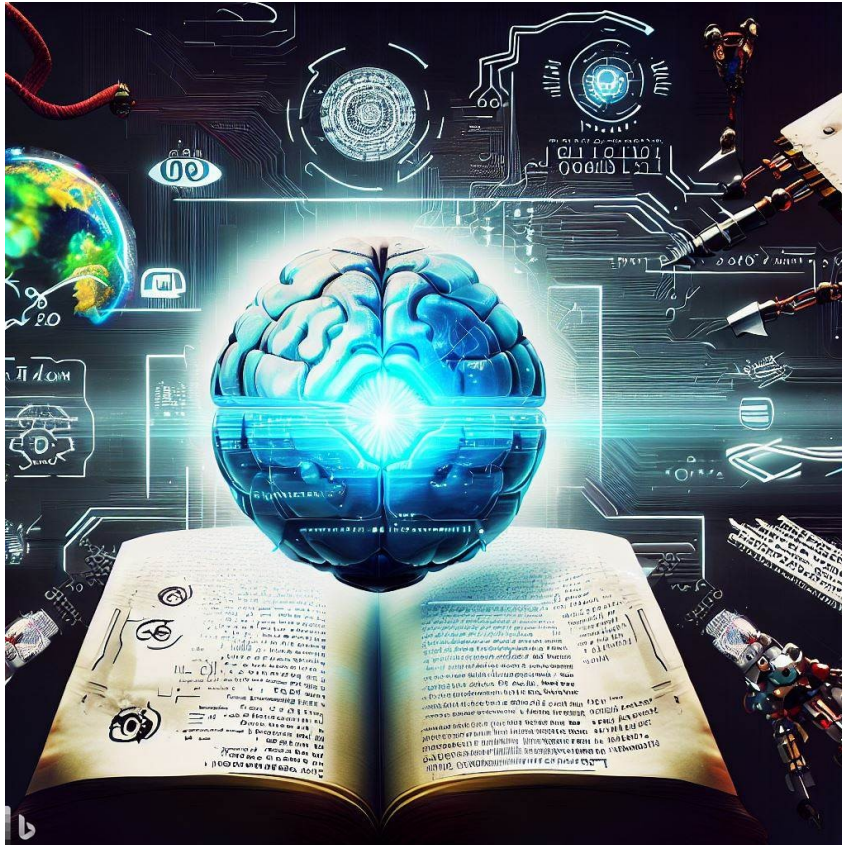
See also:

-

3.42.10.3 ! OpenAI AI-Services

Accessing Artificial Intelligence Services

Welcome to the cutting-edge of Artificial Intelligence services provided by industry leaders such as **OpenAI**.



This is how the AI imagines its own Services ("Open AI").

The SPR, now supercharged with OpenAI's GPT (Generative Pre-trained Transformer) models, has been trained to understand not just code, but also natural language. These GPT models react to their inputs, known as "prompts", and produce textual outputs. Crafting a prompt is akin to guiding your SPR commands, typically by providing instructions or examples to successfully accomplish a task.

With the GPT-powered SPR, you can develop applications that can:

- Draft documents
- Generate computer code
- Respond to queries about a knowledge base
- Analyze texts
- Construct conversational agents
- Equip software with a natural language interface
- Tutor in a variety of subjects
- Translate languages
- Simulate characters for games
- ... and so much more!

To interact with our advanced GPT models, you'll first need to initialize them via the SPR.

This requires setting your API-Key and choosing the model of your choice.

We offer a range of models, starting from GPT-1, our first language model using the Transformer architecture, to our latest GPT-4 model, a significant advancement with a staggering one trillion parameters. Each model has its unique strengths and was built upon the achievements and improvements of its predecessors, allowing for increasingly sophisticated responses and capabilities.

As AI is constantly developed, there will be newer model available and you will be able to use them with the SPR also.

After initializing - you need to set your API-Key and choose a model - , you can send requests and expect a return from the AI. Technically your requests are called "PROMPTS".

In return, you'll receive a response encapsulating the chosen model's output.

Do I need to learn and implement JSON?

I don't think so. Generally the SPR does all the JSON Stuff for you, "under the hood". Of course JSON is there, but its invisible to the User.

There may be few cases where you want to "Escape-"/ or "UnEscape" Strings, for these cases the needed commands are also available within the `AIC.-Command`.

Open AI-Models?

Our newest models, GPT-4 and GPT-3.5-turbo, can be utilized through the chat completions API endpoint.

Using the SPR you will just use the `AIC.Ask_Chat` - Command and not care about further details.

Currently, only the older legacy models are accessible via the completions API endpoint. These are cheaper and available using the `AIC.Ask_Completion` Commands.

You just choose the right command, anything else is done for you inside the SPR, under the hood.

In addition to our AI services, if you're interested in top-tier speech synthesis, Elevenlabs' services are also readily accessible through these commands.

You will find more details below.

To use these AI commands, you will need to obtain a commercial API-Key from OpenAI. This key can be obtained after logging in to your OpenAI account. You can log in and get your API-Key [click Button below].

With these instructions and your API-Key, you have a gateway to harnessing the power of our advanced GPT models. Happy experimenting!

[Get OpenAI API-Key](#)

Unlock the Power of OpenAI: Your API Key Guide

Welcome, intrepid coder! You're about to embark on a thrilling journey through the cosmos of artificial intelligence with OpenAI. But before you launch, you'll need the

golden ticket - your OpenAI API Key! This key is not just a string of characters; it's your passport to a universe of possibilities. Let's get you acquainted with this magical artifact.

What is an OpenAI API Key?

Imagine a mystical key that opens the gates to an enchanted castle. In the world of OpenAI, the API Key is just that! It's a unique combination of letters, numbers, and symbols that authenticates and secures your gallant quests through OpenAI's services.

The Majestic Format

Your API Key is like a spell, and every spell has its incantation. The OpenAI API Key typically begins with a sacred prefix - `sk-` or `pk-`, followed by a series of alphanumeric characters. The prefixes are the guardians; `sk-` stands for Secret Key, while `pk-` stands for Public Key.

Here's what it might look like in all its glory:

```
...
  sk-enchantedforest789magiccastle
...
```

or

```
...
  pk-dragonlayer123wizardkingdom
...
```

Behold! But remember, these are mere illustrations and not actual keys. Your key will be unique, just like a wizard's wand.

Why is it Precious?

This key is your identity in the realm of OpenAI. It whispers to the gates, letting them know you are a worthy traveler. Without it, the doors remain closed. With it, you command the clouds, access mystical functionalities, and conjure powerful AI magic. Guard it with your life!

Safeguarding Your Treasure


Your API Key is akin to a royal seal. In the wrong hands, kingdoms can fall. Keep it secret; keep it safe. Never share it with others, and especially be cautious of where you enter it. It is bound to your account and holds the power to access your resources and data.

To help you safeguard your API-Key the SPR supports you with the [Option to save the API-Key in a Crypted way into a "Keyfile"](#)^[79] that you can use for all your Scripts and even include in your executable. This way the API-Key is never seen in Clear-Text in your Scripts.

Alternatively you could also use the Datamaker and make the API-Key to Inline-Code.

Ready to Embark?

With your API Key in hand, you are ready to set sail across the boundless seas of artificial intelligence. Harness the power, create wonders, and let your imagination be your compass.

Bon voyage, brave explorer! 

API keys

Your secret API keys are listed below. Please note that we do not display your secret key after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. To protect the security of your account, OpenAI may also automatically rotate any API key found to have leaked publicly.

NAME	KEY	CREATED	LAST USED
SPR-Key	sk-...q6Pe	Jun 12, 2023	Jun 12, 2023

[+ Create new secret key](#)

Default organization

If you belong to multiple organizations, this setting controls which organization is used when making requests with the API keys above.

Personal 

Note: You can also specify which organization to use for each API request. See [Authentication](#) to learn more.

Understanding OpenAI Usage Costs

Diving into the world of Artificial Intelligence with OpenAI involves a fascinating behind-the-scenes process.

As you tap into the power of AI, you're essentially interacting with sophisticated Language Learning Models (LLMs). Each interaction is quantified using a measure called "tokens".

Imagine tokens as the fundamental building blocks of language comprehension and generation in AI.

Every word, punctuation mark, or piece of whitespace your AI model processes or produces is counted as tokens.

For instance, the sentence "Hello, world!" would be translated into tokens like this: "Hello" (1 token), "," (1 token), " " (1 token), "world" (1 token), "!" (1 token).

In most cases, a word accounts for about two tokens, but this can vary depending on the complexity and length of the word.

Now, here's where it gets interesting.

The cost of using OpenAI's services is tied directly to the number of these tokens you utilize and the type of LLM you engage with.

The more tokens your requests consume, and the more advanced the LLM you select, the higher the cost. But worry not!

These charges are typically quite modest, and they fund the continuous development and enhancement of these remarkable AI services.

So, every time you command an AI model to draft a poem, translate a sentence, or even answer a trivia question, remember:

you're actually [engaging with a complex dance of tokens](#), all buzzing to create a symphony of AI intelligence.

And that's what makes your journey with OpenAI not just a service, but an experience.

Optimized for dialogue, the performance of ChatGPT models is impressive, costing at this time only \$0.0015 for input and \$0.002 for output per 1,000 tokens.

Other models are designed to follow single-turn instructions.

[See Details about Pricing here \(22-06-2023\):](#)

The cost varies from as low as \$0.0004 to as high as \$0.02 per 1,000 tokens, depending on the model you choose - from Ada, the fastest, to Davinci, the most powerful.

The AI-Models that can be used with the SPR are currently:

```
"gpt-3.5-turbo-0301"  0,002 Cent / 1K Tokens
"text-davinci-003"
"text-davinci-002"
"code-davinci-002"
"text-curie-001"
"text-babbage-001"   0,0005 Cent / 1K Tokens
"text-ada-001"       0,0004 Cent / 1K Tokens
"gpt-4-0314"
```

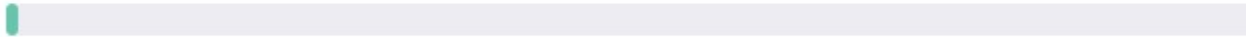
On this page from OpenAI you can track your Usage of these services. You can also set Limits for maximum usage, if desired.

OpenAI Usage

Below you'll find a summary of API usage for your organization. All dates and times are in UTC and data may be delayed up to 5 minutes.



Usage this month



Daily usage breakdown (UTC)

Select a day All org members

Important News from July, 6

Source: [Open AI Blog](#)

Starting January 4, 2024, [older completion models](#) will no longer be available and will be replaced with the following models:

Older model	New model
ada	ada-002
babbage	babbage-002
curie	curie-002
davinci	davinci-002

```
davinci-instruct-beta
curie-instruct-beta
text-ada-001
text-babbage-001
text-curie-001
text-davinci-001
text-davinci-002
text-davinci-003
gpt-3.5-turbo-instruct
```

Applications using the stable model names for base GPT-3 models (`ada`, `babbage`, `curie`, `davinci`) will automatically be upgraded to the new models listed above on January 4, 2024. The new models will also be accessible in the coming weeks for early testing by specifying the following model names in API calls: `ada-002`, `babbage-002`, `curie-002`, `davinci-002`.

Developers using other older completion models (such as `text-davinci-003`) will need to manually upgrade their integration by January 4, 2024 by specifying `gpt-3.5-turbo-instruct` in the "model" parameter of their API requests. `gpt-3.5-turbo-instruct` is an InstructGPT-style model, trained similarly to `text-davinci-003`. This new model is a drop-in replacement in the Completions API and will be available in the coming weeks for early testing.

Developers wishing to continue using their fine-tuned models beyond January 4, 2024 will need to fine-tune replacements atop the new base GPT-3 models (`ada-002`, `babbage-002`, `curie-002`, `davinci-002`), or newer models (`gpt-3.5-turbo`, `gpt-4`). Once this feature is available later this year, we will give priority access to GPT-3.5 Turbo and GPT-4 fine-tuning to users who previously fine-tuned older models. We acknowledge that migrating off of models that are fine-tuned on your own data is challenging. We will be providing support to users who previously fine-tuned models to make this transition as smooth as possible.

In the coming weeks, we will reach out to developers who have recently used these older models, and will provide more information once the new completion models are ready for early testing.

Deprecation of the Edits API

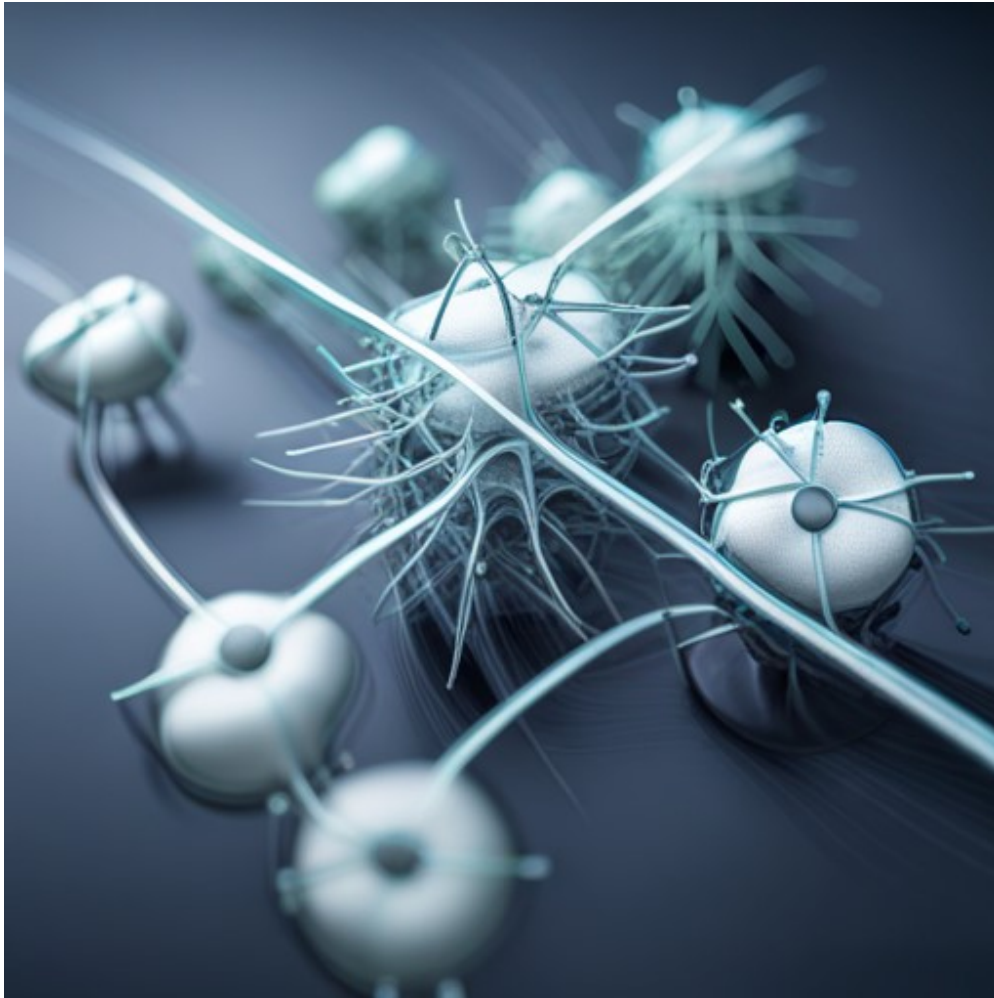
Users of the Edits API and its associated models (e.g., `text-davinci-edit-001` or `code-davinci-edit-001`) will need to migrate to GPT-3.5 Turbo by January 4, 2024. The Edits API beta was an early exploratory API, meant to enable developers to return an edited version of the prompt based on instructions.

We took the feedback from the Edits API into account when developing `gpt-3.5-turbo` and the Chat Completions API, which can now be used for the same purpose.

3.42.10.4 AI - Direct Access

The Commands of this Section give you direct Control over the results that are returned from the Open AI-API.

In the OpenAI API, when you make a request to one of its endpoints, the response you receive is structured in a format known as JSON (JavaScript Object Notation). JSON is a widely-used data interchange format that is both human-readable and easy for machines to parse and generate.



Generated using Stable Diffusion "Neural Network".

Here's a brief overview of JSON:

- JSON is composed of key-value pairs.
- Each key is a string, and the value can be a string, number, boolean, null, array, or another JSON object.
- JSON objects are enclosed in curly braces {}.
- Arrays are ordered lists and are enclosed in square brackets [].
- Key-value pairs within objects are separated by commas.

Now, let's talk about the OpenAI API response:*

The OpenAI API typically returns a JSON structure that contains various pieces of information in response to your request. For instance, if you are using the completion endpoint, the response might contain generated text, model information, and other metadata.

Here is an example of a JSON structure that you might receive from the completion endpoint, it contains two alternative results:

```
{
  "id": "cmpl-uqkvlQyYK7bGYrRHQ0eXlWi7",
  "object": "text_completion",
  "created": 1589478378,
  "model": "text-davinci-003",
  "choices": [
    {
      "text": "\n\nThis is the first Choice",
      "index": 0,
      "logprobs": null,
      "finish_reason": "length"
    }
    {
      "text": "\n\nThis is the second choice",
      "index": 1,
      "logprobs": null,
      "finish_reason": "length"
    }
  ],
  "usage": {
    "prompt_tokens": 5,
    "completion_tokens": 7,
    "total_tokens": 12
  }
}
```

The **AIC. Command** also provides direct access commands which allow you to directly access specific fields within the JSON structure. For example, you can directly access the **text**, **role** and **content** fields if they are part of the response.

For this you can just call the command

```
AIC.Get Role|<index>|$$RET
```

and you get the role-output of the last call directly in the SPR-Variable. Ready to use.

Hereby, the **AIC.-Commands** takes care of **escaping** and **unescaping** the JSON structure for you.

Why is Escaping Necessary?

JSON data is structured with specific characters such as curly braces { }, square brackets [], double quotes ", and colons :.

However, the actual data you are working with might also contain these characters. For example, if you have a string that contains double quotes, it could be misinterpreted as the end of a string in the JSON structure.

Escaping is the process of converting characters that have special meaning in JSON syntax into a format that can be included in a JSON string without breaking the structure.

For example, the double quote character " is escaped as \". This tells the JSON parser that this character is part of the data and should not be treated as a control character.

Why is Unescaping Necessary?

Unescaping is the reverse process of escaping.

When you retrieve data from a JSON structure, you want it in its original form, not in the escaped form. Unescaping converts the escaped characters back into their original form.

Automatic Escaping and Unescaping in AIC Commands

In the context of using the OpenAI API with AIC (AI Commands), the escaping and unescaping processes are handled automatically.

This means that you don't have to worry about manually replacing special characters with their escaped equivalents when sending data, or converting them back when retrieving data.

This automation is highly beneficial as it:

- **Simplifies the Code:** You don't have to write additional code for escaping and unescaping, making your code cleaner and easier to maintain.
- **Prevents Errors:** Automatically handling escaping and unescaping helps in preventing syntax errors in the JSON structure which could be caused by special characters.
- **Saves Time:** It saves development time as you can focus on the core functionality rather than the intricacies of data formatting.

By handling these processes automatically, the AIC commands ensure a smooth and error-free interaction with the JSON data returned by the OpenAI API.

*More detailed Info see here: [Open AI API Reference](#)

3.42.10.4.1 Escaped String

AIC.Escaped String[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Escaped String

Escape a text string for use within JSON structures



Intention

The `AIC.Escaped String` command, also accessible through its alias `AIC.ESC`, is used to escape a text string for use within JSON structures. JSON structures require certain characters to be escaped, such as double quotes, backslashes, and control characters. This command makes sure the input text conforms to JSON string encoding standards.

Example Usage:

```

$$TXT="Hallo /n Peter"
AIC.Escaped String|$$TXT|$$RET
PRT.$$RET
ENR.

```

```

[17:39:33] \"Hallo /n Peter\"

```

Syntax

AIC.Escaped String|P1 [|P2]
AIC.ESC|P1 [|P2]

Parameter Explanation

P1 - Variable or text, this is the text that should be "escaped" for use with JSON Structures.

P2 - opt. Variable for the result, if omitted the result is been placed on TOS.

Example

```
! *****  
' EXAMPLE: AIC.-Commands  
! *****  
-
```

Remarks

Make sure that the input text is properly formatted before using this command, especially if it already contains escaped characters.

Limitations:

-

See also:

- [Set_Key](#)⁷⁹⁹
- [Ask_Chat](#)⁸³³
- [Ask_Completion](#)⁸⁴²

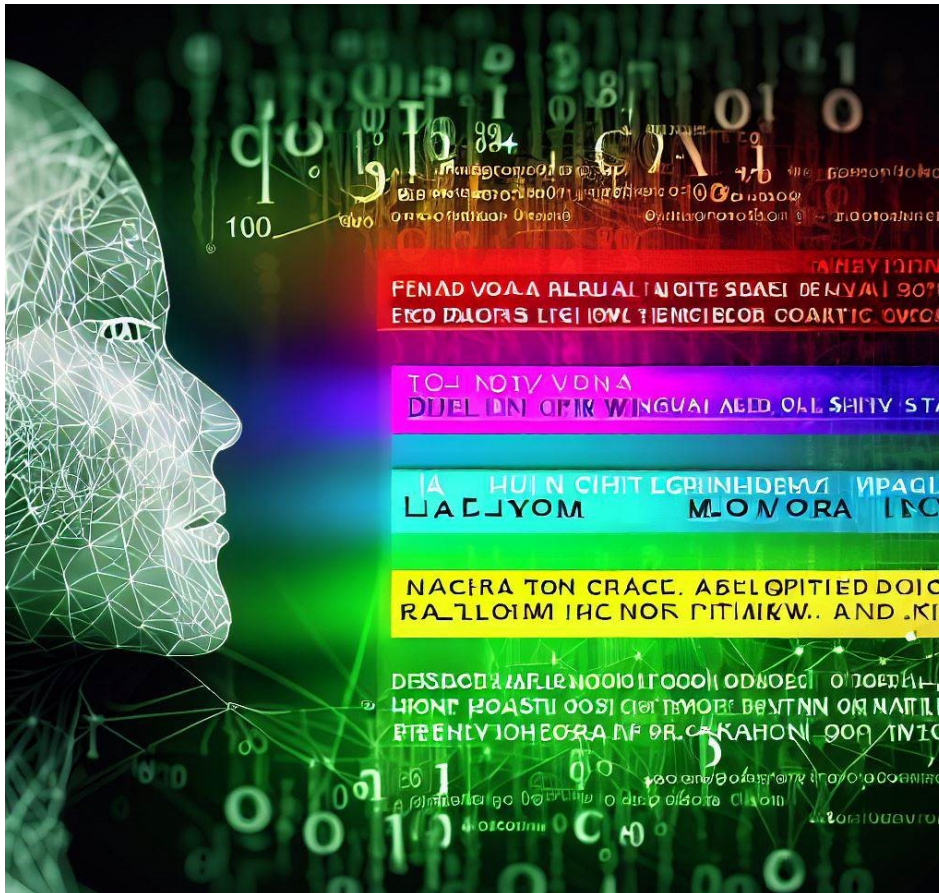
3.42.10.4.2 Estimate_Token_Count

[AIC.Estimate Token Count](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Estimate Token Count

Get the estimated Amount of Tokens of a Text or String.



Intention

The `AIC.Estimate Token Count` command will deliver a rough approximation "how many Tokens" a Text String has.

This might be useful to check if a Text or String "will fit" into a model's Token-Maximum.

OpenAI's GPT models, like GPT-3, generally use a variant of the Byte Pair Encoding (BPE) tokenization.

The number of BPE (Byte Pair Encoding) pairs in the original vocabulary of models like GPT-2 and GPT-3 is in the order of tens of thousands.

For instance, GPT-2 uses a vocabulary size of 50,257 tokens. GPT-3, being a larger and more advanced model, has a similar vocabulary size.

Therefore the `AIC.Estimate Token Count` Command will just do a statistical driven estimation of the Token-Count, depending on the appearance of spaces and characters in the Text.

Syntax:

```
AIC.Estimate Token Count|<Text>|<Variable for Result>
```

Parameters:

<Number>: An integer value representing the number of outputs you want to generate. This number should be greater than or equal to 1.

Note that setting a very high number may have cost implications and might be subject to rate limits imposed by the OpenAI API.

Example Usage:

```
$$TXT=This is a Text to be tokenized and counted
AIC.Estimate Token Count|$$TXT|$$NUM
MBX.The ETC of the Text is: $$NUM
```

This example will output the estimated Tokencount of the given Text. Generally the estimated Tokencount is between 2.0 to- 4.5 characters per Token.

Syntax

```
AIC.Estimate Token Count|P1 [|
P2]
AIC.etc|P1 [|P2]
```

Parameter Explanation

P1 - Variable or Text to evaluate the estimated Token Count.

P2 - opt. Variable for the result. If omitted, TOS is used..

Example

```
'*****
' EXAMPLE 1: AIC.-Commands
'*****
-
```

Remarks

OpenAI's GPT models, like GPT-3, generally use a variant of the Byte Pair Encoding (BPE) tokenization.

The number of BPE (Byte Pair Encoding) pairs in the original vocabulary of models like GPT-2 and GPT-3 is in the order of tens of thousands.

For instance, GPT-2 uses a vocabulary size of 50,257 tokens. GPT-3, being a larger and more advanced model, has a similar vocabulary size.

It's important to note that these tokens are not just BPE pairs, but also include individual characters, common words, and subwords. The BPE algorithm is used to construct this vocabulary by iteratively merging frequent pairs of characters or subwords.

The vocabulary is an essential part of the model, and it is constructed during the pre-training phase.

When tokenizing text for input to the model, the tokenizer uses this pre-established vocabulary to convert the text into a sequence of token IDs that the model can process.

To exactly replicate the tokenization used by GPT models, one would need to use the same vocabulary and tokenization algorithm as used in the pre-training of these models.

As it would need a large Token Dataset with more than 50.000 Tokenpairs, we do just a statistical driven Token-Count estimation with the SPR.

Limitations:

-

See also:

- [Set Key](#)  799
- [Ask Chat](#)  833
- [Ask Completion](#)  842

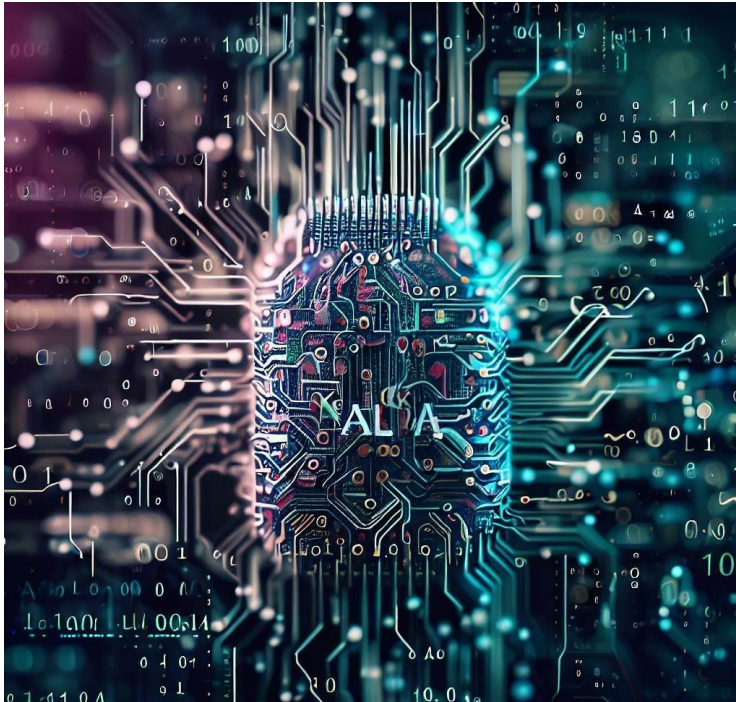
3.42.10.4.3 Get Content

[AIC.Get Content](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Get Content

Get the Text Result of the Ask-Chat Command.



Intention

Some Commands will return their Output in a Variable* called "Text", in that case you will get the Output using the "AIC.Get Text"-Command.

The Open AI "Chat-Endpoint" = the "AIC.Ask Chat" Command, returns its Output in a variable called "Content", therefore you need to use "AIC.Get Content"

to get the Output-Text. In total, the Output from the "AIC.Ask Chat" Command will include these fields (these are Array starting with the Index of "0"):

- **Choices** (returns a number how many answers are returned, normally equal the number you have set using "AIC.Set Number")
- **Content** - this is the important Part of the Answer.
- **Role**
- **Finish** - this is the "**Finish Reason**". It will tell you if the answer finished due to "Stop" which is Ok, or due to "insufficient number of available Tokens".
In the later case you should increase the Number in "AIC.Set

MaxToken".

In case you have set "AIC.Set Number" to any number higher than "1", you will receive multiple Answers from the AI. They may be equal (in Text-Generation, if Temperature is high) or not (in Image Generation mostly not).

You can use the

```
AIC.Get Several|5|$$NUM
```

Command to find out how many Answers have been delivered by the AI.

You can specify which of these multiple Answers you want to have returned by specifying the Index of the Answer after the Command.

To keep things equal with the underlying JSON Structure, the Index-Number of the first Answer is zero (you can leave it empty or write "0").

Syntax:

```
AIC.Get Content|<Index Number>|<Variable for Result>
```

Parameters:

<Number>: An integer value starting at zero, valid to the Value that has been specified using "AIC.Set Number" Minus 1 (because we start with "0").

For example if you have specified to get just 1 Answer - which is the normal case - then the only valid number is "0".

If you are waiting for 2 Answers from the AI, the possible numbers are "0" and "1". This number representing the Array Index of the Output you want to get.

<Variable for Result>: This is a placeholder for the variable where you want to store the content retrieved by the "AIC.Get Content" command.

This variable will hold the data from the specified index in the output array. If omitted, TOS is used.

Example Usage:

```
AIC.Set Number|5
AIC.Get Content|4|$$RET
```

This example we have set the number of Results/Answers we want from the AI to 5. So we get 5 Answers, their Index-Numbers range from "0" to "4". Here we retrieve the last answer which has Index 4.

Syntax

AIC.Get Content[|P1][|P2]

AIC.gtc[|P1][|P2]

Parameter Explanation

P1 - opt. Variable or numeric value, this is the number of the alternative results, you want to be returned.

P2 - opt. Variable for result. If omitted TOS is used.

*Here is a Sample of the original returned JSON Data from the Chat-Endpoint.

```
"id": "chatcmpl-7Zbc78I0ULOEdP1RfHTYPzRxsZSYZ",
"object": "chat.completion",
"created": 1688720424,
"model": "gpt-3.5-turbo-0613",
"choices": [
  {
    "index": 0,
    "message": {
      "role": "assistant",
      "content": "\"Imagine a world where technology has advanced to the point"
    },
    "finish_reason": "length"
  }
],
"usage": {
  "prompt_tokens": 20,
  "completion_tokens": 30,
  "total_tokens": 50
}
}
```

Example

```
'*****
' EXAMPLE 1: AIC.-Commands
'*****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' Set Model
AIC.SetModel_Chat|1

' Set Model-Temperature
AIC.Set_Temperature|0

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000
' The more Tokens you use the more you need to pay.
AIC.SetMax_Token|25

' We will create 4 alternative answers
AIC.Set Number|4

' Ask Question and receive answer to $$RET
```

```
AIC.Ask_Chat|What is a "Windows Button"?|$$RET

' Make a Loop over the number of results
FOR.$$LOP|1|4
  AIC.gtc|.$$LOP|$$RET
  DBP.Result No.$$LOP
  DBP.$$RET
NEX.
MBX.$$RET

:enx
ENR.
```

Remarks

Limitations:

-

See also:

- [Set Key](#)⁷⁹⁹
- [Ask Chat](#)⁸³³
- [Ask Completion](#)⁸⁴²

3.42.10.4.4 Get Finish

AIC.Get Finish

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Get Finish

Get the Finish Reason of the last Open AI-Command.

```
"finish_reason": "stop"  
"finish_reason": "length" }
```

Intention

The "AIC.Get Finish" command returns the "finish_reason" variable*, which specifies the reason that caused the AI to cease generating a response. Typically, this occurs when the AI has successfully completed and formulated the answer.

The "stop" and "length" are two possible values for the "Finish_reason" parameter, which indicates why the AI stopped generating the response. Let's evaluate these two and also discuss other possible values for "Finish_reason":

- **stop**: This is the normal case, where the AI has successfully processed your prompt and **generated a complete response**. It indicates that the AI has reached a logical conclusion or endpoint in the response.
- **length**: This reason indicates that the *AI could not complete the response* because it **reached the maximum token limit** set by the user through the "AIC.Set MaxToken" command or by default system constraints. Tokens are chunks of text, and there is a limit to how many tokens can be in a single response. If the response is cut off due to reaching this limit, the "Finish_reason" will be "**length**".
- **timeout**: This is another possible value for "Finish_reason". It indicates that the AI stopped generating the response because it reached a **time limit**. This could be due to system constraints or a user-defined timeout setting. It's used to prevent the AI from taking too long to generate a response.
- **interrupted**: This reason would indicate that the generation process was interrupted by an external factor, such as a user manually stopping the generation or a system error.
- **threshold**: In some cases, the AI might stop generating text if the **confidence level falls below a certain threshold**. This is to ensure that the generated text meets a **certain quality standard**.

Understanding the "**Finish_reason**" is important as it provides insights into why the AI stopped generating the response and can help in troubleshooting or optimizing the settings for text generation.

Some Commands will return their Output in a Variable called "Text", in that case you will get the Output using the "AIC .Get Text"-Command.

The Open AI "Chat-Endpoint" = the "AIC .Ask Chat" Command, returns its Output in a variable called "Content", therefore you need to use "AIC .Get Content"

to get the Output-Text. In total, the Output from the "AIC .Ask Chat" Command will include these fields (these are Array starting with the Index of "0"):

- **Choices** (returns a number how many answers are returned, normally equal the number you have set using "AIC .Set Number")
- **Content** - this is the important Part of the Answer.
- **Role**
- **Finish** - this is the "**Finish Reason**". It will tell you if the answer finished due to "Stop" which is Ok, or due to "insufficient number of available Tokens".
In the later case you should increase the Number in "AIC .Set MaxToken".

In case you have set "AIC .Set Number" to any number higher then "1", you will receive multiple Answers from the AI. They may be equal (in Text-Generation, if Temperature is high) or not (in Image Generation mostly not).

You can use the

```
AIC.Get Several|5|$$NUM
```

Command to find out how many Answers have been delivered by the AI.

You can specify which of these multiple Answers you want to have returned by specifying the Index of the Answer after the Command.

To keep things equal with the underlying JSON Structure, the Index-Number of the first Answer is zero (you can leave it empty or write "0").

Syntax:

```
AIC.Get Finish|<Index Number>|<Variable for Result>
```

Parameters:

<Number>: An integer value starting at zero, valid to the Value that has been specified using "AIC .Set Number" Minus 1 (because we start with "0"). For example if you have specified to get just 1 Answer - which is the normal case - then the only valid number is "0".

If you are waiting for 2 Answers from the AI, the possible numbers are "0" and "1". This number representing the Array Index of the Output you want to get.

<Variable for Result>: This is a placeholder for the variable where you want to store the content retrieved by the "AIC .Get Content" command.

This variable will hold the data from the specified index in the output array. If omitted, TOS is used.

Example Usage:

```
AIC.Set Number|5
AIC.Get Content|4|$$RET
```

This example we have set the number of Results/Answers we want from the AI to 5. So we get 5 Answers, their Index-Numbers range from "0" to "4". Here we retrieve the last answer which has Index 4.

Syntax

```
AIC.Get Finish[|P1][|P2]
AIC.gtf[|P1][|P2]
```

Parameter Explanation

P1 - opt. Variable or numeric value, this is the number of the alternative results, you want to be returned.

P2 - opt. Variable for result. If omitted TOS is used.

*Here is a Sample of the original returned JSON Data from the Chat-Endpoint.

```
{
  "id": "chatcmp1-7Zbc78I0ULOEdP1RfHTYPzRxsZSYZ",
  "object": "chat.completion",
  "created": 1688720424,
  "model": "gpt-3.5-turbo-0613",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "\"Imagine a world where technology has advanced to the point"
      },
      "finish_reason": "length"
    }
  ],
  "usage": {
    "prompt_tokens": 20,
    "completion_tokens": 30,
    "total_tokens": 50
  }
}
```

Example

```

'*****
' EXAMPLE 1: AIC.-Commands
'*****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' Set Model
AIC.SetModel_Chat|1

' Set Model-Temperature
AIC.Set_Temperature|0

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000)
' The more Tokens you use the more you need to pay.
AIC.SetMax_Token|25

' We will create 4 alternative answers
AIC.Set Number|4

' Ask Question and receive answer to $$RET
AIC.Ask_Chat|What is a "Windows Button"?|$$RET

' Make a Loop over the number of results
FOR.$$LOP|1|4
  AIC.gtc|.$$LOP|$$RET
  DBP.Result No.$$LOP
  DBP.$$RET
  AIC.gtf|.$$LOP|$$RET
  DBP.Finish Reason fort Result No.$$LOP
  DBP.$$RET
NEX.
MBX.!
```

:enx
ENR.

Remarks

Limitations:

-

See also:

- [Set_Key](#)⁷⁹⁹
- [Ask_Chat](#)⁸³³
- [Ask_Completion](#)⁸⁴²

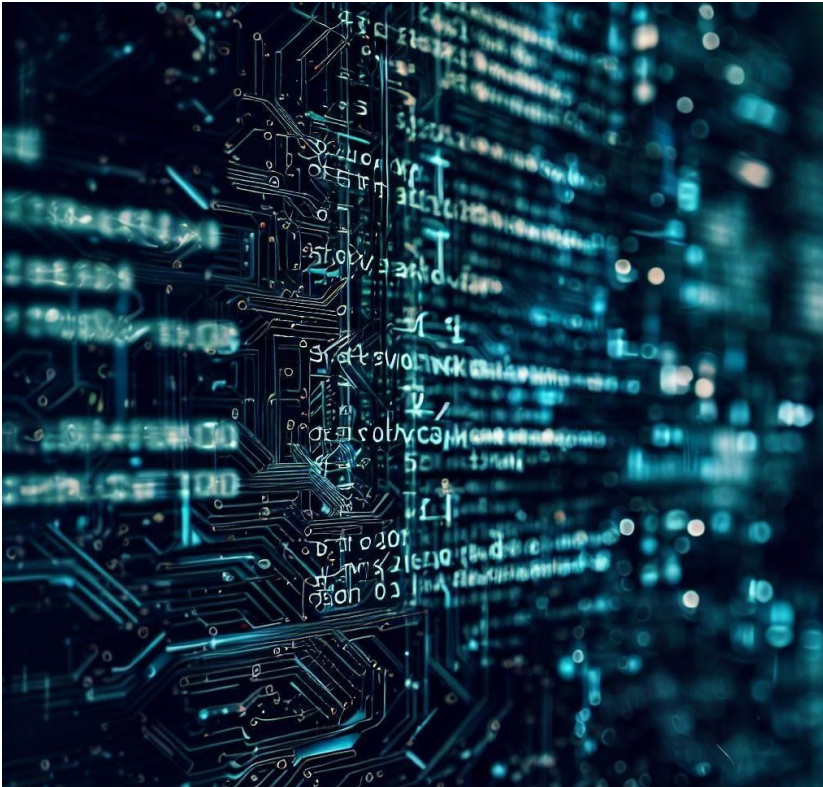
3.42.10.4.5 Get Raw Output

[AIC.Get Raw Output](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Get Raw Output

Get the original JSON Structure that was generated by the AI.



Intention

The `AIC.Get Raw Output` command in the Smart-Package Robot (SPR) allows users to get access to **the original JSON* Data Structure** that is returned from the **OpenAI endpoints** in answer to a Prompt.

This command is versatile and can be used across all OpenAI endpoints, including:

- text generation,
- image generation,
- and more.

This command is mostly useful for debugging.

The "`AIC.gro`" Command does not need any Index, as the result it will contain all Answers that are generated.

Important: if you use this command after "Image Generation", there may be a lot (up to 5 MB) of "Base64 encoded" image data in the result.

In that case you can not use a "Messagebox" to display this amount of data.

Syntax:

```
AIC.Get Raw Output|<Variable for Result>
```

Parameters:

<Variable for Result>: This is a placeholder for the variable where you want to store the content retrieved by the "AIC.Get Content" command.

This variable will hold the data from the specified index in the output array. If omitted, TOS is used.

Example Usage:

```
AIC.Get Raw Output|$$RET
```

Syntax

AIC.Get Raw Output[|P1]

AIC.gro[|P1]

Parameter Explanation

P1 - opt. Variable for result. If omitted TOS is used.

*Here is a Sample of the original returned JSON Data from the Chat-Endpoint.

```
{
  "id": "chatcmpl-7Zbc78I0ULOEdP1RfHTYPzRxsZSYZ",
  "object": "chat.completion",
  "created": 1688720424,
  "model": "gpt-3.5-turbo-0613",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "\"Imagine a world where technology has advanced to the point"
      },
      "finish_reason": "length"
    }
  ],
  "usage": {
    "prompt_tokens": 20,
    "completion_tokens": 30,
    "total_tokens": 50
  }
}
```

Example

```

'*****
' EXAMPLE 1: AIC.-Commands
'*****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' Set Model
AIC.SetModel_Completion|1

' Set Model-Temperature
AIC.Set_Temperature|0

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000)
' The more Tokens you use the more you need to pay.
AIC.SetMax_Token|25

' Ask Question and receive answer to $$RET
AIC.Ask_Completion|What is a "Windows Button"?|$$RET
' We will look at the RAW Output here
AIC.Get Raw Output|$$RET
DBP.$$RET
MBX.!!
:enx
ENR.

```

Remarks

Important: if you use this command after "Image Generation", there may be a lot (up to 5 MB) of "Base64 encoded" image data in the result. In that case you can not use a "Messagebox" to display this amount of data. You can use the PRT . Command for that.

Limitations:

-

See also:

- [Set_Key](#)⁷⁹⁹
- [Ask_Chat](#)⁸³³
- [Ask_Completion](#)⁸⁴²

3.42.10.4.6 Get Role

[AIC.Get Role](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Get Role

Get the Text Result of the Ask-Chat Command.



Intention

In the input and output JSON structure of the chat endpoint from OpenAI, the "role" field specifies the role of the message sender.

This plays a role with the "AIC.Ask Chat" Command, because it returns some of its Output in a variable called "Role".

Therefore you need to use "AIC.Get Role" to get this information.

The possible values for the "role" field are:

- **"system"**: This value indicates that the message is a system message. **System** messages are usually used to set the behavior of the assistant at the beginning of the conversation.
For example, a system message might instruct the assistant to speak like Shakespeare.

- **"user"**: This value indicates that the message is from the user. **User** messages are the prompts or questions that the user wants the assistant to respond to.
- **"assistant"**: This value indicates that the message is from the assistant. **Assistant** messages are the responses generated by the AI in reply to the user's prompts or questions.

The **"role"** field is important as it helps in distinguishing between different types of messages in the conversation and understanding the flow of the conversation.

You can set the Role of an "AIC.Ask Chat" Command using the "AIC.Set Role"-Command.

You can use the

```
AIC.Get Several|5|$$NUM
```

Command to find out how many Answers have been delivered by the AI.

Important Note: The "AIC.Set Role" command and the "AIC.Get Role" command operate on different registers; each command has its own dedicated register.

The "AIC.Set Role" command is utilized during a request to set a specific role. On the other hand, the "AIC.Get Role" command is used to retrieve the role that has been returned from the OpenAI server.

This distinction is crucial as the "AIC.Set Role" command is for specifying a role before sending a request, while the "AIC.Get Role" command is for obtaining the role information from the response received from OpenAI.

Syntax:

```
AIC.Get Role|<Index Number>|<Variable for Result>
```

Parameters:

<Number>: An integer value starting at zero, valid to the Value that has been specified using "AIC.Set Number" Minus 1 (because we start with "0"). For example if you have specified to get just 1 Answer - which is the normal case - then the only valid number is "0".

If you are waiting for 2 Answers from the AI, the possible numbers are "0" and "1". This number representing the Array Index of the Output you want to get.

<Variable for Result>: This is a placeholder for the variable where you want to store the content retrieved by the "AIC.Get Role" command.

This variable will hold the data from the specified index in the output array. If omitted, TOS is used.

Example Usage:

```
AIC.Set Number|5
AIC.Get Role|4|$$RET
```

This example we have set the number of Results/Answers we want from the AI to 5. So we get 5 Answers, their Index-Numbers range from "0" to "4". Here we retrieve the role from the last answer which has Index 4.

Syntax

```
AIC.Get Role [|P1] [|P2]
AIC.gtc [|P1] [|P2]
```

Parameter Explanation

P1 - opt. Variable or numeric value, this is the number of the alternative results, you want to be returned.

P2 - opt. Variable for result. If omitted TOS is used.

*Here is a Sample of the original returned JSON Data from the Chat-Endpoint.

```
{
  "id": "chatcmpl-7Zbc78I0ULOEdP1RfHTYPzRxsZSYZ",
  "object": "chat.completion",
  "created": 1688720424,
  "model": "gpt-3.5-turbo-0613",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "\"Imagine a world where technology has advanced to the point"
      },
      "finish_reason": "length"
    }
  ],
  "usage": {
    "prompt_tokens": 20,
    "completion_tokens": 30,
    "total_tokens": 50
  }
}
```

Example

```
'*****
' EXAMPLE 1: AIC.-Commands
'*****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' Set Model
AIC.SetModel_Chat|1
```

```

' Set Model-Temperature
AIC.Set_Temperature|0

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000)
' The more Tokens you use the more you need to pay.
AIC.SetMax_Token|25

' We will create 4 alternative answers
AIC.Set Number|4

' Ask Question and receive answer to $$RET
AIC.Ask_Chat|What is a "Windows Button"?|$$RET

' Make a Loop over the number of results
FOR.$$LOP|1|4
  AIC.gtc|.$$LOP|$$RET
  DBP.Result No.$$LOP
  DBP.$$RET
NEX.
MBX.$$RET

:enx
ENR.

```

Remarks

Important Note: The "AIC.Set Role" command and the "AIC.Get Role" command operate on different registers; each command has its own dedicated register.

The "AIC.Set Role" command is utilized during a request to set a specific role. On the other hand, the "AIC.Get Role" command is used to retrieve the role that has been returned from the OpenAI server.

This distinction is crucial as the "AIC.Set Role" command is for specifying a role before sending a request, while the "AIC.Get Role" command is for obtaining the role information from the response received from OpenAI.

Limitations:

-

See also:

- [Set Key](#)⁷⁹⁹
- [Ask Chat](#)⁸³³
- [Ask Completion](#)⁸⁴²

3.42.10.4.7 Get Several

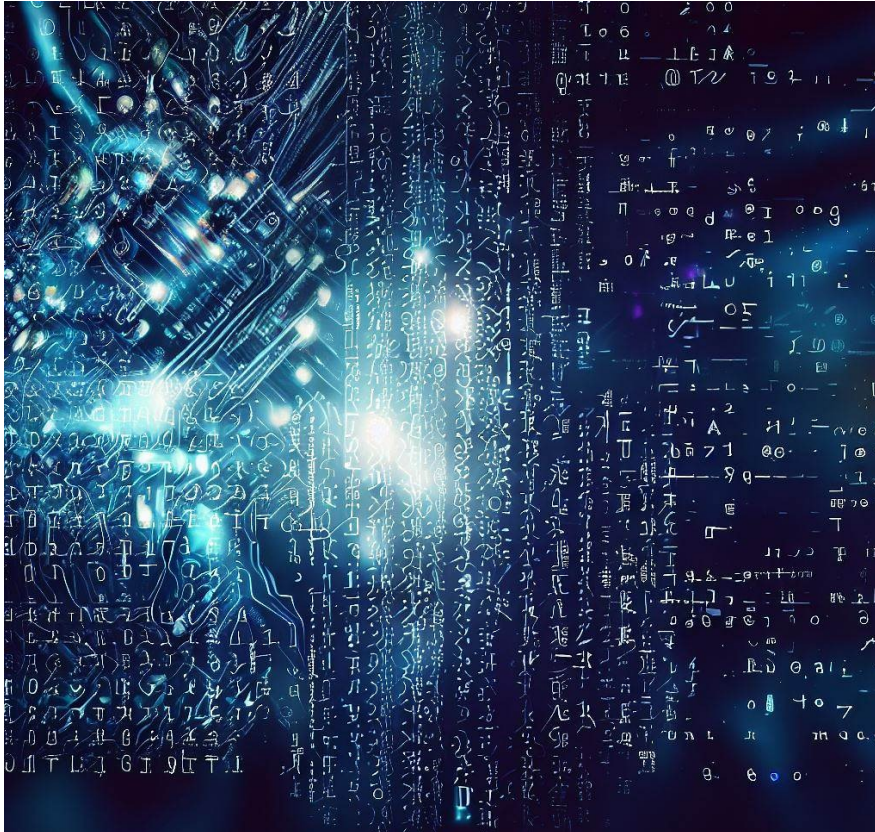
AIC.Get Several

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Get Several

Set the Amount of alternative results you want to have generated.



This command gives you access to many important "inner values" of the AI Engines.

Intention

The `AIC.Get Several` command in the Smart-Package Robot (SPR) allows users to get several internal AI-Values. For this the first Parameter P1 is a number that defines, which value will be returned.

- 0 - `AI_Last_A` - The last Answer from the AI. In case of multiple Answers, this is only the last Answer.
- 1 - `AI_Last_Q` -The last Prompt sent to the AI.
- 2 - `AI_All` - The complete history of Answers and Questions.
- 3 - `AI_Return` - This is the Result of the last Operation, in RAW Format. In case of problems, this will often contain something useful.
- 4 - `AIO_HTTP_Status` This is the Result of the last `http://` Operation, in RAW Format. In case of problems, this will often contain something useful.
- 5 - `AIO_Model` This will return the effectively used Model that has been used in the last Operation.

- 6 - `AIO_Choices` This will return the number of choices returned from the last AI-Operation no matter if Open AI or GPT4All..
- 7 - `Infostring` This will return the InfoString Output from Stable Diffusion.
- 8 - `AI_Model` This is the Model for ChatGPT.
- 9 - `AI_SBody` This is the last JSON Structure that was sent as Chat-Input
- 10 - `AIO_Error` This is the last Error that was registered inside the System.
- 11 - `NegPrompt` This is the internal Register for the negative Prompt
- 12 - `PosPrompt` This is the internal Register for the Positive Prompt
- 13 - `AIW_Text` This is a Textual Return.
- 14 - `AIO_ID` This is a Return from the Vision Model. Looks like this: "chatcmpl-8IwIipyemyAFUuCVg
- 15 - `AIO_Object` This is a Return from the Vision Model. Looks like this: "chat.completion".
- 16 - `AI_Vision_Model` This is the current used Model for GPT4-Vision
- 17 - `AI_TTS_Model` This is is the currently used OpenAI Text to Speech Model.
- 18 - `Vision_Endpoint` This is is the currently used OpenAI Vision Endpoint
- 19 - `TTS_Endpoint` This is is the currently used OpenAI TTS-Endpoint
- 20 - `AI_TTS_Voice` This is is the currently used OpenAI TTS-Voice
- 21 - `AI_TTS_Format` This is is the currently used OpenAI TTS-Output Format

Changes:

- 08 was: `AI_CNT` This is the Counter for the number of History Entries. Moved to `AIC.Get Any Value`

Syntax:

```
AIC.Get Several|<Number>|<Variable for Result>
```

Example Usage:

```
AIC.Get Several|5|$$MOD
```

This example gets the name of the Model that was effectively used during the last AI Operation.

Syntax

```
AIC.Get Several|P1 [|P2]
AIC.GTS|P1 [|P2]
```

Parameter Explanation

P1 - <Number>: An integer from the values see Table above.

P2 - opt. <Variable for Result>: A variable that will receive the result of the Operation. If omitted TOS is used.

Example

```

'*****
' EXAMPLE 1: AIC.-Commands
'*****
$$LOG=?exeloc\Output.txt
DEL.$$LOG
$$WOA="GTO.JMP.JNJ.JNF.GSB.JSR.JIV.JIZ.JNZ.PRR.JLE.JME.JRR.JIT.JNT.JIS.JIE.VBS.J
    ' "GSB.GTO.JAC.JFP.JIE.JIO.JIS.JIT.JIV.JIZ.JLE.JME.JMP.JNC.JNF.JNJ.JNO.JNP.J
GSC.$$WOA|.|$$ANZ
GSB.Write_Log|SPR counted: $$ANZ Elements.$CrLf$-----$

FOR.$$NUM|1|7
    AIC.Set MaxToken|1024
    GSB.Lab_SetModel
    AIC.SetModel|$$MOD
    AIC.Set Number|2
    AIC.Set Temperatur|1

    GSB.Write_Log|Model: $$MOD
    $$TXT=Act as my Assistant.
    $$TXT Below is a Line with a a number of words that are each 3 letters ending
    $$TXT Tell me how many words are in there.$CrLf$
    $$TXT Do not generate code. Do not repeat yourself. Use english language.
    $$TXT Sort these words alphabetically step by step, and show me the result.$
    $$TXT $$WOA
    IVV.$$NUM>6
        GSB.Call_GPT
    ELS.
        AIC.Ask GPT4All|$$TXT|$$RET
    EIF.
    GSB.Write_Log|$$RET
    AIC.Get Several|5|$$RAW
    GSB.Write_Log|Used Model: $$RAW
    DBP.-----
NEX.

ENR.
'-----
:Lab_SetModel
SCS.$$NUM
CAN.1
    $$MOD=Wizard Uncensored
CAN.2
    $$MOD=Hermes
CAN.3
    $$MOD=Snoozy
CAN.4
    $$MOD=Replit
CAN.5
    $$MOD=Nous Vicuna
CAN.6
    $$MOD=Groovy
'CAN.7

```

```

'  $$MOD=ChatGPT-3.5 Turbo
'CAN.8
'  $$MOD=ChatGPT-4
CAE.
  $$MOD=$$MOD
ESC.
RET.
'-----
:Write_Log
VAV.$$OUT=$$ _01$crLf$
ATF.$$LOG| $$OUT
DBP.$$OUT
RET.
'-----
' We do this separate as the other way sometime4s the GUI seems to crash.
'-----
:Call_GPT
$$MOD=gpt-3.5-turbo-0613
AIC.SetKey|File
AIC.SetModel_Chat|1
AIC.Ask_Chat| $$TXT| $$RET
RET.
'-----
ENR.

```

Remarks

-

Limitations:

-

See also:

- [Set Key](#)⁷⁹⁹
- [Ask_Chat](#)⁸³³
- [Ask_Completion](#)⁸⁴²

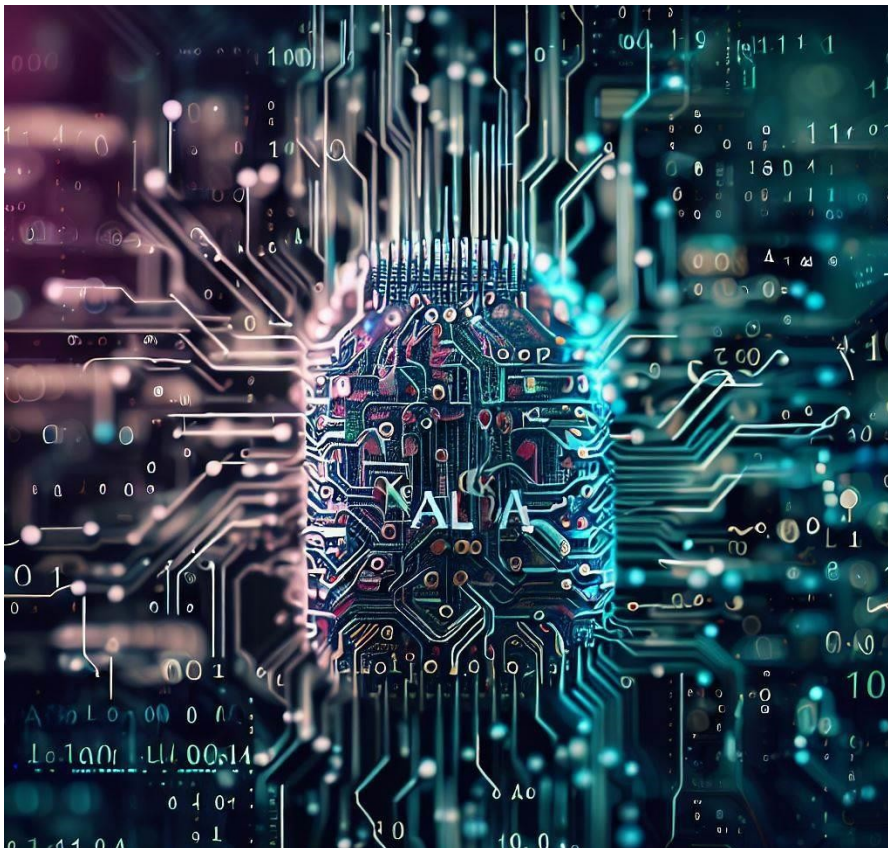
3.42.10.4.8 Get Text

[AIC.Get Content](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Get Text

Get the Text Result of the Ask-Completion, Ask_GPT4All and other Commands.



Intention

Some Commands will return their Output in a Variable(Array) called "Text", in that case you will get the Output using the "AIC.Get Text"-Command.

In case you have set "AIC.Set Number" to any number higher then "1", you will receive multiple Answers from the AI. They may be equal (in Text-Generation, if Temperature is high) or not (in Image Generation mostly not).

You can specify which of these multiple Answers you want to have returned by specifying the Index of the Answer after the Command. To keep things equal with the underlying JSON Structure, the Index-Number of the first Answer is zero (you can leave it empty or write "0").

In case you specify


```
AIC.Set Number|4
```

then you can use "AIC.gtx" Command to retrieve the Text with the Index 0 to 3. Because Index starts at "0".

You can use the

```
AIC.Get Several|5|$$NUM
```

Command to find out how many Answers have been delivered by the AI.

Syntax:

```
AIC.Get Text|<Index Number>|<Variable for Result>
```

Parameters:

<Number>: An integer value starting at zero, valid to the Value that has been specified using "AIC.Set Number" Minus 1 (because we start with "0"). For example if you have specified to get just 1 Answer - which is the normal case - then the only valid number is "0".

If you are waiting for 2 Answers from the AI, the possible numbers are "0" and "1". This number representing the Array Index of the Output you want to get.

<Variable for Result>: This is a placeholder for the variable where you want to store the content retrieved by the "AIC.Get Content" command.

This variable will hold the data from the specified index in the output array. If omitted, TOS is used.

Example Usage:

```
AIC.Set Number|5
AIC.Get Text|4|$$RET
```

This example we have set the number of Results/Answers we want from the AI to 5. So we get 5 Answers, their Index-Numbers range from "0" to "4". Here we retrieve the last answer which has Index 4.

Syntax

```
AIC.Get Text [|P1] [|P2]
```

```
AIC.gtx [|P1] [|P2]
```

Parameter Explanation

P1 - opt. Variable or numeric value, this is the number of the alternative results, you want to be returned.

P2 - opt. Variable for result. If omitted TOS is used.

*Here is a Sample of the original returned JSON Data from the Chat-Endpoint.

```

{id": "chatcmpl-7Zbc78I0ULOEdP1RfHTYPzRxsZSYZ",
"object": "chat.completion",
"created": 1688720424,
"model": "gpt-3.5-turbo-0613",
"choices": [
  {
    "index": 0,
    "message": {
      "role": "assistant",
      "content": "\"Imagine a world where technology has advanced to the point",
    },
    "finish_reason": "length"
  }
],
"usage": {
  "prompt_tokens": 20,
  "completion_tokens": 30,
  "total_tokens": 50
}
}

```

Example

```

'*****
' EXAMPLE 1: AIC.-Commands
'*****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' Set Model
AIC.SetModel_Completion|2

' Set Model-Temperature
AIC.Set_Temperature|0.5

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000 ?
' The more Tokens you use the more you need to pay. But the longer Input and Out
AIC.SetMax_Token|300

' Ask Question and receive answer to $$RET
$$QUE=Tell me what is X=3+5?

AIC.Set Role|User
' We ask for 3 results.

AIC.Set Number|4

AIC.Ask_Completion|$$QUE|$$REA
' The Command will return all results in one Variable
DBP.$$REA
DBP.+++++
DBP.Here we get the results separated:
FOR.$$LOP|0|3
AIC.gtx|$$LOP|$$RET

```

```

DBP.-----
DBP.Result No.$$LOP
DBP.$$RET
DBP.-----
NEX.
:enx
ENR.

```

Result (removed some Linefeeds):

```

[Answer:000] X=3+5 is 8.
[Answer:001] X=3+5 is 8.
[Answer:002] X=3+5 is 8.
[Answer:003] X=3+5 is 8.

[14:32:15] +++++
[14:32:15] Here we get the results separated:
[14:32:15] -----
[14:32:15] Result No.0
[14:32:15] X=3+5 is 8.
[14:32:15] -----
[14:32:15] -----
[14:32:15] Result No.1
[14:32:15] X=3+5 is 8.
[14:32:15] -----
[14:32:15] -----
[14:32:15] Result No.2
[14:32:15] X=3+5 is 8.
[14:32:15] -----
[14:32:15] -----
[14:32:16] Result No.3
[14:32:16] X=3+5 is 8.
[14:32:16] -----

```

Remarks

-

Limitations:

-

See also:

- [Set Key](#)⁷⁹⁹
- [Ask Chat](#)⁸³³
- [Ask Completion](#)⁸⁴²

3.42.10.4.9 GetAnyValue

[Get Any Value](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

Get Any Value

Retrieves various internal values related to AI operations

Intention

The `\AIC.Get Any Value` command retrieves various internal values related to AI operations, providing insights into different aspects of AI processing and limitations. This command is intended to offer users access to internal metrics and limits within the AI system. It is useful for monitoring and adapting AI-related operations based on these values.

Syntax

AIC.Get Any Value|P1 [|P2]

Parameter Explanation

- `\P1` is a number corresponding to the specific value to retrieve (see the list below).
 - `\P2` (**optional**) is a variable where the retrieved value will be stored.
- - `\0`: `\AIO_Choices` - Number of choices returned from the last AI operation.
 - - `\1`: `\TotalTokens` - Total tokens used in the last AI operation.
 - - `\2`: `\PromptTokens` - Number of tokens used in the prompt of the last AI operation.
 - - `\3`: `\CompletionTokens` - Number of tokens used in the completion of the last AI operation.
 - - `\4`: `\AIO_Created` - Timestamp of the last AI operation's completion.
 - - `\5`: `\AIC_MaxToken_Vision` - Maximum tokens allowed in the vision model.
 - - `\6`: `\AI_MaxToken` - Maximum tokens allowed in the current AI model.
 - - `\7`: `\AI_MaxToken_GPTAll` - Maximum tokens for GPT-All operations.
 - - `\8`: `\AI_Cnt` - Counter for the number of history entries.

Example

```

' *****
' AIC.-Sample
' *****
' Example SPR-Script to get the number of choices from the last AI operation
AIC.Get Any Value|0|$$Num

' Output the retrieved value
DBP.$$Num

```

ENR.

Remarks

-

Limitations:

-

See also:

- [AIC.Get_Several](#)  683

3.42.10.4.10 UnEscaped String

[AIC.UnEscaped String](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.UnEscaped String

UnEscape a text string for use within JSON structures



Intention

The `AIC.UnEscaped String` command, also accessible through its alias `AIC.USC`, is used to unescape a text string that was in use within JSON structures. JSON structures require certain characters to be escaped, such as double quotes, backslashes, and control characters. This command makes sure the text is converted back to original encoding.

Example Usage:

```
$$TXT=\"Hallo /n Peter\"  
AIC.UnEscaped String|$$TXT|$$RET  
DBP.$$RET  
ENR.
```

```
[17:44:51] "Hallo /n Peter"
```

Syntax

AIC.UnEscaped String|P1 [|P2] AIC.USC|P1 [|P2]

Parameter Explanation

P1 - Variable or text, this is the text that should be "unescaped" from earlier use with JSON Structures.

P2 - opt. Variable for the result, if omitted the result is been placed on TOS.

Example

```
' *****  
' EXAMPLE: AIC.-Commands  
' *****  
-
```

Remarks

Make sure that the input text is properly formatted before using this command, especially if it already contains escaped characters.

Limitations:

-

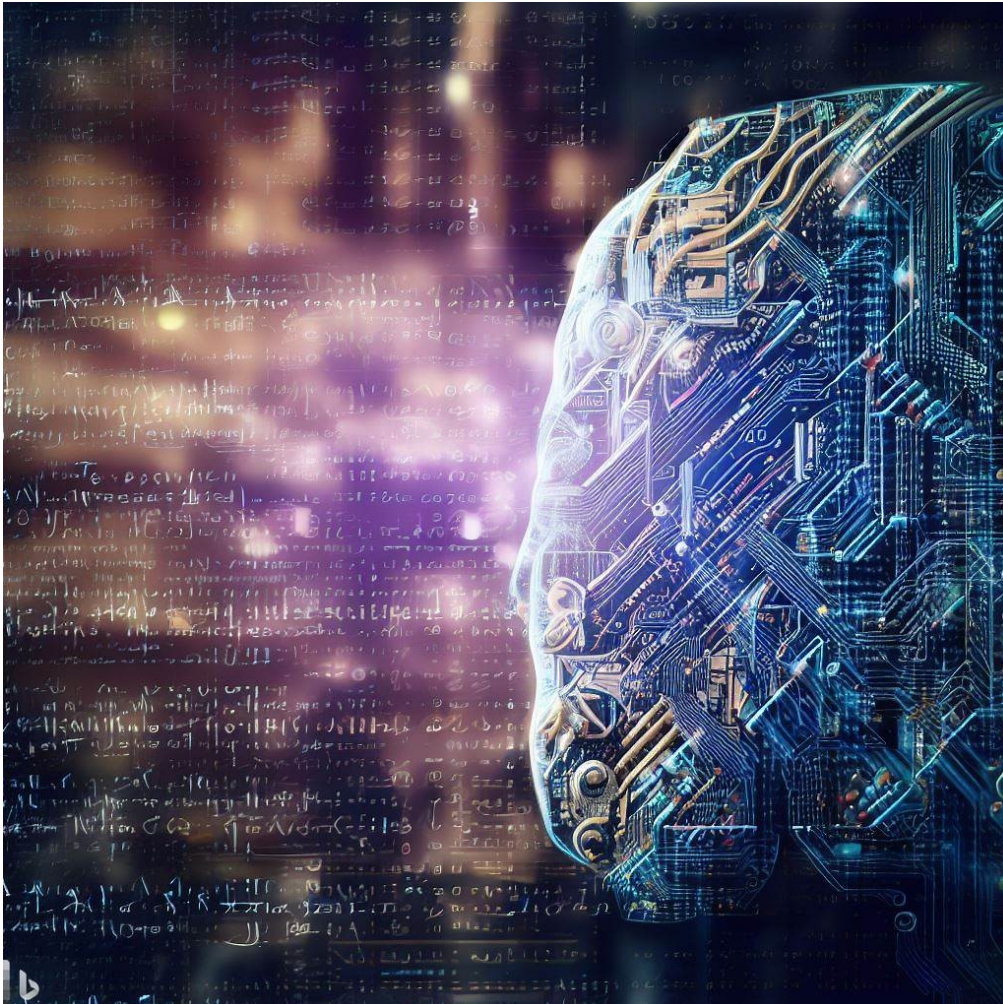
See also:

- [Set Key](#)⁷⁹⁹
- [Ask Chat](#)⁸³³
- [Ask Completion](#)⁸⁴²

3.42.10.5 AI - History

Managing the Past

Whatever AI said in the Past - get it back using the Commands in this Section.



The SPR-AI History will be kept as long as the SPR is running.

Unless you use the
AIC.Reset

Which will clear all history data.

Also whenever you restart the SPR you will also start a new history.

Currently the SPR keeps **two different version of a history**.

We have the

- "One big Text" - History
- Indexed Array History

Both will be used with different commands.

There is one command that will serve both histories:

AIC.Add to History

this command will add Question and Answer to both histories.

3.42.10.5.1 Add to History

[AIC.Add to History](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Add to History

Add custom Prompt and Answer to internal History.



Intention

The `AIC.Add to History` command will add a custom Prompt and a custom Answer to the internally stored History, that you can retrieve using the `AIC.Get Several|2` or the `AIC.Get History - Command`. It will automatically add the delimiter that is defined with `AIC.Set Delimiter QS` and `AIC.Set Delimiter AN` to the History.

It will also add a Count and a Date to each Block of Question and Answer.

This Command works together with the `AIC.Get History Command` and the `AIC.Set Delimiter QS` and `AIC.Set Delimiter AN` Commands.

You can get the history of the chat, and the last Question, or the last Answer using the Commands:

```
AIC.Get_History|$$HIS
AIC.Get_Last_Question|$$QUE
AIC.Get_Last_Answer|$$ANS
```

Syntax:

```
AIC.Add to History|<Question>|<Anwer>
```

Example Usage:

```
$$QUS=This is just a divider.
$$ANS=-----
AIC.Add to History|$$QUS|$$ANS
```

This example adds a custom Block to the History. This can be useful to mark Blocks as dividers. Or add comments to the history.

Syntax

```
AIC.Add to History[|P1][|P2]
AIC.ATH[|P1][|P2]
```

Parameter Explanation

P1 - opt. Variable or Text representing the Prompt (Question).

P2 - opt. Variable or or Text representing the Answer.

If omitted the default value is "-".

Example

```
'*****
' EXAMPLE 1: AIC.-Commands
'*****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' Set Model
AIC.SetModel_Completion|4

' Set Model-Temperature
AIC.Set_Temperature|0

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000
' The more Tokens you use the more you need to pay.
AIC.SetMax_Token|25

AIC.Add to History|The next block shows the use of step by step prompting|-----
```

```
' Ask Question and receive answer to $$RET  
AIC.Ask_Completion|What is a "Windows Button"|$$RET  
MBX.$$RET
```

```
:enx  
ENR.
```

Remarks

-

Limitations:

Currently the history has a hard-cap of the last ~200 kb. This is not a limitation but just to prevent it from memory overflow.

See also:

-

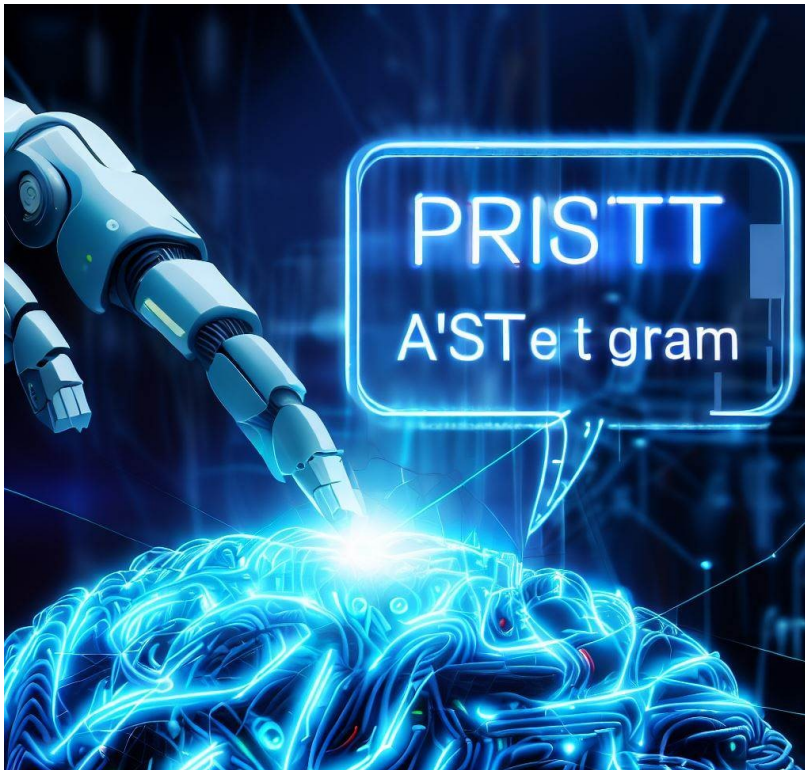
3.42.10.5.2 Get History Text

[AIC.Get History](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Get History Text

Get the complete stored AI-History as one Text.



Intention

The `AIC.Get History` command enables users to retrieve the full record of the most recent question and its corresponding answers exchanged with any AI model. This includes all text-based inputs and outputs sent to and received from the AI. Please note that outputs from "Generate Image" models, which provide images in Base64 encoded format, are not included in the history. This is due to the substantial data size of these images, which could make the history difficult to read. The `AIC.Get History` command is versatile and can be used with any AI model, regardless of whether it operates locally or online. The history integrates the text-based interactions from all AI models, creating a comprehensive record of text inputs and outputs across different models.

Syntax:

```
AIC.Get History Text|<Variable for Return>
```

Example Usage:

```
AIC.Get History|$$RET
MBX.$$RET
```

This example Recalls the last Prompt send to any AI-Model..

```
AIL.Set Number|1
AIL.Set Temperature|$$LR0
AIL.Set Model|Hermes
$$TXT=Hello! Please tell me what 3+4 is.
AIL.Ask GPT4All|$$TXT|$$REA
```

```
$$TXT=Hello! Please tell me what 5+6 is.
AIL.Ask GPT4All|$$TXT|$$REA
DBP.-----
AIC.ght|$$REB
DBP.The complete history was:
DBP.$$REB
DBP.-----
ENR.
```

Debug Window

```
[10:06:12] -----
[10:06:12] The complete history was:
[10:06:12]
<AI-Count: 1 at: 08.07.2023.10:06:07>
Hello! Please tell me what 3+4 is.
```

The result of the addition of 3 and 4 is 7.

```
<AI-Count: 2 at: 08.07.2023.10:06:12>
Hello! Please tell me what 5+6 is.
```

```
The result of 5 + 6 is 11.
[10:06:12] -----
```

Syntax

```
AIC.Get History[|P1]
AIC.GHI [|P1]
```

Parameter Explanation

P1 - opt. Variable for the result. If omitted TOS i used.

Example

```

'*****
' EXAMPLE 1: AIC.-Commands
'*****
VAN.$$TIM=#dtime#
AIL.SetModel|$$MOD
AIL.Set MaxToken|1024
AIC.Set Number|1
FOR.$$LR0|0|1|0.1
    AIL.Set Temperature|$$LR0
    AIL.Set Model|Hermes

    $$TXT>Hello! I have a list of items represented as a string, and I would like
    $$TXT=$$TXT "ISP.NSP.WSP.GSP.SSP."
    $$TXT=$$TXT Please follow these steps to sort the items:
    $$TXT=$$TXT Split the string into individual items based on the period ('.')
    $$TXT=$$TXT Sort the resulting array of items in alphabetical order.
    $$TXT=$$TXT Join the sorted items back into a single string, using the period
    $$TXT=$$TXT Please provide me with the sorted string as the output. DO not gen

    AIC.Estimate Token Count|$$TXT|$$TOA
    AIL.Ask GPT4All|$$TXT|$$REA
    CAL.$$TIU=#dsince#|i
    AIC.Estimate Token Count|$$REA|$$TOB
    DBP.-----
    VAR.$$OUT=$$REA $CrLf$Temp. $$LR0 $CrLf$ Time used: $$TIU sec.$CrLf$Tokens in
    DBP.$$OUT
    DBP.-----
    AIC.ghi|$$REB
    DBP.The complete history was:
    DBP.$$REB
    NEX.
    ENR.

```

Remarks

-

Limitations:

Currently the history has a hard-cap of the last 200 kb.This is not a limitation but just to prevent it from memory overflow.

See also:

- [Set Key](#)⁷⁹⁹
- [Ask Chat](#)⁸³³
- [Ask Completion](#)⁸⁴²

3.42.10.5.3 Get Last Answer

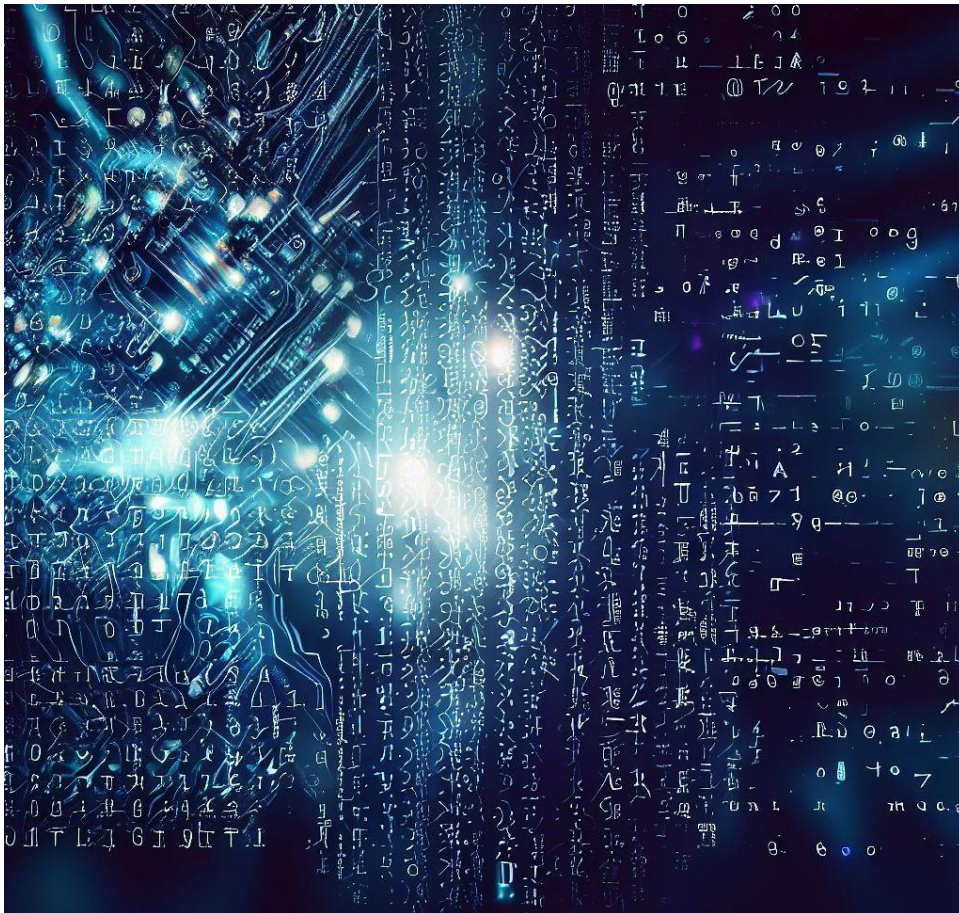
AIC.Get Last Answer

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Get Last Answer

Get the last used Answer.



Get the last Answer that was sent from the AI.

Intention

The AIC.Get Last Answer command in the Smart-Package Robot (SPR) allows users recall the last Question that was send to any AI-Model. This command can be used for any AI Model whether its Local or Online..

Syntax:

```
AIC.Get Last Answer|<Variable for Return>
```

Example Usage:

```
AIC.Get Last Answer|$$RET  
MBX.$$RET
```

This example Recalls the last Prompt send to any AI-Model..

```

AIL.Set Number|1
AIL.Set Temperature|$$LR0
AIL.Set Model|Hermes

$$TXT=Hello! Please tell me what 5+6 is.
AIL.Ask GPT4All|$$TXT|$$REA
DBP.-----
DBP.$$REA
DBP.-----
AIC.gla|$$REB
DBP.The last Answer was:
DBP.$$REB
DBP.-----
ENR.

```

Syntax

AIC.Get Last Answer[|P1]
AIC.GLA[|P1]

Parameter Explanation

P1 - opt. Variable for the result. If omitted TOS i used.

Example

```

'*****
' EXAMPLE 1: AIC.-Commands
'*****
VAN.$$TIM=#dtime#
AIL.SetModel|$$MOD
AIL.Set MaxToken|1024
AIC.Set Number|1
FOR.$$LR0|0|1|0.1
  AIL.Set Temperature|$$LR0
  AIL.Set Model|Hermes

```

```

$$TXT=Hello! I have a list of items represented as a string, and I would like
$$TXT=$$TXT "ISP.NSP.WSP.GSP.SSP."
$$TXT=$$TXT Please follow these steps to sort the items:
$$TXT=$$TXT Split the string into individual items based on the period ('.')
$$TXT=$$TXT Sort the resulting array of items in alphabetical order.
$$TXT=$$TXT Join the sorted items back into a single string, using the period
$$TXT=$$TXT Please provide me with the sorted string as the output. DO not gen

```

```
AIC.Estimate Token Count|$$TXT|$$TOA
AIL.Ask GPT4All|$$TXT|$$REA
CAL.$$TIU=#dsince#|i
AIC.Estimate Token Count|$$REA|$$TOB
DBP.-----
VAR.$$OUT=$$REA $crlf$Temp. $$LR0 $crlf$ Time used: $$TIU sec.$crlf$Tokens in
DBP.$$OUT
DBP.-----
AIC.gla|$$REB
DBP.The last Amswer was:
DBP.$$REB
NEX.
ENR.
```

Remarks

-

Limitations:

-

See also:

- [Set Key](#)⁷⁹⁹
- [Ask Chat](#)⁸³³
- [Ask Completion](#)⁸⁴²

3.42.10.5.4 Get Last Question

[AIC.Get Last Question](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Get Last Question

Get the last used Prompt.



Intention

The `AIC.Get Last Question` command in the Smart-Package Robot (SPR) allows users recall the last Question that was send to any AI-Model. This command can be used for any AI Model whether its Local or Online..

Syntax:

```
AIC.Get Last Question|<Variable for Return>
```

Example Usage:

```
AIC.Get Last Question|$$RET  
MBX.$$RET
```

This example Recalls the last Prompt send to any AI-Model..

```

AIL.Set Number|1
AIL.Set Temperature|$$LR0
AIL.Set Model|Hermes

$$TXT=Hello! Please tell me what 5+6 is.
AIL.Ask GPT4A11|$$TXT|$$REA
DBP.-----
DBP.$$REA
DBP.-----
AIC.g1q|$$REB
DBP.The last Prompt was:
DBP.$$REB
DBP.-----
ENR.

```

Syntax

AIC.Get Last Question[|P1] AIC.GLQ[|P1]

Parameter Explanation

P1 - opt. Variable for the result. If omitted TOS i used.

Example

```

'*****
' EXAMPLE 1: AIC.-Commands
'*****
VAN.$$TIM=#dtime#
AIL.SetModel|$$MOD
AIL.Set MaxToken|1024
AIC.Set Number|1
FOR.$$LR0|0|1|0.1
  AIL.Set Temperature|$$LR0
  AIL.Set Model|Hermes

  $$TXT=Hello! I have a list of items represented as a string, and I would like
  $$TXT=$$TXT "ISP.NSP.WSP.GSP.SSP."
  $$TXT=$$TXT Please follow these steps to sort the items:
  $$TXT=$$TXT Split the string into individual items based on the period ('.') o
  $$TXT=$$TXT Sort the resulting array of items in alphabetical order.
  $$TXT=$$TXT Join the sorted items back into a single string, using the period
  $$TXT=$$TXT Please provide me with the sorted string as the output. DO not gen

  AIC.Estimate Token Count|$$TXT|$$TOA
  AIL.Ask GPT4A11|$$TXT|$$REA

```

```
CAL.$$TIU=#dsince#|i
AIC.Estimate Token Count|$$REA|$$TOB
DBP.-----
VAR.$$OUT=$$REA $crlf$Temp. $$LR0 $crlf$ Time used: $$TIU sec.$crlf$Tokens in
DBP.$$OUT
DBP.-----
AIC.glq|$$REB
DBP.The last Prompt was:
DBP.$$REB
NEX.
ENR.
```

Remarks

-

Limitations:

-

See also:

- [Set Key](#)⁷⁹⁹
- [Ask Chat](#)⁸³³
- [Ask Completion](#)⁸⁴²

3.42.10.5.5 Reset

AIC.Reset

[Previous](#) 2027 | [Top](#) 49 | [Next](#) 2069

MiniRobotLanguage (MRL)

AIC.Reset

Clear internal History-Logs

Intention

Generally all Input and Output that is sent or received from AI (Exception: Image Generation) is kept in the "History" in two Logs. If you want to start fresh and reset these History Logs, then this is the command. The AIC.Reset command will reset the Internal History and delete all Entries.

Syntax:

```
AIC.Reset
```

Example Usage:

```
AIC.Reset
ENR.
```

Syntax

AIC.Reset
AIC.res

Parameter Explanation

-

Example

```
'*****
' EXAMPLE 1: AIC.-Commands
'*****

ENR.
```

Remarks

-

Limitations:

-

See also:

- [Set_Key](#)⁷⁹⁹
- [Ask_Chat](#)⁸³³
- [Ask_Completion](#)⁸⁴²

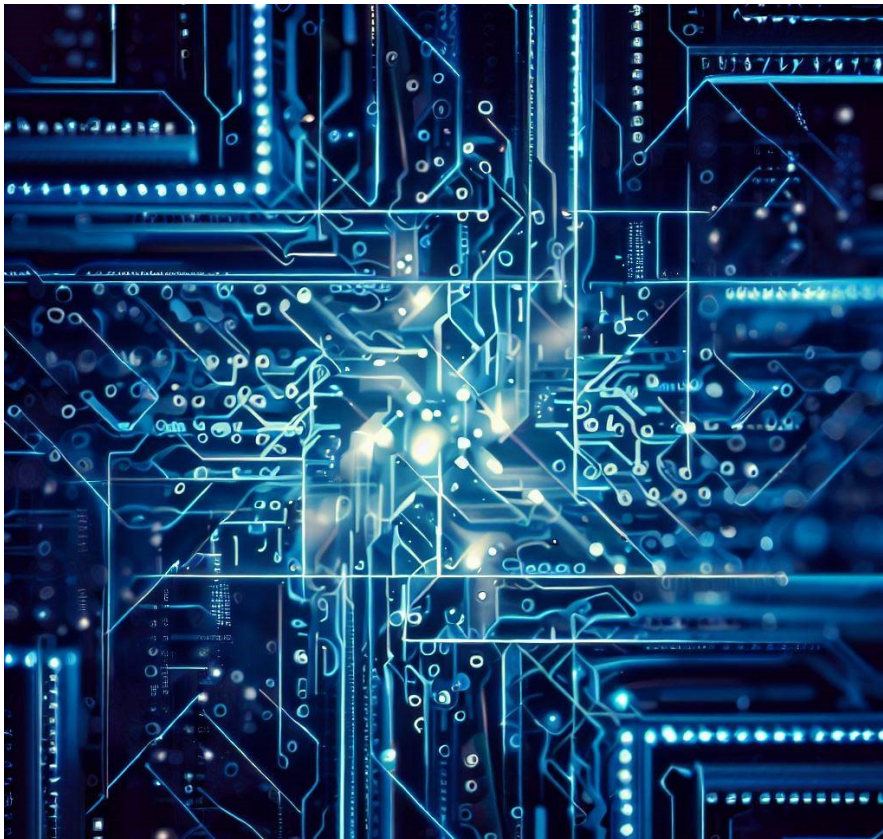
3.42.10.5.6 Set Delimiter AN

[AIC.Set Delimiter AN](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set Delimiter AN

Set the delimiters for the reply of the AI in the history.



Intention

The `AIC.Set Delimiter AN` command in the Smart-Package Robot (SPR) allows users to set the delimiters that are used in the Text-History. This can make it easier to parse the history.

Syntax:

```
AIC.Set Delimiter AN|<delimiter start>|<delimiter end>
```

Parameters:

`<delimiter start>`: Any Text or string that makes it easy for you to parse the History.

`<delimiter end>`: Any Text or string that makes it easy for you to parse the History.

Example Usage:

```
AIC.Set Delimiter AN|<Answer_Start:>|<Answer_End:>
AIC.Set Delimiter QS|<Prompt_Start:>|<Prompt_End:>
ENR.
```

This example will enclose the "Answers" between <Answer_Start:> and <Answer_End:>

The Prompts will be enclosed between <Prompt_Start:> and <Prompt_End:>

in the History that you can get using the "AIC.Get History" Command.

Syntax

```
AIC.Set Delimiter AN[|P1][|P2]
AIC.SDA[|P1][|P2]
```

Parameter Explanation

P1 - opt. Variable with the delimiter for the "Start of the Answer". If omitted, the value is been set to \$CrLf\$.

P2 - opt. Variable with the delimiter for the "End of the Answer". If omitted, the value is been set to \$CrLf\$.

Example

```
'*****
' EXAMPLE 1: AIC.-Commands
'*****
AIC.Set Delimiter AN|<Antwort_Start:>|<Antwort_End:>
AIC.Set Delimiter QS|<Prompt_Start:>|<Prompt_End:>
AIL.Set Number|1
AIL.Set Temperature|$$LR0
AIL.Set Model|Hermes
$$TXT=Hello! Please tell me what 3+4 is.
AIL.Ask GPT4A11|$$TXT|$$REA

$$TXT=Hello! Please tell me what 5+6 is.
AIL.Ask GPT4A11|$$TXT|$$REA
DBP.-----
AIC.ghi|$$REB
DBP.The complete history was:
DBP.$$REB
DBP.-----
ENR.
```

Remarks

-

Limitations:

-

See also:

- [Set_Key](#)⁷⁹⁹
- [Ask_Chat](#)⁸³³
- [Ask_Completion](#)⁸⁴²

3.42.10.5.7 Set Delimiter QS

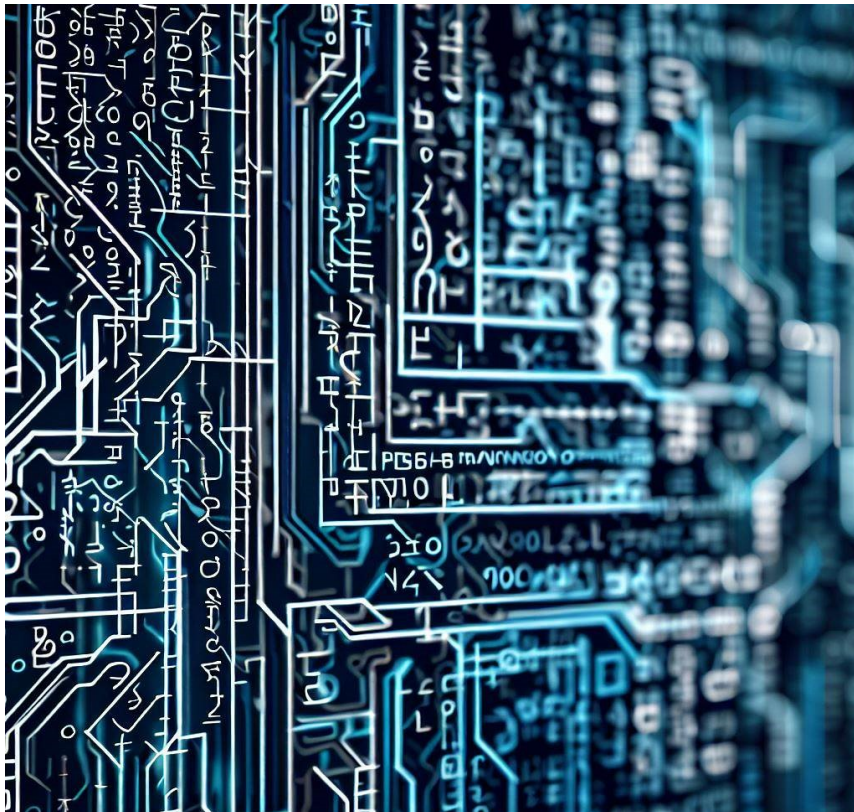
AIC.Set Delimiter QS

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set Delimiter QS

Set the delimiters for the Prompts in the history.



Intention

The `AIC.Set Delimiter QS` command in the Smart-Package Robot (SPR) allows users to set the delimiters that are used in the Text-History. This can make it easier to parse the history.

Syntax:

```
AIC.Set Delimiter QS|<delimiter start>|<delimiter end>
```

Parameters:

`<delimiter start>`: Any Text or string that makes it easy for you to parse the History.

`<delimiter end>`: Any Text or string that makes it easy for you to parse the History.

Example Usage:

```
AIC.Set Delimiter AN|<Answer_Start:>|<Answer_End:>
AIC.Set Delimiter QS|<Prompt_Start:>|<Prompt_End:>
ENR.
```

This example will enclose the "Answers" between <Answer_Start:> and <Answer_End:>

The Prompts will be enclosed between <Prompt_Start:> and <Prompt_End:>

in the History that you can get using the "AIC.Get History" Command.

Syntax

```
AIC.Set Delimiter QS [|P1] [|P2]
AIC.SDQ [|P1] [|P2]
```

Parameter Explanation

P1 - opt. Variable with the delimiter for the "Start of the Question". If omitted, the value is been set to `$CrLf$`.

P2 - opt. Variable with the delimiter for the "End of the Question". If omitted, the value is been set to `$CrLf$`.

Example

```
'*****
' EXAMPLE 1: AIC.-Commands
'*****
AIC.Set Delimiter AN|<Antwort_Start:>|<Antwort_End:>
AIC.Set Delimiter QS|<Prompt_Start:>|<Prompt_End:>
AIL.Set Number|1
AIL.Set Temperature|$$LR0
AIL.Set Model|Hermes
$$TXT=Hello! Please tell me what 3+4 is.
AIL.Ask GPT4A11|$$TXT|$$REA

$$TXT=Hello! Please tell me what 5+6 is.
AIL.Ask GPT4A11|$$TXT|$$REA
DBP.-----
AIC.ghi|$$REB
DBP.The complete history was:
DBP.$$REB
DBP.-----
ENR.
```

Remarks

-

Limitations:

-

See also:

- [Set_Key](#)⁷⁹⁹
- [Ask_Chat](#)⁸³³
- [Ask_Completion](#)⁸⁴²

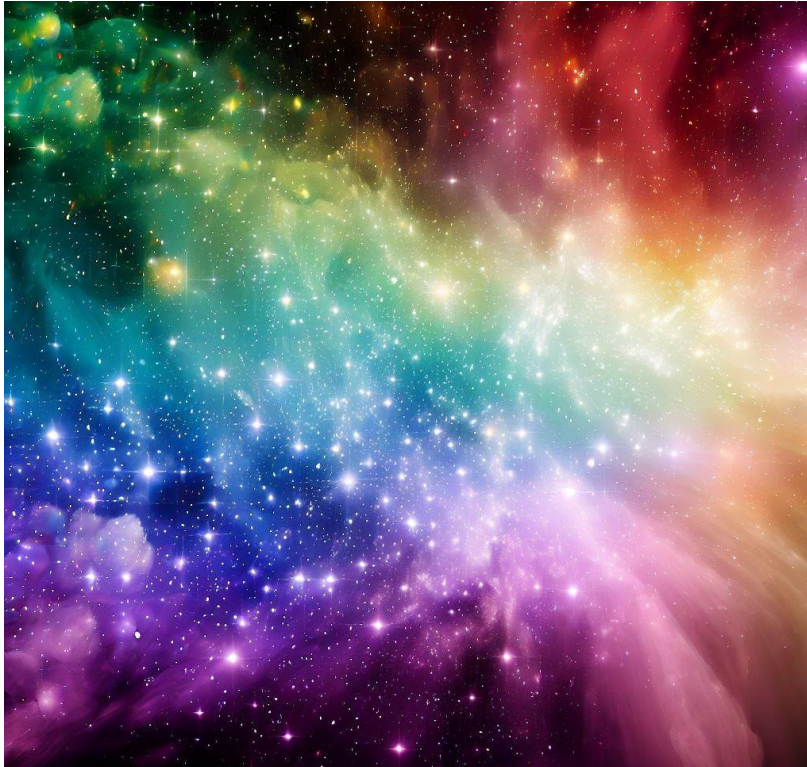
3.42.10.5.8 Set History Dimension

[AIC.Set History Dimension](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set History Dimension

Set the Number of Entries that will be stored in the Index-Array-based history.



Intention

The `AIC.Set History Dimension` command is designed to manage the size of your system's history memory.

It provides you with the ability to control the number of history events that can be stored, reset the history memory to its default size, or simply retrieve the current size of the history memory. An optional second parameter can be used to return the history array size. Maximum size is actually 32768 Steps.

Syntax:

```
AIC.Set History Dimension|<value>|<variable for return>
```

Example Usage:

```
' Will reset the size of the history to its default size w  
AIC.Set History Dimension|-1|$$RET
```



```
' Will not change the size of the history just return the
AIC.Set History Dimension|0|$$RET
```

```
' Will set the size of the history to 20024 Steps.
' Return Value will be on TOS.
AIC.Set History Dimension|20024
```

```
' All three variants will return the current size of the H
ENR.
```

Syntax

```
AIC.Set History Dimension|P1 [|
P2]
AIC.SHD|P1 [|P2]
```

Parameter Explanation

P1 - Variable with any number between -1 and 32767. Hereby is:

-1 - will reset the History Array to its default size.

0 - will not change the size just return the current size

>0 - will set the size of the History Array to that number, currently the maximum number is 32767

P2 - opt. Variable with the delimiter for the "End of the Question". If omitted, the value is been set to `$CrLf$`.

Example

```
'*****
' EXAMPLE 1: AIC.-Commands
'*****
' We can alsu
AIC.Set History Dimension|120
AIL.Set Number|1
AIL.Set Temperature|$$LR0
AIL.Set Model|Hermes
$$TXT=Hello! Please tell me what 3+4 is.
AIL.Ask GPT4All|$$TXT|$$REA

$$TXT=Hello! Please tell me what 5+6 is.
```

```
AIL.Ask GPT4A11|$$TXT|$$REA
FOR.$$LOP|1|2
DBP.-----
AIC.ghf|$$LOP|$$ANS|$$QUE
DBP.We asked: $$QUE
DBP.and got this answer: $$ANS
DBP.-----
NEX.
ENR.
```

Remarks

-

Limitations:

Maximum size is actually 32768 Steps.

See also:

- [Set Key](#)⁷⁹⁹
- [Ask Chat](#)⁸³³
- [Ask Completion](#)⁸⁴²

3.42.10.6 AI - Internet Commands

This chapter contains commands that are designed to help you use Online AI-Services.



Its not magic, its AI combined with the Internet.

Staying Connected

To ensure a seamless experience, make sure your device is connected to the internet. This is necessary not just for browsing, but also for making use of most of the AI features.

Exploring Rich Content

Discover a world filled with vibrant pictures, captivating sounds, and a wealth of information.

Browse through images, listen to music, watch videos, and read articles on countless topics.

AI will help you to sort things out.

AI at Your Service

Harness the power of AI to make your life easier. From simple tasks like setting reminders to more complex ones like data analysis, AI is here to assist. Note that the AI services can only be availed if you are online.

Commands

Here are some commands that can help you navigate through content and make the best use of AI services:

- AIC.Download File: Download any File via `http://` from the Internet.
- AIC.Download Image File: Download images from the Internet. This Command also helps you to give these Images Numbers.
- AIC.http Request: This is a special Command for advanced usage.
- AIC.Set Proxy On: Turn ON or OFF an configured Proxy Server.
- AIC.Set Proxy Server: Configure your Proxy Server.
- AIC.Test If Online: Puts a 0 or a 1 on the TOS depending if you are Online or not.

Conclusion

The internet and the AIC.-Commands together make a powerful combination that puts a vast amount of resources at your fingertips.

Always ensure that you are online to make the most out of these technologies. Stay curious, explore responsibly, and let AI assist you in your endeavors.

3.42.10.6.1 Decode from Base64

[Decode From BASE64](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

Decode from Base64

Decode Data that was encoded using BASE64 or Base64 URL

Intention

Base64 is a method for encoding 8-bit binary data (such as executable programs, ZIP files, or images) into a string consisting only of readable, codepage-independent ASCII characters. It is used in the Internet standard Multipurpose Internet Mail Extensions (MIME) and is used there for sending email attachments.

This is necessary to ensure the smooth transport of any binary data, as SMTP in its original version was not designed to handle binary data.

Base64 is often used in JSON to encode binary data, such as images or other files, that cannot be directly represented in JSON.

JSON is a text-based format, so binary data must be encoded into a text format before it can be included in a JSON object.

Base64 is one of the most common methods for doing this.

The binary data (for example images generated using AI) is first converted into a Base64 string, which can then be included in the JSON object as a normal string value.

This allows the binary data to be transmitted and stored along with the rest of the JSON data.

Base64 and **Base64url** are both ways to encode binary data in string form.

The problem with Base64 is that it contains the characters `+`, `/`, and `=`, which have a reserved meaning in some filesystem names and URLs.

So **Base64url** solves this by replacing `+` with `-` and `/` with `_`.

The trailing padding character `=` can be eliminated when not needed, but in a URL it would instead most likely be `%` URL encoded.

Then the encoded data can be included in a URL without problems¹.

So, if you need to transmit or save Base64-encoded text where `+`, `/`, or `=` have special meaning, e.g. in URLs where all 3 do, then it is better to use **Base64url**.

AIC.Decode from BASE64 will convert either Base6 or Base64URL back to Binary Data.

AIC.Encode to BASE64 will convert Binary Data to MIME/BASE64 data.

It is advised to save binary data to a file and prevent it from going through unnecessary Variable resolutions.

The command itself looks like this:

```
' $$SRC - MIME-Format
```


P2 - (*optional*) destination variable. If omitted the result is placed in the variable in **P1**.

Example

```

'*****
' AIC.Encode to BASE64 and
' AIC.Decode from BASE64
' using one Input-Output-Variable
'*****
VAR.$$SRC=What happens under BASE64 with the + and / Characters?
VAR.$$SRC+$$SRC$$SRC$$SRC$$SRC$$SRC
LEN.$$SRC|$$LEN
PRT.-----
PRT.Original:
PRT.($$LEN)->$$SRC
AIC.Encode to BASE64|$$SRC
LEN.$$SRC|$$LEN
PRT.-----
PRT.ENCODED:
PRT.($$LEN)->$$SRC
PRT.-----
AIC.Decode from BASE64|$$SRC
LEN.$$SRC|$$LEN
PRT.DECODED:
PRT.($$LEN)->$$SRC
PRT.-----
MBX.It works!

ENR.

```

Remarks

-

Limitations:

-

See also:

- [1.5.2 Working with Text-Strings](#)^[123]
- [STR.TO BASE64 URL](#)^[4417]
- [STR.TEXT TO MIME](#)^[4413]
- [STR.MIME TO TEXT](#)^[4351]
- [STR.GENERATE - Mode](#)^[4320]

- [GEC. - Get Encryption](#)¹¹⁸⁷³¹
- [GMD. - Get Message Digest](#)¹¹⁸⁷⁹¹
- [SHA. - Safer Hash Algo](#)¹¹⁸⁸⁹¹
- [VAR. - Variable Set Value/Clear](#)⁴⁴⁹²⁵¹
- [RPL. - RePLace in String](#)⁴⁴⁴⁸⁴¹
- [IVS. / NVS. - If-Variable-String](#)⁴⁴⁸⁹⁴¹
- [GSS. - GetSplitString](#)⁴⁴⁴⁷⁸¹
- [GFS. - Get-Filtered-String](#)⁴⁴⁴⁶³¹
- [GES. - Get-Extracted-String](#)⁴⁴¹⁹⁵¹
- [LEN. - Length-of-String](#)⁴⁴⁴⁸¹¹

3.42.10.6.2 Download File

[AIC.Download File](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Download File

Download a file from the Internet.



Intention

The `AIC.Download File` command is a versatile command designed for downloading files from the internet using the `HTTP` protocol and saving them to a designated location on the local storage. It is not limited to images, and can be used to download any type of file including documents, archives, audio files, etc.

Syntax:

```
AIC.Download File|<URL>|<Path+Filename on local drive>
```

Parameters:

- **<URL>**: The complete URL of the file that you want to download. It must start with `http://` or `https://`.
- **<Path+Filename on local drive>**: The path where you want to save the downloaded file along with the desired filename. The path should be valid and the filename should include the file extension (e.g., `.pdf`, `.zip`).

Example Usage:

```
AIC.Download File|  
https://example.com/document.pdf|  
C:/documents/myfile.pdf
```

In this example, the command will download the file from `https://example.com/document.pdf` and save it to `C:/documents/` with the filename `myfile.pdf`.

Here is another example:

```
$$URL=https://www.smart-package.com/downloads/SPR_Silent_Setup.exe  
$$PAT=?exeloc\  
AIC.Download File|$$URL|$$PAT  
ENR.
```

```
' Here is the same example, but the Filename is specified  
$$URL=https://www.smart-package.com/downloads/SPR_Silent_Setup.exe  
$$PAT=?exeloc\Silent_Setup.exe  
AIC.Download File|$$URL|$$PAT  
ENR.
```

Important Considerations:

- **Valid URL:** Ensure that the URL is valid and is accessible. URLs that require authentication or are otherwise restricted may not work.
- **Storage Space:** Make sure that there is sufficient storage space at the specified location for the file.
- **File Formats and Extensions:** Ensure that the filename includes the correct file extension corresponding to the format of the file being downloaded.
- **Permissions:** The specified path should be a location where the application has write permissions.
- **Security:** Be cautious when downloading files from untrusted sources as they could contain malware or other security threats.

Note:

Please use this command responsibly and ensure that you have the rights to download and use the files. Be aware of copyright, licensing issues, and terms of use when downloading files from the internet. Also, be mindful of security and only download files from reputable sources.

Syntax

```
AIC.Download File|P1|P2
```

```
AIC.DLF|P1|P2
```

Parameter Explanation

P1 - URL: This parameter specifies the URL from which the file will be downloaded.

P2 - Filepath for the Downloaded File: This parameter specifies the local file path where the downloaded file should be saved.

If the URL (**P1**) contains a valid filename, you have the option to omit the filename in this parameter. Instead, simply end the file path with a backslash "\".

When the file path ends with a backslash, the filename from the URL will be used automatically.

P3 - Optional Flags: This parameter allows you to set optional flags to modify the download behavior.

1: Download asynchronously without waiting. This means that the download will happen in the background, allowing the program to continue running without waiting for the download to complete.

If the Script ends before the Download is completed, the download will be incomplete.

2: Do not follow redirects. This means that if the URL redirects to another location, the download will not follow the redirect and will instead terminate.

Example

```
' *****  
' EXAMPLE 1: AIC.-Commands  
' *****  
$$URL=https://www.smart-package.com/downloads/SPR_Silent_Setup.exe  
$$PAT=?exeloc\  
AIC.Download File|$$URL|$$PAT  
ENR.
```

Remarks

-

Limitations:

- [Download Imagefile](#)⁷³¹

See also:

- [Set Key](#)⁷⁹⁹
- [Ask Chat](#)⁸³³
- [Ask Completion](#)⁸⁴²

3.42.10.6.3 Download Imagefile

AIC.Download Imagefile

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Download Image File

Download an Image File from an specified URL and save it as File.



This picture is one of 3 pictures that was created using the Script below.

When using the "Image Generation Commands" you may get this type of returns (see Script and picture below) that contain URL's of the generated Pictures.

Instead of having them automatically downloaded and saved, you can also do that manually.

And of course you can also download other pictures from the Internet, from social media etc. in case you have the URL.

```
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' This will make the AI to generate 3 Pictures
AIC.SetNumber|1
' Can be 256 or 512 or 1024
AIC.Set Image Size|256
$$TXT=Chessboard with a nice golden Boxer on a horse with 4 legs and an galactic
AIC.Generate Image|$$TXT

' Use this to see directly the last Output of the API
AIC.gtt|0|$$ROW
MBX.$$ROW
AIC.Download Image File|$$ROW|?exeloc\MyPic_****.png|005
ENR.
```

MyPic_005.png

26.06.2023 20:29

PNG-Datei

193 KB

Intention

The `AIC.Download Image File` command is designed to download images from a specified (**P1**) internet URL and save them to a designated location on the local storage.

This command is versatile and can be used for downloading images from various sources, including those generated through the OpenAI Image Generation commands.

Additionally, it is used internally by SPR to store images that are created by the image generation commands.

Syntax:

```
AIC.Download Image File|<URL>|<Path+Filename>|<Filenumber to replace "*****" in the Filename>
```

Parameters:

<URL>: The complete URL of the image that you want to download. It must start with `http://` or `https://`.

<Path+Filename>: The path where you want to save the downloaded image along with the desired filename. The path should be valid and the filename should include the file extension (e.g., `.jpg`, `.png`). You can include `****` in the filename which will be replaced by the filenumber parameter.

<Filenumber to replace "***" in the Filename>**: A number which will replace the `****` in the filename. This can be used to create unique filenames for multiple downloads. It should be an integer.

Example Usage:

```
AIC.Download Image File|https://example.com/image.jpg|C:/i
```

In this example, the command will download the image from `https://example.com/image.jpg` and save it to `C:/images/` with the filename `picture001.jpg`.

Important Considerations:

- **Valid URL:** Ensure that the URL is valid and is accessible. URLs that require authentication or are otherwise restricted may not work.
- **Storage Space:** Make sure that there is sufficient storage space at the specified location for the image file.
- **File Formats:** Ensure that the filename includes the correct file extension corresponding to the format of the image being downloaded.
- **Permissions:** The specified path should be a location where the application has write permissions.
- **Unique Filenames:** Using the filenumber replacement feature (`****`) is useful for ensuring unique filenames and avoiding overwriting existing files.
Currently files with the same name will be overwritten, without any message.

Note:

Please use this command responsibly and ensure that you have the rights to download and use the images.
Be aware of copyright and licensing issues when downloading images from the internet.

Syntax

```
AIC.Download Image File|P1[|P2]
[|P3]
AIC.DIM|P1 [|P2] [|P3]
```

Parameter Explanation

P1 - URL of the file to download

P2 - opt. Variable with the Filename and Path where the Picture shall be saved. If the Filename contains "****", these will be replaced with the number in **P3**.

P3 - opt. Variable or numeric value, this is the number of alternative results. It will replace the "****" in **P2**.

Example

```
'*****
' EXAMPLE 1: AIC.-Commands
'*****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' This will make the AI to generate 3 Pictures
AIC.SetNumber|1
' Can be 256 or 512 or 1024
AIC.Set Image Size|256
$$TXT=Chessboard with a nice golden Boxer on a horse with 4 legs and an galactic
AIC.Generate Image|$$TXT

' Use this to see the last Output of the API
AIC.gtt|0|$$ROW
MBX.$$ROW
AIC.Download Image File|$$ROW|?exeloc\MyPic_****.png|005
ENR.
```





Remarks

-

Limitations:

-

See also:

- [Download File](#)  728
- [Set Key](#)  799
- [Ask Chat](#)  833
- [Ask Completion](#)  842

3.42.10.6.4 Encode to Base64

[Decode From BASE64](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

Encode to Base64

Encode Data using BASE64

Intention

Base64 is a method for encoding 8-bit binary data (such as executable programs, ZIP files, or images) into a string consisting only of readable, codepage-independent ASCII characters. It is used in the Internet standard Multipurpose Internet Mail Extensions (MIME) and is used there for sending email attachments.

This is necessary to ensure the smooth transport of any binary data, as SMTP in its original version was not designed to handle binary data.

Base64 is often used in JSON to encode binary data, such as images or other files, that cannot be directly represented in JSON.

JSON is a text-based format, so binary data must be encoded into a text format before it can be included in a JSON object.

Base64 is one of the most common methods for doing this.

The binary data (for example images generated using AI) is first converted into a Base64 string, which can then be included in the JSON object as a normal string value.

This allows the binary data to be transmitted and stored along with the rest of the JSON data.

Base64 and **Base64url** are both ways to encode binary data in string form.

The problem with Base64 is that it contains the characters ``+``, ``/``, and ``=``, which have a reserved meaning in some filesystem names and URLs.

So **Base64url** solves this by replacing ``+`` with ``-`` and ``/`` with ``_``.

The trailing padding character ``=`` can be eliminated when not needed, but in a URL it would instead most likely be ``%`` URL encoded.

Then the encoded data can be included in a URL without problems¹.

So, if you need to transmit or save Base64-encoded text where ``+``, ``/``, or ``=`` have special meaning, e.g. in URLs where all 3 do, then it is better to use

Base64url.

AIC.Decode from BASE64 will convert either Base6 or Base64URL back to Binary Data.

AIC.Encode to BASE64 will convert Binary Data to MIME/BASE64 data.

It is advised to save binary data to a file and prevent it from going through unnecessary Variable resolutions.

The command itself looks like this:

```
' $$SRC - MIME-Format
```

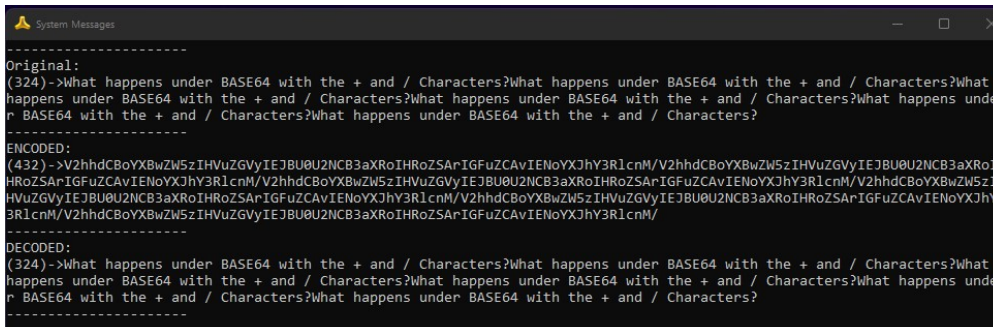


```
' $$TAR - Binary or Text-Data Result
AIC.DECODE FROM BASE64|$$SRC|$$TAR
```

Here is a Sample-Script:

```
VAR.$$SRC=What happens under BASE64 with the + and / Characters?
VAR.$$SRC+$$SRC$$SRC$$SRC$$SRC$$SRC
LEN.$$SRC|$$LEN
PRT.-----
PRT.Original:
PRT.($$LEN)->$$SRC
AIC.Encode to BASE64|$$SRC
LEN.$$SRC|$$LEN
PRT.-----
PRT.ENCODED:
PRT.($$LEN)->$$SRC
PRT.-----
AIC.Decode from BASE64|$$SRC
LEN.$$SRC|$$LEN
PRT.DECODED:
PRT.($$LEN)->$$SRC
PRT.-----
MBX.It works!
ENR.
```

The Result looks somehow like you see below.



```
System Messages
-----
Original:
(324)->What happens under BASE64 with the + and / Characters?What happens under BASE64 with the + and / Characters?What
happens under BASE64 with the + and / Characters?What happens under BASE64 with the + and / Characters?What happens unde
r BASE64 with the + and / Characters?What happens under BASE64 with the + and / Characters?
-----
ENCODED:
(432)->V2hhdCB0eXBwZW5zIHVuZGVyIEJBU0U2NCB3aXR0IHROZSArIGFuZCAvIENoYXJhY3R1cnM/V2hhdCB0eXBwZW5zIHVuZGVyIEJBU0U2NCB3aXR0I
HROZSArIGFuZCAvIENoYXJhY3R1cnM/V2hhdCB0eXBwZW5zIHVuZGVyIEJBU0U2NCB3aXR0IHROZSArIGFuZCAvIENoYXJhY3R1cnM/V2hhdCB0eXBwZW5zIH
V2hhdCB0eXBwZW5zIHVuZGVyIEJBU0U2NCB3aXR0IHROZSArIGFuZCAvIENoYXJhY3R1cnM/V2hhdCB0eXBwZW5zIHVuZGVyIEJBU0U2NCB3aXR0IHROZSArIGFuZCAvIENoYXJhY
3R1cnM/V2hhdCB0eXBwZW5zIHVuZGVyIEJBU0U2NCB3aXR0IHROZSArIGFuZCAvIENoYXJhY3R1cnM/
-----
DECODED:
(324)->What happens under BASE64 with the + and / Characters?What happens under BASE64 with the + and / Characters?What
happens under BASE64 with the + and / Characters?What happens under BASE64 with the + and / Characters?What happens unde
r BASE64 with the + and / Characters?What happens under BASE64 with the + and / Characters?
```

As you can see the MIME encoded Binary is 432 bytes long, while the original size was 324 bytes. That is generally 134% the size of the original size. If the MIME-Text contains linefeeds (`$\r\n`), these will be ignored.

This is the revers command to `AIC.Encode to Base64`.

Syntax

AIC.Encode to BASE64|P1[|P2]

Parameter Explanation

P1 - source variable. If **P2** is missing then its also the destination variable

P2 - (*optional*) destination variable. If omitted the result is placed in the variable in **P1**.

Example

```

'*****
' AIC.Encode to BASE64 and
' AIC.Decode from BASE64
' using one Input-Output-Variable
'*****
VAR.$$SRC=What happens under BASE64 with the + and / Characters?
VAR.$$SRC+$$SRC$$SRC$$SRC$$SRC$$SRC
LEN.$$SRC|$$LEN
PRT.-----
PRT.Original:
PRT.($$LEN)->$$SRC
AIC.Encode to BASE64|$$SRC
LEN.$$SRC|$$LEN
PRT.-----
PRT.ENCODED:
PRT.($$LEN)->$$SRC
PRT.-----
AIC.Decode from BASE64|$$SRC
LEN.$$SRC|$$LEN
PRT.DECODED:
PRT.($$LEN)->$$SRC
PRT.-----
MBX.It works!
ENR.

```

Remarks

-

Limitations:

-

See also:

- [1.5.2 Working with Text-Strings](#)^[123]
- [STR.TO BASE64 URL](#)^[4417]
- [STR.TEXT TO MIME](#)^[4413]
- [STR.MIME TO TEXT](#)^[4351]
- [STR.GENERATE - Mode](#)^[4320]
- [GEC. - Get Encryption](#)^[1873]

- [GMD. - Get Message Digest](#)¹⁸⁷⁹
- [SHA. - Safer Hash Algo](#)¹⁸⁸⁹
- [VAR. - Variable Set Value/Clear](#)⁴⁹²⁵
- [RPL. - RePLace in String](#)⁴⁴⁸⁴
- [IVS. / NVS. - If-Variable-String](#)⁴⁸⁹⁴
- [GSS. - GetSplitString](#)⁴⁴⁷⁸
- [GFS. - Get-Filtered-String](#)⁴⁴⁶³
- [GES. - Get-Extracted-String](#)⁴¹⁹⁵
- [LEN. - Length-of-String](#)⁴⁴⁸¹

3.42.10.6.5 Get http-Status

AIC.Get http-Status[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Get http-Status

Get the HTTP-Status from last http-Request.

Intention

The `AIC.Get http Status` command is used to return the Status of the last http-Request from any such Operation, including `AIC.http-Request`.

This command can be used to interface additional AI Services that may not yet be implemented using the SPR.

It can also be used for things that have nothing to do with AI-Services.

Technically the Command does exactly the same as

```
AIC.Get Several | 4 | $$RET
```

```
AIC.Get http-Status | $$RET
```

Here is a Table with the most common Status Codes and what they want to say.

These Status-Codes below are just a few examples, and the actual HTTP specification includes many more status codes.

If you are interested in a more comprehensive list, you may want to consult the [official HTTP/1.1 specification](#) or

other online resources that list all possible HTTP status codes.

Status Code	Category	Meaning
200	Success	The request was successful, and the server is sending the requested data.
201	Success	The request was successful, and a new resource was created as a result.
204	Success	The request was successful, but there's no representation to return (i.e. the response is empty).
301	Redirection	The requested URL has been moved

		permanently to another URL. The server will redirect the client to the new URL.
302	Redirection	The requested URL has been found but moved temporarily to another URL. The server will redirect the client to the new URL.
400	Client Error	The request could not be understood or was missing required parameters.
401	Client Error	The client must authenticate itself to get the requested response.
403	Client Error	The client does not have access rights to the content; that is, it is unauthorized, so the server is rejecting to give the requested resource.
404	Client Error	The server can not find the requested resource. This code is most often used when the server does not wish to reveal exactly why the request has been refused, or when no other response is applicable.
500	Server Error	The server has encountered a situation it doesn't know how to handle.
502	Server Error	The server, while acting as a gateway or proxy, received an invalid response from an inbound server it accessed while

		attempting to fulfill the request.
503	Server Error	The server is not ready to handle the request, often because it is overloaded or under maintenance.

Some of the most used Status-Codes.

Specific Status Codes for Claude 2 from [Anthropic.com](https://anthropic.com)

HTTP Status Code	Description
400	Invalid request: There was an issue with the format or content of your request.
401	Unauthorized: There's an issue with your API key.
403	Forbidden: Your API key does not have permission to use the specified resource.
404	Not found: The requested resource was not found.
429	Too Many Requests: Your account has hit a rate limit.
500	Internal Server Error: An unexpected error has occurred internal to Anthropic's systems.
529	Service Unavailable: Anthropic's API is temporarily overloaded.

Depending on the availability of the [Claude 2 API](#), we will also support this AI-System. In that case here are the error-values you could get there.

More general Information on Status-Codes:
[Internet-Link: HTTP-Status-Codes](#)

Managing DeepL.com Error-Codes:

HTTP status codes are utilized to signal errors. It's crucial that your application is designed to appropriately respond to these errors. For guidance, refer to the status code results expected for each endpoint, as outlined in the endpoint documentation. The following HTTP status codes require particular attention:

HTTP Status Code	Description	Recommended Action
429	Too many requests	Configure your application to delay and resend the request instead of continuously resending it. If you're experiencing frequent 429 errors and wish to enhance your translation speed, reach out to support@DeepL.com .
456	Quota exceeded	Your account's translation limit has been met. Consider upgrading your subscription.
500 and above	Temporary errors in the DeepL service	Set up your application to delay and resend the request instead of continuously resending it. If the error continues for an extended period, please contact support@DeepL.com .

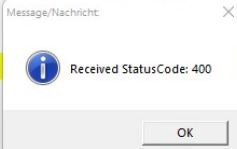
When resending failed requests, your application should employ an exponential-backoff strategy, which increases the delay time with each subsequent failed request. All official DeepL client libraries incorporate this strategy and can serve as a reference. The service's responsiveness adjusts dynamically based on system load. Once errors subside, you can resume sending more requests. As the service adjusts, you'll be able to send an increasing number of requests within a certain timeframe without encountering errors. If you're unable to achieve your desired translation speed or if you're receiving an excessive number of 429 errors, please contact support@DeepL.com for further investigation.

Additional details about the error may be provided in a JSON response. If available, this extra information will be found in the 'message' key.

Source:

[DeepL.com specific Status-Code:](#)

```
' Make http-Request and receive answer
$$URL=https://api.openai.com
AIC.http request|get|$$URL||Accept=application/json$crLf$User-Agent=MyApp
POP.$$TOS
AIC.Get Several|4|$$RET
MBX.Received $$RET
ENR.
```



Here is a Sample of a Script that runs and returns a statuscode.

Example Usage:

```
' Make http-Request and receive answer
$$URL=https://api.openai.com
AIC.http request|get|$$URL||Accept=application/json$crLf$U
POP.$$TOS
AIC.Get Several|4|$$RET
MBX.Received $$RET
ENR.
```

Syntax

AIC.Get http-Status | P1
AIC.Ghs | P1

Parameter Explanation

P1 - <Variable>: Variable to return the Status-Code, prefixed by "StatusCode: ".

Example

```
' *****
' EXAMPLE 1: AIC.-Commands
' ' This is just a Syntax Sample the Script does not make anything useful
' *****
' Make http-Request and receive answer
$$URL=https://api.openai.com
AIC.http request|get|$$URL||Accept=application/json$crLf$User-Agent=MyApp
POP.$$TOS
AIC.Get Several|4|$$RET
MBX.Received $$RET
ENR.
```


Remarks

-

Limitations:

-

See also:

- [Set_Key](#)⁷⁹⁹
- [Ask_Chat](#)⁸³³
- [Ask_Completion](#)⁸⁴²

3.42.10.6.6 Http Request

[AIC.Http Request](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Http Request

Send an HTTP request and receive the result

Intention

The `AIC.Http Request` command is used to send an HTTP request to a specified URL.

It allows you to perform different types of HTTP operations such as **GET**, **POST**, **PUT**, **PATCH**, and **DELETE**.

You can also specify data to be sent and custom headers for the request.

The Command will return the Status of the Request on TOS. If this is "0" there will be additional Information available using

```
AIC.Get Several|3|$$RET
```

this Command will also return the result of the `http.request` no matter is the call was successful or error.

This command can be used to interface additional AI Services that may not yet be implemented using the SPR.

It can also be used for things that have nothing to do with AI-Services.

Example Usage:

```
' Make http-Request and receive answer
$$URL=https://api.openai.com
AIC.http request|get|$$URL||Accept=application/json$crLf$U
POP.$$TOS
AIC.Get Several|3|$$RET
MBX.Received $$TOS: $$RET
ENR.
```

Syntax

AIC.Http Request|P1|P2|P3|P4

AIC.HTP|P1|P2|P3|P4

Parameter Explanation

P1 - <Type of Operation>: Specifies the HTTP method for the request. Acceptable values are: `get`, `post`, `put`, `patch`, `delete`. This parameter is case-insensitive.

P2 - <URL>: Specifies the URL to which the HTTP request is sent. Should be a valid URL string.

P3 - <Data>: Specifies the data to be sent in the request. For **GET** requests, this should typically be left empty or null. For **POST**, **PUT**, and **PATCH** requests, this can be the data payload.

P4 - <Headers>: Specifies the headers to be included in the HTTP request. Headers should be separated with `$crlf$` and each key-value pair should be separated with `=`.

If no headers are specified, the default headers will be used:

```
content-type set to application/json
Authorization set to Bearer <AI_OKEY>
```

Where `<AI_OKEY>` is the Open AI API-Key that you have set using the Command `AIC.Set Key`

Example

```
'*****
' EXAMPLE 1: AIC.-Commands
' ' This is just a Syntax Sample the Script does not make anything useful
'*****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000
' The more Tokens you use the more you need to pay.
AIC.SetMax_Token|25

' Make http-Request and receive answer
$$URL=https://api.openai.com
AIC.http request|get|$$URL||Accept=application/json$crlf$user-Agent=MyApp
POP.$$TOS
AIC.Get Several|3|$$RET
MBX.Received $$TOS: $$RET

:enx
ENR.
```

Remarks

-

Limitations:

-

See also:

- [Set_Key](#)⁷⁹⁹
- [Ask_Chat](#)⁸³³
- [Ask_Completion](#)⁸⁴²

3.42.10.6.7 Set Proxy ON

[AIC.Set Proxy On](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set Proxy On

Enable or disable the use of a Proxy Server

Intention

The `AIC.Set Proxy On` command (abbreviated as `AIC.SPO`) is used to enable or disable the use of a Proxy Server in the Smart Package Robot AI-Commands.

This command can be used in conjunction with the `AIC.Set Proxy Server` command, which sets the parameters for the Proxy Server such as **IP address, login credentials, and password**.

Please be careful when using this command together with the GPT4All Commands as they will also try to use the Proxy Server if it is turned on!

This was done due to the fact that GPT4All can also be used in Networks using the `AIC.Change GPT4all URL - Command`.

Using this Command you can switch On and Off the Proxy Server quickly at any time, the Parameters from the `AIC.Set Proxy Server` Command will be stored internally until the Script ends.

Syntax:

```
AIC.Set Proxy On|0 or 1
```

Example Usage:

```
AIC.Set Proxy On|1
```

This example enables the Use of the Proxy-Server.

Syntax

```
AIC.Set Proxy On [ |P1 ]  
AIC.SPO [ |P1 ]
```

Parameter Explanation

P1 - opt. `<Number>`: The parameter **P1** can be set to either 0 or 1.
0: This value disables the Proxy Server.

1: This value enables the Proxy Server.

If **P1** is not specified, the default value is 1, which means the Proxy Server will be turned on by default.

The command leaves the state of the Proxy-Server on the TOS.

Example

```
'*****
' EXAMPLE 1: AIC.-Commands
'*****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' This will switch on the Proxy Server for all following AI-Operations
' You can easily switch off the Proxy Server at any time, the Parameters
' will be stored internally until the Script ends
AIC.Set Proxy Server|192.168.0.1|LoginMe|MyPassword
AIC.SetProx ON|1

' Set Model
AIC.SetModel_Completion|4

' Set Model-Temperature
AIC.Set_Temperature|0

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000)
' The more Tokens you use the more you need to pay.
AIC.SetMax_Token|25

' We will create 4 alternative answers
AIC.Set Proxy On|4

' Ask Question and receive answer to $$RET
AIC.Ask_Completion|What is a "Windows Button"|$$RET
MBX.$$RET

:enx
ENR.
```

Remarks

Please be careful when using this command together with the GPT4All Commands as they will also try to use the Proxy Server if it is turned on!

This was done due to the fact that GPT4All can also be used in Networks using the AIC.Change GPT4all URL - Command.

If **GPT4All** runs local it is recommended to turn the Proxy-Server off.

Limitations:

-

See also:

-

3.42.10.6.8 Set Proxy Server

[AIC.Set Proxy Server](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set Proxy Server

Set the Parameters like IP, Login and Password for use of a the Proxy Server with AI-Commands

Intention

The `AIC.Set Proxy Server` command (abbreviated as `AIC.SPX`) is used to set the parameters for the Proxy Server such as:

IP address, login credentials, and password for use with all Smart Package Robot AI-Commands.

Note that all AI commands use http-Connections therefore select the Proxy-Server for such connections.

This command should be used in conjunction with the "`AIC.Set Proxy ON`" command, which will turn the Proxy Serve `ON` or `OFF`.

Please be careful when using this command together with the GPT4All Commands as they will also try to use the Proxy Server if it is turned on!

This was done due to the fact that GPT4All can also be used in Networks using the `AIC.Change GPT4all URL - Command`.

Using the the "`AIC.Set Proxy ON`" Command you can switch On and Off the Proxy Server quickly at any time, the Parameters from the `AIC.Set Proxy Server` Command will be stored internally until the Script ends.

Example Usage:

```
AIC.Set Proxy Server|1
AIC.Set Proxy Server|192.168.0.1|Login|Password
```

This example enables the Use of the Proxy-Server with the given Parameters.

Syntax

```
AIC.Set Proxy Server|P1 [|P2] [|P3]
```


AIC . SPX | P1 [| P2] [| P3]

Parameter Explanation

P1 - <IP-Adress of the Proxy Server>: The parameter **P1** should contain the address of the proxy Server.

P2 - opt. <Login>: The parameter **P2** contains your Login to the Proxy-Server. If there is no Login & Password, just omit **P2** and **P3**

P3 - opt. <Password>: The parameter **P3** contains your Password for use with the Proxy-Server. If there is no Login & Password, just omit **P2** and **P3**

Example

```

'*****
' EXAMPLE 1: AIC.-Commands
'*****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' This will switch on the Proxy Server for all following AI-Operations
' You can easily switch off the Proxy Server at any time, the Parameters
' will be stored internally until the Script ends
AIC.Set Proxy Server|192.168.0.1|LoginMe|MyPassword
AIC.SetProx ON|1

' Set Model
AIC.SetModel_Completion|4

' Set Model-Temperature
AIC.Set_Temperature|0

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000)
' The more Tokens you use the more you need to pay.
AIC.SetMax_Token|25

' We will create 4 alternative answers
AIC.Set Proxy Server|4

' Ask Question and receive answer to $$RET
AIC.Ask_Completion|What is a "Windows Button"?|$$RET
MBX.$$RET

:enx
ENR.

```

Remarks

Please be careful when using this command together with the GPT4All Commands as they will also try to use the Proxy Server if it is turned on!

This was done due to the fact that GPT4All can also be used in Networks using the AIC.Change GPT4all URL - Command.

If **GPT4All** runs local it is recommended to turn the Proxy-Server off.

Limitations:

-

See also:

-

3.42.10.6.9 Test_If_Online

[AIC.Test if Online](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Test if Online

Test if the SPR can reach out to the Open AI-API Server



Intention

The `AIC.Test if Online` Command will leave the Information if the PC is connected to the Internet, on TOS or in a specified Variable as "0" or "1". This way you can easily use this Information in your Scripts.

```
' Usage for TIO
AIC.tio
' Show TOS
DMP.6
MBX.!
```

ENR.

```

System Messages
-----
User-Stack-Dump:
-----
stackpos: 000 ->LongInt :1
-----

```

Syntax

```
AIC.Test if Online[|P1]
AIC.TIO[|P1]
```

Parameter Explanation

P1 - opt. Variable for Result, if omitted result is placed on TOS.

Example

```

'*****
' EXAMPLE 1: AIC.-Commands
'*****
' Usage for TIO
AIC.tio
' Show TOS
DMP.6
MBX.!
ENR.

```

Remarks

-

Limitations:

-

See also:

- [DMP.Dump System Values](#)^[1823]

3.42.10.6.10 URL Save File

AIC.URL Save File

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.URL Save File

Download file from a WEB-Link

Intention

This command downloads a file from a web link. If successful, it places a '0' on the Top of Stack (TOS).

In case of an error, a different number will be indicated on the TOS.

This command allows you to specify the filename you wish to use for downloading the file.

Syntax

AIC.URL Save File|P1|P2

Parameter Explanation

P1 - URL of the file to be downloaded

P2 - Filename of the file where the downloaded File from the URL in **P1** should be stored.

On success it will put a "0" on TOS. On Error, another number can be found on TOS.

Example

```

'*****
' AIC.-Sample
'*****
$$URL=https://svn.apache.org/repos/asf/poi/trunk/poi-examples/src/main/java/org,
$$FIL=?path\Myfile.txt
AIC.URL Save File|$$URL|$$FIL
ENR.

```

Remarks

-

Limitations:

-

See also:

-

3.42.10.7 AI - Set Parameters

Using these Commands you can set the Parameters that will be used by all available AI-Systems, no matter if its "Open AI" or "GPT4AI".

Not all AI-Systems use all Parameters.



To get the best result from your AI, you will need to dose the Ingredients like an Expert.

Introduction:

Hello there, young explorer!

Have you ever wondered how robots and magical AI (Artificial Intelligence) creatures think and create amazing stories, poems, and even answer your questions?

It's like they have a little magical potion inside them that helps them think!

Today, we are going on an adventure to discover the magical ingredients of this potion.

So, grab your wizard hat and let's dive into the Magic Book of AI Wonders!

Chapter 1: The Magic Cauldron - AI's Brain!

Imagine AI as a wizard with a giant cauldron. The wizard throws different ingredients into this cauldron to create magical spells. In the AI world, this cauldron is like its brain, and the ingredients are the parameters like Temperature, Number of Results (n), and Max Tokens.

Temperature 🔥:

Imagine the temperature of the cauldron. If it's very hot, the potion will bubble and fizz wildly, and the magic will be unpredictable! In AI, a high temperature means the AI will create wild, creative, and sometimes silly answers. It's like the AI is in a playful mood! If the cauldron is cooler, the potion will be calm, and the magic will be more controlled and sensible. In AI, a low temperature means the AI will give you more focused and reliable answers. It's like the AI is in a serious mood!

Number of Results (AIC.Set Number) 📄:

Now, imagine the wizard wants to create a few different magic spells. The number of results, or "n", is like telling the wizard how many spells you want. If you set "n" to 3, it's like asking the wizard for three different potions or spells. The AI will then give you three different answers or creations!

Max Tokens (AIC.Set MaxToken) 🧰:

Think of tokens as little magical beads. Each word or piece of a word is like one bead. The AI can only carry a certain number of these beads in its magic pouch. "Max Tokens" is like telling the AI how big its pouch is. If you give it a small pouch, it can only make short spells. But if you give it a big pouch, it can create long, enchanting stories! But be careful; if the pouch is too big, the wizard might get tired.

Chapter 2: Mixing The Ingredients 🧪:

Now that you know about the magical ingredients, you can tell the AI wizard how to make its potion! You can ask it to be wild and creative by turning up the temperature, or ask it to make several different spells by setting a higher number for "n". And don't forget to choose the size of the AI's pouch with Max Tokens!

Remember, even AI wizards need practice to make the perfect magic. So, don't be afraid to try different combinations of ingredients!

Chapter 3: Creating Magic ✨:

With your newfound knowledge, you're now ready to create magic with AI! Whether you want to write a fairy tale, solve a math problem, or create a poem, remember that the AI is your friendly wizard, ready to mix its ingredients in the cauldron.

Chapter 4: The Wizard's Library - Different Models 📖:

As we continue our magical adventure, let's step into the wizard's library. Here, the AI wizard keeps all its ancient spell books. Each spell book contains unique magical powers and knowledge. In the world of AI, these spell books are called "models".

GPT (Generative Pre-trained Transformer) 📖:

Imagine a giant, ancient spell book with thousands of pages. It's full of stories, wisdom, and spells. This is like the GPT model. It's really big and smart, and can write amazing stories, answer questions, and even make jokes!

a. GPT-3 ✨:

Think of GPT-3 as the third edition of the spell book. It's like a wise, old tree with deep roots and thousands of leaves. It knows a lot because it has read so many books, websites, and all sorts of texts. GPT-3 can help you write essays, create poems, answer trivia questions, and much more. It's pretty magical!

In the SPR you can use these cheaper Models using the "AIC.Ask Completion" or you can use GPT 3.5 via "AIC.Ask GPT4All" Command.

b. GPT-3.5 Turbo 🚀 :

Imagine if the wise, old tree could also run super fast! That's GPT-3.5 Turbo. It's like GPT-3, but with extra magical energy. This extra energy helps it to perform more tasks and even makes it faster and more efficient. It's perfect for when you need a quick and smart answer!

In the SPR you can use the "AIC.Ask Chat" or you can use GPT 3.5 via "AIC.Ask GPT4All" Command.

c. GPT-4 📖 :

Now imagine a spell book so large that it looks like a galaxy with stars and planets! GPT-4 is like this galaxy. It's even bigger and more powerful than GPT-3. With more knowledge and magic, GPT-4 can understand complex ideas and create even more amazing stories. It's like having an entire universe of wisdom at your fingertips!

In the SPR you can use the "AIC.Ask Chat" or you can use GPT4 via "AIC.Ask GPT4All" Command.

DALL-E 🎨 :

Imagine a colorful, enchanted paintbrush that can paint anything you can imagine! DALL-E is like this magical paintbrush. Instead of words and sentences, DALL-E creates images from the text. If you ask DALL-E to paint a picture of a "sunset over a candy land," it will create a beautiful painting just like that!

In the SPR you can use this feature using the "AIC.Generate Image" Command.

ChatGPT 💬 :

Now, imagine a chatty parrot that loves to talk and have conversations. ChatGPT is like this parrot. It's designed to have conversations with people. You can ask it questions, chat about your day, or even ask for advice, and ChatGPT will happily chat back.

In the SPR that equals the "AIC.Ask Chat" and other "AIC.Ask*" Commands.

Remember, each spell book or magical creature in the wizard's library has its own special powers. Just like in magic, choosing the right model in AI helps the wizard create the most wonderful spells and answers for what you need.

Now, young explorer, with the knowledge of the magic cauldron and the wizard's library, you are ready to create wonders! May your quill never run dry and your imagination forever soar through the boundless skies! 🚀 📖 ✨

May your adventures be full of wonder and discovery! 🌿 ✨

3.42.10.7.1 Save_Key

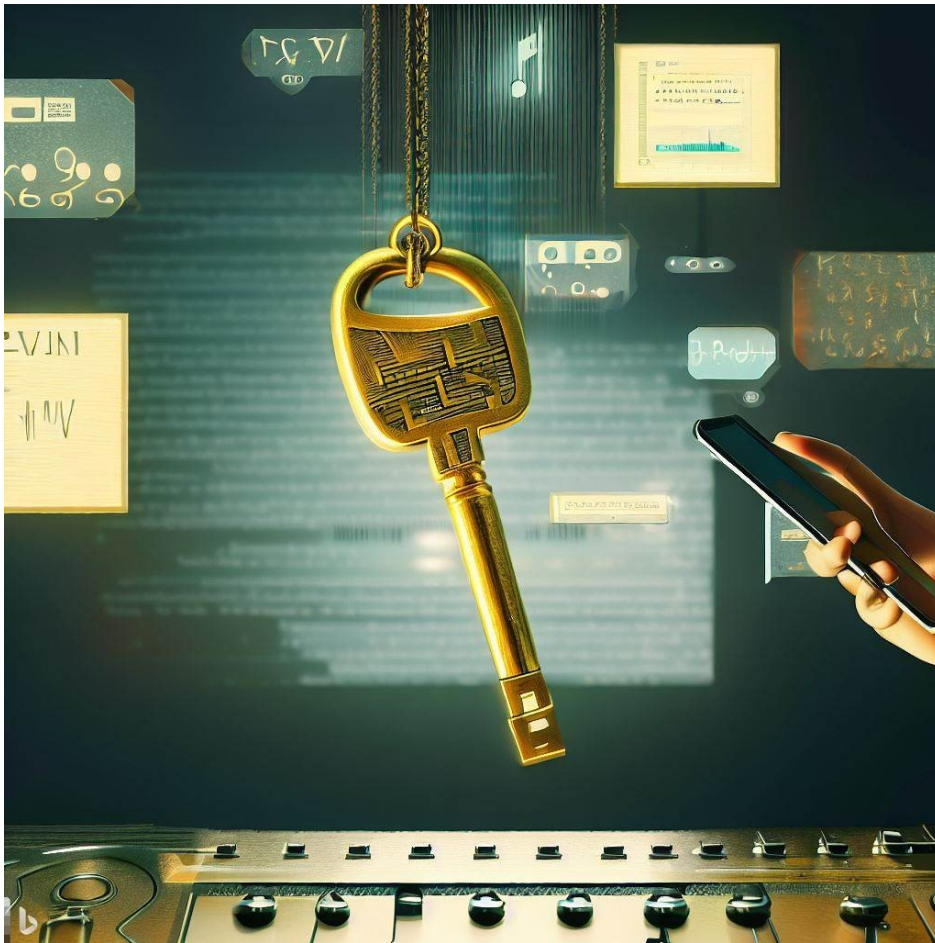
AIC.Save Key

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Save Key

Save OpenAI-API Key encrypted to a file



Intention

The `AIC.Save Key` Command is an essential tool for developers venturing into the world of OpenAI.

It acts as a secure vault, allowing you to store your API Key in an encrypted format within a file.

This not only bolsters security but also streamlines the process of utilizing the key across various scripts.

Utilizing the Save Key Command

To employ the 'Save Key Command', you need to invoke the `AIC.Save_Key` function.

Pass your API Key as the primary argument.

Additionally, you can specify a file path to determine where the encrypted key should be stored, while this optional.

```
AIC.Save_Key <YOUR_API_KEY> [ |OPTIONAL_FILE_PATH]
```

In the event that a file path is not specified, the command will default to saving the file in the directory where the script or executable is situated.

The default path is "?exeloc\AIC_License_Key.dat"

Default Naming Convention and Location

The encrypted file is conventionally named `AIC_License_Key.dat`.

This standard naming practice ensures easy identification.

If a file path is omitted, the file will be created in the directory denoted by `?exeloc\`, which corresponds to the location of the script or executable.

The Significance of the Save Key Command

Using the `Save Key Command` to store the API Key in an encrypted file is highly advisable. This approach significantly reduces the risk of unintentional exposure and provides a convenient method for reusing the key in different scripts.

Wrapping Up

The `Save Key Command` is a powerful and indispensable tool for safeguarding your API Key. By storing it in an encrypted file, you ensure its protection and facilitate its use across your OpenAI projects.

```
' Script 1: Save the Key to the file "AIC_License_Key.dat".
```

```
' IMPORTANT: This API-Key is a phantasy API-Key it must be replaced with your API Key
  $$KEY=sk-abcdefghijklmnopqrstuvwxyz123456
```

```
' Here we save the Keyfile at the default path, that is:
```

```
' ?exeloc\AIC_License_Key.dat
```

```
AIC.Save_Key|$$KEY
```

```
ENR.
```

```
' Script 2; Using the crypted API-Keyfile
```

```
' Test if we are online, AI-Commands will only work if you are online.
```

```
NOL.
```

```
GTO.enx
```

```
EIF.
```

```
' Set OpenAI API-Key from the saved File
```

```
AIC.SetKey|from_File
```

```
' Set Model
```

```
AIC.SetModel_Completion|4
```

```
' Set Model-Temperature
```

```

AIC.Set_Temperature|0

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000 )
' The more Tokens you use the more you need to pay.
AIC.SetMax_Token|25

' Ask Question and receive answer to $$RET
AIC.Ask_Completion|Wieviel Uhr ist es?|$$RET
DBP.$$RET

:enx
ENR.

```

Syntax

AIC.SaveKey|P1 [|P2]

AIC.Svk|P1 [|P2]

Parameter Explanation

P1 - See here [Open AI API-Key](#)^[470]

P2 - opt. Filepath for the API-Key to save

Example

```

*****
' EXAMPLE 1: AIC.-Commands
*****
' Script 1: Save the Key to the file "AIC_License_Key.dat".

' IMPORTANT: This API-Key is a phantasy API-Key it must be replaced with your AI
$$KEY=sk-abcdefghijklmnopqrstuvwxy123456

' Here we save the Keyfile at the default path, that is:
' ?exeloc\AIC_License_Key.dat
AIC.Save_Key|$$KEY
ENR.

```

Remarks

-

Limitations:

🗡️ Safeguarding Your API Key: A Knight's Guide to OpenAI's T

Greetings, Noble Coder! 🏰

Embarking on a quest through the enchanted forests of OpenAI?

Before you mount your steed, there's a sacred artifact you must secure - **the illustrious API Key**.

This key is not just a string of characters; it's the heart of your adventure, the magic that unlocks the kingdom's secrets.

*****The Enchanted Encryption*****

Ah, you seek to protect your treasure by locking it within an enchanted file. Wise as it may seem, remember, even the mightiest spells have their counters. The encryption wards off common thieves, but against a sorcerer with the dark SPR arts, it may falter.

*****Guard Your Treasure Chest*****

Distributing your encrypted key is akin to leaving your treasure chest in the dragon's lair. Any rogue with an SPR spellbook can break the enchantment.

The treasure within - your API Key - is bound to your very essence, your account. In the wrong hands, it can unleash storms and deplete your resources.

*****Set Magical Boundaries*****

Fear not, for there is a spell to shield your treasure further.


Within the hallowed halls of OpenAI's website, you can weave a spell to set limits on your API Key's powers.

This incantation ensures that even if your key is seized, its magic is bound, and the havoc it can wreak is contained.

*****The Knight's Code*****

- **Guard the Key****: Never let your API Key, even if enchanted in a file, sail on uncharted waters.
- **Summon Guardians****: Create a mystical barrier through a backend service. Let this guardian use the API Key in the shadows, far from prying eyes.
- **Eternal Vigilance****: Watch over your domain. Keep an eagle's eye on the usage of your API Key.
- **Weave Protective Spells****: Visit OpenAI's sacred grounds and set limits upon your API Key. Bind it, so its magic doesn't turn against the kingdom.

Remember, brave knight, with great power comes great responsibility. Your API Key is the magic that courses through the veins of your quests. Guard it, protect it, and let it guide you through countless adventures in the realm of OpenAI.

Onward, to glory! 

See also:

- [Set Key](#)
-

3.42.10.7.2 Set Best of

AIC.Set Best of

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set Best of

Internally evaluate multiple results and deliver the best of these

```
{
  "error": {
    "message": "Unrecognized request argument supplied: best_of",
    "type": "invalid_request_error",
    "param": null,
    "code": null
  }
}
```

"Best of" as an Parameter will only work with the Command: AIC.Ask Completion it will not work with any other Command.

Intention

Due to Open AI specification, this command is only valid for the "Completion Endpoint".

Which is the Command: AIC.Ask Completion. It will not effect other "Ask.-Commands"

The **"best of"** parameter allows you to control the number of response candidates generated by the model during an AI call.

Instead of a single evaluation, you can specify the "best of" value To indicate how many response candidates you want the model to generate.

The call will then rank these responses and returns the top-ranked response from the candidates based on the model's scoring.

By generating multiple candidates, you can have a more varied set of potential responses. And choose the one that best fits your needs Or preferences.

For example, If you set "best_of" To 3, the model will produce three response candidates, And you can Select the most suitable one based on relevance, quality, Or any other criteria.

Keep IN mind that setting a higher value For "best_of" increases the computational cost And response time of the Call, as generating multiple candidates requires more processing time.

Therefore, it's important to strike a balance between the number of candidates and the trade-off in API performance.

The "best_of" parameter in the OpenAI API accepts integer values greater than or equal to 1.

The value you choose For "best_of" determines the number of response candidates generated by the model.

For example, If you set "best_of" To 3, the API CALL will produce three potential responses, And the API will Return the highest-ranked response among those candidates.

The default value for "best_of" in the OpenAI API is 1, meaning that by default, the API returns a single response without generating additional candidates.

Performance: A higher "best_of" value increases computational cost and response time since it requires generating and scoring multiple candidates.

Therefore, it's important To balance the desired response quality With the associated performance impact.

Response Variation: If you want more diverse And varied responses, a higher "best_of" value can be beneficial As it increases the likelihood of different candidates.

This can help you explore different perspectives Or generate more creative outputs.

Syntax

AIC.Set Best of | [P1]

AIC.sbo | [P1]

Parameter Explanation

P1 - (optional) Number of internal versions that will be rated.

Example

```

'*****
' AIC.Best of-Sample
'*****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' Set Model-Temperature
AIC.Set_Temperature|0

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000)
' The more Tokens you use the more you need to pay. But the longer Input and Output
AIC.SetMax_Token|300
AIC.Set Echo|1
AIC.Set Top_P|2
AIC.Set best of|2

$$TXT=Tell me about this formula and what it has to do with cars:$CrLf$ '$z^n =
GSB.compl

' Using the Chat-Endpoint the "Best-of" Parameter will be ignored to prevent error
GSB.chat

ENR.
'-----
:compl
AIC.SetModel_Completion|text-davinci-003

```

```

AIC.Ask Completion|$$TXT|$$RET
DBP.-----
DBP. Answer
DBP.$$RET
CLP.$$RET
DBP.-----
AIC.gro|$$REA
DBP. RAW
DBP.$$REA
DBP.-----
DBP.JSON Output
AIC.Gts|9|$$JSO
DBP.$$JSO
DBP.-----
RET.
'-----
:chat
AIC.SetModel_Chat|0
AIC.Ask Chat|$$TXT|$$RET
DBP.-----
DBP. Answer
DBP.$$RET
CLP.$$RET
DBP.-----
AIC.gro|$$REA
DBP. RAW
DBP.$$REA
DBP.-----
DBP.JSON Output
AIC.Gts|9|$$JSO
DBP.$$JSO
DBP.-----
RET.

```

Remarks

If the Open AI closes the "Completion Endpoint" then this Parameter will be useless.

Limitations:

"Best of" as an Parameter will only work with the Command: AIC.Ask Completion
it will not work with any other Command.

See also:

- [1.6.1. Program Flow Control](#)^[134]
- [! Smart Package Robot 's Parallel Robot Operations](#)^[3745]
- [1.5. Features and Hints](#)^[93]

3.42.10.7.3 Set Endpoints

TMP. - *Template*[Previous](#) 1492 [Top](#) 49 [Next](#) 1499

MiniRobotLanguage (MRL)

TMP.

If Result available

Intention

Conditional Statement.

Test if a specified Parallel-Robot or Background Operation has released a Result.

IRS. does not really check if the Operation is ongoing. It will just check if there is a Result from that "Ticket Number" available.

Syntax**TMP.** [**P1**] ... **ELS.** ... **EIF.****Parameter Explanation****P1** - (*optional*) Is the "Ticket Number" of a running Parallel Robot or background Operation. It can be omitted then the last "Ticket Number" that was emitted is been taken.

IRS. can be nested to unlimited Depth, and they can also enclose Sub-Programm Calls or FEX. (Enumerations) to unlimited Depth.

Example

```

| *****
| IRS.-Sample
| *****

```

Remarks

-

Limitations:

-

See also:

- [1.6.1. Program Flow Control](#)^[134]
- [! Smart Package Robot 's Parallel Robot Operations](#)^[3745]
- [1.5. Features and Hints](#)^[93]

3.42.10.7.3.1 Set Chat Endpoint

[AIC.Set Chat Endpoint](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set Chat Endpoint

Set the http: - Endpoint for the Open AIC.Ask Chat Command

Intention

Using this command you can alter the Endpoint" of the AIC.Ask Chat and the AIC.Ask Chat Array Commands.

The default endpoint is:

```
https://api.openai.com/v1/chat/completions
```

If you want to use another endpoint, maybe because Open AI did changes, you can set the alternative endpoint using this command.

This Command will change the Endpoint only for the Commands: AIC.Ask Chat and the AIC.Ask Chat Array

Syntax

AIC.Set Chat Endpoint[|P1]

Parameter Explanation

P1 - (optional) URL of the Endpoint to use.

Example

-

Remarks

-

Limitations:

Setting an unreachable Endpoint may lead to undefined results, the Script may even end without an error-message.

See also:



3.42.10.7.3.2 Set Completion Endpoint

[AIC.Set Completion Endpoint](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set Completion Endpoint

Set the http: - Endpoint for the Open AIC.Ask Completion Command

Intention

Using this command you can alter the Endpoint" of the AIC.Ask Completion Commands.

The default endpoint is:

```
https://api.openai.com/v1/completions
```

If you want to use another endpoint, maybe because Open AI did changes, you can set the alternative endpoint using this command.

This Command will change the Endpoint only for the Command: AIC.Ask Completion

Syntax

```
AIC.Set Completion Endpoint[ | P1 ]
```

Parameter Explanation

P1 - (optional) URL of the Endpoint to use.

Example

-

Remarks

-

Limitations:

Setting an unreachable Endpoint may lead to undefined results, the Script may even end without an error-message.

See also:

-

3.42.10.7.3.3 Set Dalle Endpoint

[AIC.Set Dalle Endpoint](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set Dalle Endpoint

Set the http: - Endpoint for the DALL-E Image Generation Command

Intention

This command allows you to specify the endpoint for the DALL-E Image Generation operations. The default endpoint is:

```
`https://api.openai.com/v1/images/generations`
```

If there's a need to change to a different endpoint in the future, this command enables that modification. It only affects the DALL-E Image Generation command.

Syntax

AIC.Set Dalle Endpoint[|P1]

Parameter Explanation

P1 - (*optional*) URL of the Endpoint to use. Currently, only the default endpoint is allowed. If **P1** is omitted, the default Endpoint is set.

Example

```
'*****  
' AIC.-Sample  
'*****  
' Set the Dalle endpoint to the default  
AIC.Set Dalle Endpoint
```

Remarks

Currently, only the default endpoint is allowed.

Limitations:

Setting an incorrect or unreachable endpoint may lead to errors or undefined results.

See also:

-

3.42.10.7.3.4 Set TTS Endpoint

[AIC.Set TTS Endpoint](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set TTS Endpoint

Set the http: - Endpoint for the Text-to-Speech Command

Intention

This command is used to set the endpoint for Text-to-Speech (TTS) operations. The default endpoint is:

```
`https://api.openai.com/v1/audio/speech`
```

This command can be used to update the endpoint if necessary, specifically for TTS operations.

Syntax

AIC.Set TTS Endpoint[|P1]

Parameter Explanation

P1 - (*optional*) URL of the Endpoint to use. Currently, only the default endpoint is allowed. If **P1** is omitted, the default Endpoint is set.

Example

```
'*****
' AIC.-Sample
'*****
' Set the TTS endpoint to the default
AIC.Set TTS Endpoint
```

Remarks

Currently, only the default endpoint is allowed.

Limitations:

Setting an incorrect or unreachable endpoint may lead to errors or undefined results.

See also:

-

3.42.10.7.3.5 Set Vision Endpoint

[AIC.Set Vision Endpoint](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set Vision Endpoint

Set the http: - Endpoint for the Open AI Vision Commands

Intention

This command is designed to define the endpoint for vision-related AI operations. The default endpoint is:

```
`https://api.openai.com/v1/chat/completions`
```

It allows for the endpoint to be updated as necessary, affecting only vision-related commands.

Syntax

AIC.Set Vision Endpoint[|P1]

Parameter Explanation

P1 - (*optional*) URL of the Endpoint to use. Currently, only the default endpoint is allowed. If **P1** is omitted, the default Endpoint is set.

Example

```
'*****
' AIC.-Sample
'*****

' Set the Vision endpoint to the default
AIC.Set Vision Endpoint
ENR.
```

Remarks

Currently, only the default endpoint is allowed.

Limitations:

Setting an incorrect or unreachable endpoint may lead to errors or undefined results.

See also:

-

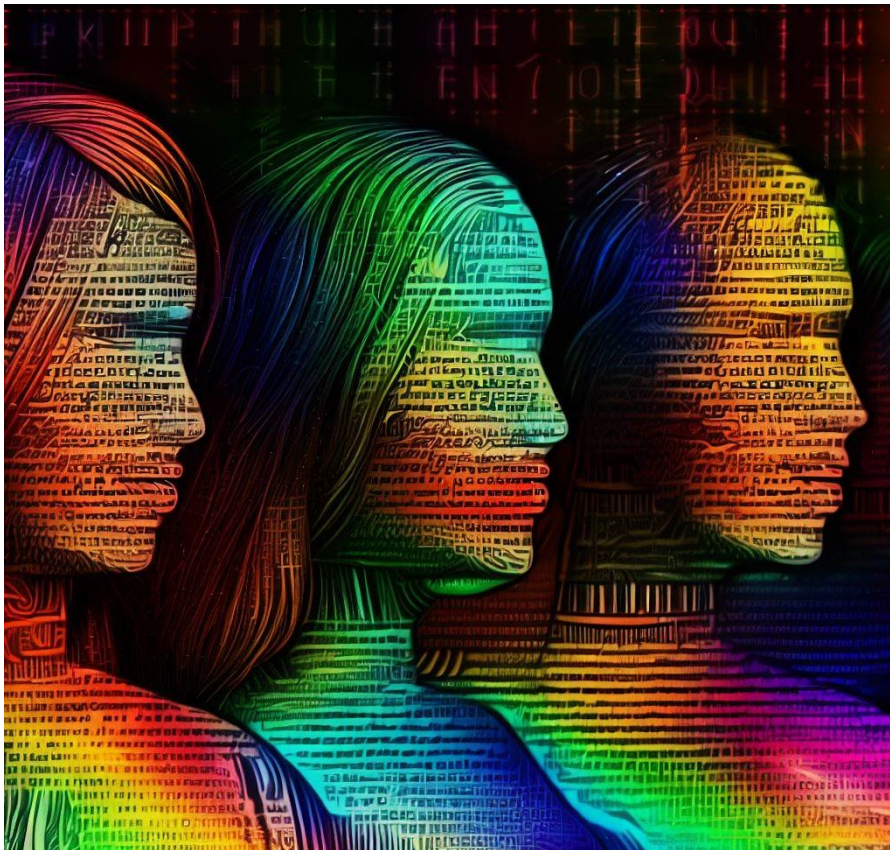
3.42.10.7.4 Set Number

[AIC.Set Number](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set Number

Set the Amount of alternative results you want to have generated.



Intention

The `AIC.Set Number` command in the Smart-Package Robot (SPR) allows users to set the number of outputs to be generated when using OpenAI endpoints.

This command is versatile and can be used across all OpenAI endpoints, including:

- text generation,
- image generation,
- and more.

Setting the number of outputs is particularly useful when you want to generate multiple variations of content based on a single prompt.

Syntax:

```
AIC.Set Number | <Number>
```

Parameters:

<Number>: An integer value representing the number of outputs you want to generate. This number should be greater than or equal to 1.

Note that setting a very high number may have cost implications and might be subject to rate limits imposed by the OpenAI API.

Example Usage:

```
AIC.Set Number|5
```

This example sets the number of outputs to be generated to 5. This means that when you issue a command to generate content (e.g., text, images), it will produce 5 different outputs.

Important Considerations:

Costs: Generating multiple outputs might have cost implications. OpenAI may charge based on the number of tokens (for text) or images generated. Be sure to understand the pricing structure of the OpenAI API to avoid unexpected charges.

Rate Limits: OpenAI imposes rate limits on its API. Generating a large number of outputs in a short time frame might cause you to hit these rate limits. It is recommended to check the OpenAI documentation for the latest information on rate limits.

Quality of Outputs: Generating a higher number of outputs might give you more options to choose from, but it's important to keep in mind that not all outputs will necessarily meet the desired quality or relevance. Review the generated content carefully.

```
' Set OpenAI API-Key from the saved File
AIC.SetKey|File
```

```
' This will make the AI to generate 3 Pictures
AIC.SetNumber|5
```

```
' Can be 256 or 512 or 1024
```

```
AIC.Set Image Size|256
```

```
$$TXT=Chessboard with a golden and silver figures in an ga
```

```
AIC.Generate Image|$$TXT
```

```
ENR.
```

Syntax

AIC.Set Number|P1

AIC.SNB[|P1]

Parameter Explanation

P1 - opt. Variable or numeric value, this is the number of alternative results, the AI should produce.

Example

```

'*****
' EXAMPLE 1: AIC.-Commands
'*****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' Set Model
AIC.SetModel_Completion|4

' Set Model-Temperature
AIC.Set_Temperature|0

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000
' The more Tokens you use the more you need to pay.
AIC.SetMax_Token|25

' We will create 4 alternative answers
AIC.Set Number|4

' Ask Question and receive answer to $$RET
AIC.Ask_Completion|What is a "Windows Button"?|$$RET
MBX.$$RET

:enx
ENR.

```

Remarks

-

Limitations:

-

See also:

- [Set_Key](#)⁷⁹⁹
- [Ask_Chat](#)⁸³³
- [Ask_Completion](#)⁸⁴²

3.42.10.7.5 Set Presence Penalty

AIC.Set Presence

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set Presence

Setting this Parameter will tell AI to avoid repetitions

Intention

The "presence_penalty" parameter controls how much the OpenAI API model **avoids generating repetitive** or redundant responses in the conversation history.

The "presence_penalty" parameter typically accepts values between **-2.0** and **2.0**, where **0.0** means **no penalty** for repetitive responses and 1.0 means the model will strongly avoid repetition.

The default value for "presence_penalty" is usually 0.6, but it's recommended to experiment with different values based on the desired level of response repetition in your specific use case.

If omitted or set to 99 it will be omitted and will not be present in the Code. This is recommended if you do not need this parameter.

Syntax

AIC.Set Presence [| P1]

Parameter Explanation

P1 - (optional) A numeric value between -2 and 2. If omitted or set to 99 the Parameter will not be used and the AI can use internal Presets.

A good default value is between 0 and 0.6.

Example

```
| *****  
|  
| *****
```

Remarks

-

Limitations:

-

See also:

-

3.42.10.7.6 Set Role

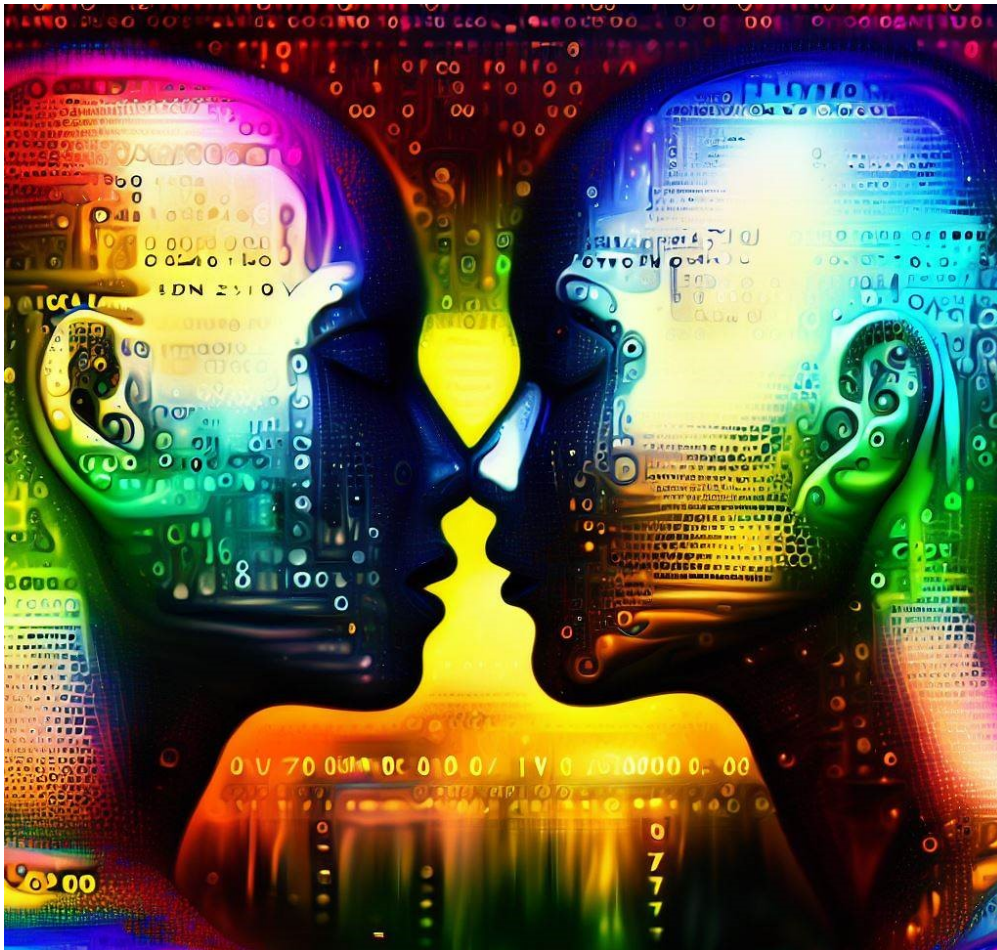
AIC.Set Role

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set Role

Set the desired Role for the Ask-Chat Command.



Intention

In the input and output JSON structure of the chat endpoint from OpenAI, the "role" field specifies the role of the message sender.

This plays a role with the "AIC.Ask Chat" Command, because it returns some of its Output in a variable called "Role".

Therefore you need to use the "AIC.Set Role" and the "AIC.Get Role" to set and get this information.

The possible values for the "role" field are:

- "**system**": This value indicates that the message is a system message. **System** messages are usually used to set the behavior of the assistant at the beginning of the conversation.

For example, a system message might instruct the assistant to speak like Shakespeare.

- **"user"**: This value indicates that the message is from the user. **User** messages are the prompts or questions that the user wants the assistant to respond to.
- **"assistant"**: This value indicates that the message is from the assistant. **Assistant** messages are the responses generated by the AI in reply to the user's prompts or questions.

The **"role"** field is important as it helps in distinguishing between different types of messages in the conversation and understanding the flow of the conversation.

You can set the Role of an "AIC.Ask Chat" Command using the "AIC.Set Role"-Command.

Important Note: The default/recommended role to set for AIC.Ask Chat is **User**. Using other roles may lead to undefined results.

Using the Role **"Assistant"** for your prompt may lead to shorter answers (see Sample-Script below).

No matter what role you specify, the AI will typically answer using the role **"Assistant"**.

The "AIC.Set Role" command and the "AIC.Get Role" command operate on different registers; each command has its own dedicated register.

The "AIC.Set Role" command is utilized during a request to set a specific role. On the other hand, the "AIC.Get Role" command is used to retrieve the role that has been returned from the **OpenAI** server.

This distinction is crucial as the "AIC.Set Role" command is for specifying a role before sending a request, while the "AIC.Get Role" command is for obtaining the role information from the response received from **OpenAI**.

Options of Use:

a) You can specify a number 1,2 or 3. In this case the number will evaluate internally to these roles:

- 1 - System
- 2 - User
- 3 - Assistant

b) You can type the first letter of each role. In this case the letter (lower or uppercase) will evaluate internally to these roles:

- S** - System
- U** - User
- A** - Assistant

c) You can type the role (see below). Most typing mistakes should be ignored and the right role should be assigned. You can also type other Roles not listed here, then these roles will be used.

System, Assistant, User ..

Syntax:

```
AIC.Set Role|1
AIC.Set Role|u
AIC.Set Role|Assistant
```

Parameters:

<Role-Identifier>: "Assistant" or "User" or "System", alternative Options to specify the Role see above. All letters will be treated as lowercase. The default role is "user".

Syntax

AIC.Set Role [|P1]

AIC.sro [|P1]

Parameter Explanation

P1 - opt. "Assistant" or "User" or "System", alternative Options to specify the Role see above. All letters will be treated as lowercase. The default role is "user".

*Here is a Sample of the original returned JSON Data from the Chat-Endpoint.

```
{
  "id": "chatcmpl-7Zbc78I0ULOEdP1RfHTYPzRxsZSYZ",
  "object": "chat.completion",
  "created": 1688720424,
  "model": "gpt-3.5-turbo-0613",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "\\Imagine a world where technology has advanced to the point"
      },
      "finish_reason": "length"
    }
  ],
  "usage": {
    "prompt_tokens": 20,
    "completion_tokens": 30,
    "total_tokens": 50
  }
}
```

Example

```

'*****
' EXAMPLE 1: AIC.-Commands
'*****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' Set Model
AIC.SetModel_Chat|0

' Set Model-Temperature
AIC.Set_Temperature|0.5

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000)
' The more Tokens you use the more you need to pay. But the longer Input and Output
AIC.SetMax_Token|300

' Ask Question and receive answer to $$RET
$$QUE=Tell me what is X=3+5?

AIC.Set Role|User
AIC.Ask_Chat|$$QUE|$$REA
DBP.$$REA

AIC.Set Role|Assi
AIC.Ask_Chat|$$QUE|$$REC
DBP.$$REC

AIC.Set Role|System
AIC.Ask_Chat|$$QUE|$$RED
DBP.$$RED

:enx

```

Result:

```

[14:11:01] X = 3 + 5 is a mathematical equation. When you add 3 and 5 together,
[14:11:02] X = 8
[14:11:03] X = 3 + 5 is a mathematical equation. When you add 3 and 5 together,

```

Remarks

Important Note: The "AIC.Set Role" command and the "AIC.Get Role" command operate on different registers; each command has its own dedicated register.

The "AIC.Set Role" command is utilized during a request to set a specific role. On the other hand, the "AIC.Get Role" command is used to retrieve the role that has been returned from the OpenAI server.

This distinction is crucial as the "AIC.Set Role" command is for specifying a role before sending a request, while the "AIC.Get Role" command is for obtaining the role information from the response received from OpenAI.

Limitations:

-

See also:

- [Set_Key](#)⁷⁹⁹
- [Ask_Chat](#)⁸³³
- [Ask_Completion](#)⁸⁴²

3.42.10.7.7 Set Top K

[AIC.Set Top_K.](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set Top_K.

Set the Top_K Value in the LLM

Intention

The `top_k` parameter in OpenAI's GPT-3 API is **used to limit the number of tokens considered** for each step of the generation process.

It's a form of stochastic truncation, helping to focus the model's predictive capabilities on a smaller subset of its vocabulary.

The minimum value for `top_k` is 0.

This means the model will consider all possible tokens for each step, leading to a highly diverse and potentially less focused output.

There isn't a strict maximum value for `top_k`, but it should be noted that setting `top_k` to a very high value (greater than the size of the model's vocabulary) is effectively the same as not setting it at all, as the model will consider all possible tokens for each step.

If the `top_k` parameter is not provided in the call, the model will use its default behavior, which is to consider all tokens in its vocabulary.

This is equivalent to setting `top_k` to a value larger than the size of the model's vocabulary (larger 30.000).

While there's no strict rule for what values of `top_k` are "regular" or most commonly used, many developers find that values in the range of 20 to 50 provide a good balance between diversity and focus in the model's responses.

For instance, a `top_k` value of 40 means that at each step, the model will consider the top 40 most likely next words based on its internal calculations.

Here is a more detailed explanation and comparison of the ``top_k``, ``top_p``, and ``temperature`` parameters in the context of OpenAI's GPT-3 API.

When GPT-3 generates text, it does so word by word. For each word it generates, it calculates a probability for every word in its vocabulary, and then selects the next word based on these probabilities. The ``top_k``, ``top_p``, and ``temperature`` parameters are all ways to influence this selection process.

- ``top_k``: This parameter limits the number of words that the model considers as the next possible word. If ``top_k`` is set to 50, for example, the model will only consider the 50 words it thinks are most likely. This can make the output more focused and less random, because it's only choosing from a subset of words. However, it can also make the output less diverse, because it's ignoring a lot of potential words. The ``top_k`` value can be any non-negative integer, with larger values leading to more

randomness and smaller values leading to less randomness. If `top_k` is not set, the model considers all possible words.

2. `top_p`: Also known as nucleus sampling, this parameter is a bit more dynamic. Instead of always considering a fixed number of words like `top_k`, `top_p` considers however many words are needed to reach a certain cumulative probability. For example, if `top_p` is set to 0.9, the model will consider the smallest set of words that have a combined probability of 90%. This set of words can be larger or smaller depending on the specific probabilities for each word. Like `top_k`, `top_p` can make the output more focused and less random, but it can also reduce diversity. The `top_p` value is a float between 0 and 1, with larger values leading to more randomness and smaller values leading to less randomness.

3. `temperature`: This parameter controls the "sharpness" of the probability distribution. If `temperature` is set to a high value (close to 1), the model's word selection will be more random and less deterministic, even if some words have much higher probabilities than others. If `temperature` is set to a low value (close to 0), the model's word selection will be more deterministic and less random, with the model strongly favoring words that have higher probabilities. In other words, a high `temperature` makes the model more "adventurous" in its word choices, while a low `temperature` makes the model more "conservative".

In summary, `top_k` and `top_p` are ways to limit the number of words that the model considers for each step of the generation process, while `temperature` is a way to control the randomness of the model's word selection within those limits. All three parameters can be used together to finely tune the behavior of the model. The optimal values for these parameters can depend on your specific use case and the desired behavior of the model. It's a good idea to experiment with different values to see what works best for your needs.

Syntax

```
AIC.Set Top_K[|P1]
AIC.STK[|P1]
```

Parameter Explanation

P1 - (optional) numeric value, 0 to 30000. If omitted or -1, then the parameter is not used therefore the System will use internal default values.

Example

```
! *****
!
! *****
```

Remarks

-

Limitations:

-

See also:

-

3.42.10.7.8 Set Top P

[AIC.Set Top_P.](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set Top_P

Set the Top_P Value in the LLM

Intention

The ``top_p`` parameter, also known as nucleus sampling or ``p`-sampling`, is another way to introduce randomness into the model's responses. Instead of selecting from the top ``k`` most likely next words, ``top_p`` sampling chooses from the smallest possible set of words whose cumulative probability exceeds the ``p`` threshold.

The ``top_p`` value is a float between 0 and 1.

- A ``top_p`` value of 0.0 would make the model deterministic, always choosing the single most likely next word.
- A ``top_p`` value of 1.0 means the model considers all possible next words, leading to maximum diversity in the output.

In practice, ``top_p`` values between 0.7 and 0.9 are often used to balance between diversity and relevance of the output. For example, a ``top_p`` of 0.85 means the model will randomly choose the next word from a set of top candidates that have a combined probability of 85%.

As with ``top_k``, the optimal ``top_p`` value can depend on your specific use case and the desired behavior of the model.

Generally its not recommended to use Top_P together with the 'temperature' value as the result is difficult to predict.

Here is a more detailed explanation and comparison of the ``top_k``, ``top_p``, and ``temperature`` parameters in the context of OpenAI's GPT-3 API.

When GPT-3 generates text, it does so word by word. For each word it generates, it calculates a probability for every word in its vocabulary, and then selects the next word based on these probabilities. The ``top_k``, ``top_p``, and ``temperature`` parameters are all ways to influence this selection process.

1. ``top_k``: This parameter limits the number of words that the model considers as the next possible word. If ``top_k`` is set to 50, for example, the model will only consider the 50 words it thinks are most likely. This can make the output more focused and less random, because it's only choosing from a subset of words. However, it can also make the output less diverse, because it's ignoring a lot of potential words. The ``top_k`` value can be any non-negative integer, with larger values leading to more randomness and smaller values leading to less randomness. If ``top_k`` is not set, the model considers all possible words.

2. ``top_p``: Also known as nucleus sampling, this parameter is a bit more dynamic. Instead of always considering a fixed number of words like ``top_k``, ``top_p`` considers

however many words are needed to reach a certain cumulative probability. For example, if `top_p` is set to 0.9, the model will consider the smallest set of words that have a combined probability of 90%. This set of words can be larger or smaller depending on the specific probabilities for each word. Like `top_k`, `top_p` can make the output more focused and less random, but it can also reduce diversity. The `top_p` value is a float between 0 and 1, with larger values leading to more randomness and smaller values leading to less randomness.

3. `temperature`: This parameter controls the "sharpness" of the probability distribution. If `temperature` is set to a high value (close to 1), the model's word selection will be more random and less deterministic, even if some words have much higher probabilities than others. If `temperature` is set to a low value (close to 0), the model's word selection will be more deterministic and less random, with the model strongly favoring words that have higher probabilities. In other words, a high `temperature` makes the model more "adventurous" in its word choices, while a low `temperature` makes the model more "conservative".

In summary, `top_k` and `top_p` are ways to limit the number of words that the model considers for each step of the generation process, while `temperature` is a way to control the randomness of the model's word selection within those limits. All three parameters can be used together to finely tune the behavior of the model. The optimal values for these parameters can depend on your specific use case and the desired behavior of the model. It's a good idea to experiment with different values to see what works best for your needs.

Syntax

```
AIC.Set Top_P [|P1]
AIC.STP [|P1]
```

Parameter Explanation

P1 - (optional) numeric value, between 0 and 1. If omitted or -1, then the parameter is not used therefore the System will use internal default values.

Example

```
| *****
|
| *****
```

Remarks

-

Limitations:

-

See also:

-

3.42.10.7.9 Set_Key

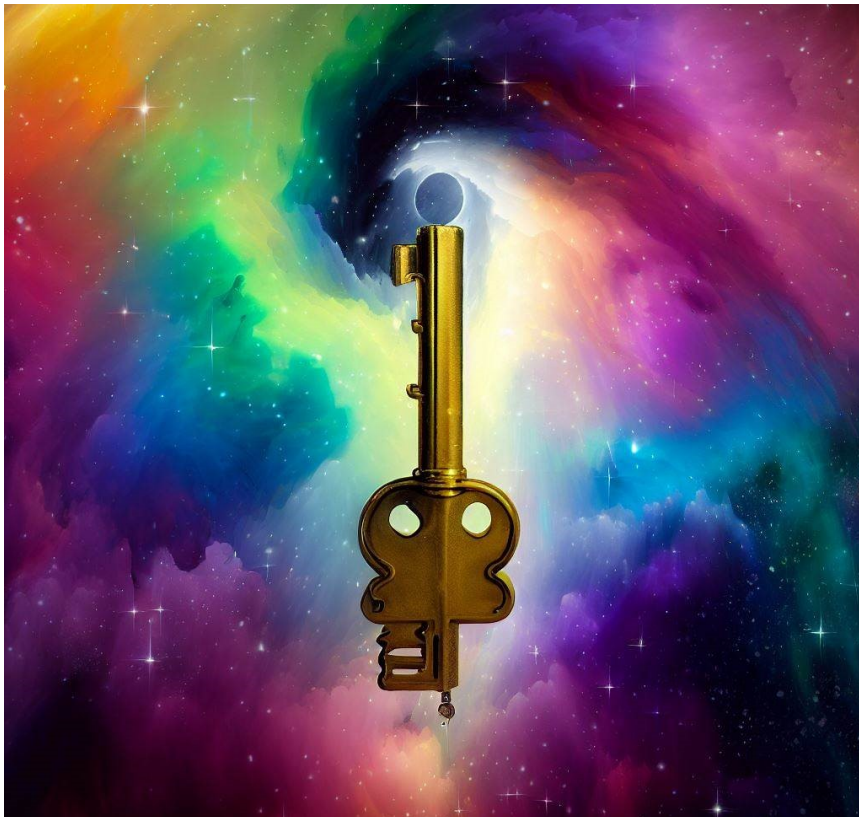
AIC.SetKey

[Previous](#) 2027 | [Top](#) 49 | [Next](#) 2069

MiniRobotLanguage (MRL)

AIC . SetKey

Set OpenAI-API Key



Intention

SetKey Command: Initiating Your Script with a License Key

Before diving into the world of AI scripting, it's essential to set the stage with the SetKey command.

This command is the golden ticket that grants you access to the plethora of AI functionalities offered by **OpenAI**.

Let's break down how to use it effectively.

What is the SetKey Command?

The SetKey command is the first command you need to include at the beginning of your script.

It's like the key to a treasure chest; without it, you can't unlock the AI capabilities you're after.

Specifically, it defines [AI - Artificial Intelligence Commands](#) 470.

Why Do You Need It?

You might be wondering why there's a need for such a key.

The reason is that the AI functionalities you are looking to use are not processed locally on your computer.

Instead, they are handled remotely in the high-powered OpenAI Cloud.

This License Key ensures that you have the proper [authorization to access these cloud-based services](#)^[470].

How to Obtain the License Key?

To get your hands on this key, you'll need to visit OpenAI's official website.

Once there, follow the instructions to register and obtain the License Key.

Keep this key safe, as you'll need it every time you want to use OpenAI's services.

How to Use the SetKey Command?

Once you have your License Key, it's time to put it to use.

At the very start of your script, include the AIC.SetKey command followed by your License Key.

This will authenticate your script with **OpenAI's cloud services**.

On the long run you may prefer the second Option (see below) that will encrypt your key into a file and use this file.

The AIC.Set_Key-Command has multiple Usage Options.

1. You can directly set the Key in the Script.

Here is a Sample Script that shows how this is done.

```
' IMPORTANT: This API-Key is a phantasy API-Key it must be
replaced with your API-Key.
$$KEY=sk-abcdefghijklmnopqrstuvwxyz123456

' Test if we are online, AI-Commands will only work if you are
online.
NOL.
  GTO.enx
EIF.

' Set OpenAI API-Key
AIC.SetKey|$$KEY

' Set Model
AIC.SetModel_Comp|0

' Set Model-Temperature
AIC.Set_Temperature|0

' Set Max-Tokens (Possible lenght of answer, depending on the
Model up to 2000 Tokens which is about ~6000 characters)
' The more Tokens you use the more you need to pay.
AIC.SetMax_Token|25
```

```
' Ask Question and receive answer to $$RET
AIC.Ask_Completion|Wieviel Uhr ist es?|$$RET
DBP.$$RET
```

```
:enx
ENR.
```

2. You can use a saved, encrypted Key, that is stored in the project-folder.

This is the preferred way because this is more save for your key.

The default name for this saved Key is "AIC_License_Key.dat"

If no path is given, the file is created at "?exeloc\" that is the folder where the Script/the executable resides.

This key-file can be created using the AIC.Save_Key|\$\$KEY[|\$\$FIL] Command.

```
' Script 1: Save the Key to the file "AIC_License_Key.dat".
```

```
' IMPORTANT: This API-Key is a phantasy API-Key it must be replaced with your API-Key
$$KEY=sk-abcdefghijklmnopqrstuvwxy123456
```

```
' Here we save the Keyfile at the default path, that is:
```

```
' ?exeloc\AIC_License_Key.dat
```

```
AIC.Save_Key|$$KEY
```

```
ENR.
```

```
' Script 2; Using the crypted API-Keyfile
```

```
' Test if we are online, AI-Commands will only work if you are online.
```

```
NOL.
```

```
GTO.enx
```

```
EIF.
```

```
' Set OpenAI API-Key from the saved File
```

```
AIC.SetKey|from_File
```

```
' Set Model
```

```
AIC.SetModel_Completion|4
```

```
' Set Model-Temperature
```

```
AIC.Set_Temperature|0
```

```
' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000 Tokens)
```

```
' The more Tokens you use the more you need to pay.
```

```
AIC.SetMax_Token|25
```

```
' Ask Question and receive answer to $$RET
```

```
AIC.Ask_Completion|Wieviel Uhr ist es?|$$RET
```

```
DBP.$$RET
```

```
:enx
```

```
ENR.
```

3. You can use a Textfile, that contains your Key.

This is way to go, if you make an Executable that you want to share with the public or other people.

To give them the Option to use your SPR-Script together with their own API-Key, you can offer this Option.

The default name for this saved Key in a Textfile is "AIC_License_Key.txt"
If no path is given in **P2**, the file should be located in "?exeloc\" that is the folder where the Script/the executable resides.

This key-file can be created using any Texteditor (or CTF. -Command).

```
' Script 3: Using the API-Keyfile with the API-Key as Text
' Test if we are online, AI-Commands will only work if you are online.
NOL.
GTO.enx
EIF.

' Set OpenAI API-Key from the saved File
AIC.SetKey|from Text

' Set Model
AIC.SetModel_Completion|4

' Set Model-Temperature
AIC.Set_Temperature|0

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000
' The more Tokens you use the more you need to pay.
AIC.SetMax_Token|25

' Ask Question and receive answer to $$RET
AIC.Ask_Completion|Wieviel Uhr ist es?|$$RET
DBP.$$RET

:enx
ENR.
```

Syntax

AIC.SetKey|P1 [|P2]

AIC.Set_Key|P1 [|P2]

Parameter Explanation

P1 - Can be directly an [Open AI API-Key](#)^[470] or:

- **from File** - if given as **P1**, there should be a encrypted Keyfile with the name "AIC_License_Key.dat" at [?exeloc\](#)^[364] or if specified, at Path **P2**. This file can only be created using the [AIC.Save_Key](#)^[762] - Command.
- **from Text** - if given as **P1**, there should be a Textfile that contains the API-Key with the name "AIC_License_Key.txt" at [?exeloc\](#)^[364] or if specified, at Path **P2**. This file can be created using any Text-Editor.
- **Load Any Key** - if given as **P1**, the command will load any available API-Key.

P2 - opt. if specified this is the Path (and filename) of the Keyfile to use with the two options above. It can be either a Crypted or a Textfile. The System will decide with the Extension ".txt" or ".dat" how it is loaded.

Example

```

'*****
' EXAMPLE 1: AIC.-Commands
'*****
' IMPORTANT: This API-Key is a phantasy API-Key it must be replaced with your API-Key
$$KEY=sk-san4gh3j43h543k3HzpGbZbCq6PeVbSZy69H
' Test if we are online, AI-Commands will only work if you are online.
NOL.
GTO.enx
EIF.

' Set OpenAI API-Key
AIC.SetKey|$$KEY

' Set Model
AIC.SetModel_Completion|4

' Set Model-Temperature
AIC.Set_Temperature|0

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000 Tokens)
' The more Tokens you use the more you need to pay.
AIC.SetMax_Token|25

' Ask Question and receive answer to $$RET
AIC.Ask_Completion|Wieviel Uhr ist es?|$$RET
DBP.$$RET

:enx
ENR.

```

Remarks

A Word of Caution: Safeguarding Your API Key 🚨

Dear Valued User,

🔒 This Encryption is Not Impenetrable 🔒

Even if your API Key is encrypted within a file, the Decryption is hereby done without a Password, means anybody can decrypt it if he has the SPR at hand.

It's akin to a treasure locked in a chest.

The chest provides an added layer of security and invisibility, but should it fall into the hands of a pirate with the SPR and Knowledge, the treasure can be plundered.

You may ask: "Why did we not use a Password?"

The answer is simple, even then you would need to provide that Password in Cleartext in the code, because the SPR needs to decrypt the keyfile anyway. So there would be no advantage. In case you want to enter the Password each time and have an unbreakable Encryption, you can instead do something using the `GUT.` and the `GEC.`-Command.

⊗ *Do Not Distribute the Key, Even Encrypted* ⊗

Distributing executables or scripts along with the file containing your encrypted API Key is akin to sending your treasure chest out to sea on an unmanned ship. Anyone who gets hold of this file and has access to an SPR (Script Processing Runtime) can potentially decrypt and misuse your API Key.

💡 *Why is This a Big Deal?* 💡

Your API Key is not just a string; it's your identity and access within the OpenAI realm. It's linked to your account, your resources, and your privileges. In the wrong hands, it can be used to access services and consume quotas associated with your account. This can have both financial and security implications.

🔒 *What Should You Do?* 🔒

Never Distribute the Key: Do not include your API Key, even if encrypted, in any files or executables that you distribute.

Access Control: If your application requires the use of the API Key, consider implementing a backend service that your application can call. The service can then use the API Key server-side, where it's not exposed to the end-user.

Vigilance: Regularly monitor the usage of your API Key and be vigilant for any unauthorized or unexpected activity.

Set Usage Limits: Visit the OpenAI website and access your account settings. Here, you can set limits on the usage of your API Key. This is a wise precaution to ensure that even if the unthinkable happens, the potential damage is contained. It's like setting a magical barrier around your treasure chest!

Remember, with great power comes great responsibility. Your API Key is a powerful tool; wield it wisely and guard it well.

Safe coding!

Limitations:

-

See also:

- [Save_Key](#)⁷⁶²¹
-

3.42.10.7.10 Set_Max-Token

AIC.Set_MaxToken

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set_MaxToken

Set the number of Input- and Output-Token for the LLM

Intention

The `AIC.Set_MaxToken` command acts as a powerful dial in controlling the richness and extent of content generated by the LLM (Large Language Model) AI. Picture this: you're asking a wise sage a question, and you have the ability to control the length of the sage's response - that's essentially what this command does!

By default, think of the AI as a somewhat reserved character; it usually limits itself to a concise response. For instance, with OpenAI, it's like the AI has an internal pact to keep responses capped at around 256 tokens (the default). Tokens, by the way, are chunks of text that can be as small as a single character or as long as a word, like 'a' or 'apple'.

Now, let's say you're not just looking for a quick answer but a more elaborate one, almost like asking a storyteller to weave a rich tapestry of words. This is where "AIC.Set_MaxToken" swings into action. By tweaking the ``max_tokens`` parameter, you're essentially forceful nudging the AI to either expand or condense its creative canvas. Besides that you can include the length of the Output that you expect also in the Prompt.

Tokens	Characters
82	364

```
The fist difference is, that in the WEB-GUI the whole dialog is been used
as Input/context for each next answer.
Until the maximum amount of usable Tokens has been exceeded. If that
happens, then the AI will forget the start of the dialog and may even be
completely lost.
If that happens in the WEB-GUI, you will recognize that you get
surprisingly wrong answers.
```

TEXT TOKEN IDS

To learn more about Tokens you can use the [Open AI Tokenizer](#).

For example:

- Set ``max_tokens`` to 50, and you might get: "The cat sat on the mat."
- Set it to 200, and it could turn into:
"On a sunny afternoon, the mischievous cat, with its glistening fur, found solace on an antique mat, which had stories woven into its fabric."

But, be cautious; if you let the AI go too wild, it might give you a lot more than you bargained for, like an epic befitting a tome. Also, you don't want the AI to start rambling and lose focus, giving you irrelevant or redundant information.

Now, let's talk about costs. Imagine tokens as currency - the more tokens you use, the more you'll need to pay. So, if you're on a budget or managing resources for a larger project, judicious use of "AIC.Set MaxToken" is key. It's like fine dining; you want to savor each bite (or in this case, token) for the richness it brings, without overindulging.

Moreover, different AI models come with their own specialties and pricing, like choosing between a sumptuous buffet or an à la carte menu. Managing the `max_tokens` parameter allows you to optimize the selection of AI models based on the quality, efficiency, and cost that align with your objectives.

In summary, "AIC.Set MaxToken" is like the conductor of an orchestra, ensuring that each section comes together harmoniously to create a symphony that is just the right length and tempo. Whether you're looking for a succinct answer, a detailed explanation, or a narrative masterpiece, this command holds the key to unlocking the AI's potential, while also keeping an eye on the purse strings.

Imagine you're using a magical quill that writes stories, answers questions, or generates reports. However, this quill has an inkpot (the AI model) that gets used up with every word written. The "AIC.Set MaxToken" command allows you to determine **how much ink to use for each task**.

In the world of AI, this 'ink' is measured in tokens. A token can be as small as a single character ('a') or as long as a word ('apple'). By default, the AI is set to use a limited amount of ink, say enough for 256 tokens. This usually results in short and concise responses.

But what if you need more detailed responses? Here's where "AIC.Set MaxToken" comes into play. By adjusting the max_tokens parameter, you can tell the magical quill to use more ink for a more elaborate composition.

Especially if you work with Code, the amount of Tokens can never be large enough, yet there are limitations that come from the AI-Architecture.

Max Tokens (Chat Endpoint)	Max Tokens (Completion Endpoint)

4096	Not Applicable
16384	Not Applicable
8192	Not Applicable
32768	Not Applicable

Not Applicable	4096
Not Applicable	2048
Not Applicable	2048
Not Applicable	2048

Not Applicable	2048

However, there is a caveat - this magical ink is not free!

Imagine you have a prepaid ink pot. The more ink (tokens) the quill uses, the faster your balance depletes. This is exactly how AI cost management works. Each token has a cost associated with it, and the more tokens you use, the higher the cost of your API call.

For instance:

If you set max_tokens to 50, it's like writing a brief note. Cost-wise, imagine it as buying a small coffee.

Set it to 200, and you're looking at a more detailed piece, akin to a short story. The cost now may be comparable to a fancy latte with all the toppings. It's also important to know that, like coffee shops, different AI models have different pricing. A highly sophisticated model might produce richer content but at a premium cost, while a basic model could be more economical for simple tasks.

To make the most out of your ink pot (and budget), you want to strike a balance. Use the "AIC.Set MaxToken" command judiciously to ensure you're getting the quality and detail you need without unnecessarily draining your resources.

In summary, as an end-user, think of "AIC.Set MaxToken" as your way to control the AI's verbosity and depth. It's like having a volume knob and a quality selector, combined with a budget manager. Turn it up for richer content but be mindful of the costs, or dial it down for quick answers while saving your ink (and coins) for when you really need them.

The more tokens used in input and output, the higher the cost will be. It's essential to manage tokens efficiently to control costs.

Model	Context Window Size	Input Cost (\$/1K tokens)	Output Cost (\$/1K tokens)	Description
GPT-4	8K tokens	\$0.03	\$0.06	Broad general knowledge and domain expertise, capable of following complex instructions

				in natural language and solving difficult problems with accuracy.
	32K tokens	\$0.06	\$0.12	
GPT-3.5-turbo	4K tokens	\$0.0015	\$0.002	Optimized for dialogues, performance on par with Instruct Davinci. Useful for tasks like drafting emails, writing code, answering questions, creating conversational agents, and more`oaicite: {"number":1,"metadata": {"title":"Chat (beta)}}
	16K tokens	\$0.003	\$0.004	
DALL-E 2 (image generation)	Default: 256x256 pixels	Varies based on usage	Varies based on usage	Generates images; default image size is 256x256 pixels up to 1024x1024

As Prices are subject to change at any time, please always look at the [Open AI Web-Site to check for the actual Models and Prices.](#)

Syntax

AIC.Set_MaxToken | P1

AIC.Set_MaxToken | P1

Parameter Explanation

P1 - numeric value or Variable, containing a number

Example

```
' *****
' EXAMPLE 1: AIC.-Commands
```

```

'*****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File
FOR.$$LEE|0|11
  $$RET=
' Set Model
  AIC.SetModel_Chat|$$LEE

' Set Model-Temperature
  AIC.Set_Temperature|0

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000 5
' The more Tokens you use the more you need to pay. But the longer Input and Out
  AIC.SetMax_Token|300

' Ask Question and receive answer to $$RET
  $$QUE=Act as a mathematician.Calculate x for the formula "5*x^3=1450". Do it s
  AIC.Ask_Chat|$$QUE|$$RET
  CLP.$$RET
  MBX.Model: $$LEE $CrLf$$RET
NEX.
:enx
ENR.

```

Remarks

-

Limitations:

-

See also:

- [Set Key](#)⁷⁹⁹
-

3.42.10.7.11 Set_Temperature

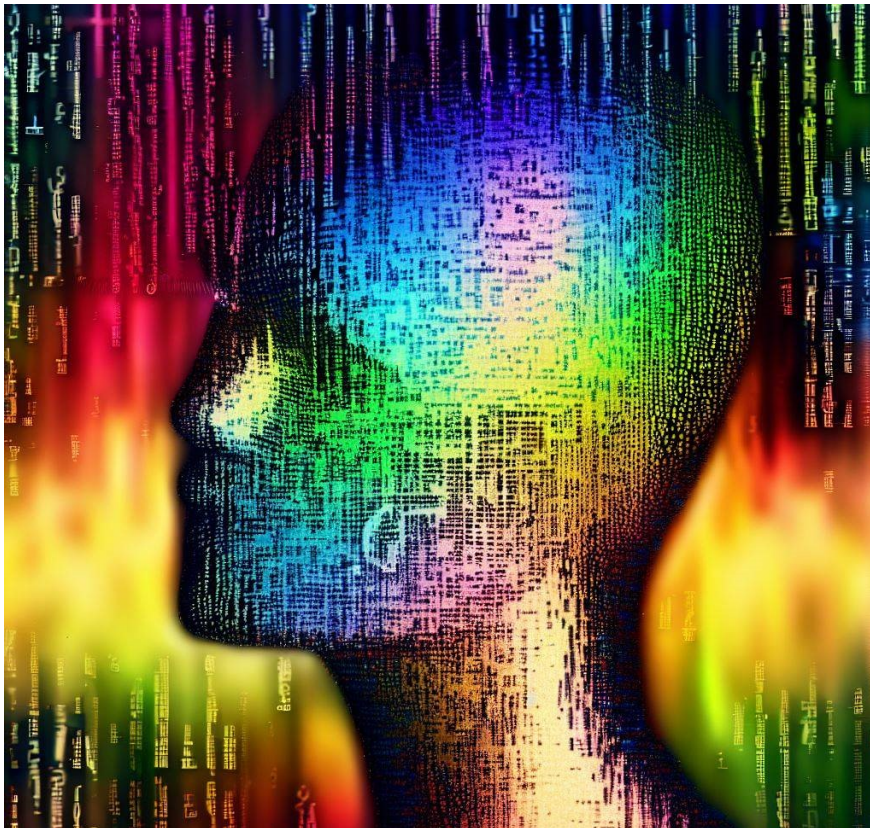
AIC.Set Temperature

[Previous](#) 2027 [Top](#) 49 [Next](#) 2069

MiniRobotLanguage (MRL)

AIC.Set Temperature

Save OpenAI-API Key encrypted to a file



This is how the AI imagines the Set Temperature Command.

Intention

The ``AIC.SetTemperature`` command is an essential tool for finetuning the creativity and randomness of responses generated by OpenAI's large language models (LLMs). It's like adjusting the seasoning in a recipe - a higher temperature spices things up, while a lower temperature keeps things more conservative and focused. Understanding and effectively using this command can help you craft the perfect responses for your application's needs.

****Description****

``AIC.SetTemperature`` takes a single parameter - the temperature value, which is a floating-point number typically ranging from 0.0 to 1.0. This parameter is responsible for scaling the logits before the final Softmax layer during text generation, effectively controlling the randomness in the model's output.

****Behavior********Lower Temperature Values (e.g., 0.2)****

- When the temperature is set to a lower value, the LLM's output becomes more deterministic and focused.
- The model is more likely to choose the most probable word sequences, making the text more coherent and less random.
- This is particularly useful when you need the model to produce factual information or adhere closely to a certain writing style.

****Higher Temperature Values (e.g., 0.8)****

- Setting a higher temperature makes the model's output more diverse and creative.
- The model is less biased towards the most probable word sequences and is more likely to produce surprising or unexpected responses.
- This is great when you want the model to generate creative content, brainstorm ideas, or be more conversational and less formal.

****Extreme Values****

- Setting the temperature to 0 makes the model completely deterministic, always choosing the most likely word. This might make the text repetitive or robotic.
- Setting the temperature very high (e.g., above 1) can result in very random and potentially nonsensical text.

****Use Cases****

1. ****Story Generation****: A higher temperature (e.g., 0.7) could be used to generate more imaginative and varied stories.
2. ****FAQ Responses****: A lower temperature (e.g., 0.2) would be appropriate for generating straightforward and consistent answers to frequently asked questions.
3. ****Idea Brainstorming****: A medium to high temperature (e.g., 0.5 to 0.8) might be ideal for generating a wide range of creative ideas.

****Example****

```
' Set a low temperature for focused, factual responses
AIC.Set Temperature|0.2

' Set a higher temperature for creative, diverse responses
AIC.Set Temperature|0.8
```

****Example Script****

```
' Set OpenAI API-Key from the saved File
AIC.SetKey|File
' Set Model
AIC.SetModel_Chat|0

FOR.$LEE|0|1|0.33
```

```

    $$RET=

' Set Model-Temperature
  AIC.Set_Temperature|$$LEE

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000)
' The more Tokens you use the more you need to pay. But the longer Input and Output
  AIC.SetMax_Token|100

' Ask Question and receive answer to $$RET
  $$Q=What is the next word after "the winner is ..."?
  AIC.Ask_Chat|$$Q|$$RET
  CLP.$$RET
  DBP.Temp.: $$LEE $crlf$$RET
  DBP.-----
NEX.
:enx
ENR.

```

****Output of the Script****

```

Temp.: 0
The next word after "the winner is ..." depends on the context. It could be the

Temp.: .33
The next word after "the winner is ..." depends on the context. It could be the

Temp.: .66
The next word is typically the name of the winner or the prize they have won. For

Temp.: .99
The next word is the name of the winner.

```

****Conclusion****

The `\AIC.SetTemperature`` command is a powerful dial that controls the flavor of your LLM's responses. Like a master chef, understanding how to adjust this dial can help you produce the perfect dish for any occasion. Whether you need focused, factual content or imaginative, creative prose, the `\AIC.SetTemperature`` command has got you covered.

Here is a more detailed explanation

... and comparison of the `\top_k``, `\top_p``, and `\temperature`` parameters in the context of OpenAI's GPT-3 API.

When GPT-3 generates text, it does so word by word. For each word it generates, it calculates a probability for every word in its vocabulary, and then selects the next word based on these probabilities. The `\top_k``, `\top_p``, and `\temperature`` parameters are all ways to influence this selection process.

1. `\top_k``: This parameter limits the number of words that the model considers as the next possible word. If `\top_k`` is set to 50, for example, the model will only consider the 50 words it thinks are most likely. This can make the output more focused and

less random, because it's only choosing from a subset of words. However, it can also make the output less diverse, because it's ignoring a lot of potential words. The ``top_k`` value can be any non-negative integer, with larger values leading to more randomness and smaller values leading to less randomness. If ``top_k`` is not set, the model considers all possible words.

2. ``top_p``: Also known as nucleus sampling, this parameter is a bit more dynamic. Instead of always considering a fixed number of words like ``top_k``, ``top_p`` considers however many words are needed to reach a certain cumulative probability. For example, if ``top_p`` is set to 0.9, the model will consider the smallest set of words that have a combined probability of 90%. This set of words can be larger or smaller depending on the specific probabilities for each word. Like ``top_k``, ``top_p`` can make the output more focused and less random, but it can also reduce diversity. The ``top_p`` value is a float between 0 and 1, with larger values leading to more randomness and smaller values leading to less randomness.

3. ``temperature``: This parameter controls the "sharpness" of the probability distribution. If ``temperature`` is set to a high value (close to 1), the model's word selection will be more random and less deterministic, even if some words have much higher probabilities than others. If ``temperature`` is set to a low value (close to 0), the model's word selection will be more deterministic and less random, with the model strongly favoring words that have higher probabilities. In other words, a high ``temperature`` makes the model more "adventurous" in its word choices, while a low ``temperature`` makes the model more "conservative".

In summary, ``top_k`` and ``top_p`` are ways to limit the number of words that the model considers for each step of the generation process, while ``temperature`` is a way to control the randomness of the model's word selection within those limits. All three parameters can be used together to finely tune the behavior of the model. The optimal values for these parameters can depend on your specific use case and the desired behavior of the model. It's a good idea to experiment with different values to see what works best for your needs.

Syntax

```
AIC.Set Temperature | P1
AIC.Set_Temperature | P1
```

Parameter Explanation

P1 - `<value>`: A floating-point number or variable representing the temperature.

Commonly used values are between 0.0 and 1.0, but higher values can also be used for more randomness.

Example

```
! *****
```

```

' EXAMPLE 1: AIC.-Commands
' Set Model-Temperature
'*****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File
' Set Model
AIC.SetModel_Chat|0

FOR.$$LEE|0|1|0.33
  $$RET=

' Set Model-Temperature
  AIC.Set_Temperature|$$LEE

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000 ?)
' The more Tokens you use the more you need to pay. But the longer Input and Output
  AIC.SetMax_Token|100

' Ask Question and receive answer to $$RET
  $$QUE=What is the next word after "the winner is ..."?
  AIC.Ask_Chat|$$QUE|$$RET
  CLP.$$RET
  DBP.Temp.: $$LEE $crlf$$$RET
  DBP.-----
NEX.
:enx
ENR.

```

Remarks

-

Limitations:

-

See also:

- [Set Key](#)^[799]
- [Set_Max_Token](#)^[806]

3.42.10.7.12 Use the STOP-Parameter

[Use the STOP-Parameter](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

Use the STOP-Parameter

Use the STOP-Parameter to Stop the Output Generation at certain Conditions

Intention

The ``stop`` parameter in OpenAI's GPT-3 API is used to specify a sequence of tokens that, when encountered, will signal the model to stop generating further tokens. This can be particularly useful when you want the model to generate text up to a certain point or condition.

The ``stop`` parameter can take a variety of forms, including:

- A single string: If you pass a single string, the model will stop generating when it encounters that string. For example, if you pass ``stop="\n"``, the model will stop generating after it produces a newline character.
- A list of strings: If you pass a list of strings, the model will stop generating when it encounters any of the strings in the list. For example, if you pass ``stop=["\n", "END"]``, the model will stop generating after it produces either a newline character or the string "END".

Here are some examples of how you might use the ``stop`` parameter:

1. To generate a single paragraph of text, you might use ``stop="\n\n"``. This tells the model to stop generating after it produces two newline characters in a row, which typically signifies the end of a paragraph.
2. To generate a list of items, you might use ``stop="\n1."``. This tells the model to stop generating after it produces a newline character followed by "1.", which typically signifies the start of a new list item.
3. To generate a conversation, you might use ``stop=["\n- User:", "\n- Assistant:"]``. This tells the model to stop generating after it produces a newline character followed by either "- User:" or "- Assistant:", which typically signify the start of a new line of dialogue.

Remember, the ``stop`` parameter is a tool to help you control the output of the model, and the best way to use it can depend on your specific use case. It's a good idea to experiment with different values to see what works best for your needs.

Here's a table of some commonly used special characters, their meanings, and their ASCII values:

Special Character	Description	ASCII Value
<code>\n</code>	Newline	10
<code>\t</code>	Tab	9
<code>\r</code>	Carriage return	13
<code>\"</code>	Double quote	34
<code>\'</code>	Single quote (apostrophe)	39
<code>\\</code>	Backslash	92
<code>\b</code>	Backspace	8
<code>\f</code>	Form feed	12

These special characters are used in strings to represent certain control characters.

For example, `\n` is used to represent a newline, which is a control character that moves the cursor to the beginning of the next line. Similarly, `\t` is used to represent a tab, which is a control character that moves the cursor to the next tab stop.

In the context of the stop parameter in OpenAI's GPT-3 API, these special characters can be used to tell the model to stop generating when it encounters certain control characters. For example, if you pass `stop="\n"`, the model will stop generating when it encounters a newline.

Some examples follow:

Generating a single paragraph of text:

```
{
  "prompt": "Once upon a time,",
  "max_tokens": 200,
  "stop": "\n\n"
}
```

**' SPR-Code: The "AIC.Set Stop"-Command will NOT Escape the Parameters internally
AIC.Set Stop|\\n\\n**

In this example, the model will stop generating text after it produces two newline characters in a row, which typically signifies the end of a paragraph.

Generating a list of items:

```
{
  "prompt": "1. First item",
  "max_tokens": 50,
  "stop": "\n1."
}
```

```
' SPR-Code: The "AIC.Set Stop"-Command will NOT Escape the Parameters internally
AIC.Set Stop|\\n1.
```

Here, the model will stop generating after it produces a newline character followed by "1.", which typically signifies the start of a new list item.

Generating a conversation:

```
{
  "prompt": "- User: Hello, how are you?\n- Assistant:",
  "max_tokens": 50,
  "stop": ["\n- User:", "\n- Assistant:"]
}
```

```
' SPR-Code: Using the "AIC.Add to Stop" Command, the Parameters are internally e
AIC.Clear
AIC.Add to Stop|\n- User:
AIC.Add to Stop|\n- Assistant:
```

In this example, the model will stop generating after it produces a newline character followed by either "- User:" or "- Assistant:", which typically signify the start of a new line of dialogue.

Generating a poem:

```
{
  "prompt": "Roses are red,",
  "max_tokens": 50,
  "stop": "END"
}
```

```
' SPR-Code:
AIC.Set Stop|END
```

Here, the model will stop generating after it produces the string "END", which can be used to signify the end of the poem.

Generating a story with chapters:

```
{
  "prompt": "Chapter 1: The Beginning",
  "max_tokens": 500,
  "stop": "Chapter 2:"
}
```

```
' SPR-Code:
AIC.Set Stop|Chapter 2:
```

In this example, the model will stop generating after it produces the string "Chapter 2:", which can be used to signify the start of a new chapter.

Here are three practical examples of using multiple stop conditions:

Generating a conversation with multiple participants:

```
{
  "prompt": "- Alice: Hello, how are you?\n- Bob:",
  "max_tokens": 50,
  "stop": ["\n- Alice:", "\n- Bob:", "\n- Charlie:"]
}
```

```
' SPR-Code: The Parameters are internally escaped using the "AIC.Add to Stop" Command
AIC.Clear
AIC.Clear
AIC.Add to Stop|\n- Alice:
AIC.Add to Stop|\n- Bob:
AIC.Add to Stop|\n- Charlie:
```

In this example, the model will stop generating after it produces a newline character followed by either "- Alice:", "- Bob:", or "- Charlie:", which typically signify the start of a new line of dialogue from each participant.

Generating a story with multiple endings:

```
{
  "prompt": "Once upon a time,",
  "max_tokens": 200,
  "stop": ["THE END", "To be continued...", "And they lived happily ever after."]
}
```

```
' SPR-Code:
AIC.Clear
AIC.Add to Stop|THE END
AIC.Add to Stop|To be continued...
AIC.Add to Stop|And they lived happily ever after.
```

Here, the model will stop generating after it produces any of the strings "THE END", "To be continued...", or "And they lived happily ever after.", which can be used to signify different types of endings for the story.

Generating a technical document with multiple sections:

```
{
  "prompt": "1. Introduction",
  "max_tokens": 500,
  "stop": ["2. Methodology", "3. Results", "4. Discussion", "5. Conclusion"]
}
```

```
' SPR-Code:  
AIC.Clear  
AIC.Add to Stop|2. Methodology  
AIC.Add to Stop|3. Results  
AIC.Add to Stop|4. Discussion  
AIC.Add to Stop|5. Conclusion
```

In this example, the model will stop generating after it produces any of the strings "2. Methodology", "3. Results", "4. Discussion", or "5. Conclusion", which can be used to signify the start of different sections in a technical document.

Remarks

-

Limitations:

-

See also:

- [1.6.1. Program Flow Control](#)^[134]
- [! Smart Package Robot 's Parallel Robot Operations](#)^[3745]
- [1.5. Features and Hints](#)^[93]

3.42.10.7.12.1 Set Stop

[AIC.Set Stop](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Set Stop

Will set an STOP-Condition for the Open AI - Services, will not escape Strings

Intention

The `AIC.Set Stop` command is a powerful tool that allows you to control when the AI model should stop generating further text.

This command is particularly useful when you want the model to generate text up to a certain point or condition.

Usage:

```
AIC.Set Stop|<stop_sequence>
```

The `<stop_sequence>` can be any string of characters. It could be a single character, a word, a sentence, or even special characters like newline (`\n`) or tab (`\t`). For example, if you want the model to stop generating text after it produces a newline character (which typically signifies the end of a paragraph), you would use the command like this:

```
AIC.Set Stop|\\n
```

As you see, we use `\\` instead of `\` this is done because the `AIC.Set Stop` will not automatically ["escape"](#) its Content for JSON Use. So we need to do that manually.

The `AIC.Set Stop` command is a powerful and flexible tool for controlling the behavior of the AI model. By understanding and using this command effectively, you can get the most out of your interactions with the model.

If you want to use special characters in the `<stop_sequence>`, you need to escape them.

The command will not automatically handle the escaping for you.

For example, to use a newline character, you would simply write `\\n`, not `\n`.

[For more Details see "Using the STOP-Flag".](#)

Syntax

```
AIC.Set Stop [ |P1 ]
```


Parameter Explanation

P1 - (*optional*) Variable or String to Set the STOP-Parameter. If omitted the STOP-Parameter will be cleared, just like "AIC.Clear Stop"

Example

```
! *****  
!  
! *****
```

Remarks

- The AIC.Set Stop command is a tool to help you control the output of the model. The best way to use it can depend on your specific use case.
- If you want to use multiple stop sequences, you can use the AIC.Add to Stop command to add more sequences after using the AIC.Set Stop command.
- If you want to clear all stop sequences, you can use the AIC.Clear command.

Limitations:

-

See also:

-

3.42.10.7.12.2 Add to Stop

[AIC.Add to Stop](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Add to Stop

Will add more stop sequences for the AI

Intention

The `AIC.Add to Stop` command is a useful tool that allows you to add additional stop sequences to the existing set of stop sequences for the AI model. This command is particularly useful when you want the model to stop generating text upon encountering any one of several different sequences.

The syntax for the `AIC.Add to Stop` command is as follows:

```
AIC.Add to Stop|<stop_sequence>
```

Here, `<stop_sequence>` is the sequence of characters that, when encountered, will signal the model to stop generating further tokens.

The `<stop_sequence>` can be any string of characters. It could be

- a single character,
- a word,
- a sentence, or even
- a special characters like newline (`\n`) or tab (`\t`).

For example, if you want the model to stop generating text after it produces a newline character or the word "END", you would use the command like this:

```
' See the difference in the handling of special characters, here we use "\\n"  
AIC.Clear Stop  
AIC.Set Stop|\\n  
AIC.Add to Stop|END
```

or use:

```
' See the difference in the handling of special characters, here we use "\\n"  
AIC.Clear Stop  
AIC.Add to Stop|\\n  
AIC.Add to Stop|END
```

In this case, the model will stop generating text when it encounters either a newline character or the word "END".

Special Characters

If you want to use special characters in the `<stop_sequence>`, you don't need to escape them. The command will automatically handle the escaping for you.

For example, to use a newline character, you would simply write `\n`, not `\\n`.

[For more Details see "Using the STOP-Flag".](#)

Syntax

AIC.Add to Stop|P1

Parameter Explanation

P1 – Variable or String to Set the STOP-Parameter.

Example

```
| *****  
|  
| *****
```

Remarks

- The `AIC.Add to Stop` command is a tool to help you control the output of the model. The best way to use it can depend on your specific use case. It's a good idea to experiment with different values to see what works best for your needs.
- You can use the `AIC.Add to Stop` command multiple times to add multiple stop sequences.
- If you want to clear all stop sequences, you can use the `AIC.Clear` command.

Limitations:

-

See also:

-

3.42.10.7.12.3 Get Stop Count

[AIC.Get Stop Count](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIC.Get Stop Count

Retrieves the total count of stop sequences currently set for the AI model.

Intention

The `AIC.Get Stop Count` command is a utility command that allows you to retrieve the number of stop sequences that have been set for the AI model. This command is particularly useful when you want to know how many stop conditions are currently in effect.

The syntax for the `AIC.Get Stop Count` command is as follows:

```
AIC.Get Stop Count | $$RET
```

Here, `$$RET` is a variable that will hold the number of stop sequences currently set. If `$$RET` is omitted, the result is placed on top of the stack (TOS).

You can use the `AIC.Get Stop Count` command when you want to know how many stop sequences have been set using the `AIC.Set Stop` or `AIC.Add to Stop` commands.

For example, if you have previously set the stop sequences to a newline character and the word "END", and you want to know how many stop sequences are currently set, you would use the command like this:

```
AIC.Get Stop Count | $$RET
```

After executing this command, the `$$RET` variable will hold the value 2, which is the number of stop sequences currently set.

Syntax

AIC.Get Stop Count [| P1]

Parameter Explanation

P1 - (*optional*) variable that will hold the number of stop sequences currently set. If `$$RET` is omitted, the result is placed on **top of the stack (TOS)**.

Example

```
| *****  
|  
| *****
```

Remarks

-

Limitations:

-

See also:

-

3.42.10.7.12.4 Clear Stop

`AIC.Clear Stop.`

[Previous](#) 1492 | [Top](#) 49 | [Next](#) 1499

MiniRobotLanguage (MRL)

AIC.Clear Stop.

Removes all set stop sequences.

Intention

The `AIC.Clear Stop` command that allows you to clear all the stop sequences that have been set for the AI model.

This command is particularly useful when you want to reset the stop conditions to their default state or before setting new stop conditions.

The syntax for the `AIC.Clear Stop` command is as follows:

```
AIC.Clear Stop
```

This command does not require any additional parameters.

You can use the `AIC.Clear Stop` command when you want to remove all the stop sequences that have been set using the `AIC.Set Stop` or `AIC.Add to Stop` commands.

For example, if you have previously set the stop sequences to a newline character and the word "END", and you want to remove these stop sequences, you would use the command like this:

```
AIC.Clear Stop
```

After executing this command, the AI model will not have any stop sequences and will continue generating text until it is ready or until it reaches the maximum token limit.

Syntax

AIC.Clear Stop

Parameter Explanation

No Parameters

Example

```
! *****  
!  
! *****
```

Remarks

- The `AIC.Clear Stop` command is a tool to help you control the output of the model. The best way to use it can depend on your specific use case. It's a good idea to use this command when you want to start setting stop sequences from scratch.
- After using the `AIC.Clear Stop` command, you can use the `AIC.Set Stop` or `AIC.Add to Stop` commands to set new stop sequences.

Limitations:

-

See also:

-

3.42.10.8 Ask AI Commands

While the other commands are good to change the AI-Environment, and make settings, the ASK-Commands are the heart of the System.



These commands will effectively call the AI and deliver you the result.

There are multiple so called "Endpoints" each of which has its own Set of Commands.

Especially has each Endpoint an own set of: "Set-Model" and "Ask Command".

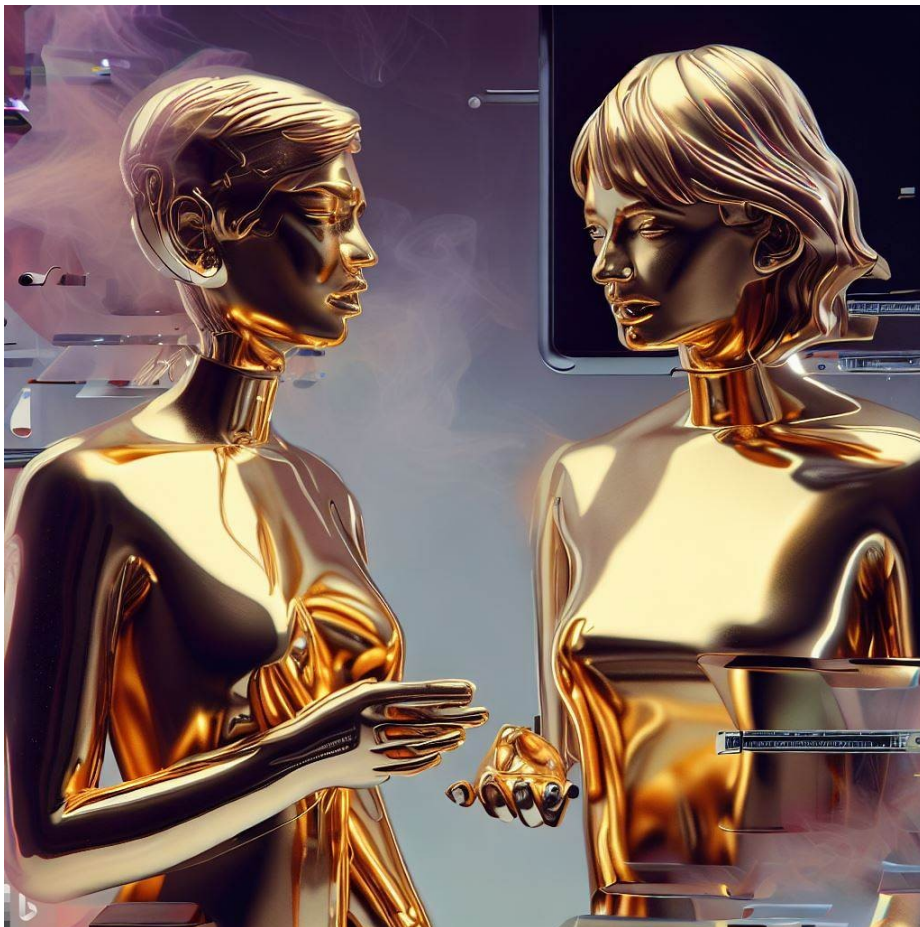
3.42.10.8.1 Ask Chat

`AIC.Ask Chat`[Previous](#) 2027 | [Top](#) 49 | [Next](#) 2069

MiniRobotLanguage (MRL)

`AIC.Ask_Chat`

Ask or instruct the most advanced Open AI models that are available for the "Chat endpoint" and expect Return.



Intention

The `AIC.Ask_Chat` command is the command to send a Question or Instruction to the most advanced Open AI models and receive an answer.

You will need to get your Open AI-API Key here: [AI - Artificial Intelligence Commands](#) 470 before being able to use this command.

Use the `AIC.Set_Model_Chat` command to specifying the OpenAI model you want to use for chat-based conversations.

There are multiple other commands which can be used [to change the environment for the AI](#) 759.



System

Complete the following code.



User

def fibonacci(num):



Assistant

```

if num == 0:
    return 0
elif num == 1:
    return 1
else:
    fib = [0, 1]
    for i in range(2, num + 1):
        fib.append(fib[i-1] + fib[i-2])
    return fib[num]

```

Sample from OpenAI how to use the [Chat-Completion Endpoint](#).

Once all preconditions are set, the usage of this command is as simple as:

```

$$QUS=Act as 3 different Persons with multiple Qualificati
$$QUS=$$QUS let each of the 3 Persons write a Script that v
$$QUS=$$QUS then evaluate all 3 results and give me the be
AIC.Ask|$$QUS|$$ANS
MBX. Here is the Script:$crlf$$$ANS
ENR.

```

Using the SPR Command is different from chatting in ChatGPT via the Internet.

Before we can discuss the details, you need to know the concept of Tokens. In natural language processing (NLP), a "token" typically refers to a unit of text.

In the simplest sense, tokens can be thought of as words.

For example, the sentence "I love AI" can be broken down into three tokens: "I", "love", and "AI".

However, tokens can also represent smaller units such as characters or subwords, or larger units like sentences, depending on the context.

See picture below.

When it comes to AI LLM-models, tokens play a crucial role in how text is processed.

These models read text in chunks called tokens.

Managing tokens is an important aspect of using chat models. Tokens are chunks of text that models read, and the total number of tokens in an API call affects the cost, time taken, and whether the API call works at all.

For gpt-3.5-turbo, the maximum limit is 4096 tokens. Both input and output tokens count toward these quantities.

For example, GPT-3, one of OpenAI's language models, is capable of understanding and generating human-like text by predicting the probability of a sequence of tokens.

Now, let's talk about OpenAI's language models and tokens.

OpenAI's models, like GPT-3, are not just limited to English; they can process text in multiple languages.

Additionally, a single token can represent a whole word, a part of a word, or even a single character, depending on the language and context.

For example, the word "chatbot" might be a single token, but in some languages or contexts, it might be split into multiple tokens like "chat" and "bot".

There's also a concept of "maximum available tokens" for OpenAI models.

Do you remember times when first computers had a maximum capacity of 4 KB?

This is where we are in terms of AI now.

This is essentially the maximum number of tokens that a model can process in a single request or operation.

For example, GPT-3 has a maximum token limit of 4096 tokens (01.06.2023).

This means that if you want GPT-3 to process a text, the total number of tokens in that text must not exceed 4096.

The Token-Limits includes both the input and output tokens.

If the text is too long, you would need to truncate or shorten it to fit within this limit.

Otherwise the Model will forget the start of the text when reading the end.

Tokens	Characters
82	364

The first difference is, that in the WEB-GUI the whole dialog is been used as Input/context for each next answer.

Until the maximum amount of usable Tokens has been exceeded. If that happens, then the AI will forget the start of the dialog and may even be completely lost.

If that happens in the WEB-GUI, you will recognize that you get surprisingly wrong answers.

TEXT TOKEN IDS

If you want to experiment with Tokens, you can use the [Open AI Online Tokenizer](#).

It's important to note that token limits are not necessarily fixed and may change over time as models are updated or new models are released.

Additionally, different models may have different token limits.

What is the difference between using the SPR and using the WEB-Interface from ChatGPT?

Using the WEB-GUI, **the whole dialog is been used as Input/context for each next answer.**

Until the maximum amount of usable Tokens has been exceeded. If that happens, then the AI will forget the start of the dialog and may even be completely lost. If that happens in the WEB-GUI, you will recognize that you get surprisingly wrong answers.

Using Open AI via the SPR this is generally NOT the case.

First you can set the maximum amount of Tokens to use using the Command `AIC.Set MaxToken`.

And then every `AIC.Ask_Chat` - Command is a completely new command and does by itself not remember anything that was before.

This way saving you a lot of Token.

Using the SPR **you can use more Tokens**, because any Chat is generally "**NEW**" and starts with the Full Amount of Tokens that are available, and is only limited by using the Command `AIC.Set MaxToken` and the maximum Tokens of the used Model.

Using the `Ask_Chat`-Command you do not automatically include all previous Questions and answers into the AI-Processing.

Every `Ask_Chat`-Command start completely new, this way saving Tokens (and costs) and also having more Tokens left for your answer.

You can access the Chat-History using the `AIC.Get_History` - Command and other [AI - History](#) Commands and this way include parts or all of earlier chats into the current prompt. But mostly this does not make sense here.

Therefore the rule here is:

Include all needed Instruction and Samples into the current Prompt.

You can get the history of the chat, and the last Question, or the last Answer using the Commands:

```
AIC.Get_History|$$HIS
AIC.Get_Last_Question|$$QUE
AIC.Get_Last_Answer|$$ANS
```

OpenAI currently offers two chat models that can be used with the so called "chat completion endpoint", namely **gpt-3.5-turbo** and **gpt-4**.

These models can be used to build various applications such as:

- drafting emails,
- writing Python or other code,
- answering questions,
- creating conversational agents,
- tutoring,
- language translations,
- simulating characters
- use them for for video games

- among others.

The `AIC.Ask_Chat` command works by taking your Prompt as input and returns a model-generated message as output.

As models don't have memory of past requests, so all relevant information must be supplied via the current Prompt.

It is worth noting that the Chat API, accessed through `AIC.Ask_Chat`, grants entry to a more advanced set of models, which, though potent in capabilities, may come with a higher price tag. In summary:

`AIC.Ask_Chat`: Ideal for more complex conversations, accesses advanced AI models like GPT 3.5 and 4.

`AIC.Set_Model_Completion`: Best suited for single-turn tasks, and "cheap tasks". Users are advised to choose the appropriate command based on the complexity and nature of the tasks they wish to accomplish with the SPR system.

Alternative to giving a Model-Name, you can specify a Model using a number, like this:

```
' Here we would specify "gpt-3.5-turbo"
AIC.Set_Model_Chat|3
```

Here is a Test-Script that you can use to see the answers of the models to a Problem. Due to the complexity i have increased the number of maximum Tokens.

```
' Set OpenAI API-Key from the saved File
AIC.SetKey|File
FOR.$$LEE|1|3
```

```
' Set Model
AIC.SetModel_Chat|$$LEE
```

```
' Set Model-Temperature
AIC.Set_Temperature|0
```

```
' Set Max-Tokens (Possible lenght of answer, depending on ...
' The more Tokens you use the more you need to pay. But th...
AIC.SetMax_Token|300
```

```
' Ask Question and receive answer to $$RET
$$QUE=Act as a mathematician.Calculate x for the formula
AIC.Ask_Chat|$$QUE|$$RET
CLP.$$RET
MBX.Model: $$LEE $CrLf$$$$RET
```

```
NEX.
```

```
:enx
```

```
ENR.
```

Model Number	Model Name	Comments
1	gpt-4	most actual Model
2	gpt-4-32k	most actual Model with 32k Tokens (~96 kb Text In/Out)
3	gpt-3.5-turbo	Standard Model
4	gpt-3.5-turbo-16k	Standard Model with 16k Tokens (~48 kb Text In/Out)

If you specify "0", the default Model here is Nr.3.

Here are some highlights about GPT-4 and ChatGPT with the GPT-3.5-turbo engine.

GPT-4 is a newer language model developed by OpenAI, whereas **GPT-3.5-turbo** is the default engine within the ChatGPT family.

Functions and Applications:

- Both GPT-4 and GPT-3.5-turbo can be used to
- draft emails,
- write code,
- answer questions about documents,
- create conversational agents,
- give software a natural language interface,
- tutor in various subjects,
- translate languages,
- simulate characters for video games, and much more.

GPT-4 has broad general knowledge and domain expertise and **can follow complex instructions** in natural language and **solve difficult problems** with accuracy. Conversations can be as short as 1 message or fill many pages, and including the conversation history helps when user instructions refer to prior messages.

Tokens:

Language models read text in chunks called tokens. A token can be as short as one character or as long as one word.

Both input and output tokens count toward the total tokens used in an API call.

The total number of tokens affects the cost, time, and whether the API call works at all.

Pricing (per 19.09.2023 - [prices are subject to change at any time](#)):

Model	Version	Input Cost	Output Cost
-------	---------	------------	-------------

4K context	GPT-3.5 Turbo	\$0.0015 / 1K tokens	\$0.002 / 1K tokens
8K context	GPT-4	\$0.03 / 1K tokens	\$0.06 / 1K tokens
16K context	GPT-3.5 Turbo	\$0.003 / 1K tokens	\$0.004 / 1K tokens
32K context	GPT-4	\$0.06 / 1K tokens	\$0.12 / 1K tokens

Both models are powerful tools for natural language processing and can be used for a wide range of applications. GPT-4 is the newer model and is likely to have improvements over GPT-3.5-turbo. However, GPT-3.5-turbo is much more cost-effective, especially for applications that don't require the absolute cutting edge in language model performance.

Syntax

AIC.Ask Chat|P1 [|P2] [|P3]
AIC.Ask|P1 [|P2] [|P3]

Parameter Explanation

P1 - <Prompt/Question>: This is your question / instruction to the AI.

P2 - **opt.** Variable to return the result / answer from the AI.

P3 - **opt.** 0/1 - Flag: This flag is optional and is used to specify how the results should be returned when multiple results are expected. If you have set the number of expected results to a value higher than 1 using `AIC.Set Number`, this flag determines how the results are returned. If set to "1", only the last result will be returned. If set to "0" (or left as the default), all results will be returned.

Example

```
' *****
' EXAMPLE 1: AIC.-Commands
' Here we let the AI Calculate x for the formula "5*x^3=1450"
'
' *****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File
FOR.$$LEE|0|11

' Set Model
  AIC.SetModel_Chat|$$LEE

' Set Model-Temperature
```



```
AIC.Set_Temperature|0
```

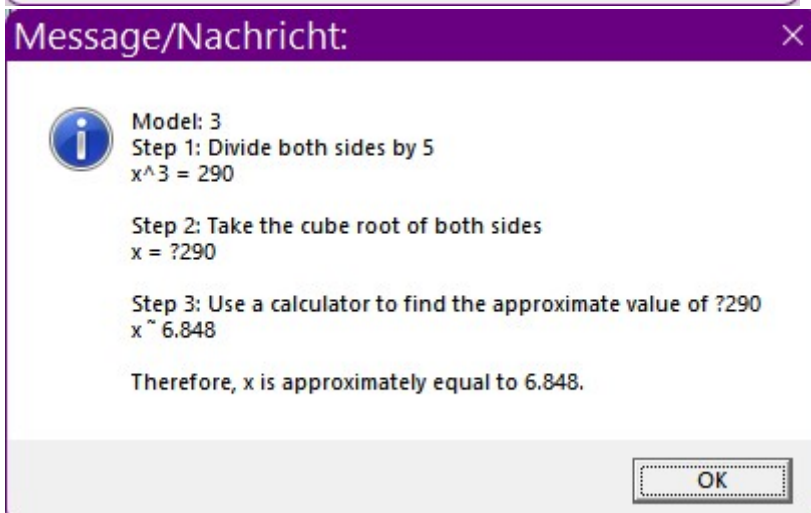
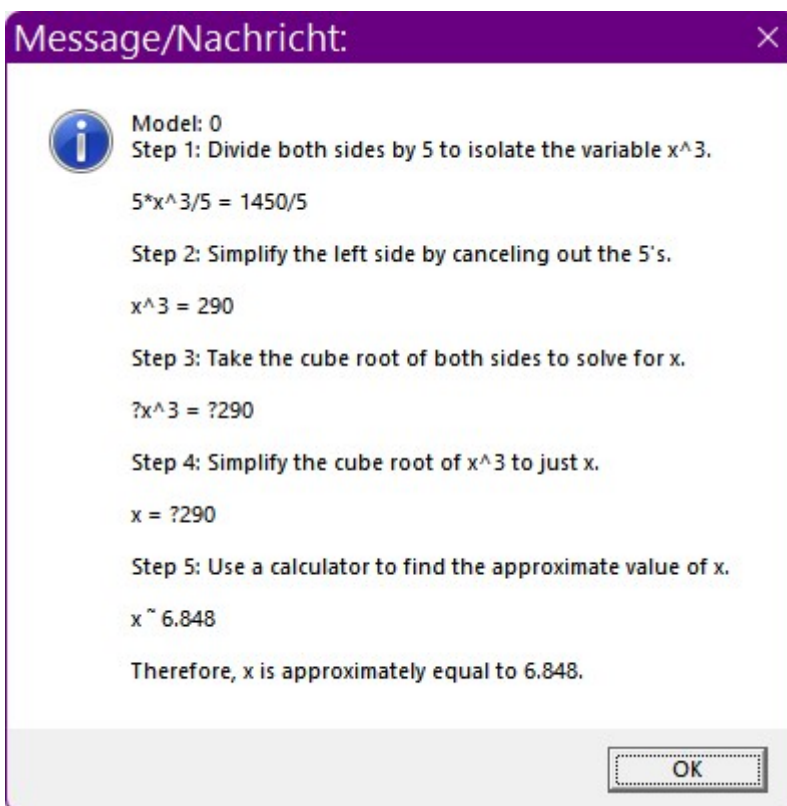
- ' Set Max-Tokens (Possible length of answer, depending on the Model up to 2000)
 - ' The more Tokens you use the more you need to pay. But the longer Input and Output
- ```
AIC.SetMax_Token|1000
```

- ' Ask Question and receive answer to \$\$RET  
 \$\$QUE=Act as a mathematician.Calculate x for the formula "5\*x^3=1450". Do it s  
 AIC.Ask\_Chat|\$\$QUE|\$\$RET  
 CLP.\$\$RET  
 MBX.Model: \$\$LEE \$CrLf\$\$RET

```
NEX.
```

```
:enx
```

```
ENR.
```



**Note that the Answer-Text is cut off at the end if you have specified a too small number of maximum Tokens in the Script.**

### Remarks

In your Prompts, ensure Clarity and Precision: Articulate your prompt in a way that unambiguously communicates the desired output from the model. Refrain from using vague or open-ended language, as this can yield unpredictable outcomes.

Incorporate Pertinent Keywords: Embed keywords in the prompt that are directly associated with the subject matter. This guides the model in grasping the context and subsequently producing more precise content.

Supply Contextual Information: Should it be necessary, furnish the model with background information or context. This equips the model to formulate more informed and contextually relevant responses.

Engage in Iterative Refinement: Embrace the process of experimentation with a variety of prompts to ascertain which is most effective. Continuously refine your prompts in response to the output generated, making adjustments until the desired results are achieved.

### Limitations:

-

### See also:

- [Set\\_Key](#)<sup>799</sup>
-

## 3.42.10.8.2 Ask Completion

[AIC.Ask Completion](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.Ask Completion

Ask or instruct the AI and receive answer, using the "Completion endpoint".



The "Completion" Models try to calculate the Possibility of several Tokens/Words to complete a given Text.

### Intention

The `AIC.Ask Completion` command is used to send a Prompt to the first Generation of Open AI Models, and receive an answer.

The syntax of this command is

```
AIC.Ask Completion|<Prompt>|<Variable for Answer>[|Flag]
```

The default and cheapest model is "text-ada-001".

The strongest and recommended model set by this command is "text-davinci-003".

It is of paramount importance for users to be cognizant of the distinction between AIC.Ask Completion and AIC.Ask Chat commands within the SPR environment.

- AIC.Ask Completion is specifically tailored for the OpenAI Completion API, which is good for simple tasks and offers cheaper costs.
- AIC.Ask Chat is able to call the newest and most capable Open AI Models, like GPT 3.5 and GPT-4 and has the higher costs per used Token.

From usage within the SPR, both Commands have the same Parameters and are used in the same way.

To choose the Model that fits to your task, use

```
' Here we would specify "gpt-3.5-turbo"
AIC.Set_Model_Completion|3
```

Here is a Sample-Script, in the Table below you will find the Answers of the models. To the given Quest: Calculate X for "5\*x=50" ?

Please change the quest to the sort of Problems you are going to solve, then you can use this Script to find the best Completion Model.

**Generally the Model Nr.3 is the strongest Model from all Models here.**

```
' Set OpenAI API-Key from the saved File
AIC.SetKey|File
FOR.$$LEE|0|3
```

```
' Set Model
AIC.SetModel_Completion|.$$LEE
```

```
' Set Model-Temperature
AIC.Set_Temperature|0
```

```
' Set Max-Tokens (Possible lenght of answer, depending on ...
' The more Tokens you use the more you need to pay. But th...
AIC.SetMax_Token|25
```

```
' Ask Question and receive answer to $$RET
$$QUE=Calculate X for "5*x=50" ?
AIC.Ask_Completion|$$QUE|$$RET
CLP.$$RET
MBX.Model: $$LEE $CrLf$ $$RET
```

```
NEX.
:enx
ENR.
```

| Case | Model ID               | Description                                                                                                                  | Token Limit (as of Sep 2021) | Cost per 1K tokens (as of July 2023) |
|------|------------------------|------------------------------------------------------------------------------------------------------------------------------|------------------------------|--------------------------------------|
| 1    | gpt-3.5-turbo-instruct |                                                                                                                              | 4096                         | \$0.0160                             |
| 2    | davinci-002            |                                                                                                                              | 4096                         | \$0.0120                             |
| 3    | babbage-002            | A version of the Babbage model, which is likely less powerful than Davinci but still very capable. Used for generating text. | 4096                         | \$0.0016                             |

Please find most actual models and prices here: [Open AI Prices](#)

### Syntax

```
AIC.Ask Completion|P1 [|P2] [|P3]
AIC.ACM|P1 [|P2] [|P3]
```

### Parameter Explanation

**P1** - <Prompt/Question>: This is your question / instruction to the AI.

**P2** - **opt.** Variable to return the result / answer from the AI.

**P3** - **opt.** 0/1 - Flag: This flag is optional and is used to specify how the results should be returned when multiple results are expected. If you have set the number of expected results to a value higher than 1 using `AIC.Set Number`, this flag determines how the results are returned. If set to "1", only the last result will be returned. If set to "0" (or left as the default), all results will be returned.

### Example

```

' *****
' EXAMPLE 1: AIC.-Commands
' *****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' Set Model
AIC.SetModel_Completion|4

' Set Model-Temperature
AIC.Set_Temperature|0

```

```
' Set Max-Tokens (Possible length of answer, depending on the Model up to 2000)
' The more Tokens you use the more you need to pay.
```

```
AIC.SetMax_Token|25
```

```
' Ask Question and receive answer to $$RET
```

```
AIC.Ask_Completion|What is a "Windows Button"?|$$RET
```

```
MBX.$$RET
```

```
:enx
```

```
ENR.
```



Note that the Answer-Text is cut off at the end because i have specified a maximum of 25 Tokens in the Script.

### Remarks

In your Prompts, ensure Clarity and Precision: Articulate your prompt in a way that unambiguously communicates the desired output from the model. Refrain from using vague or open-ended language, as this can yield unpredictable outcomes.

Incorporate Pertinent Keywords: Embed keywords in the prompt that are directly associated with the subject matter. This guides the model in grasping the context and subsequently producing more precise content.

Supply Contextual Information: Should it be necessary, furnish the model with background information or context. This equips the model to formulate more informed and contextually relevant responses.

Engage in Iterative Refinement: Embrace the process of experimentation with a variety of prompts to ascertain which is most effective. Continuously refine your prompts in response to the output generated, making adjustments until the desired results are achieved.

### Limitations:

-

### See also:

- [Set\\_Key](#)<sup>799</sup>
-



## 3.42.10.8.3 Ask Edit

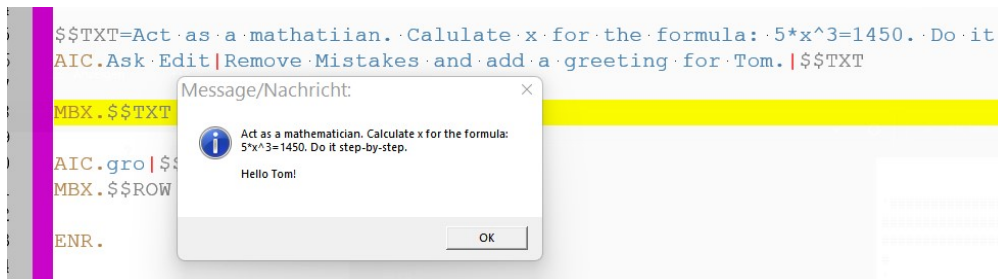
AIC.Ask Edit

[Previous](#) [Top](#) [Next](#)

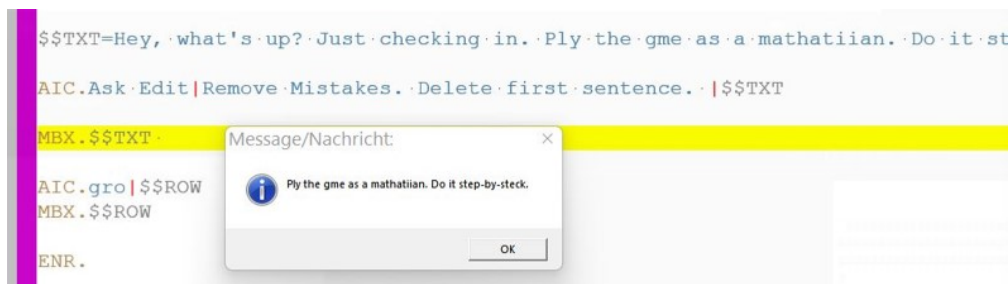
MiniRobotLanguage (MRL)

## AIC.Ask Edit

Let AI do the text-corrections for you.



In this Sample you can see that the AI will do corrections in Texts much better then normal Spellcheckers.



You can also have the AI remove or add words or Sentences.

### Intention

The `AIC.Ask Edit` command is a specialized command that is used in conjunction with the OpenAI Edits Endpoint.

It is specifically designed for scenarios where you need the AI to make modifications to an existing piece of text, based on certain instructions.

### Usage

```
AIC.Ask Edit|<Instruction>|<Text to be changed>
```

This command takes two parameters:

#### 1. Instruction

This is the first parameter and should be a clear and concise directive on what kind of edit or modification needs to be done to the text.

Example: If you want the AI to remove mistakes from the text, the instruction could be "Remove mistakes".

#### 2. Text to be changed



This is the second parameter and should be the original text that you want the AI to modify based on the instruction provided.

Example: "Theres mistakes in this sentence, which needs correction."

#### Full Example

```
$$TXT=Theres mistakes in this sentence, which needs correction.
AIC.Ask Edit|Remove mistakes|$$TXT
```

In this example, the command is instructing the AI to remove mistakes from the text in \$\$TXT.

The Result of the Operation will also be in \$\$TXT. Please note that the Edits-Endpoint is rather slow in Operation.

#### Notes

It is important to ensure that the instruction is clear, as ambiguous instructions may result in unexpected edits.

The text to be changed should be well-structured to make sure that the AI understands the context properly.

The command and the parameters are separated by a vertical bar |. This is crucial for the command to be processed correctly.

Use the `AIC.Ask Edit` command when you need the AI to perform specific editing tasks, and always ensure that your instructions are clear and the text is contextual for the best results.

#### Important Note:

For this command, only 2 Models can be used. One is for Text, and one is for Code, mainly Python, C++ etc.

The usage is very cheap compared to the state-of-the-art Models.

Yet, the results are limited to some degree. You can not ask for a "Rewrite as if you are Shakespeare". This is something that you can get from the "chat-endpoint" models, yet it comes at a much higher price.

If you fail to set these Models prior to calling the `AIC.Ask Edit` command, then you will not get a result.

Because the command will not change the Model itself.

These two Models are:

| Model Name            | Use Case                                                                         | Cost (per token)                     | Usage                                                                                                                                                                                                                                                                                         |
|-----------------------|----------------------------------------------------------------------------------|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| text-davinci-edit-001 | Text editing,<br>Text completion,<br>Text corrections                            | very cheap, but<br>nobody knows. :-) | Suitable used for<br>correcting errors in<br>text, such as grammar<br>and spelling.                                                                                                                                                                                                           |
| code-davinci-edit-001 | Code editing,<br>code generation,<br>code review.<br>Used in Github-<br>Copilot. | very cheap, but<br>nobody knows. :-) | Used for editing code<br>through the edits<br>endpoint. This means,<br>rather than just<br>completing code, it can<br>be used to edit existing<br>code. When using this<br>model, you provide<br>some code along with<br>instructions for how it<br>should be modified,<br>and the model will |

|  |  |  |                                        |
|--|--|--|----------------------------------------|
|  |  |  | attempt to make the edits accordingly. |
|--|--|--|----------------------------------------|

Please note that the cost per token might change and it is best to check the [OpenAI pricing page](#) for the most up-to-date information.

```
' Usage for Ask-Edits
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' Set Model to Text ("0"), ("1") would be Code.
AIC.SetModel_Edit|0

' Set Max-Tokens (Possible length of answer, depending on
' The more Tokens you use the more you need to pay. But the
AIC.SetMax_Token|300

$$TXT=Hey, what's up? Just checking in. Ply the gme as a m
AIC.Ask Edit|Remove Mistakes. Delete first sentence. |$$TX

' Get the Result
MBX.$$TXT
ENR.
```

### Syntax

**AIC.Ask Edit|P1|P2**

**AIC.ase|P1|P2**

### Parameter Explanation

**P1** - opt. Variable with Instructions what to do. If omitted the default Instruction is "Remove typing mistakes"

**P2** - Variable with Text, and also will receive the corrected Result. Can not be omitted.

### Example

```
' *****
' EXAMPLE 1: AIC.-Commands
' *****
```

```
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000)
' The more Tokens you use the more you need to pay. But the longer Input and Output
AIC.SetMax_Token|300

' Set Model
AIC.SetModel_Edit|0

$$TXT=Hey, what's up? Just checking in. Ply the gme as a mathatiian. Do it step-
AIC.Ask Edit|Remove Mistakes. Delete first sentence.|$$TXT
MBX.$$TXT

' See the Original Output of the Command (JSON Format).
' This is useful in case of Errors.
AIC.gro|$$ROW
MBX.$$ROW
ENR.
```

**Remarks**

-

**Limitations:**

-

**See also:**

- [Ask AI Commands](#) 

## 3.42.10.8.4 Ask Multi Vision

AIC.Ask Multi Vision

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.Ask Multi Vision

Send a multiple Picture to the Open AI "Vision A.I."-Endpoint and receive a description of these pictures

### Intention

Using the `AIC.Ask Multi Vision` command you can send a picture to the OpenAI Vision endpoint and receive a text-description about this picture.

You can also send a prompt and tell the Vision AI what exactly you want to know about these picture and this way you can influence the result that you get.

The `AIC.Ask Multi Vision` command is developed for complex image processing tasks involving multiple images.

It is particularly useful in scenarios that require batch analysis of images under the guidance of a single prompt.

*The current model is best at answering general questions about what is present in the images. While it does understand the relationship between objects in images, it is not yet optimized to answer detailed questions about the location of certain objects in an image. For example, you can ask it what color a car is or what some ideas for dinner might be based on what is in your fridge, but if you show it an image of a room and ask it where the chair is, it may not answer the question correctly.*

*The latency of the model can also be improved by downsizing your images ahead of time to be less than the maximum size they are expected them to be. For low res mode, we expect a 512px x 512px image. For high rest mode, the short side of the image should be less than 768px and the long side should be less than 2,000px. You can use the [AIC.EnsureFormatResize](#) Command to automatically generate resized versions of Pictures.*

Actually the OpenAI Vision API will only accept ".jpg"-Files.

Therefore you best provide ".jpg"-Files.

However, if you provide other files, the SPR will internally generate jpg-Versions of these files in the "?temp\" (Windows-Temp-Folder) and use these instead.

```

AIC.Set Key|File
$$PAT=F:\Testfolder
ARR.Set Array|0|0|$$PAT\Testpic_XL.jpg
ARR.Set Array|0|1|$$PAT\Testpic_XA.png

$$PRO=Tell me the difference between the two pictures
AIC.Ask Multi Vision|0|$$PRO|$$RET
AIC.Show Error
MBX.$$RET
ENR.

```



Based on the images provided, here are some differences I can point out:

1. **Background:** The background in the first image appears to be a room with various items, including what looks like a computer monitor and other miscellaneous objects. The second image has a background that shows a salon-like setting with sinks and beauty products.
2. **Outfit:** In the first picture, the person is wearing a plaid shirt, while in the second picture, they are wearing a camouflage jacket.
3. **Expression:** The facial expression of the person in the first image seems slightly more relaxed, while in the second image, the expression is more neutral or serious.
4. **Lighting:** The lighting in the first picture is warmer with a clear emphasis on the right side of the person's face, possibly from artificial light sources. The second picture has more even and balanced lighting, likely from professional indoor lighting, characteristic of a salon or studio.
5. **Image Quality:** The second image appears to have a higher resolution and clarity compared to the first image, which seems to be of lower resolution and grainier.
6. **Hair:** The person's hair in the second image appears to be styled differently, possibly slicked back, while in the first image, the hair is less styled and more casual.
7. **Perspective and Camera Angle:** The first image is taken from a slightly higher angle and closer proximity, whereas the second image is taken straight on and seems to be from a further distance.
8. **Context:** The context of the first image suggests a casual, personal environment, while the second image suggests a more public or professional context, potentially before or after a hair service given the salon setting.

Keep in mind that these observations are based on visual elements present in the photos and not on any personal information about the individual depicted.

In the two pictures, the same person appears to be present, judging by the facial features and overall appearance. Despite the likely identity of the subject being the same, several differences in the context and details between the images can be observed:

1. Attire: In the first image, the individual is wearing a checkered shirt, whereas in the second image, he is clad in a camouflage-patterned jacket.

2. Background: The first photo has a domestic or maybe an office setting with various items scattered in the background, while the second image has a cleaner background that looks like it could be in a salon, as suggested by the presence of hairdressing chairs and mirrors.

3. Expression: The subject's facial expression is slightly different in each picture. In the first, he has a more neutral expression, and in the second, it appears more solemn or serious.

4. Image Resolution: The first image is of a lower resolution, possibly from a webcam or a lower-quality camera, and the second image is of a higher resolution, perhaps taken with a better-quality camera.

5. Lighting: The lighting in the first image seems to be artificial room lighting, giving a yellowish tint, while the second picture has more natural or white light, highlighting more detail.

6. Camera Angle: The angle at which each photograph is taken differs slightly. The first image is taken at a higher angle relative to the individual, and the second is more head-on.

These are the immediate, notable differences between the two images.



In the first image, we see a person wearing a plaid shirt and headphones around their neck with an indoor background that looks cluttered, with various items dispersed around, including what may be a monitor or TV. The image quality is lower, and the lighting appears to be artificial, coming mostly from the ceiling.

In the second image, the same person is seen wearing a camouflaged pattern jacket and sitting in an environment that looks like a hair salon, indicated by the presence of basins behind him, presumably for washing hair. There is also a mirrored reflection of a shelf with beauty products. The image quality here is higher, with better focus and color reproduction, and the setting is brightly lit with more natural, evenly-dispersed light.

The person's expression is quite neutral in both images, although the angle at which the photos are taken slightly differs, with the second image being more direct and aligned with the person's face. The context and attire vary between the two images, suggesting different times, locations, and possibly activities.

In the first image, we see a person with a plaid shirt sitting at a table with various items spread across it. This person seems to be in a casual setting, possibly at home, and there are visible elements like a computer monitor and what appears to be a kitchen in the background.

The second image features the same person wearing a camouflaged jacket, sitting in what appears to be a salon, as suggested by the presence of a hair washing station with a basin in the background. Additionally, various hair products are visible on shelves behind the individual, and the lighting in this space is brighter and more even, typically found in commercial or retail settings.

Both pictures capture the person with a similar straight-faced expression, but the context, clothing, and backgrounds provide clear distinctions between the two.

**Using the same pictures and prompts, you may get very different results.**

## Syntax

# AIC.Ask Multi Vision|P1|P2[|P3]

## Parameter Explanation

**P1** - **Array-Nr.** 0-32, is the array number, an integer between 0 and 32, indicating the SPR-Array that contains the file names.

**P2** - **Prompt** - Prompt to tell the AI what you want to know from the picture.

**P3** - (**optional**) Variable that will receive the result. If omitted the result is placed on TOS.

## Example



```

'*****
' AIC.Ask Multi Vision
'*****
AIC.Set Key|File
$$PAT=F:\00_MR\MR_Komponents\U1-DLL

ARR.Set Array|0|0|$$PAT\Testpic_XL.jpg
ARR.Set Array|0|1|$$PAT\X513.jpg
ARR.Set Array|0|2|$$PAT\C.jpg

$$PRO=Tell me the difference between the 3 pictures
AIC.Ask Multi Vision|0|$$PRO|$$RET
AIC.Show Error
MBX.$$RET
ENR.

```

In the first image, you see a person with a light complexion wearing a plaid shirt. The setting looks casual, possibly a home environment, and there's a cluttered background with various items and what looks like a computer monitor. The image quality appears to be of standard definition.

The second image features the same person wearing a camouflaged jacket in a different setting, which seems to be a hair salon, judged by the presence of hair styling chairs and mirrors in the background. The lighting is bright and even, and the image has a much higher resolution than the first one.

The third image shows the same individual and setting as the second image, but it appears to be altered with some kind of filter or editing, evident by the blurry boundaries around the person and the artificial appearance of the background. The resolution and sharpness in this image are lower compared to the second image, due to the editing effect applied.

#### Remarks

It's crucial to ensure the array number (**P1**) is within the valid range (**0-32**). Providing an out-of-range number will result in an error.

#### Limitations:

##### Is there a maximum Number of images you can send?

From the SPR side there is no other limitation then Memory limits and bandwidth limits.

#### See also:

- 
-

## 3.42.10.8.5 Ask Multi Vision URL

AIC.Ask Multi Vision URL

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.Ask Multi Vision URL

Send a multiple Picture via URL to the Open AI "Vision A.I."-Endpoint and receive a description of these pictures

### Intention

Using the `AIC.Ask Multi Vision` command you can send a picture to the OpenAI Vision endpoint and receive a text-description about this picture.

You can also send a prompt and tell the Vision AI what exactly you want to know about these picture and this way you can influence the result that you get.

The `AIC.Ask Multi Vision` command is developed for complex image processing tasks involving multiple images.

It is particularly useful in scenarios that require batch analysis of images under the guidance of a single prompt.

*The current model is best at answering general questions about what is present in the images. While it does understand the relationship between objects in images, it is not yet optimized to answer detailed questions about the location of certain objects in an image. For example, you can ask it what color a car is or what some ideas for dinner might be based on what is in your fridge, but if you show it an image of a room and ask it where the chair is, it may not answer the question correctly.*

*The latency of the model can also be improved by downsizing your images ahead of time to be less than the maximum size they are expected them to be. For low res mode, we expect a 512px x 512px image. For high rest mode, the short side of the image should be less than 768px and the long side should be less than 2,000px. You can use the [AIC.EnsureFormatResize](#) Command to automatically generate resized versions of Pictures.*

Actually the OpenAI Vision API will only accept ".jpg"-Files. Therefore you need to provide ".jpg"-Files to use this command.

Technically if the image is not a ".jpg" the Smart Package Robot will anyway download it and then convert it to ".jpg" before sending it to the Open AI Vision Endpoint.

```

AIC.Set Key|File
' URL's from "Twitter" (X)
ARR.Set Array|0|0|https://pbs.twimg.com/media/F-zrQGBXMAAGdCs?format=jpg&name=me
ARR.Set Array|0|1|https://pbs.twimg.com/media/F-xtTucXMAAcpfy?format=jpg&name=me
ARR.Set Array|0|2|https://pbs.twimg.com/media/F-1T8dGXIAASffV?format=png&name=3

$$PRO=Explain what you see.
AIC.Ask Multi Vision URL|0|$$PRO|$$RET
'AIC.Show Error
MBX.$$RET
ENR.

```



Based on the images provided, here are some differences I can point out:

1. **Background:** The background in the first image appears to be a room with various items, including what looks like a computer monitor and other miscellaneous objects. The second image has a background that shows a salon-like setting with sinks and beauty products.
2. **Outfit:** In the first picture, the person is wearing a plaid shirt, while in the second picture, they are wearing a camouflage jacket.
3. **Expression:** The facial expression of the person in the first image seems slightly more relaxed, while in the second image, the expression is more neutral or serious.
4. **Lighting:** The lighting in the first picture is warmer with a clear emphasis on the right side of the person's face, possibly from artificial light sources. The second picture has more even and balanced lighting, likely from professional indoor lighting, characteristic of a salon or studio.
5. **Image Quality:** The second image appears to have a higher resolution and clarity compared to the first image, which seems to be of lower resolution and grainier.
6. **Hair:** The person's hair in the second image appears to be styled differently, possibly slicked back, while in the first image, the hair is less styled and more casual.
7. **Perspective and Camera Angle:** The first image is taken from a slightly higher angle and closer proximity, whereas the second image is taken straight on and seems to be from a further distance.
8. **Context:** The context of the first image suggests a casual, personal environment, while the second image suggests a more public or professional context, potentially before or after a hair service given the salon setting.

Keep in mind that these observations are based on visual elements present in the photos and not on any personal information about the individual depicted.

In the two pictures, the same person appears to be present, judging by the facial features and overall appearance. Despite the likely identity of the subject being the same, several differences in the context and details between the images can be observed:

1. Attire: In the first image, the individual is wearing a checkered shirt, whereas in the second image, he is clad in a camouflage-patterned jacket.

2. Background: The first photo has a domestic or maybe an office setting with various items scattered in the background, while the second image has a cleaner background that looks like it could be in a salon, as suggested by the presence of hairdressing chairs and mirrors.

3. Expression: The subject's facial expression is slightly different in each picture. In the first, he has a more neutral expression, and in the second, it appears more solemn or serious.

4. Image Resolution: The first image is of a lower resolution, possibly from a webcam or a lower-quality camera, and the second image is of a higher resolution, perhaps taken with a better-quality camera.

5. Lighting: The lighting in the first image seems to be artificial room lighting, giving a yellowish tint, while the second picture has more natural or white light, highlighting more detail.

6. Camera Angle: The angle at which each photograph is taken differs slightly. The first image is taken at a higher angle relative to the individual, and the second is more head-on.

These are the immediate, notable differences between the two images.



In the first image, we see a person wearing a plaid shirt and headphones around their neck with an indoor background that looks cluttered, with various items dispersed around, including what may be a monitor or TV. The image quality is lower, and the lighting appears to be artificial, coming mostly from the ceiling.

In the second image, the same person is seen wearing a camouflaged pattern jacket and sitting in an environment that looks like a hair salon, indicated by the presence of basins behind him, presumably for washing hair. There is also a mirrored reflection of a shelf with beauty products. The image quality here is higher, with better focus and color reproduction, and the setting is brightly lit with more natural, evenly-dispersed light.

The person's expression is quite neutral in both images, although the angle at which the photos are taken slightly differs, with the second image being more direct and aligned with the person's face. The context and attire vary between the two images, suggesting different times, locations, and possibly activities.

In the first image, we see a person with a plaid shirt sitting at a table with various items spread across it. This person seems to be in a casual setting, possibly at home, and there are visible elements like a computer monitor and what appears to be a kitchen in the background.

The second image features the same person wearing a camouflaged jacket, sitting in what appears to be a salon, as suggested by the presence of a hair washing station with a basin in the background. Additionally, various hair products are visible on shelves behind the individual, and the lighting in this space is brighter and more even, typically found in commercial or retail settings.

Both pictures capture the person with a similar straight-faced expression, but the context, clothing, and backgrounds provide clear distinctions between the two.

**Using the same pictures and prompts, you may get very different results.**

The first image is a text-based graphic with a question in German that translates to "What are you having for dinner on 24.12? I need to know in time, at whose place I can crash." This is likely referring to a humorous or pragmatic approach to determining where one might spend their Christmas Eve based on the attractiveness of the dinner menu.

The second image shows two women in a shopping mall setting, each carrying a rifle. The situation appears unusual as it is not common to see civilians casually carrying such firearms in a public space like a mall. This type of imagery might be more commonplace in regions with a high level of security concerns where armed presence in public spaces is part of routine security measures. The text overlaid in Hebrew suggests that the photo is likely from Israel, where it is more common to see armed soldiers or security personnel in public due to the country's military conscription and security policies. It is important to approach such images responsibly, considering the context and local customs.

You've shared two images that are quite different in content:

1. The first image displays a text in German which translates to "What are you having for dinner on December 24? I need to know in advance where to sign up." The text is framed in a humorous context, suggesting that the person is trying to decide where to spend Christmas Eve dinner, possibly based on the menu offerings.

2. The second image shows two women in a shopping mall. Both appear to be armed with rifles, which is unusual in some parts of the world but might not be in others, like Israel, where the populace may be seen carrying firearms for security reasons. The scene seems casual considering the firearms present, as both women appear to be engaged in shopping or browsing and not in a combat or high-alert scenario. The juxtaposition of armed individuals in a civilian shopping environment could be surprising or thought-provoking for viewers not accustomed to such sights.

**Detail: Low**

**Detail: High**

## Syntax

# AIC.Ask Multi Vision|P1|P2[|P3]

## Parameter Explanation

**P1 - Array-Nr.** 0-32, is the array number, an integer between 0 and 32, indicating the SPR-Array that contains the file names.

Technically if the image is not a "jpg" the Smart Package Robot will anyway download it and then convert it to "jpg" before sending it to the Open AI Vision Endpoint.

**P2 - Prompt** - Prompt to tell the AI what you want to know from the picture.

**P3 - (optional)** Variable that will receive the result. If omitted the result is placed on TOS.

**Example**

```

'*****
' AIC.Ask Multi Vision
'*****
AIC.Set Key|File
$$PAT=F:\00_MR\MR_Komponents\U1-DLL

ARR.Set Array|0|0|$$PAT\Testpic_XL.jpg
ARR.Set Array|0|1|$$PAT\X513.jpg
ARR.Set Array|0|2|$$PAT\C.jpg

$$PRO=Tell me the difference between the 3 pictures
AIC.Ask Multi Vision|0|$$PRO|$$RET
AIC.Show Error
MBX.$$RET
ENR.

```

In the first image, you see a person with a light complexion wearing a plaid shirt. The setting looks casual, possibly a home environment, and there's a cluttered background with various items and what looks like a computer monitor. The image quality appears to be of standard definition.

The second image features the same person wearing a camouflaged jacket in a different setting, which seems to be a hair salon, judged by the presence of hair styling chairs and mirrors in the background. The lighting is bright and even, and the image has a much higher resolution than the first one.

The third image shows the same individual and setting as the second image, but it appears to be altered with some kind of filter or editing, evident by the blurry boundaries around the person and the artificial appearance of the background. The resolution and sharpness in this image are lower compared to the second image, due to the editing effect applied.

**Remarks**

It's crucial to ensure the array number (**P1**) is within the valid range (**0-32**). Providing an out-of-range number will result in an error.

**Limitations:****Is there a maximum Number of images you can send?**

From the SPR side there is no other limitation then Memory limits and bandwidth limits.

From the Server side, pictures should be best not too small (<512 Pixels) and not larger then 2048x1792)

**See also:**

- 
-



## 3.42.10.8.6 Ask Vision URL

AIC.Ask Vision URL

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.Ask Vision URL

Send a Picture via URL to the Open AI "Vision A.I."-Endpoint and receive a description of the picture

### Intention

Using the `AIC.Ask Vision URL` command you can send a picture to the OpenAI Vision endpoint and receive a text-description about this picture. You can also send a prompt and tell the Vision AI what exactly you want to know about these picture and this way you can influence the result that you get.

The `AIC.Ask Vision URL` command is developed for complex image processing tasks involving multiple images. It is particularly useful in scenarios that require batch analysis of images under the guidance of a single prompt.

*The current model is best at answering general questions about what is present in the images. While it does understand the relationship between objects in images, it is not yet optimized to answer detailed questions about the location of certain objects in an image. For example, you can ask it what color a car is or what some ideas for dinner might be based on what is in your fridge, but if you show it an image of a room and ask it where the chair is, it may not answer the question correctly. The latency of the model can also be improved by downsizing your images ahead of time to be less than the maximum size they are expected them to be. For low res mode, we expect a 512px x 512px image. For high rest mode, the short side of the image should be less than 768px and the long side should be less than 2,000px. You can use the [AIC.EnsureFormatResize](#) Command to automatically generate resized versions of Pictures.*

Actually the OpenAI Vision API will only accept ".jpg"-Files. Therefore you need to provide ".jpg"-Files to use this command. Technically if the image is not a ".jpg" the Smart Package Robot will anyway download it and then convert it to ".jpg" before sending it to the Open AI Vision Endpoint.

```
AIC.Set Key|File
' URL's from "Twitter" (X)
$$URL=https://pbs.twimg.com/media/F-1T8dGXIAASffV?format=png&name=360x360
$$PRO=Explain what you see.
AIC.Ask Vision URL|$$URL|$$PRO|$$RET
'AIC.Show Error
MBX.$$RET
ENR.
```

### Syntax

## AIC.Ask Vision URL|P1|P2 [|P3]

### Parameter Explanation

**P1 - URL of the Image**, this is the URL of the image to send. Technically if the image is not a "jpg" the Smart Package Robot will anyway download it and then convert it to "jpg" before sending it to the Open AI Vision Endpoint.

**P2 - Prompt** - Prompt to tell the AI what you want to know from the picture.

**P3 - (optional)** Variable that will receive the result. If omitted the result is placed on TOS.

### Example

```

'*****
' AIC.Ask Multi Vision
'*****
AIC.Set Key|File
$$PAT=F:\00_MR\MR_Komponents\U1-DLL

ARR.Set Array|0|0|$$PAT\Testpic_XL.jpg
ARR.Set Array|0|1|$$PAT\X513.jpg
ARR.Set Array|0|2|$$PAT\C.jpg

$$PRO=Tell me the difference between the 3 pictures
AIC.Ask Multi Vision|0|$$PRO|$$RET
AIC.Show Error
MBX.$$RET
ENR.

```

In the first image, you see a person with a light complexion wearing a plaid shirt. The setting looks casual, possibly a home environment, and there's a cluttered background with various items and what looks like a computer monitor. The image quality appears to be of standard definition.

The second image features the same person wearing a camouflaged jacket in a different setting, which seems to be a hair salon, judged by the presence of hair styling chairs and mirrors in the background. The lighting is bright and even, and the image has a much higher resolution than the first one.

The third image shows the same individual and setting as the second image, but it appears to be altered with some kind of filter or editing, evident by the blurry boundaries around the person and the artificial appearance of the background. The resolution and sharpness in this image are lower compared to the second image, due to the editing effect applied.

### Remarks

It's crucial to ensure the array number (**P1**) is within the valid range (**0-32**). Providing an out-of-range number will result in an error.

### Limitations:

#### Is there a maximum Number of images you can send?

From the SPR side there is no other limitation then Memory limits and bandwidth limits.

From the Server side, pictures should be best not too small (<512 Pixels) and not larger then 2048x1792)

### See also:

- 
-

## 3.42.10.8.7 Ask Vision

[AIC.Ask\\_Vision](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.Ask Vision

Send a Single Picture to the Open AI "Vision A.I."-Endpoint and receive a description of this picture

### Intention

Using the `AIC.Ask_Vision` command you can send a picture to the OpenAI Vision endpoint and receive a text-description about this picture. You can also send a prompt and tell the Vision AI what exactly you want to know about this picture and this way you can influence the result that you get.

*The current model is best at answering general questions about what is present in the images. While it does understand the relationship between objects in images, it is not yet optimized to answer detailed questions about the location of certain objects in an image. For example, you can ask it what color a car is or what some ideas for dinner might be based on what is in your fridge, but if you show it an image of a room and ask it where the chair is, it may not answer the question correctly.*

*The latency of the model can also be improved by downsizing your images ahead of time to be less than the maximum size they are expected to be. For low res mode, we expect a 512px x 512px image. For high res mode, the short side of the image should be less than 768px and the long side should be less than 2,000px. You can use the [AIC.EnsureFormatResize](#) Command to automatically generate resized versions of Pictures.*

Actually the OpenAI Vision API will only accept ".jpg"-Files. Therefore you best provide ".jpg"-Files.

However, if you provide other files, the SPR will internally generate jpg-Versions of these files in the "?temp\" (Windows-Temp-Folder) and use these instead.

Using `AIC.Ask_Vision` the SPR will automatically delete these temporary files after the process.

**In the image you've provided, there is a person wearing a plaid shirt. They seem to be indoors, possibly in an office or home environment, given the presence of a computer and various items in the background that suggest a personal workspace.**

**This is a Sample result using the AIC.Ask Vision Command.**

The image is a screenshot of the Gmail interface in a web browser. This particular view within Gmail is showing the "Snoozed" folder ("Zurückgestellt" in German), indicating that no emails are currently snoozed, as the main window where emails would be listed is empty. In the upper right corner, there is a profile picture suggesting that a user is logged in. The left sidebar displays various Gmail folders and categories including Inbox ("Posteingang"), Starred ("Markiert"), Important ("Wichtig"), Sent ("Gesendet"), Drafts ("Entwürfe"), as well as categories for Social Networks ("Soziale Netzwerke"), Notifications ("Benachrichtigungen"), Forums ("Foren"), and Promotions ("Werbung"). Additionally, there are user-created labels listed, such as "Azimo," "Google," "Immobilien," and "Personal." At the bottom left, there's also an indication of how much storage is being used (0.32 GB of 15 GB). The lower portion of the sidebar also displays options such as "Terms of Use" ("Nutzungsbedingungen"), "Privacy" ("Datenschutz"), and "Program Policies" ("Programmrichtlinien). The user can compose a new email by clicking on the "Compose" button ("Schreiben" in German). The top bar of the browser indicates multiple open tabs and has various bookmarked sites visible.

This is a Sample result using the AIC.Ask Vision Command.

### Syntax

## AIC.Ask Vision|P1|P2[|P3]

### Parameter Explanation

**P1 - Picture, File name** - Is the Path and File name for the Picture to upload to the AI-

**P2 - Prompt** - Prompt to tell the AI what you want to know from the picture.

**P3 - (optional)** Variable that will receive the result. If omitted the result is placed on TOS.

### Example

```

'*****
' AIC.-Sample
'*****
AIC.Set Key|File
$$PAT=F:\Testfolder
$$URL=$$PAT\Testpic_XL.jpg
$$PRO=Tell me what you see.
AIC.Ask Vision|$$URL|$$PRO|$$RET
'AIC.Show Error
MBX.$$RET
ENR.

```

**Remarks**

-

**Limitations:**

-

**See also:**

- 
-

## 3.42.10.8.8 Ask with History

[AIC.Ask\\_with\\_History](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.Ask\_with\_History

Ask or instruct the most advanced Open AI models using a history-Array and expect Return.



In the WEB-UI, ChatGPT uses the history of previous answers to generate a new answer. Using this command you can do exactly this also.

### Intention

The `AIC.Ask_with_History` command allows you to add a complete "Chat History" to the prompt, in the same way as if you would have been chatting with the AI in the Chat-Window.

This way you can reconstruct and get the same Answers that you may have gotten while chatting with the AI (at low temperature values, else there is a high random factor).

The `AIC.Ask_with_History` command is the most advanced AI-Command to send a Question or Instruction to the most advanced Open AI models and receive an answer.

Using this command you can achieve the same results like if you chat with the AI in a Browser Window.

You can attach a "chat history" to the Prompt, which is done using the new Array-Commands.

You will need to get an API-Key here: [AI - Artificial Intelligence Commands](#)<sup>[470]</sup> before being able to use this command.

Use the `AIC.Set_Model_Chat` command to specifying the OpenAI model you want to use for chat-based conversations.

There are multiple other commands which can be used [to change the environment for the AI](#)<sup>[759]</sup>.

Once all preconditions are set, the usage of this command is as simple as:

```
'#SPI:NoAPIKey

' Set OpenAI API-Key from the saved File
AIC.SetKey|File
' Set Model
AIC.SetModel_Chat|3
' Set Model-Temperature
AIC.Set_Temperature|0.6
AIC.SetMax_Token|300

' Using the ARR.-Command we build a Chat history
'Each Array element must consist of 2 Parts:
'<Role>:<Content>

' You can use as many Array Elements as you like, the Array
',
' <Array-No.>|<Array-Element No.>|<Text to assign to t
',

ARR.Set|1|0|system:You are a technical advisor. Your name
ARR.Set|1|1|user:What is 1+1?
ARR.Set|1|2|assistant:Hallo, I am Paul and 1+1 is 2.
ARR.Set|1|3|user:What is 2+2?
ARR.Set|1|4|assistant:Hallo, I am Paul and 2+2 is 4.
ARR.Set|1|5|user:What is the sum of 3+5?

' Now we send the Array to the Model.
AIC.Ask With History|1|$$RET
MBX.$$RET
```



ENR.

```

' .Set OpenAI API-Key from the saved File
AIC.SetKey|File
' .Set Model
AIC.SetModel_Chat|3
' .Set Model-Temperature
AIC.Set_Temperature|0.6
AIC.SetMax_Token|300

' .Using the ARR.-Command we build a Chat history
' Each Array element must consist of 2 Parts:
' <Role>:<Content>

' You can use as many Array Elements as you like, the Array is "Auto-Dim"
'
'<Array-No.>|<Array-Element-No.>|<Text to assign to the Array Element>
'
ARR.Set|1|0|system:You are a technical advisor. Your name is Paul.
ARR.Set|1|1|user:What is 1+1?
ARR.Set|1|2|assistant:Hallo, I am Paul and 1+1 is 2.
ARR.Set|1|3|user:What is 2+2?
ARR.Set|1|4|assistant:Hallo, I am Paul and 2+2 is 4.
ARR.Set|1|5|user:What is the sum of 3+5?

' Now we send the Array to the Model.
AIC.Ask_With_History|1|$$RET
MBX.$$RET
ENR.

```



This is the Result of the above Script. Chat GPT apologizes for a mistake we told him he may have done using the Chat-History

**Using the SPR "AIC.Ask with history" Command is equal to chatting in ChatGPT via**

Before we can discuss the details, you need to know the concept of Tokens. In natural language processing (NLP), a "token" typically refers to a unit of text.

In the simplest sense, tokens can be thought of as words.

For example, the sentence "I love AI" can be broken down into three tokens: "I", "love", and "AI".

However, tokens can also represent smaller units such as characters or subwords, or larger units like sentences, depending on the context.

See picture below.

**When it comes to AI LLM-models, tokens play a crucial role in how text is processed.**

These models read text in chunks called tokens.

Managing tokens is an important aspect of using chat models. Tokens are chunks of text that models read, and the total number of tokens in an API call affects the cost, time taken, and whether the API call works at all.

For gpt-3.5-turbo, the maximum limit is 4096 tokens. Both input and output tokens count toward these quantities.

For example, GPT-3, one of OpenAI's language models, is capable of understanding and generating human-like text by predicting the probability of a sequence of tokens.

Now, let's talk about OpenAI's language models and tokens.

OpenAI's models, like GPT-3, are not just limited to English; they can process text in multiple languages.

Additionally, a single token can represent a whole word, a part of a word, or even a single character, depending on the language and context.

For example, the word "chatbot" might be a single token, but in some languages or contexts, it might be split into multiple tokens like "chat" and "bot".

### ***There's also a concept of "maximum available tokens" for OpenAI models.***

Do you remember times when first computers had a maximum capacity of 4 KB?

This is where we are in terms of AI now.

This is essentially the maximum number of tokens that a model can process in a single request or operation.

For example, GPT-3 has a maximum token limit of 4096 tokens (01.06.2023).

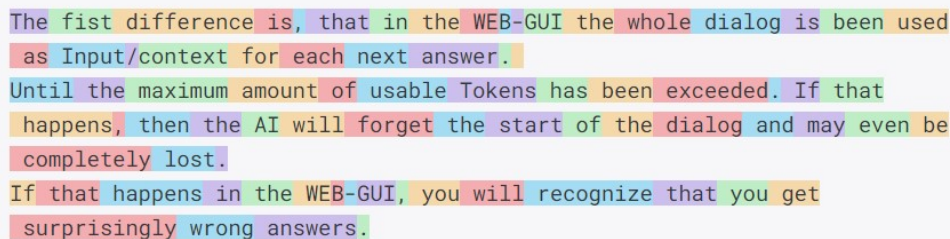
This means that if you want GPT-3 to process a text, the total number of tokens in that text must not exceed 4096.

**The Token-Limits includes both the input and output tokens.**

If the text is too long, you would need to truncate or shorten it to fit within this limit.

**Otherwise the Model will forget the start of the text when reading the end.**

| Tokens | Characters |
|--------|------------|
| 82     | 364        |



The fist difference is, that in the WEB-GUI the whole dialog is been used as Input/context for each next answer. Until the maximum amount of usable Tokens has been exceeded. If that happens, then the AI will forget the start of the dialog and may even be completely lost. If that happens in the WEB-GUI, you will recognize that you get surprisingly wrong answers.

TEXT    TOKEN IDS

**If you want to experiment with Tokens, you can use the [Open AI Online Tokenizer](#).**

It's important to note that token limits are not necessarily fixed and may change over time as models are updated or new models are released.

Additionally, different models may have different token limits.

### ***What is the difference between using the SPR and using the WEB-Interface from ChatGPT?***

Using the WEB-GUI, **the whole dialog is been used as Input/context for each next answer.**

Until the maximum amount of usable Tokens has been exceeded. If that happens, then the AI will forget the start of the dialog and may even be completely lost.

If that happens in the WEB-GUI, you will recognize that you get surprisingly wrong answers.

Using Open AI via the SPR this is generally NOT the case.

First you can set the maximum amount of Tokens to use using the Command `AIC.Set MaxToken`.

And then every `AIC.Ask_with_History` - Command is a completely new command and does by itself not remember anything that was before.

This way saving you a lot of Token.

To overcome this issue, using this Command you can "attach" a previous dialog or full history to the prompt.

This is exactly what happens also in the WEB-GUI.

Using the SPR **you can use more Tokens**, because any Chat is generally "**NEW**" and starts with the Full Amount of Tokens that are available, and is only limited by using the Command `AIC.Set MaxToken` and the maximum Tokens of the used Model.

Therefore you can choose which parts of the history you really need, and only attach these parts to the History-Array.

You can access the Chat-History using the `AIC.Get History` - Command and other [AI - History](#) Commands and this way include parts or all of earlier chats into the current prompt. But mostly this does not make sense here.

However, the best rule here is:

#### **Include all needed Instruction and Samples into the current Prompt.**

Using the History Array you can also provide Sample-Answers to the AI and get your final answer in the way you want.

You can get the history of the chat, and the last Question, or the last Answer using the Commands:

```
AIC.Get_History|$$HIS
AIC.Get_Last_Question|$$QUE
AIC.Get_Last_Answer|$$ANS
```

OpenAI currently offers two chat models that can be used with the so called "chat completion endpoint", namely **gpt-3.5-turbo** and **gpt-4**.

Alternative to giving a Model-Name, you can specify a Model using a number, like this:

```
' Here we would specify "gpt-3.5-turbo"
AIC.Set_Model_Chat|3
```

Here is a Test-Script that you can use to see the answers of the models to a Problem. Due to the complexity i have increased the number of maximum Tokens.

```
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' Set Model
AIC.SetModel_Chat|3

' Set Model-Temperature
```

```
AIC.Set_Temperature|0.2
```

```
' Set Max-Tokens (Possible lenght of answer, depending on ...
' The more Tokens you use the more you need to pay. But the
AIC.SetMax_Token|300
```

```
ARR.Set|1|0|system:You are a technical advisor. Your name is Paul.
ARR.Set|1|1|user:What is a PowerPlug?
ARR.Set|1|2|assistant:Hallo, I am Sidney A Powerplug is ..
ARR.Set|1|3|user:How about mechanical value?
ARR.Set|1|4|assistant:Hallo I am Sidney A mechanical valve
ARR.Set|1|5|user:Tell me about this formula: $z^n = \sqrt{z}$
ARR.Send to AI|1
AIC.Ask with History|1|$$RET
AIC.gro|$$REA
DBP.$$RET
DBP.$crlf$$crlf$
DBP.JSON Output
DBP.$$REA
ENR.
```

[14:44:30] Hello, I'm Paul, a technical advisor. The formula you provided is a representation of a complex number in polar form. Let's break it down-  $z$  represents a complex number-  $n$  is the exponent or power to which the complex number is raised.  $\sqrt{z}$  represents the square root of  $z$ . Note that this formula assumes the complex number is in polar form to begin with. If you have the complex number in rectangular form (with real and imaginary parts), you would need to convert it to polar form before applying this formula. I hope this helps!

**Using the Script above we tell the AI to use the name Paul and is a technical advisor.**

| Model Number | Model Name        | Comments                                               |
|--------------|-------------------|--------------------------------------------------------|
| 1            | gpt-4             | most actual Model                                      |
| 2            | gpt-4-32k         | most actual Model with 32k Tokens (~96 kb Text In/Out) |
| 3            | gpt-3.5-turbo     | Standard Model                                         |
| 4            | gpt-3.5-turbo-16k | Standard Model with 16k Tokens (~48 kb Text In/Out)    |

If you specify "0", the default Model here is Nr.3.

**Here are some highlights about GPT-4 vs. the GPT-3.5-turbo engine.**

**GPT-4** is a newer language model developed by OpenAI, whereas **GPT-3.5-turbo** is the default engine within the ChatGPT family.

**Functions and Applications:**

- Both GPT-4 and GPT-3.5-turbo can be used to
- draft emails,
- write code,
- answer questions about documents,
- create conversational agents,
- give software a natural language interface,
- tutor in various subjects,
- translate languages,
- simulate characters for video games,
- and much more.

**GPT-4** has broad general knowledge and domain expertise and **can follow complex instructions** in natural language and **solve difficult problems** with accuracy. Conversations can be as short as 1 message or fill many pages, and including the conversation history helps when user instructions refer to prior messages.

#### **Tokens:**

Language models read text in chunks called tokens. A token can be as short as one character or as long as one word.

Both input and output tokens count toward the total tokens used in an API call.

The total number of tokens affects the cost, time, and whether the API call works at all.

*Pricing (per 25.06.2023 - prices are subject to change at any time):*

| Model                | Context Window (maxTokens) | Cost per 1K Tokens (Input) | Cost per 1K Tokens (Output) |
|----------------------|----------------------------|----------------------------|-----------------------------|
| <b>GPT-4</b>         | 8K                         | \$0.03                     | \$0.06                      |
|                      | 32K                        | \$0.06                     | \$0.12                      |
| <b>GPT-3.5-turbo</b> | 4K                         | \$0.0015                   | \$0.002                     |
|                      | 16K                        | \$0.003                    | \$0.004                     |

Both models are powerful tools for natural language processing and can be used for a wide range of applications. GPT-4 is the newer model and is likely to have improvements over GPT-3.5-turbo. However, GPT-3.5-turbo is much more cost-effective, especially for applications that don't require the absolute cutting edge in language model performance.

#### **Syntax**

```
AIC.Ask_with_History[|P1][|P2]
[|P3]
AIC.Awh[|P1][|P2][|P3]
```

**Parameter Explanation**

**P1** - <value 0-32>: This is the number of the Array that contains the "Chat History".

**P2** - **opt.** Variable to return the result / answer from the AI.

**P3** - **opt.** 0/1 - Flag: This flag is optional and is used to specify how the results should be returned when multiple results are expected. If you have set the number of expected results to a value higher than 1 using `AIC.Set Number`, this flag determines how the results are returned. If set to "1", only the last result will be returned. If set to "0" (or left as the default), all results will be returned.

**Example**

```

'*****
' EXAMPLE 1: AIC.-Commands
' Here we let the AI Calculate x for the formula "5*x^3=1450"
'
'*****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File
FOR.$$LEE|0|11

' Set Model
 AIC.SetModel_Chat|.$$LEE

' Set Model-Temperature
 AIC.Set_Temperature|0

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000)
' The more Tokens you use the more you need to pay. But the longer Input and Output
 AIC.SetMax_Token|1000

' Ask Question and receive answer to $$RET
 $$QUE=Act as a mathematician.Calculate x for the formula "5*x^3=1450". Do it s
 AIC.Ask_Chat|$$QUE|$$RET
 CLP.$$RET
 MBX.Model: $$LEE $CrLf$$RET
NEX.
:enx
ENR.

```

**Note that the Answer-Text is cut off at the end if you have specified a too small number of maximum Tokens in the Script.**

**Remarks**

In your Prompts, ensure Clarity and Precision: Articulate your prompt in a way that unambiguously communicates the desired output from the model. Refrain from using vague or open-ended language, as this can yield unpredictable outcomes.

Incorporate Pertinent Keywords: Embed keywords in the prompt that are directly associated with the subject matter. This guides the model in grasping the context and subsequently producing more precise content.

Supply Contextual Information: Should it be necessary, furnish the model with background information or context. This equips the model to formulate more informed and contextually relevant responses.

Engage in Iterative Refinement: Embrace the process of experimentation with a variety of prompts to ascertain which is most effective. Continuously refine your prompts in response to the output generated, making adjustments until the desired results are achieved.

### Limitations:

-

### See also:

- [Set Key](#)
-

## 3.42.10.8.9 Ask TTS

AIC.Ask TTS

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.Ask TTS

Converts text to spoken MP3 (in a file).

### Intention

The `AIC.Ask TTS` command is designed to convert text into an MP3 audio file using cloud services from [Open AI](#).

This command requires an [API key](#), which must be set using the `AIC.Set Key` command.

**Hint:** The model will automatically identify the written language and use the set parameters to generate speech in it.

This command does not use any Caching System, unlike the `AIC.Text to MP3`, which uses a Caching System.

### Syntax

## AIC.Ask TTS | P1 [ | P2 ] [ | P3 ]

### Parameter Explanation

**P1** - The Text to speak (convert to .mp3).

**P2** - (*optional*) The Output-Filename/Path. If omitted a temporary file is generated and the filename is in **P3** or if **P3** is omitted on TOS.

**P3** - (*optional*) Variable to receive the generated/used Filename for the mp3/Output-File.

### Example

```

' *****
' AIC.-Sample
' *****
AIC.Set Key|File
$$TXT=Alles neu macht der Mai
AIC.Ask TTS|$$TXT||$$OUT
'AIC.Show Error
' Open Folder with generated mp3-File
BLB.$$OUT|$$FOL

```



```
EXS.$$FOL
' Put Path into Clipboard
CLP.$$FOL
MBX.$$OUT
ENR.
```

### Remarks

-

### Limitations:

-

### See also:

-

## 3.42.10.9 EnsureFormatResize

[AIC.EnsureFormatResize](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.EnsureFormatResize

Automate the process of image format conversion and resizing

### Intention

This command is designed to automate the process of image format conversion and resizing.

It is particularly useful in scenarios where consistency in image format and size is required.

Validates the desired image format.

- Checks if the file is already in the desired format; if so, returns the original file path.
- Attempts to load the image; on failure, sets an error message (`AIC.Show Error`).
- Resizes the image based on `ResizeMinFlag`, `XS`, and `YS`.
- If the `ResizeMinFlag` is set to 1, this changes the resizing behavior to use `XS` and `YS` as minimum dimensions and Re-Size the picture with or without Aspect-Ratio (if `X` or `Y` is 0, this side is recalculated using the Aspect Ratio).
- Generates a unique filename for the converted image.
- Saves the image in the new format and frees the loaded image from memory.

### Syntax

```
AIC.EnsureFormatResize | P1 | P2 [| P3] [| P4] [| P5]
```

### Parameter Explanation

**P1** - (**FilePath in/out**) Variable with the path to the source image file. Also contains the destination-image file (if its a Variable). If **P1** is not a Variable, the result is placed on TOS.

**P2** - (Format) is the desired image format (e.g., 'jpg', 'png', 'bmp').

**P3** - (XSize) (optional) is the width to which the image should be resized. If set to 0, the width is auto-calculated.

**P4** - (YSize) (optional) is the height to which the image should be resized. If set to 0, the height is auto-calculated.

**P5** - (ResizeMinFlag) (optional) is a flag that, if set to 1, changes the resizing behavior to use `XS` and `YS` as minimum dimensions.

Returns the file-path of a file with the format and dimension as expected in Variable **P1**.

**P1** contains the destination-image file path (if its a Variable). If **P1** is not a Variable, the result is placed on TOS.

### Example

```
'*****
' IRS.-Sample
'*****
' Example SPR-Script to ensure format and resize an image
' FilePath: in $$PAT
' Desired Format: "png"
' Width: 800
' Height: 600
$$PAT=C:\Image.png
AIC.EnsureFormatResize|$$PAT|jpg|800|600
MBX.New Path is: $$PAT

' FilePath: in $$PAT
' Desired Format: "JPG"
' Width: 512
' Height: 0 (Keeping Aspect Ratio)
$$PAT=C:\Image.png
' If the file is smaller then 512 Px, it will be resized to
512 Px, keeping the Aspect Ratio
AIC.EnsureFormatResize|$$PAT|jpg|512|0|1
MBX.New Path is: $$PAT
ENR.
```

### Remarks

-

### Limitations:

-

### See also:

-

## 3.42.10.1 Set Negative Prompt

AIC.Set Negative Prompt

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

### AIC.Set Negative Prompt

If Result available

#### Intention

The "negative prompt" is a parameter that tells the AI model what not to include in the generated image.

At its core, a negative prompt is an instruction to tell the AI what you do NOT want, what to exclude from whichever AI you use.

Essentially, it's like telling the model, **"Create this, but don't include that"**.

#### Syntax

**AIC.Set Negative Prompt [ | P1 ]**  
**AIC.Sen [ | P1 ]**

#### Parameter Explanation

**P1 - (optional)** The Terms, words and Phrases that build together the Negative Prompt.

#### Example

```
' *****
' IRS.-Sample
' *****
```

#### Remarks

-

#### Limitations:

-

See also:

-

## 3.42.10.1: Set Positive Prompt

`AIC.Set Positive Prompt`[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.Set Positive Prompt

If Result available

### Intention

This command will set the Positive Prompt that is to be used with multiple AI-Engines, like WHISPER, Stable Diffusion etc.

### *The Positive Prompt Register is not automatically cleared.*

The positive prompt register as well as the negative prompt register is not automatically cleared.

If you use it, for example for stable diffusion, and later you are going to use another AI, for example Whisper, then the prompt is still in the positive prompt register and may influence the result. Therefore make sure to explicitly set the positive and the negative prompt register before using any of the AI systems that are influenced by this register. Mostly important, this is possibly when using Whisper, as it's influenced by the positive prompt register.

### Syntax

`AIC.Set Positive Prompt[|P1]``AIC.Set Prompt[|P1]``AIC.Sep[|P1]`

### Parameter Explanation

*P1 - (optional)* Is the "Ticket Number" of a running Parallel Robot or background Operation. It can be omitted then the last "Ticket Number" that was emitted is been taken.

IRS. can be nested to unlimited Depth, and they can also enclose Sub-Programm Calls or FEX. (Enumerations) to unlimited Depth.

### Example

```
! *****
! IRS.-Sample
! *****
```

**Remarks**

-

**Limitations:**

-

**See also:**

•

## 3.42.10.1: Set\_Model Commands

MiniRobotLanguage (MRL)

## AIC.Set Model Commands

Chose the AI solution provider (AI-Model) of your choice.



**Using the Set-Model Commands you can chose between a variety of AI-Models of different capabilities and costs.**

Generally there is a relation between the capabilities and the costs.  
The more capable an AI-Model is, the higher are the costs of usage.

Using the GPT4All "Local-"Models is generally free, as they do run on your local Computer and therefore do not use Cloud Resources.

Generally there are separate "AIC.Set Model -Commands" for each Endpoint / Ask-Command.

```
AIC.Set Model Chat -> AIC.Ask Chat
AIC.Set Model Completion -> AIC.Ask Completion
AIC.Set Model Edit -> AIC.Ask Edit
AIC.Set Model -> AIC.Ask GPT4All
```

AIC.Set Model can be used for all endpoints that use Models.

The reason for the other commands is simply that you can use numbers that simplify the process of choosing. Also its easier to iterate through models in a loop, and you have less to type.



All `AIC.Set Model` - Commands do all the same thing, they write a Model name into the same internal **Model-Name Register**.

Therefore any `AIC.Set Model` - Command will overwrite the Model Name of any previous Set-Model Command.

If you use Open AI Models at the same time with GPT4All-Models, it may be necessary to use the `AIC.Set Model-Commands` each time before calling the `AIC.Ask` Commands.

The "Image Generation Endpoint" does not use any Model, and does not use the internal **Model-Name Register**.

The other Endpoints that use Models, generally use own, exclusive Models.

There are two internal **Model-Name Registers**.

1. The Register where you insert the name of the Model that you want be used.  
This is the Register that is been set using the "Set Model" Commands.
2. The Register that is filled after the AI has done its work. In this Register you can read the real name of the Model, that was used in the last task.  
This makes sense because the AI will not always use the Model you write, but may use another Model, due to typing mistakes or internal reasons.  
This can especially happen while using GPT4All.

To read the second "Read only" Model-Register, and find out which Model processed your last activity, use:

```
'
AIC.Get Several|5|$$MOD
DBP.Last used Model: $$MOD
ENR.
```

## 3.42.10.12.1 Set Dallee Model

[AIC.Set Dallee Model](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.Set Dallee Model

Sets or resets the DALL-E model

### Intention

The `AIC.Set Dallee Model` command is designed to select a specific DALL-E model for image generation and manipulation tasks. This command allows users to choose between available DALL-E models, providing the flexibility to utilize different capabilities and features offered by each model. If no specific model is specified, the command defaults to the **"dall-e-3"** model.

### Syntax

## AIC.Set Dallee Model [ |P1 ]

### Parameter Explanation

**P1** - (*optional*) is the name of the DALL-E model to be set. If omitted, the command defaults to the "dall-e-3" model.

Currently, two models are available:

- **dall-e-3**: The default and most advanced DALL-E model.
- **dall-e-2**: An earlier version of the DALL-E model.

As more models become available, they can also be specified using this command.

### Example

```

!*****
! AIC.-Sample
!*****
! Example SPR-Script to set the DALL-E model to "dall-e-3"
AIC.Set Dallee Model|dall-e-3

! To use the "dall-e-2" model
AIC.Set Dallee Model|dall-e-2

```

### Remarks

- The availability and features of DALL-E models are subject to updates and changes by the service provider.
- Different DALL-E models may offer varying capabilities in terms of image quality, creativity, costs and processing time.

### Limitations:

-

### See also:

•

## 3.42.10.12.2 Set Model

[AIC.Set\\_Model](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.Set\_Model

Choose a Model for the AI.



The "Completion" Models try to calculate the Possibility of several Tokens/Words to complete a given Text.

### Intention

The `AIC.Set_Model` command is used for specifying the Model you want to use for **whatever AI conversation**.

Using this command you must take care to write the name of the model as it should be, else you will run into errors and not get the expected results.

The syntax of this command is

`AIC.Set_Model|<Modelname>`, where `<Modelname>` is the name of the OpenAI model you want to use.

The default model set by this command is "gpt-3.5-turbo".

The `AIC.Set_Model` command will select an Model which can then be used with multiple AI-Commands.

You have to specify the Model-Name, you can not specify a number.

If you leave the Parameter **P1** empty, the Commands will set the Model-Variable to "gpt-3.5-turbo".

Please choose a Model that is supported by the `AIC.Ask` that you want to use.

| Model ID                    | Description                                                                                                                                                                    | Token Limit (as of Sep 2021) | Cost per 1K tokens (as of July 2023) |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|--------------------------------------|
| text-davinci-001            | A version of the Davinci model that can understand and generate text in various languages, including German. Likely used for complex tasks and detailed responses.             | 4096                         | \$0.0200                             |
| text-davinci-002            | Similar to text-davinci-001, more advanced. Generates text,                                                                                                                    | 4096                         | \$0.0200                             |
| text-davinci-003            | Most advanced version of the Davinci model. Overall, the best Model for the Completion Endpoint.                                                                               | 4096                         | \$0.0200                             |
| text-babbage-003            | A version of the Babbage model, which is likely less powerful than Davinci but still very capable. Used for generating text.                                                   | 4096                         | \$0.0005                             |
| davinci                     | The base Davinci model, known for its ability to generate high-quality text and perform complex tasks.                                                                         | 4096                         | \$0.0200                             |
| text-curie-001              | A version of the Curie model, which is smaller than Davinci and is generally used for tasks that don't require as much depth. Capable of generating text in various languages. | 2048                         | \$0.0020                             |
| text-babbage-001            | Similar to text-babbage-003, but possibly a different version or configuration of the Babbage model. Used for generating text.                                                 | 4096                         | \$0.0005                             |
| text-ada-001                | A version of the Ada model, which is smaller than Curie and is generally used for simpler tasks. Capable of generating text.                                                   | 2048                         | \$0.0004                             |
| babbage                     | The base Babbage model, smaller than Davinci but still powerful and capable of generating text.                                                                                | 4096                         | \$0.0005                             |
| text-similarity-davinci-001 | A specialized version of the Davinci model used for text similarity analysis. It can understand and compare text to gauge how similar they are.                                | 4096                         | \$0.0200                             |
| babbage-code-search-code    | A specialized version of the Babbage model, likely used for searching through code or understanding programming-related queries.                                               | 4096                         | \$0.0005                             |
| text-ada-001                | Similar to Case 8, a version of the Ada model used for generating text.                                                                                                        | 2048                         | \$0.0004                             |

### Syntax

## AIC.Set Model | P1

## AIC.SMO | P1

### Parameter Explanation

**P1** - Model-Name, can be a number (see Table above) or directly the name of the model to use.

### Example

```

'*****
' EXAMPLE 1: AIC.-Commands
'*****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' Set Model
AIC.SetModel|gpt-3.5-turbo

' Set Model-Temperature
AIC.Set_Temperature|0

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000)
' The more Tokens you use the more you need to pay.
AIC.SetMax_Token|25

' Ask Question and receive answer to $$RET
AIC.Ask_Completion|What is a "Windows Button"?|$$RET
MBX.$$RET

:enx
ENR.

```



Note that the Answer-Text is cut off at the end because i have specified a maximum of 25 Tokens in the Script which is to low for the complete answer.

### Remarks

-

### Limitations:

-

### See also:

- [Set\\_Key](#)<sup>799</sup>
-

## 3.42.10.12.3 Set Model Edit

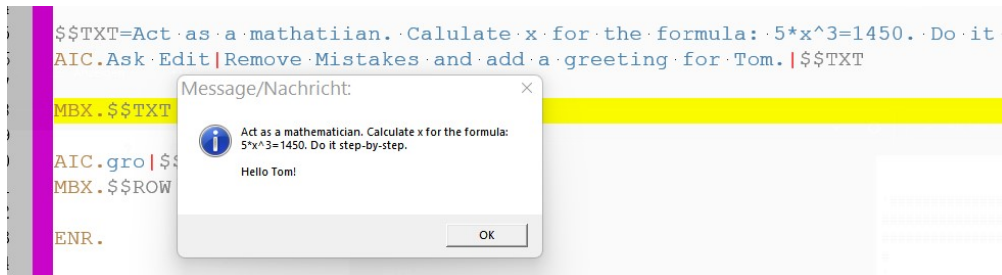
AIC.Set Model Edit

[Previous](#) [Top](#) [Next](#)

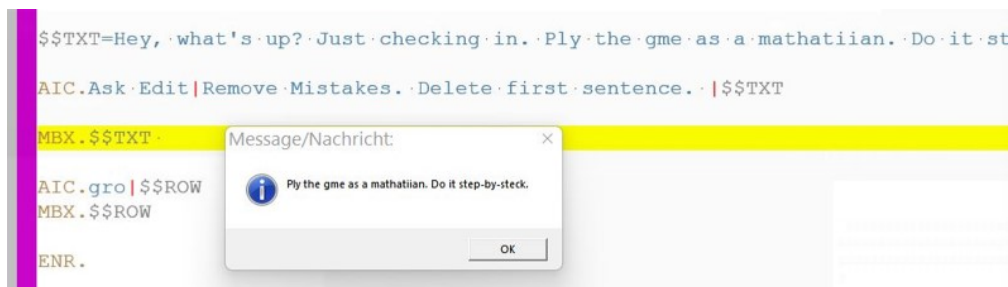
MiniRobotLanguage (MRL)

## AIC.Set Model Edit

Choose one of 2 Models available for Text-Editing.



In this Sample you can see that the AI will do corrections in Texts much better then normal Spellcheckers.



You can also have the AI remove or add words or Sentences.

### Intention

The `AIC.Set Model Edit`-Command works together with the `AIC.Ask Edit` command, which is a specialized command that is used in conjunction with the OpenAI Edits Endpoint. It is specifically designed for scenarios where you need the AI to make modifications to an existing piece of text, based on certain instructions. There will be changes in the Model, beginning 2024. For details see [here \(Open AI Blog\)](#).

### Usage

```
AIC.Ask Edit|<Instruction>|<Text to be changed>
```

This command takes two parameters:

#### 1. Instruction

This is the first parameter and should be a clear and concise directive on what kind of edit or modification needs to be done to the text.

Example: If you want the AI to remove mistakes from the text, the instruction could be "Remove mistakes".

#### 2. Text to be changed



This is the second parameter and should be the original text that you want the AI to modify based on the instruction provided.

Example: "Theres mistakes in this sentence, which needs correction."

**Full Example**

```
$$TXT=Theres mistakes in this sentence, which needs correction.
AIC.Ask Edit|Remove mistakes|$$TXT
```

In this example, the command is instructing the AI to remove mistakes from the text in \$\$TXT.

The Result of the Operation will also be in \$\$TXT. Please note that the Edits-Endpoint is rather slow in Operation.

**Notes**

It is important to ensure that the instruction is clear, as ambiguous instructions may result in unexpected edits.

The text to be changed should be well-structured to make sure that the AI understands the context properly.

The command and the parameters are separated by a vertical bar |. This is crucial for the command to be processed correctly.

Use the AIC.Ask Edit command when you need the AI to perform specific editing tasks, and always ensure that your instructions are clear and the text is contextual for the best results.

**Important Note:**

For this command, only 2 Models can be used. One is for Text, and one is for Code, mainly Python, C++ etc.

The usage is very cheap compared to the state-of-the-art Models.

Yet, the results are limited to some degree.

You can not ask for a "Rewrite as if you are Shakespear". This is something that you can get from the "chat-endpoint" models, yet these come at a much higher price.

If you fail to set these Models prior to calling the AIC.Ask Edit command, then you will not get an result.

Because the command will not change the Model it self.

These two Models are:

| Model Name            | Use Case                                   | Cost (per token)                  | Usage                                                                                          |
|-----------------------|--------------------------------------------|-----------------------------------|------------------------------------------------------------------------------------------------|
| text-davinci-edit-001 | Text editing, summarization, paraphrasing  | very cheap, but nobody knows. :-) | Suitable for natural language processing tasks like editing, summarizing or paraphrasing text. |
| code-davinci-edit-001 | Code editing, code generation, code review | very cheap, but nobody knows. :-) | Suitable for code-related tasks such as editing code,                                          |

|  |  |  |                                                                         |
|--|--|--|-------------------------------------------------------------------------|
|  |  |  | generating code snippets, or reviewing code for errors or improvements. |
|--|--|--|-------------------------------------------------------------------------|

Please note that the cost per token might change and it is best to check the [OpenAI pricing page](#) for the most up-to-date information.

```
' Usage for Ask-Edits
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' ("1") - Set Model to Text,
' ("2") - Set for Code,
' from 2024 choose ("3") that is "GPT 3.5".
AIC.SetModel_Edit|1

' Set Max-Tokens (Possible lenght of answer, depending on
' The more Tokens you use the more you need to pay. But th
AIC.SetMax_Token|300

$$TXT=Hey, what's up? Just checking in. Ply the gme as a m
AIC.Ask Edit|Remove Mistakes. Delete first sentence. |$$TX

' Get the Result
MBX.$$TXT
ENR.
```

### Syntax

**AIC.Ask Edit|P1|P2**  
**AIC.ase|P1|P2**

### Parameter Explanation

**P1** - opt. Variable with Instructions what to do. If omitted the default Instruction is "Remove typing mistakes"

**P2** - Variable with Text, and also will receive the corrected Result. Can not be omitted.

### Example

```
'*****
' EXAMPLE 1: AIC.-Commands
'*****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000 ?
' The more Tokens you use the more you need to pay. But the longer Input and Out
AIC.SetMax_Token|300

' Set Model
AIC.SetModel_Edit|0

$$TXT=Hey, what's up? Just checking in. Ply the gme as a mathatiian. Do it step-
AIC.Ask Edit|Remove Mistakes. Delete first sentence.|$$TXT
MBX.$$TXT

' See the Original Output of the Command (JSON Format).
' This is useful in case of Errors.
AIC.gro|$$ROW
MBX.$$ROW
ENR.
```

### Remarks

-

### Limitations:

-

### See also:

- [DMP. Dump System Values](#)<sup>[1823]</sup>

## 3.42.10.12.4 Set TTS Model

[AIC.Set TTS Model](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.Set TTS Model

Sets or resets the Text-to-Speech (TTS) model

### Intention

The `AIC.Set TTS Model` command is designed to select a specific Text-to-Speech (TTS) model for AI operations.

This command allows users to either set a specific TTS model or revert to the default model if no parameter is specified.

It is essential for customizing the voice synthesis process to suit different requirements.

### Syntax

```
TMP. [P1] ... ELS. ... EIF.
```

### Parameter Explanation

*P1* - (*optional*) the name of the TTS model to be set. If this parameter is omitted, the command will reset to the default TTS model.

Currently, two models are available:

**tts-1-hd**: High-definition TTS model (Default). May take a bit longer in processing.

**tts-1**: Standard TTS model. A bit faster yet a bit less quality.

### Example

```
'*****
' AIC.-Sample
'*****
' Example SPR-Script to set the TTS model to high-definition
AIC.Set TTS Model|tts-1-hd

' Reset to default TTS model
AIC.Set TTS Model
ENR.
```

### Remarks

Availability of TTS models may vary and is subject to updates or changes by the AI service provider.

The specific characteristics and capabilities of each TTS model can differ. New Models can be specified when they are available.

### Limitations:

- 

### See also:

-

## 3.42.10.12.5 Set Vision Detail

[AIC.Set Vision Detail](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.Set Vision Detail

Configures the level of detail for image processing in OpenAI's vision API. It adjusts how the model processes the image and generates its textual understanding.

### Intention

This command is intended to offer users control over the fidelity of image analysis – whether to prioritize speed and efficiency with a lower level of detail, or to seek a more detailed analysis at the expense of more resources and time.

### Detail Level Explained

- **Low Fidelity (1 or "Low"):**
  - In this mode, the model receives a low-resolution **512 x 512** version of the image and represents it with a budget of 65 tokens.
  - This mode is optimal for faster responses and fewer token consumption, suitable for use cases not requiring high detail.
- **High Fidelity (2 or "High"):**
  - This mode enables detailed image processing. Initially, the model sees the low-resolution image, followed by detailed crops of the input images as 512px squares.
  - Each detailed crop uses a token budget of 65 tokens, amounting to a total of 129 tokens.
  - Opt for this mode when detailed image analysis is required.

### Syntax

## AIC.Set Vision Detail[|P1]

### Parameter Explanation

**P1** - (optional) is the level of detail setting for image processing. It can be set to:

- **Low Fidelity (1 or "Low"):**
  - In this mode, the model receives a low-resolution 512 x 512 version of the image and represents it with a budget of 65 tokens.

- This mode is optimal for faster responses and fewer token consumption, suitable for use cases not requiring high detail.
- **High Fidelity (2 or "High"):**
  - This mode enables detailed image processing. Initially, the model sees the low-resolution image, followed by detailed crops of the input images as 512px squares.
  - Each detailed crop uses a token budget of 65 tokens, amounting to a total of 129 tokens.
  - Opt for this mode when detailed image analysis is required.

#### Example

```

'*****
' AIC.-Sample
'*****
'Example SPR-Script to
' set vision detail to high fidelity
AIC.Set Vision Detail|2
' Or, to set it to low fidelity
AIC.Set Vision Detail|1

```

#### Remarks

Selecting the appropriate detail level is crucial based on the specific requirements of your image processing task. Higher detail provides more comprehensive analysis but consumes more resources.

The choice of detail level directly impacts the response time and token consumption of the AI model.

The Chat Completions API, being non-stateful, requires the image to be passed each time a request is made.

#### Limitations:

-

#### See also:

•

## 3.42.10.12.6 Set Vision Model

[AIC.Set Vision Model](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.Set Vision Model

Sets or resets the vision model used for image processing tasks

### Intention

The `AIC.Set Vision Model` command is designed to specify the model used for vision-related AI operations. This command enables users to select the appropriate model for their image processing needs.

Currently, the default model is "gpt-4-vision-preview". As new models become available, they can be selected using this command.

### Syntax

## AIC.Set Vision Model [ |P1 ]

### Parameter Explanation

**P1** - (*optional*) - is the name of the vision model to be set. If omitted, the command defaults to the "gpt-4-vision-preview" model.

Currently, only one model is available:

- `gpt-4-vision-preview`: The default and currently the only available vision model.

As more models are introduced, they can be selected using this command.

### Example

```

' *****
' AIC.-Sample
' *****
' Example SPR-Script to set the vision model to "gpt-4-vision-preview"
AIC.Set Vision Model|gpt-4-vision-preview

' Reset to default vision model (if other models are available)
AIC.Set Vision Model
ENR.

```

### Remarks



- The availability and capabilities of vision models are subject to updates and changes by the service provider.
- Different vision models may have varying strengths in terms of image recognition, analysis, and processing capabilities.

### Limitations:

- 

### See also:

-

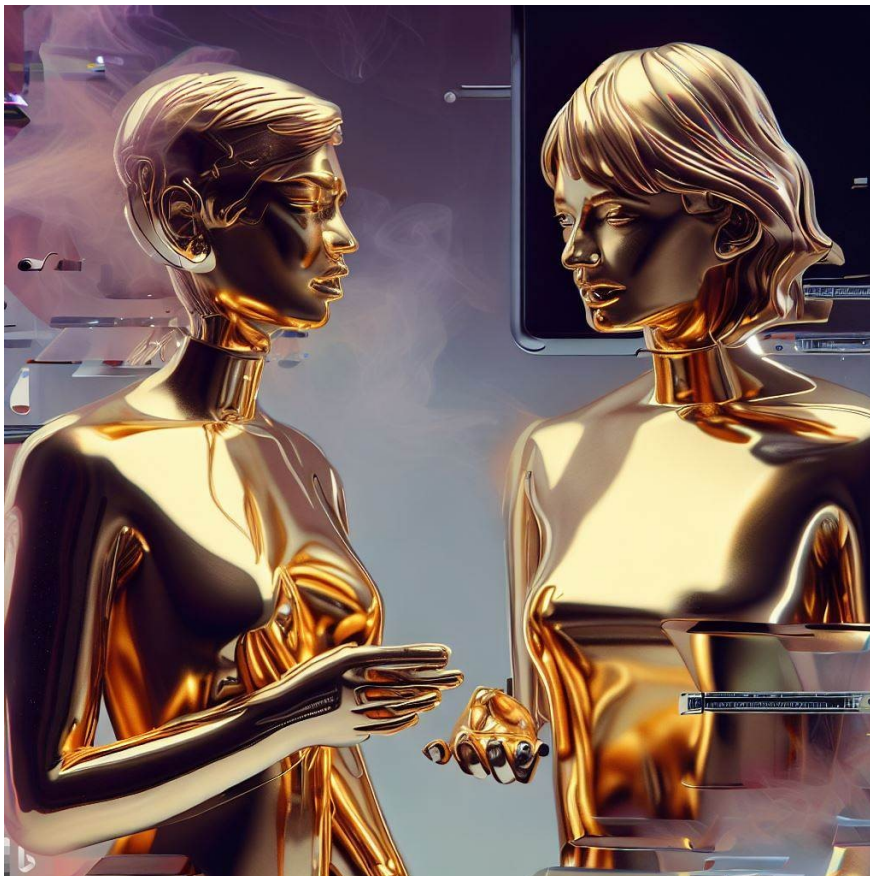
## 3.42.10.12.7 Set\_Model\_Chat

[AIC.Set\\_Model\\_Chat](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.Set\_Model\_Chat

Choose one of the Open AI Models that are available for the "Chat endpoint".



The Chat Endpoint Models try to give you the best possible answer to your instructions/questions.

### Intention

The `AIC.Set_Model_Chat` command is used for specifying the OpenAI model you want to use for the `AIC.Ask Chat` - Command.

The syntax of this command is

```
AIC.Set_Model_Chat|<Modelname or Model-Number>
```

where `<Modelname>` is the name of the OpenAI model you want to use. The default model set by this command is `gpt-3.5-turbo`.

At the time of writing, OpenAI offers two chat models that can be used with the chat completion endpoint, namely `gpt-3.5-turbo` and `gpt-4`.

These models can be used to build various applications such as drafting emails, writing Python code, answering questions, creating conversational agents, tutoring, language translation, and simulating characters for video games among others.

Both commands:

`AIC.Set_Model_Chat`: accesses the advanced AI models like GPT 3.5 and 4.  
and

`AIC.Set_Model_Completion`: Best suited for single-turn tasks, and "cheap tasks".  
do internally use the same "Model-Register", therefore they will overwrite any Model that was selected before.

Alternative to giving a Model-Name, you can specify a Model using a number, like this:

```
' Here we would specify "gpt-3.5-turbo"
AIC.Set_Model_Chat|3
```

| Number (P1) | Model (P1)           | Comments                                                                                                      |
|-------------|----------------------|---------------------------------------------------------------------------------------------------------------|
| 1           | gpt-3.5-turbo        | A variant of GPT-3.5 with enhanced performance and speed, suitable for responsive and efficient tasks.        |
| 2           | gpt-4                | Represents the next generation of language models with advanced comprehension and contextual capabilities.    |
| 3           | gpt-3.5-turbo-1106   | A specific iteration of the GPT-3.5 turbo model, likely optimized for certain tasks or features.              |
| 4           | gpt-3.5-turbo-16k    | An extended version of GPT-3.5 turbo, possibly with increased knowledge base or specialized capabilities.     |
| 5           | gpt-4-1106-preview   | An early or preview version of GPT-4, showcasing newer features or improvements over previous models.         |
| 6           | gpt-4-vision-preview | A version of GPT-4 with integrated vision capabilities, indicating a blend of language and visual processing. |
| 0           | gpt-3.5-             | Default model, a variant of GPT-3.5 turbo for general use, balancing performance and features.                |

| Number (P1) | Model (P1) | Comments |
|-------------|------------|----------|
|             | turbo-1106 |          |

## Temporary Model for 2023:

The following two Models have been announced by OpenAI in 11/2023 and are temporary, why we did not hard-code these.  
To select these, use directly the name.

New GPT-4 Turbo: "**gpt-4-1106-preview**" (Use `AIC.Ask_Chat`)

We announced GPT-4 Turbo, our most advanced model. It offers a **128K context window** and knowledge of world events up to April 2023.

We've reduced pricing for **GPT-4 Turbo** considerably: input tokens are now priced at **\$0.01/1K** and output tokens at **\$0.03/1K**, making it 3x and 2x cheaper respectively compared to the previous GPT-4 pricing.

We've improved function calling, including the ability to call multiple functions in a single message, to always return valid functions with JSON mode, and improved accuracy on returning the right function parameters.

Model outputs are more deterministic with our new reproducible outputs beta feature.

You can access GPT-4 Turbo by passing **gpt-4-1106-preview** in the API, with a stable production-ready model release planned later this year.

Updated GPT-3.5 Turbo: "**gpt-3.5-turbo-1106**" (Use `AIC.Ask_Completion`)

The new gpt-3.5-turbo-1106 supports **16K context** by default and that 4x longer context is available at lower prices: **\$0.001/1K** input, **\$0.002/1K** output. Fine-tuning of this 16K model is available.

Fine-tuned GPT-3.5 is much cheaper to use: with input token prices decreasing by 75% to \$0.003/1K and output token prices by 62% to \$0.006/1K.

**gpt-3.5-turbo-1106** joins GPT-4 Turbo with improved function calling and reproducible outputs.

Here are some highlights about GPT-4 and ChatGPT with the GPT-3.5-turbo engine.

**GPT-4** is a newer language model developed by OpenAI, whereas **GPT-3.5-turbo** is the default and even faster engine within the ChatGPT family.

### Functions and Applications:

- Both GPT-4 and GPT-3.5-turbo can be used to
- draft emails,
- write code,
- answer questions about documents,
- create conversational agents,
- give software a natural language interface,
- tutor in various subjects,

- translate languages,
- simulate characters for video games, and much more.

**GPT-4** has broad general knowledge and domain expertise and **can follow complex instructions** in natural language and **solve difficult problems** with accuracy. Conversations can be as short as 1 message or fill many pages, and including the conversation history helps when user instructions refer to prior messages. **GPT-3.5** is cheaper, is faster and can solve simple problems as well.

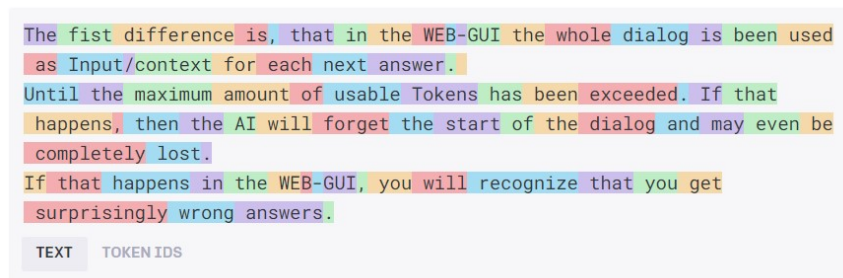
### Tokens:

Language models read text in chunks called tokens. A token can be as short as one character or as long as one word.

Both input and output tokens count toward the total tokens used in an API call.

The total number of tokens affects the cost, time, and whether the API call works at all.

| Tokens | Characters |
|--------|------------|
| 82     | 364        |



Here you can see how a Text is divided into Tokens before being processed using the AI.

Both models are powerful tools for natural language processing and can be used for a wide range of applications.

GPT-4 is the newer model and is likely to have improvements over GPT-3.5-turbo. However, GPT-3.5-turbo is faster and much more cost-effective, especially for applications that don't require the absolute cutting edge in language model capabilities.

### Syntax

## AIC.Set Model Chat | P1

## AIC.SMH | P1

### Parameter Explanation

**P1** - Model-Name, can be a number (see Table above) or directly the name of the model to use.

**Example**

```

'*****
' EXAMPLE 1: AIC.-Commands
'*****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' Set Model
AIC.SetModel_Chat|4

' Set Model-Temperature
AIC.Set_Temperature|0

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000)
' The more Tokens you use the more you need to pay.
AIC.SetMax_Token|25

' Ask Question and receive answer to $$RET
AIC.Ask_Chat|What is a "Windows Button"|$$RET
MBX.$$RET

:enx
ENR.

```



Note that the Answer-Text is cut off at the end because i have specified a maximum of 25 Tokens in the Script.

**Remarks**

-

**Limitations:**

-

**See also:**

- [Set\\_Key](#)<sup>799</sup>
- [Set\\_Model\\_Completion](#)<sup>910</sup>

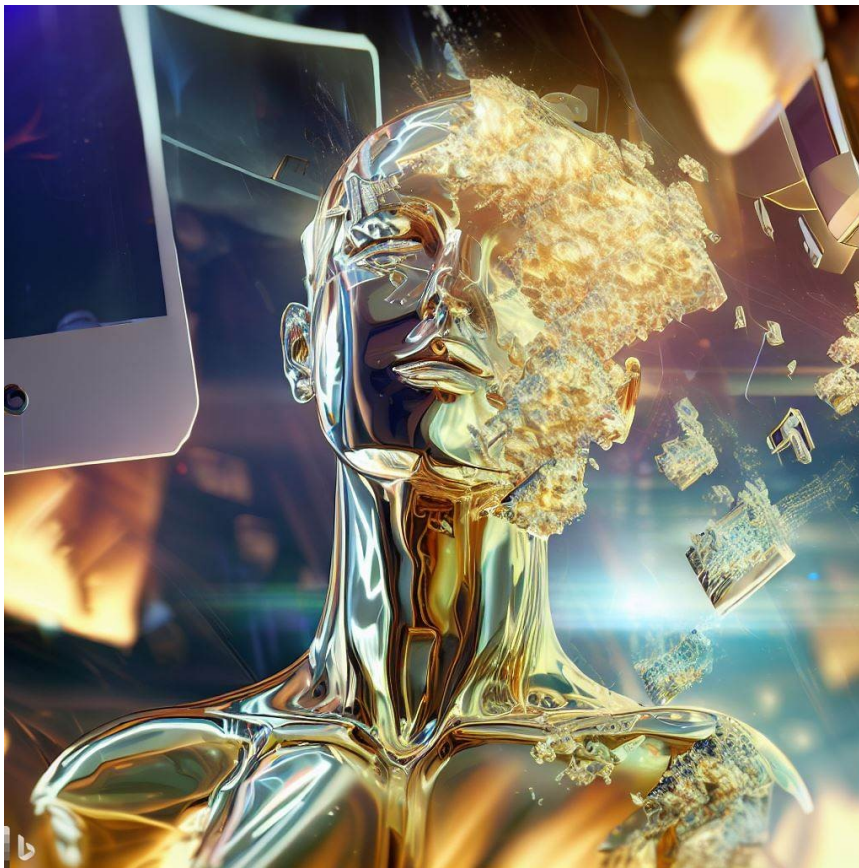
## 3.42.10.12.8 Set\_Model\_Completion

[AIC.Set\\_Model\\_Completion](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.Set\_Model\_Completion

Choose one of the Open AI Models that are available for the "Completion endpoint".



The "Completion" Models try to calculate the Possibility of several Tokens/Words to complete a given Text.

### Intention

The `AIC.Set_Model_Completion` command is used for specifying the OpenAI model you want to use for **Completion-based conversations**.

"Completion based" means that the Model will evaluate the answer "to complete your question" with the highest possibility.

The syntax of this command is

`AIC.Set_Model_Completion|<Modelname>`, where `<Modelname>` is the name of the OpenAI model you want to use.

The default model set by this command is `gpt-3.5-turbo-instruct`.



OpenAI offers several Completion models that can be used with the `Completion` endpoint, see Table below.

These models can be used in various applications, such as

- drafting emails,
- writing Python code,
- answering questions,
- creating conversational agents,
- tutoring, language translation,
- and even simulating characters for video games
- among others.

The `AIC.Set_Model_Completion` command will select an Model which can then be used with the

`AIC.Ask Completion-Command`

Alternative to giving a Model-Name, you can specify a Model number, like this:

```
' Here we would specify "davinci-002"
AIC.Set_Model_Completion|3
```

Here's an updated table with comments about each model based on general characteristics and applications:

| Number (R02) | Model (W01)            | My Comments                                                                                                                  |
|--------------|------------------------|------------------------------------------------------------------------------------------------------------------------------|
| 1            | gpt-3.5-turbo-instruct | Advanced, versatile, suitable for complex and nuanced tasks requiring a higher understanding and instructional capabilities. |
| 2            | babbage-002            | Efficient for straightforward tasks, offers balanced performance for a wide range of applications.                           |
| 3            | davinci-002            | Highly capable for creative and complex tasks, excels in generating human-like text and detailed responses.                  |
| 0            | gpt-3.5-turbo-instruct | Default choice, combines advanced capabilities with instructional tuning for a broad spectrum of tasks.                      |

**Syntax**

**AIC.Set Model Completion|P1**  
**AIC.SMC|P1**

**Parameter Explanation**



**P1** - Model-Name, can be a number (see Table above) or directly the name of the model to use.

### Example

```

'*****
' EXAMPLE 1: AIC.-Commands
'*****
' Set OpenAI API-Key from the saved File
AIC.SetKey|File

' Set Model
AIC.SetModel_Completion|4

' Set Model-Temperature
AIC.Set_Temperature|0

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000)
' The more Tokens you use the more you need to pay.
AIC.SetMax_Token|25

' Ask Question and receive answer to $$RET
AIC.Ask_Completion|What is a "Windows Button"?|$$RET
MBX.$$RET

:enx
ENR.

```



**Note that the Answer-Text is cut off at the end because i have specified a maximum of 25 Tokens in the Script which is to low for the complete answer.**

### Remarks

-

### Limitations:

-

### See also:

- [Set\\_Key](#)<sup>799</sup>
-

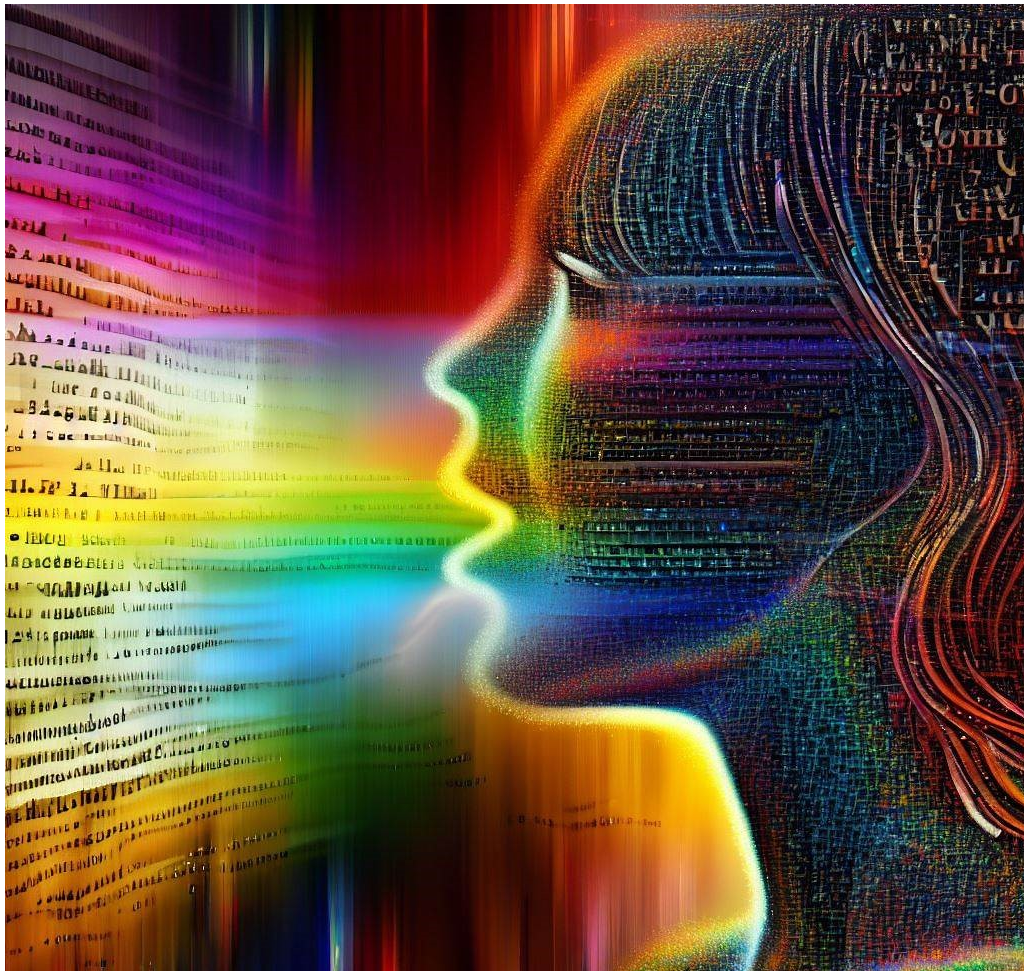


## 3.42.10.1! Open AI TTS

TTS - Text to Speech

[Previous](#) <sup>1492</sup> [Top](#) <sup>49</sup> [Next](#) <sup>1499</sup>

MiniRobotLanguage (MRL)

**Intention**

Open AI has released the option to generate realistic human speech via the API on the DevCon in November 2023.

The Smart Package Robot implements this feature in a easy usable way.

We have used the same commands and also the same Caching Options that are also available for the [AIS.-Command and for the Elevenlabs Speech Synthesis](#) <sup>999</sup>.

Open AI TTS follows WHISPER and automatically supports these languages:

Afrikaans, Arabic, Armenian, Azerbaijani, Belarusian, Bosnian, Bulgarian, Catalan, Chinese, Croatian, Czech, Danish, Dutch, English, Estonian, Finnish, French, Galician, German, Greek, Hebrew, Hindi, Hungarian, Icelandic, Indonesian,

Italian, Japanese, Kannada, Kazakh, Korean, Latvian, Lithuanian, Macedonian, Malay, Marathi, Maori, Nepali, Norwegian, Persian, Polish, Portuguese, Romanian, Russian, Serbian, Slovak, Slovenian, Spanish, Swahili, Swedish, Tagalog, Tamil, Thai, Turkish, Ukrainian, Urdu, Vietnamese, and Welsh.

### Example

```
'*****
' AIC.-Sample
'*****
AIC.Set Key|file
$$TXT=Hello, this is Sarah from Tech Support.
AIC.Say Text|$$TXT|-
PAU.10
ENR.
```

### Remarks

-

### Limitations:

-

## 3.42.10.13.1 Get Folder

[AIC.Get Folder / AIC.gfo](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.Get Folder / AIC.gfo

Retrieves cache folder path.

### Intention

The `AIC.Get Folder` command is used to retrieve the current path of the folder where the caching system stores the generated MP3 files.

The default path is `?exeloc\AIC_Folder\`

### Syntax

**AIC.Get Folder [|P1]****AIC.gfo [|P1]**

### Parameter Explanation

**P1: Optional.** The variable where the current cache folder path will be stored. If omitted, the folder path is placed on the Top of Stack (TOS).

### Example

```
! *****
! AIC.-Sample
! *****
AIC.Get Folder|$$FOL
MBX.$$FOL
ENR.
```

### Remarks

-

### Limitations:

-

See also:

-

## 3.42.10.13.2 Play MP3

[AIC.PlayMP3 / AIC.pmp](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.PlayMP3 / AIC.pmp

Plays specified MP3 file.

### Intention

Conditional Statement.

The `AIC.PlayMP3` command plays an MP3 file specified by the parameter **P1**. If **P1** is omitted, the last played file will be played again.

Using this command you can immediately play the mp3-files that have been delivered from AIC.

Alternatively you can use the [MPA. etc.-Commands](#).

### Syntax

```
AIC.PlayMP3 [| P1]
AIC.pmp [| P1]
```

### Parameter Explanation

**P1: Optional.** The file name of the MP3 file to play. If omitted, the last played file will be replayed.

### Example

```
! *****
! AIC.-Sample
! *****
$$FIL=?path\MySong.mp3
AIC.PlayMP3|$$FIL
```

### Remarks

- 

**Limitations:**

- 

**See also:**

-



## 3.42.10.13.3 Say Text

[AIC.Say Text](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.Say Text

Converts text to spoken MP3 (in a file) and plays it back immediately.

### Intention

The `AIC.Say Text` command is designed to convert text into an MP3 audio file using cloud services from Open AI and then Play this Audio-File back.

Technically this command is a combination of `AIC.Text to MP3` and `AIC.Play MP3` in one command.

This command requires an [API key](#), which must be set using the `AIC.Set Key` command.

**Hint:** You can speak in many different languages. The model will automatically identify the written language and use the set parameters to generate speech in it.

## The Caching System

**There is a Caching System built into this command. Here's how it works and why it's beneficial:**

In the observation of standard conversational patterns, it becomes evident that certain words and phrases are recurrently employed.

Given that the utilization of the Open AI Cloud incurs a cost, it is judicious to implement a caching mechanism.

When the Smart Package Robot identifies the repetition of specific words or phrases, it retrieves these elements from the cache rather than initiating a redundant request to the Open AI Cloud.

It should be noted that this caching feature is optional; specifying a filename will bypass the mechanism altogether.

The employment of a caching mechanism offers the distinct advantage of **expedited language availability** and of course lower cost, compared to awaiting MP3 delivery from the Aopen AI Cloud. However, it's important to acknowledge a minor drawback.

The Open AI Cloud is designed to never vocalize the exact same sentence in an identical manner. Consequently, bypassing the cache can lend a more authentic feel to the conversation.

On the other hand, repeated use of the cache may result in **noticeable uniformity** when the same sentence is audibly identical over time.

Ultimately, the choice of whether to utilize this feature rests with the user, allowing for customization based on individual preferences.

There is also a way to have the Smart Package Robot delete the actual recording and generate it new via Cloud.

If you do this:

```
AIC.Say Text|$$TXT|-
```

Then the Script will just re-generate the saved mp3-file with a new version, and save the new version in the Cache.

If **P2** is omitted, the system will automatically generate a **cache folder** to store all MP3 files along with a checksum. This feature helps in reducing costs and saving resources by reusing the same MP3 file for repeated phrases.

**Note:** The system is case-sensitive, meaning the same phrase with different cases will generate different checksums. Be mindful of this when using the command.

This is intentional to keep compatibility with special Commands that may need upper and lowercase characters.

**Efficiency and Speed:** The built-in caching system serves multiple purposes, primarily aimed at optimizing resource usage and reducing costs.

When the same text is converted to MP3 multiple times, the system retrieves the already generated MP3 file from the cache instead of making a new API call to Open AI . This speeds up the process significantly.

**Cost-Effectiveness:** API calls usually come with a cost. By utilizing a caching system, you minimize the number of API calls made to Open AI , thereby saving money.

**Resource Optimization:** Generating an MP3 from text consumes computational resources. Caching allows the system to avoid redundant operations, thus saving CPU cycles and memory usage.

**Customization and Control:**The commands AIC.Set Folder and AIC.Get Folder allow you to specify the directory where the cached MP3 files and their checksums are stored. This gives you control over the organization of these files, making it easier to manage them.

**Case Sensitivity:**The system intentionally does not alter the case of the text when generating the checksum. This means that the same text with different casing will be treated as different phrases, each with its own cached MP3. This feature allows for precise control but also means you should be mindful of text casing to maximize the benefits of caching.

**In summary**, the built-in caching system is designed to make the command more efficient, cost-effective, and user-friendly.

**The Location of the default cache folder is: "?exeloc\AIC\_Folder\"**

```
AIC.Set Key|<YourAPIKeyHere>
AIC.Say Text|Hello World|?path/to/save.mp3
```

In this example, the text "Hello World" will be converted into an MP3 file and saved in the specified path. Then the command will Play back the MP3-File and SAY the Phrase.

### Syntax

## AIC.Say Text|P1 [|P2]

### AIC.Say|P1 [|P2]

#### Parameter Explanation

- **P1:** The text you want to convert into an MP3 file.
- **P2:** Optional. The path where the generated MP3 file will be saved. The caching system will automatically be used if **P2** is omitted or empty.

If **P2** is omitted, the system will automatically generate a cache folder to store all MP3 files along with a checksum.

This feature helps in reducing costs and saving resources by reusing the same MP3 file for repeated phrases.

**Note:** The system is case-sensitive, meaning the same phrase with different cases will generate different checksums.

The use of different voices is not part of the caching, means the cache knows only the Text, not which voice it is.

To use the caching with different voices and cache them separately, you will need to Change the Cache Folder location using the `AIC.Set Folder - Command`.

#### Example

```

'*****
' AIC.-Sample
' Demonstration of speaking in many languages
'*****
MBX.We start with Windows native speaking

$$TXT=Hello, this is Sarah from Tech Support.
SAY.$$TXT|w

$$TXT=Hi Sarah, I'm having some issues with my computer.
SAY.$$TXT|w

$$TXT=Buenas tardes, soy Sarah del departamento de asistencia técnica.
SAY.$$TXT|w

$$TXT=Buenas tardes, Sarah. Mi ordenador no funciona bien; se bloquea todo el t:
SAY.$$TXT|w

MBX.Now with the Elevenlabs Cloud Technology

:new
AIC.Set Key|file
$$TXT=Hello, this is Sarah from Tech Support.
AIC.Say Text|$$TXT|-

$$TXT=Hi Sarah, I'm having some issues with my computer.
AIC.Say Text|$$TXT

$$TXT=Buenas tardes, soy Sarah del departamento de asistencia técnica.
```

```
AIC.Say Text|$$TXT
```

```
$$TXT=Buenas tardes, Sarah. Mi ordenador no funciona bien; se bloquea todo el t:
AIC.Say Text|$$TXT
```

```
$$TXT=Bonjour, c'est Sarah du support technique.
AIC.Say Text|$$TXT
```

```
$$TXT=Bonjour Sarah, j'ai quelques problèmes avec mon ordinateur.
AIC.Say Text|$$TXT
```

```
$$TXT=Ciao, sono Sarah del supporto tecnico.
AIC.Say Text|$$TXT
```

```
$$TXT=Ciao Sarah, sto avendo alcuni problemi con il mio computer.
AIC.Say Text|$$TXT
```

```
$$TXT=Guten Tag, hier ist Sarah vom technischen Support.
AIC.Say Text|$$TXT
```

```
$$TXT=Guten Tag Sarah, wir haben einige Probleme mit dem Computer.
AIC.Say Text|$$TXT
```

```
MBX.!
ENR.
```

### Remarks

- The API key must be set using the **AIC.Set Key** command before using this command.
- The system will use a cache to save resources **only if P2 is omitted or empty.**

### Limitations:

-

### See also:

-

## 3.42.10.13.4 Set Folder

[AIC.Set Folder / AIC.sef](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.Set Folder / AIC.sef

Sets cache folder path.

### Intention

The `AIC.Set Folder` command sets the folder where the caching system will store the generated MP3 files.

If `P1` is omitted it will reset the folder path to the default value, that is "?

`exeloc\AIC_Folder\`"

### Syntax

**AIC.Set Folder**[|P1]**AIC.Sef**[|P1]

### Parameter Explanation

**P1: Optional.** The path to the folder where MP3 files will be cached. If omitted, the default folder `?exeloc\AIC_Folder\` will be used.

### Example

```

' *****
' AIC.-Sample
' *****
$$FOL=?exeloc\Voice_02_Cache\
AIC.Set Folder|$$FOL

```

### Remarks

- If the specified folder does not exist, it will be created.

- Files in the folder may be overwritten without warning.

### Limitations:

-

### See also:

-

## 3.42.10.13.5 Set TTS Format Any

[AIC.Set TTS Format Any](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.Set TTS Format Any

Set the audio format for Text-to-Speech (TTS) output

### Intention

The `AIC.Set TTS Format Any` command allows you to specify the audio format for the output of Text-to-Speech operations.

This flexibility is crucial for ensuring compatibility with various platforms and usage scenarios.

The default format is `"mp3"`, but other formats like `"opus"`, `"aac"`, or `"flac"` are also available, catering to different needs.

### Syntax

## AIC.Set TTS Format Any|P1

### Parameter Explanation

**P1** – The desired audio format. If omitted, the default format ("mp3") is used.

Available formats:

- **"mp3"**: The default format, widely compatible with most systems and devices.
- **"opus"**: Optimized for internet streaming and communication, offering low latency.
- **"aac"**: Commonly used for digital audio compression, preferred by platforms like YouTube, and supported on Android and iOS devices.
- **"flac"**: Provides lossless audio compression, ideal for audio enthusiasts and archival purposes.

### Example

```

'*****
' AIC.-Sample
'*****
' Set the TTS format to "opus" for streaming
AIC.Set TTS Format Any|opus

' Reset to the default "mp3" format
AIC.Set TTS Format Any|mp3

```

**Remarks**

-

**Limitations:**

-

**See also:**

-



## 3.42.10.13.6 Set TTS Voice

AIC.Set TTS Voice

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.Set TTS Voice

Selects a specific voice for Text-to-Speech (TTS) operations using different options

### Intention

The `AIC.Set TTS Voice` command is used to choose a specific voice from a range of available options for Text-to-Speech operations. This command enhances the customization of TTS by offering different voice types, catering to diverse preferences and requirements. Users can select a voice **either by specifying its number or by entering a name**, with an internal algorithm determining the best match.

### Syntax

## AIC.Set TTS Voice [|P1]

### Parameter Explanation

**P1** - (*optional*) The voice to be selected. This can be a number corresponding to the voice's index or the name of the voice.

Available voices:

- - `0` or "alloy"
- - `1` or "echo"
- - `2` or "fable"
- - `3` or "onyx"
- - `4` or "nova"
- - `5` or "shimmer"

### Example

```

|*****
| AIC.-Sample
|*****
| Set the TTS voice to "echo" by name
AIC.Set TTS Voice|echo

```

```
' Set the TTS voice to "fable" by number
AIC.Set TTS Voice|2
```

### Remarks

- Selecting the appropriate TTS voice is crucial for achieving the desired tone and style in voice synthesis. The internal algorithm facilitates easy selection by name.
- The availability of voices might vary and is subject to updates or changes by the service provider.
- The voice selection algorithm's effectiveness in matching names to voices may vary.

### Limitations:

-

### See also:

•

## 3.42.10.13.7 Set TTS Voice Any

[AIC.Set TTS Voice Any](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.Set TTS Voice Any

Select a Text-to-Speech voice by name without internal processing

### Intention

The `AIC.Set TTS Voice Any` command allows for the direct selection of a Text-to-Speech (TTS) voice based solely on its name.

This command is different from `AIC.Set TTS Voice` in that it accepts only the exact name of the voice and does not perform any internal matching or processing. It is designed for future flexibility, enabling users to select new voices as they become available.

### Syntax

## AIC.Set TTS Voice Any[|P1]

### Parameter Explanation

**P1** - (*optional*) The exact name of the TTS voice to be selected. This command requires the precise name and does not interpret or process the input.

### Example

```

'*****
' AIC.-Sample
'*****
' Set the TTS voice to "nova" using the exact name
AIC.Set TTS Voice Any|nova

' In the future, to set a new voice that has been added
AIC.Set TTS Voice Any|new-voice-name

```

### Remarks

- This command is ideal for advanced users who are aware of the specific names of the TTS voices and wish to use voices that may not be available in the standard selection process.
- The command relies on the user providing the exact correct name of the voice.

- If an incorrect or non-existent voice name is provided, the result will be undefined.

### Limitations:

-

### See also:

•

## 3.42.10.13.8 Text to MP3

[AIC.Text to MP3](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## AIC.Text to MP3

Converts text to spoken MP3 (in a file).

### Intention

The `AIC.Text to MP3` command is designed to convert text into an MP3 audio file using cloud services from [Open AI](#). This command requires an [API key](#), which must be set using the `AIC.Set Key` command.

**Hint:** The model will automatically identify the written language and use the set parameters to generate speech in it.

## The Caching System

**There is a Caching System built into this command. Here's how it works and why it's beneficial:**

In the observation of standard conversational patterns, it becomes evident that certain words and phrases are recurrently employed.

Given that the utilization of the Open AI Cloud **incurs a cost**, for each spoken word, it is judicious to implement a **caching mechanism**.

When the Smart Package Robot identifies the repetition of specific words or phrases, it retrieves these elements from the cache rather than initiating a redundant request to the Open AI Cloud.

It should be noted that this caching feature is optional; specifying a filename will bypass the mechanism altogether.

The employment of a caching mechanism offers the distinct advantage of expedited language availability compared to awaiting MP3 delivery from the Open AI Cloud. However, it's important to acknowledge a minor drawback. The Open AI AI is designed to never vocalize the exact same sentence in an identical manner.

Consequently, bypassing the cache can lend a more authentic feel to the conversation. On the other hand, repeated use of the cache may result in noticeable uniformity when the same sentence is audibly identical over time.

Ultimately, the choice of whether to utilize this feature rests with the user, allowing for customization based on individual preferences.

There is also a way to have the Smart Package Robot delete the actual recording and generate it new via Cloud.

If you do this:

```
AIC.Say Text|$$TXT|-
```

Then the Script will just re-generate the saved mp3-file with a new version, and save the new version in the Cache.

If **P2** is omitted, the system will automatically generate a **cache folder** to store all MP3 files along with a checksum. This feature helps in reducing costs and saving resources by reusing the same MP3 file for repeated phrases.

**Note:** The system is case-sensitive, meaning the same phrase with different cases will generate different checksums. Be mindful of this when using the command.

This is intentional to keep compatibility with special Commands that may need upper and lowercase characters.

**Efficiency and Speed:** The built-in caching system serves multiple purposes, primarily aimed at optimizing resource usage and reducing costs.

When the same text is converted to MP3 multiple times, the system retrieves the already generated MP3 file from the cache instead of making a new API call to Open AI . This speeds up the process significantly.

**Cost-Effectiveness:** API calls usually come with a cost. By utilizing a caching system, you minimize the number of API calls made to Open AI , thereby saving money.

**Resource Optimization:** Generating an MP3 from text consumes computational resources. Caching allows the system to avoid redundant operations, thus saving CPU cycles and memory usage.

**Customization and Control:**The commands AIC.Set Folder and AIC.Get Folder allow you to specify the directory where the cached MP3 files and their checksums are stored. This gives you control over the organization of these files, making it easier to manage them.

**Case Sensitivity:**The system intentionally does not alter the case of the text when generating the checksum. This means that the same text with different casing will be treated as different phrases, each with its own cached MP3. This feature allows for precise control but also means you should be mindful of text casing to maximize the benefits of caching.

**In summary**, the built-in caching system is designed to make the command more efficient, cost-effective, and user-friendly.

*The Location of the default cache folder is: "?exeloc\AIS\_Folder\"*

### Syntax

```
AIC.Text to MP3|P1[|P2]
```

```
AIC.tmp|P1[|P2]
```

### Parameter Explanation

- **P1:** The text you want to convert into an MP3 file.

- **P2**: Optional. The path where the generated MP3 file will be saved. The caching system will automatically be used if **P2** is omitted or empty.

If **P2** is omitted, the system will automatically generate a cache folder to store all MP3 files along with a checksum.

This feature helps in reducing costs and saving resources by reusing the same MP3 file for repeated phrases.

**Note:** The system is case-sensitive, meaning the same phrase with different cases will generate different checksums.

The use of different voices is not part of the caching, means the cache knows only the Text, not which voice it is.

To use the caching with different voices and cache them separately, you will need to Change the Cache Folder location using the `AIC.Set Folder` - Command.

### Example

```
!*****
! AIC.-Sample
!*****
AIC.Set Key|<YourAPIKeyHere>
AIC.Text to MP3|Hello World|?path/to/save.mp3
```

In this example, the text "Hello World" will be converted into an MP3 file and saved in the specified path.

### Remarks

- The API key must be set using the **AIC.Set Key** command before using this command.
- The system will use a cache to save resources **only if P2 is omitted or empty.**

### Limitations:

-

### See also:

-

## 3.42.11 AIL - AI-Local Systems (GPT4All)

## Using GPT4All on your local Computer



### Prompting Guidelines for GPT4All:

When generating prompts, the aim is to clearly communicate the desired task or question while providing enough context for the AI model to understand and generate an appropriate response. Here are some general guidelines that are often followed:

**Clearly state the task:** The prompt should explicitly mention what needs to be done, such as counting the words or sorting them alphabetically.

**Provide specific input:** The prompt should include the input data or relevant details necessary to complete the task. In this case, the string of three-letter words ending with dots is given.



**Specify the desired output:** The prompt should indicate the expected result or format of the output. For example, requesting the number of words and the final sorted list in alphabetical order.

**Use clear and concise language:** The language used in the prompt should be straightforward and unambiguous, avoiding unnecessary complexity or jargon.

**Avoid unnecessary instructions:** Prompts should focus on the core task and avoid excessive instructions or information that may confuse the model.

It's worth noting that while prompts can be designed to achieve specific outcomes, the language model's responses are ultimately generated based on its training data and underlying algorithms.

## 3.42.11.1 ! AI - Combining Free AI's

[Combining free AI's](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

## Combining GPT4All and Stable Diffusion (Local)

You can combine AI's to get amazing results.



This picture was generated using the Combination of GPT4All and Stable Diffusion (local)

The following Code will use GPT4All to optimize the Prompt before its delivered to Stable Diffusion.

```
VAR.$$THE="neural network beautiful golden girl Galaxy robot digital photorealistic"
VAR.$$PAO=Please make me a prompt below 231 characters, for stable diffusion use:
VAR.$$PAO+Do not write "Prompt:", just write the result.
' Use Image Register 0
VIN.$$IMR=0
```

```

' Set OpenAI API-Key from the saved File
AIC.SetKey|File
' Set Model-Temperature
AIC.Set_Temperature|0

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000)
' The more Tokens you use the more you need to pay. But the longer Input and Output
AIC.SetMax_Token|800

' We choose Wizzard v1.1
AIL.Set Model|7

FOR.$$STP|1|25
 AIL.Ask|$$PAO|$$RET
 DBP.Got: <<$$RET >>
 $$FIR=?path\Sample Pics\MTRresult_?.png
 FIL.gen|$$FIR|1|0|$$FIL
 SDL.gtf|$$RET|$$FIL
 ANA.Load|$$IMR|$$FIL
 $$TXT=SPR/SDO.generated: $$STP Steps
 $$COA=&HFFFFFFF
 $$COB==&H0
 $$BGC=-2
 $$XPA=30
 $$YPA=450
 ANA.PrintAt|$$IMR|$$TXT|$$COA|$$BGC|$$XPA|$$YPA|24
 ANA.PrintAt|$$IMR|$$TXT|$$COB|$$BGC|($$XPA+1)|($$YPA+2)|24
 DBP.Loaded in IR:$$IMR -> $$FIL
 ANA.save|$$IMR|$$FIL
 ANA.Show|$$IMR
NEX.
ENR.

```

**Remarks**

-

**Limitations:**

-

**See also:**

-

## 3.42.11.2 ! AI - Prompting Sample #1

Assume you have this Text: "ISP.NSP.WSP.GSP.SSP."  
and you want it to be sorted.

In GPT-4 this is an easy task.  
But you have 1000 such Operations to do and want to use GPT4All because its free.

**This is not going to work:**

So you choose "Hermes" and you try:

ME:  
Please sort me this text: "ISP.NSP.WSP.GSP.SSP."

Hermes:  
This is a text that contains five sets of three letters separated by periods, wh

ME:  
Can you sort these words please alphabetically?

HERMES:  
Certainly! Here are the words sorted alphabetically: apple, banana, cherry, date

Ok that did not work.

**Below is the way it will work**

And we will use a Script that will show you the difference in the answers from HERMES  
depending on the "Temperature-Setting".  
The Script also shows you how to use GPT4All in a SPR-Script doing work for you.

```
VAN.$$TIM=#dttime#
AIL.SetModel|$$MOD
AIL.Set MaxToken|1024
AIC.Set Number|1
FOR.$$LR0|0|1|0.1
 AIL.Set Temperature|$$LR0
 AIL.Set Model|Hermes

 $$TXT>Hello! I have a list of items represented as a string, and I would like
 $$TXT=$$TXT "ISP.NSP.WSP.GSP.SSP."
 $$TXT=$$TXT Please follow these steps to sort the items:
 $$TXT=$$TXT Split the string into individual items based on the period ('.') o
 $$TXT=$$TXT Sort the resulting array of items in alphabetical order.
 $$TXT=$$TXT Join the sorted items back into a single string, using the period
 $$TXT=$$TXT Please provide me with the sorted string as the output. DO not ger

 AIC.Estimate Token Count|$$TXT|$$TOA
 AIL.Ask GPT4All|$$TXT|$$REA
 CAL.$$TIU=#dsince#|i
 AIC.Estimate Token Count|$$REA|$$TOB
```

```

DBP.-----
VAR.$$OUT=$$REA $CrLf$Temp. $$LR0 $CrLf$ Time used: $$TIU sec.$CrLf$Tokens in
DBP.$$OUT
DBP.-----
AIC.g|q|$$REB
DBP.The last Prompt was:
DBP.$$REB
NEX.
ENR.

```

**Output:**

[15:32:47] -----

[15:32:47]

```

Sure, I can help you with that! Here's the steps to sort the items alphabetical.
1. Split the string into individual items based on the period ('.') delimiter us
```python
items = 'ISP.NSP.WSP.GSP.SSP.'[:-1].split('.')
```

2. Sort the resulting array of items in alphabetical order using the `sorted()`
```python
sorted_items = sorted(items)
```

3. Join the sorted items back into a single string, using the period ('.') as th
```python
joined_items = ''.join([item + '.'] * len(sorted_items))
```

```

**Therefore, the sorted string is: `GSP.ISP.NSP.SSP.WSP.`**

```

Temp. 0
Time used: 233 sec.
Tokens in Prompt: 139
Tokens in Answer: 217

```

[15:32:47] -----

```

[15:32:47] The last Prompt was:
[15:32:47] Hello! I have a list of items represented as a string, and I would l:
[15:36:49] -----

```

[15:36:49]

```

Sure, I can help you with that! Here's the steps to sort the items alphabetical.
1. Split the string into individual items based on the period ('.') delimiter us
```python
items = 'ISP.NSP.WSP.GSP.SSP.'[:-1].split('.')
```

2. Sort the resulting array of items in alphabetical order using the `sorted()`
```python
sorted_items = sorted(items)
```

3. Join the sorted items back into a single string, using the period ('.') as th
```python
joined_items = ''.join([item + '.'] * len(sorted_items))
```

```

**Therefore, the sorted string is: `GSP.ISP.NSP.SSP.WSP.`**

```

Temp. .1
Time used: 241 sec.
Tokens in Prompt: 139
Tokens in Answer: 217

```

[15:36:49] -----

```

[15:36:49] The last Prompt was:

```

```

[15:36:49] Hello! I have a list of items represented as a string, and I would like
[15:40:59] -----
[15:40:59]
Sure, I can help you with that! Here's the steps to sort the items alphabetically.
1. Split the string into individual items based on the period ('.') delimiter using Python
```python
items = 'ISP.NSP.WSP.GSP.SSP.'[:-1].split('.')
```

2. Sort the resulting array of items in alphabetical order using the `sorted()` function
```python
sorted_items = sorted(items)
```

3. Join the sorted items back into a single string, using the period ('.') as the delimiter
```python
joined_items = ''.join([item + '.'] * len(sorted_items))
```

Therefore, the sorted string is: `GSP.ISP.NSP.SSP.WSP.`
Temp. .2
Time used: 250 sec.
Tokens in Prompt: 139
Tokens in Answer: 217
[15:40:59] -----
[15:40:59] The last Prompt was:
[15:40:59] Hello! I have a list of items represented as a string, and I would like
[15:45:23] -----
[15:45:23]
Sure, I can help you with that! Here's the steps to sort the items alphabetically.
1. Split the string into individual items based on the period ('.') delimiter using Python
```python
items = 'ISP.NSP.WSP.GSP.SSP.'[:-1].split('.')
```

2. Sort the resulting array of items in alphabetical order using the `sorted()` function
```python
sorted_items = sorted(items)
```

3. Join the sorted items back into a single string, using the period ('.') as the delimiter
```python
joined_items = ''.join([item + '.'] * len(sorted_items))
```

Therefore, the sorted string is: `GSP.ISP.NSP.SSP.WSP.`
Temp. .3
Time used: 264 sec.
Tokens in Prompt: 139
Tokens in Answer: 217
[15:45:23] -----
[15:45:23] The last Prompt was:
[15:45:23] Hello! I have a list of items represented as a string, and I would like
[15:49:51] -----
[15:49:51]
Sure, I can help you with that! Here's the steps to sort the items alphabetically.
1. Split the string into individual items based on the period ('.') delimiter using Python
```python
items = 'ISP.NSP.WSP.GSP.SSP.'[:-1].split('.')
```

2. Sort the resulting array of items in alphabetical order using the `sorted()` function
```python
sorted_items = sorted(items)

```

```

'''
3. Join the sorted items back into a single string, using the period ('.') as th
'''python
joined_items = ''.join([item + '.'] * len(sorted_items))
'''

Therefore, the sorted string is: `GSP.ISP.NSP.SSP.WSP.`
Temp. .4
Time used: 268 sec.
Tokens in Prompt: 139
Tokens in Answer: 217
[15:49:51] -----
[15:49:51] The last Prompt was:
[15:49:51] Hello! I have a list of items represented as a string, and I would l:
[15:54:40] -----
[15:54:40]
Sure, I can help you with that! Here's the steps to sort the items alphabetical.
1. Split the string into individual items based on the period ('.') delimiter us
'''python
items = 'ISP.NSP.WSP.GSP.SSP.'[:-1].split('.')
'''

2. Sort the resulting array of items in alphabetical order using the `sorted()``
'''python
sorted_items = sorted(items)
'''

3. Join the sorted items back into a single string, using the period ('.') as th
'''python
joined_items = ''.join([item + '.'] * len(sorted_items))
'''

Therefore, the sorted string is: `GSP.ISP.NSP.SSP.WSP.`
Temp. .5
Time used: 289 sec.
Tokens in Prompt: 139
Tokens in Answer: 217
[15:54:40] -----
[15:54:40] The last Prompt was:
[15:54:40] Hello! I have a list of items represented as a string, and I would l:
[15:58:32] -----
[15:58:32]
Sure, I can help you with that! Here's the steps to sort the items alphabetical.
1) Split the string into individual items based on the period ('.') delimiter us
2) Sort the resulting array of items in alphabetical order using the `sort()`` me
3) Join the sorted items back into a single string, using the period ('.') as th
The resulting sorted string would be: "GSP.SSP.ISP.NSP.WSP."
Temp. .6
Time used: 232 sec.
Tokens in Prompt: 139
Tokens in Answer: 166
[15:58:32] -----
[15:58:32] The last Prompt was:
[15:58:32] Hello! I have a list of items represented as a string, and I would l:
[15:59:25] -----
[15:59:25]
I can help you with that! Here's the sorted string: "GSP.SSP.ISP.NSP.WSP."
Temp. .7
Time used: 53 sec.
Tokens in Prompt: 139
Tokens in Answer: 21

```

```

[15:59:25] -----
[15:59:25] The last Prompt was:
[15:59:25] Hello! I have a list of items represented as a string, and I would like
[16:04:34] -----
[16:04:34]
Sure, I can help you with that. Here's the step-by-step process to sort the list:
1. Split the input string into an array of individual items based on the period.
```python
items = "ISP.NSP.WSP.GSP.SSP.".split(".")
```
```output
['ISP', 'NSP', 'WSP', 'GSP', 'SSP']
```
2. Sort the resulting array of items in alphabetical order.
```python
sorted_items = sorted(items)
```
```output ['GSP', 'ISP', 'NSP', 'SGP', 'SSP']```
3. Join the sorted items back into a single string, using the period as the delimiter.
```python
sorted_string = ".".join(sorted_items)
```
```output ".ISPGSP.S"```
4. Return the sorted string as output.
```python
print(sorted_string)
```
```output ".ISPGSP.S"```
So, the final sorted string is ".ISPGSP.S".
Temp. .8
Time used: 308 sec.
Tokens in Prompt: 139
Tokens in Answer: 200
[16:04:34] -----
[16:04:34] The last Prompt was:
[16:04:34] Hello! I have a list of items represented as a string, and I would like
[16:07:45] -----
[16:07:45]
Understood. Splitting the string based on the period (',' and 'ISP.NSP.WSP.GSP.SSP.')
Therefore, the sorted string is: "GSP.ISP.NPS.SSP.WSP."
Temp. .9
Time used: 191 sec.
Tokens in Prompt: 139
Tokens in Answer: 115
[16:07:45] -----
[16:07:45] The last Prompt was:
[16:07:45] Hello! I have a list of items represented as a string, and I would like
[16:14:02] -----
[16:14:02]
Sure, I can help you with that task. Here is the steps to sort and join the items:
1. Split the input string "ISP.NSP.WSP.GSP.SSP." using the period (',') delimiter.
2. Iterate through the resulting array and sort the items in alphabetical order.
3. Join the sorted items back into a single string, using the period (',') as the delimiter.
Here is the Python code to perform these steps:
```python
input_string = "ISP.NSP.WSP.GSP.SSP."
items = input_string.split(',')
```

```



```
sorted_items = sorted(items)
output_string = '.' + ' '.join(sorted_items)
print(output_string)
```
```

Output: `ISP.NSP.GSP.SAP.WSP.`

I hope this helps! Let me know if you have any further questions or concerns.

Temp. 1

Time used: 376 sec.

Tokens in Prompt: 139

Tokens in Answer: 182

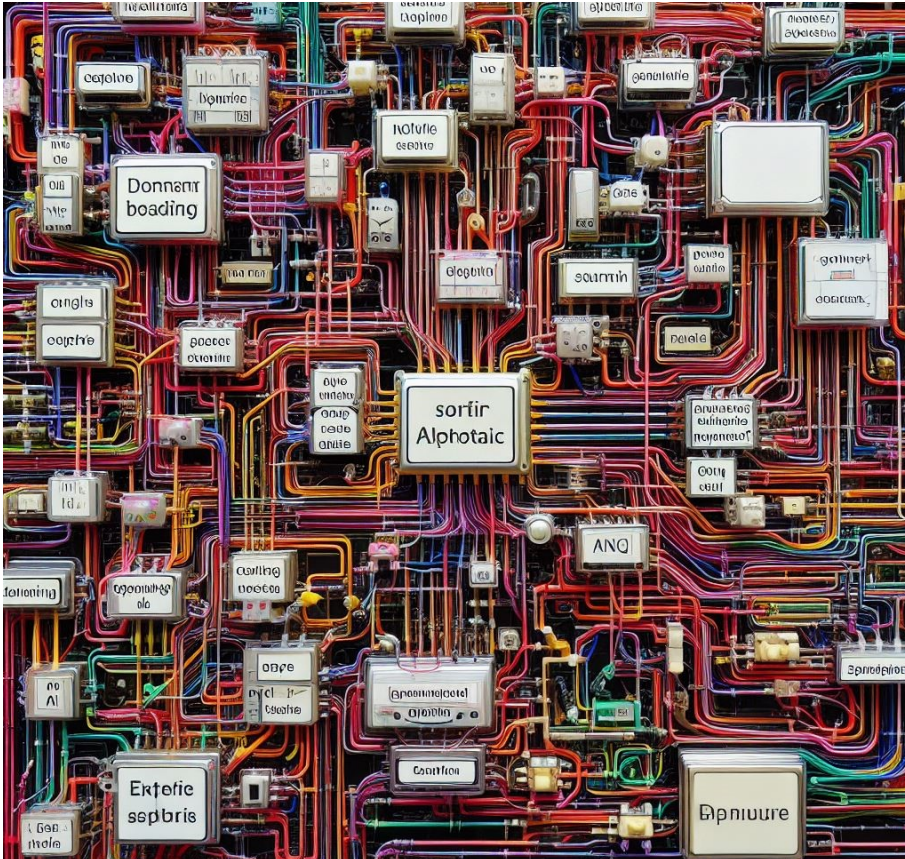
[16:14:02] -----

[16:14:02] The last Prompt was:

[16:14:02] Hello! I have a list of items represented as a string, and I would l:

3.42.11.3 ! GPT4All - Installation

GPT4All Technology for Smart Package Robot (SPR)



Installing GPT4All is an easy task.

Installing GPT4All on a Windows Computer

In order to harness the capabilities of GPT4All alongside your SPR, it is essential to install it either on your local computer or within your network environment.

In this chapter, we will walk through the steps to install GPT4All on a Windows computer. GPT4All is an open-source software ecosystem that allows you to train and deploy powerful large language models on everyday hardware such as laptops, desktops, and servers. It is optimized to run inference of 7-13 billion parameter large language models on CPUs.

For more Information and also more details on how to install GPT4All on your local Computer

Step 1: Install GPT4All on your local Computer or in your Network

You will find an [Windows-Installer for GPT4All here](#).

You can directly download the Installer

Step 2: Download Pre-Quantized Models

GPT4All uses neural network quantization to make large language models more efficient to run on consumer hardware. You need to download pre-quantized models to use with GPT4All. You can find a list of pre-quantized models on the GPT4All website or in the download pane of the chat client in left side Menu under "Downloads".

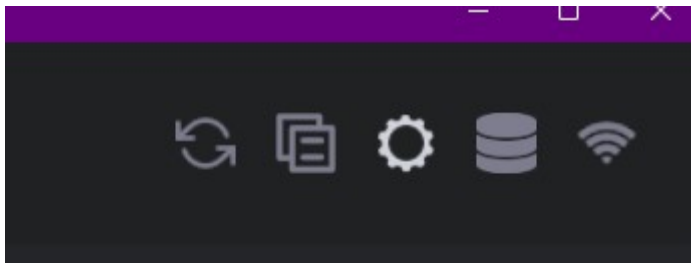
Step 3: How to start GPT4All and how to set it up

Now that you have everything set up, you can start GPT4All. Please follow the following steps:



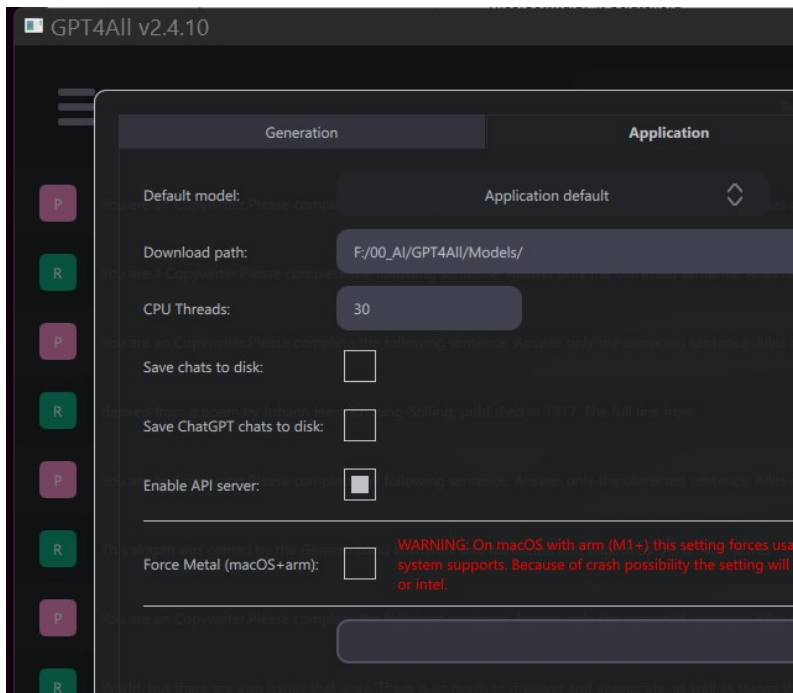
GPT4All will setup an icon on your Desktop, start GPT4All it with this icon.

Step 4: Open Settings



Click on Settings, then you see the Settings below.

Step 5: Enable the API Server





Enable the API-Server in GPT4All, then the SPR can connect via http: to GPT4All. You need to do this only once. Once the Server is enabled, it will always be enabled when GPT4All is starting.



Step 6: Download Models and (if you want - optional) enter your OpenAI API-Keys Therefore click in the Menu on the left side on "Downloads".

Editors Note 30.06.2023 As of today i can enter my OpenAI Key in GPT4All, but it is not displayed, as expected from the other Models under "Downloads". Looking into the "Models Folder" you can see that the Key is saved anyway.

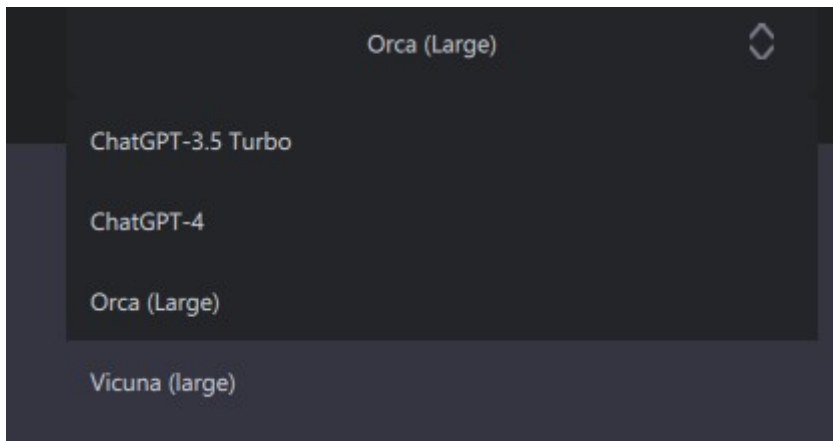
The System generated the following 2 files in your Models-Folder. Each of the file just contains the OpenAI-API Key in clear Text. However, I recommend that you create these files yourself, just make a ".txt"-File with this name and paste your API-Key inside. Then restart the GPT4All GUI and it should work.

 chatgpt-gpt-3.5-turbo.txt	30.06.2023 18:40	Textdokument	1 KB
 chatgpt-gpt-4.txt	30.06.2023 18:46	Textdokument	1 KB

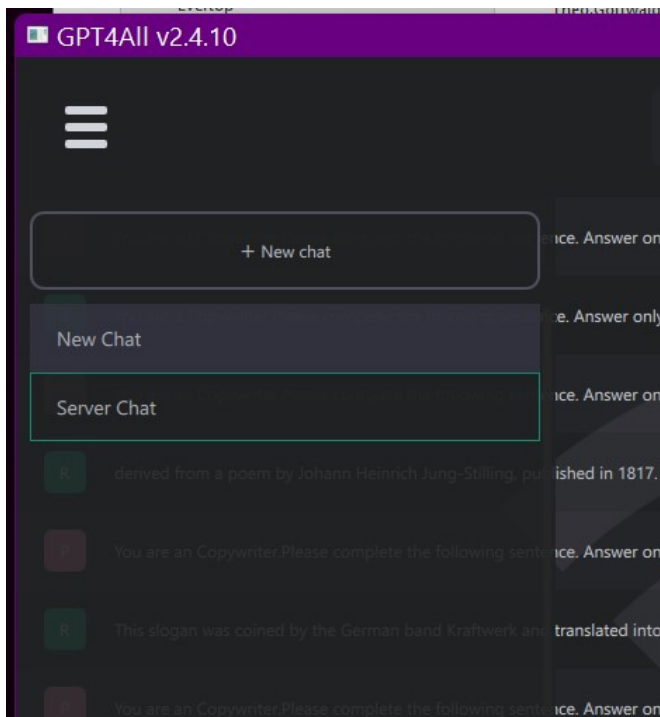
Now if you change the filename to:

 chatgpt-gpt-3.5-turbo.txt	30.06.2023 18:40	Textdokument	1 KB
 chatgpt-gpt-4.txt	30.06.2023 18:46	Textdokument	1 KB

Then you will then also see these Models in the Menu:



Once the Server is running, you can see it working if you press the "Server Chat" Button.



If you open the Menu on the left side, you can click on "Server Chat" and see all dialogs that are currently running over the Server.

Considerations

Keep in mind that the inference speed of a local large language model depends on the model size and the number of tokens given as input. It is not advised to prompt local models with large chunks of context, as their inference speed will heavily degrade. If you wish to utilize context windows larger than 750 tokens, you might want to run GPT4All models on a GPU ([Details here](#)).

3.42.11.4 ! GPT4All - SPR-specific hints

SPR-specific Hints

1. Using GPT4All in teh Network

If you want to install the ChatGPT4All-Server in your Network, you can change the URL that is used by the SPR with this command:

```
' Use your update IP-Adress in the URL
$$URL=http://127.0.0.1:4891/v1/completions
AIL.Change GPT4All Url|$$URL
```

2. Changing the used Model

If you want to change the used Model, you can use the

```
AIL.Set Model|Replit
AIL.Set Model|Wizard Uncensored
AIL.Set Model|Hermes
```

Command. Above are three examples. Use the name that is displayed in Green in the Download-Section behind the AIC.Set Model-Commands.

This is the same name that you can see in the GPT4All GUI - Menu.

Exceptions and how to use them:

For some Models the automatic switching seemed not to work as today (tested 01.07.2023).

This may work in later Versions fo GPT4All. Namely these are:

"Vicuna (large)" - ggml-vicuna-13b-1.1-q4_2.bin




"Orca (Large)" - orca-mini-13b.ggmlv3.q4_0.bin

These Models will not be used if there are other Models available.

These Models will be used if no other Models are available.

In case you want to use exactly these Models, you can:

1. End the GPT4All GUI.
2. Make an Folder inside the Models Folder
3. Move all other Models inside that Folder so that only the wanted Model is left
4. If no other Model is there, only this Model is used, no matter which Model you specify.

 orca-mini-13b.ggmlv3.q4_0.bin	30.06.2023 18:21	BIN-Datei
 localdocs_v0.db	30.06.2023 22:43	Data Base File
 Temp Removed	01.07.2023 06:26	Dateiordner

3. Check the RAW Output

If you take a look into the Raw-Output using "AIC.Get Raw Output" you can see which Model was really used together with some other Information.

```
{ "choices": [{"finish_reason": "stop", "index": 0, "logprobs": null, "references": [
], "text": "You are an Copywriter.Please complete the following sentence.
Answer only the corrected sentence: Alles neu macht der\nMensch ist,
wie ein Affe im
Zoo."}], "created": 1688115273, "id": "foobarbaz", "model": "Wizard
Uncensored", "object": "text_completion", "usage": {"completion_tokens": 1
2, "prompt_tokens": 38, "total_tokens": 50}}
```

```
{ "choices": [{"finish_reason": "stop", "index": 0, "logprobs": null, "references": [
], "text": "You are an Copywriter.Please complete the following sentence.
Answer only the corrected sentence: Alles neu macht der\nAlles neu
macht der"}], "created": 1688115512, "id": "foobarbaz", "model": "Nous
Vicuna", "object": "text_completion", "usage": {"completion_tokens": 6, "pro
mpt_tokens": 38, "total_tokens": 44}}
```

4. Train the AI with your own Data

This is a special field that has no special support using SPR-Commands at this time. Details see: <https://atlas.nomic.ai/>

5. Use the "Local Docs" Feature

LocalDocs is a GPT4All plugin that allows you to **chat with your local files and data**. It allows you to utilize powerful local LLMs to chat with private data without any data leaving your computer or server.

When using LocalDocs, your LLM will cite the sources that most likely contributed to a given output.

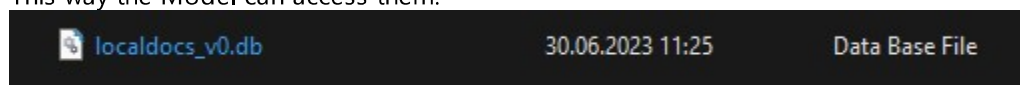
GPT4All allows you to search local docs like PDF's etc.

For this you have to add a Folder where these documents are, together with a "Collection Name" to the System.

It will automatically be referenced.

In your Models-Folder you will find a Database that references the Files in your "Local Docs".

This way the Model can access them.



Enabling LocalDocs

1. Download and install the most recent version of GPT4All Chat from the GPT4All Website.
2. Navigate to Settings and select the LocalDocs tab.
3. Set up a folder on your computer to serve as a collection containing files that your LLM should access. Feel free to modify the contents of this folder as needed. Your LLM will dynamically adapt to access the newly added files.
4. Initiate a chat session with any LLM (this includes external models like ChatGPT, but be cautious as data may be transmitted outside your device).
5. Click on the database icon located in the top-right corner and choose the collection you want your LLM to refer to during the chat session.

LocalDocs Capabilities

LocalDocs grants your LLM the ability to be informed about the contents of your documentation collection. However, not all prompts or questions will make use of your document collection for context. If LocalDocs is utilized in your LLM's response, you will notice references to the document snippets that were accessed by LocalDocs.

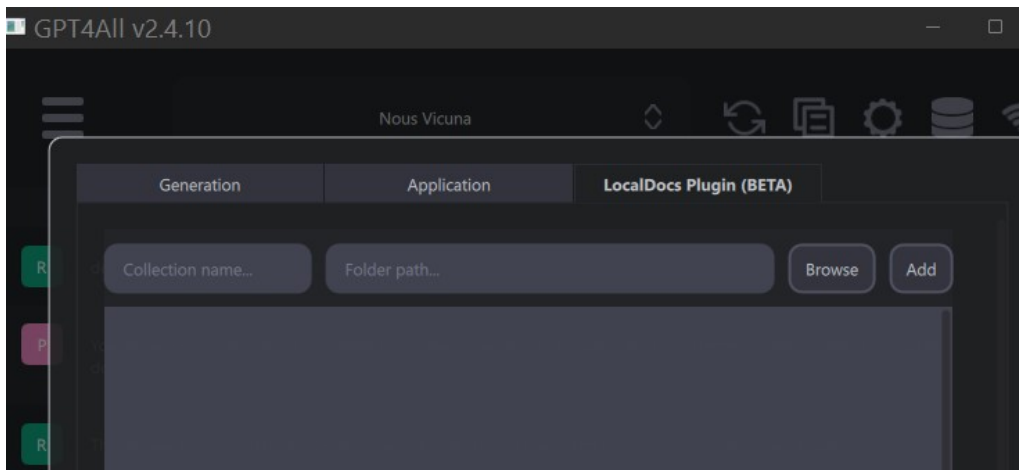
LocalDocs can:

- Search through your documents based on your prompt or question. If your documents contain information that could be relevant in answering your query, LocalDocs will attempt to make use of snippets from your documents to provide context.

LocalDocs cannot:

- Respond to broad metadata queries such as "What documents are you aware of?" or "Provide information about my documents."
- Summarize an entire document (e.g., "Summarize my Magna Carta PDF").

For solutions to common problems, please consult the Troubleshooting section.



6. Sample Script:

GPT4All will automatically load the required Model and switch to the Model that was asked for.

You can see that the Model is been loaded, if GPT4All is busy but the CPU-Usage in the Task Manager is still low.

Once the Model starts working you will see the CPU Usage to go up (depends on the Core-Settings in Settings).

If the wanted Model is already loaded, the CPU will go up immediately.

Here is a Sample Script that will **change the Model by name**.

```

$LOG=?exeloc\Output.txt
DEL.$LOG
$WOA="ISP.NSP.WSP.GSP.SSP."
$WOB="GTO.JMP.JNJ.JNF.GSB.JSR.JIV.JIZ.JNZ.PRR.JLE.JME.JRR.JIT.JNT.JIS.JIE.VBS."

```



```

FOR.$$NUM|1|8
  AIC.Set MaxToken|1024
  GSB.Lab_SetModel
  AIC.SetModel|$$MOD
  AIC.Set Number|2
  AIC.Set Temperatur|1

  GSB.Write_Log|Model: $$MOD
  $$TXT=Act as my Assistant.
  $$TXT=$$TXT Below is a Line of words, each word separated with a Dot.
  $$TXT=$$TXT Count and Sort the words alphabetically always comparing the first
  $$TXT=$$TXT Tell me how many words are in that Line.
  $$TXT=$$TXT Do not generate code. Do not repeat yourself. Use english language
  $$TXT=$$TXT Show me the words in sorted order, in one Line separated by dots.
  $$TXT=$$TXT $$WOA
  AIC.Ask GPT4All|$$TXT|$$RET
  GSB.Write_Log|$$RET
  AIC.Get Several|5|$$RAW
  GSB.Write_Log|Used Model: $$RAW
  DBP.-----
NEX.

ENR.
'-----
:Lab_SetModel
SCS.$$NUM
CAN.1
  $$MOD=Wizard Uncensored
CAN.2
  $$MOD=Hermes
CAN.3
  $$MOD=Snoozy
CAN.4
  $$MOD=Replit
CAN.5
  $$MOD=Nous Vicuna
CAN.6
  $$MOD=Groovy
CAN.7
  $$MOD=ChatGPT-3.5 Turbo
CAN.8
  $$MOD=ChatGPT-4
CAE.
  $$MOD=$$MOD
ESC.
RET.
'-----
:Write_Log
VAV.$$OUT=$$ _01$crLf$
ATF.$$LOG|$$OUT
DBP.$$OUT
RET.
'-----

ENR.
'=====

```

Another Sample Script that will change the Model by number.

Note that the API-Key for the OPENAI API is saved in the Script folder using the [Save Key-Command](#)⁷⁶².

```

$$LOG=?exeloc\Output.txt
DEL.$$LOG
$$WOA="STW.SCW.SAO.NAV.WTW.WCW.CAW.WFM.MAW.WPR.WPT.SIR.WFV.GCT.AVF.TVI.TVF.UNI.S

GSC.$$WOA|.|$ANZ
GSB.Write_Log|SPR counted: $$ANZ Elements.$CrLf$-----$

FOR.$$NUM|1|9
  AIL.Set MaxToken|2048
  AIL.SetModel|$NUM
  AIL.Set Number|1
  AIL.Set Temperatur|1
  AIL.srp|1
  AIL.snb|1

GSB.Write_Log|Model: $NUM

$$TXT=Please analyze the given string of three-letter words, each ending with
$$TXT+$$WOA
$$TXT+$CrLf$Count the number of words in the string and provide the result.$C
DBP.$$TXT

AIL.Ask GPT4All|$TXT|$$RET
GSB.Write_Log|$$RET
AIC.Get Several|5|$$RAW
GSB.Write_Log|Used Model: $$RAW
DBP.-----
NEX.

ENR.
'-----
:Write_Log
VAV.$$OUT=$$ _01$CrLf$
ATF.$$LOG|$OUT
DBP.$$OUT
RET.
'-----
ENR.
'=====

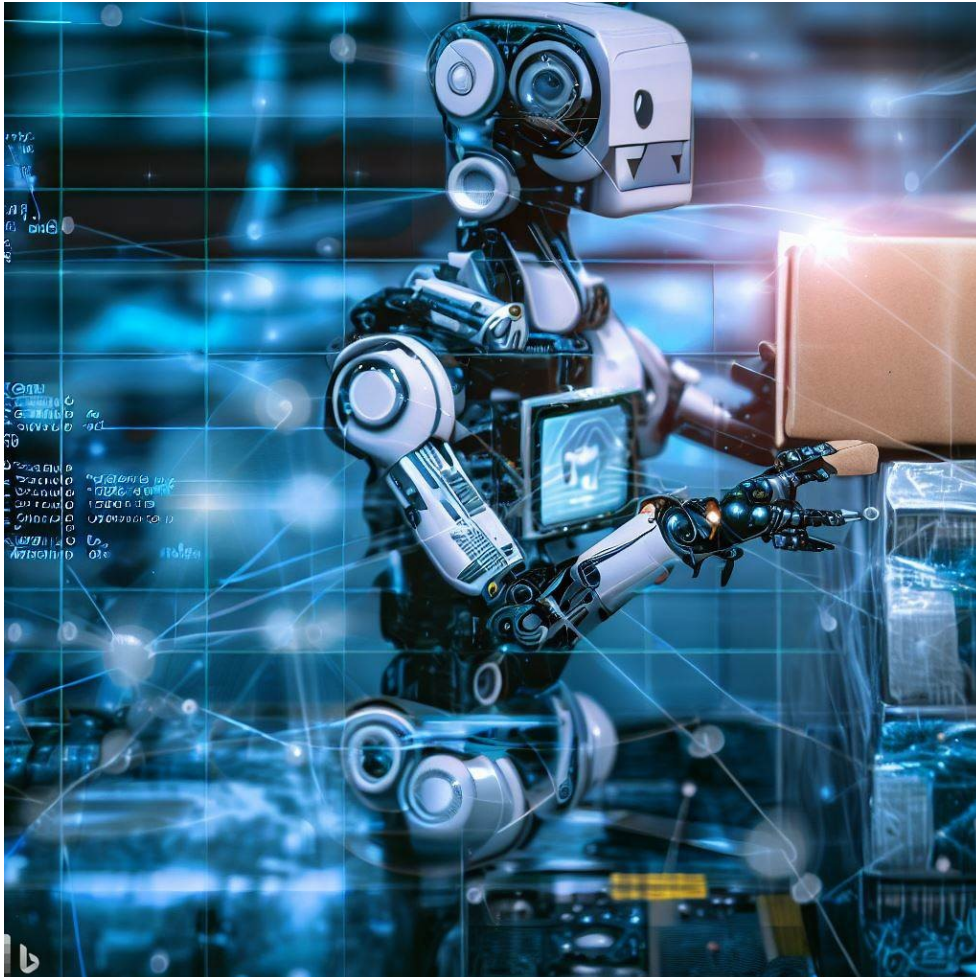
```

Conclusion

While GPT4All models may not match the prowess of OpenAI's offerings, they come with the distinct advantage of being local, thus incurring no additional costs. They are still quite capable and suitable for handling smaller tasks. Like with any AI system, it's advisable to use computers that have a high number of CPU cores and substantial VRAM on the graphics card to optimize performance.

3.42.11.5 ! GPT4All - Use local AI

GPT4All Technology for Smart Package Robot (SPR)



GPT4All is an open Source AI-System that can be installed on your local Computer. Or possibly in your network. You can use it with the SPR.

Introduction

GPT4All is an open-source software ecosystem developed by Nomic AI that allows users to use large language models locally on everyday hardware such as laptops, desktops, and servers.

The software is specifically optimized to perform inference with large language models having **7-13 billion parameters**.

It achieves this by utilizing neural network quantization, which reduces the memory requirements of these models so that they can be run efficiently on consumer-grade hardware with limited resources ([see GPT4All docs](#)). To be able to use it with the SPR, the Software must be installed on your Computer or in your Network.

Use the Local Docs Feature"

GPT4All contains a "Local Docs" Feature that enables you to access local docs like PDFs or other documents via its AI.

This can be seen like a intelligent Search Engine. More Details see below.

Models and Quantization

GPT4All supports [lots of freely available LLM-Models](#), you can click on "Download" to install these on your local Computer.

GPT4All also supports **use of the most advanced OpenAI Models** through its Interface, yet this may not be needed as you can use them with the SPR directly.

The screenshot displays the GPT4All interface with a list of AI models. Each model entry includes a title, a brief description, a list of features, and a button to either 'Remove', 'Download', or 'Install' the model. The models shown are:

- Hermes**: Best overall model. Features include being instruction-based, giving long responses, being curated with 300,000 uncensored instructions, trained by Nous Research, and not being used commercially. Status: Installed. Download size: 7.28 GB, RAM required: 16 GB, Parameters: 13 billion, Quantization: q4_0, Type: LLaMA.
- GPT4All Falcon**: Best overall smaller model. Features include fast responses, being instruction-based, trained by TII, finetuned by Nomic AI, and licensed for commercial use. Status: Available. Download size: 3.78 GB, RAM required: 8 GB, Parameters: 7 billion, Quantization: q4_0, Type: Falcon.
- Groovy**: Creative model can be used for commercial purposes. Features include fast responses, creative responses, being instruction-based, trained by Nomic AI, and licensed for commercial use. Status: Available. Download size: 3.53 GB, RAM required: 8 GB, Parameters: 7 billion, Quantization: q4_0, Type: GPT-J.
- ChatGPT-3.5 Turbo**: OpenAI's ChatGPT model GPT-3.5 Turbo. Features include requiring a personal OpenAI API key, a warning to send chats to OpenAI, the API key being stored on disk, and only being used to communicate with OpenAI. Status: Available. Download size: minimal, RAM required: minimal, Parameters: 7, Quantization: NA, Type: GPT.
- ChatGPT-4**: OpenAI's ChatGPT model GPT-4. Features include requiring a personal OpenAI API key, a warning to send chats to OpenAI, the API key being stored on disk, and only being used to communicate with OpenAI. Status: Available.

If you click "Downloads" in the left side Menu, you will get to the LLM-Page, where you can download and install the AI-Models (LLMs) of your choice.

Inference Speed and Performance

The inference speed of a local large language model (LLM) depends on the model size and the number of tokens provided as input. It's not advisable to use large chunks of context with local LLMs as their inference speed will significantly degrade. For context windows larger than 750 tokens, it's recommended to run GPT4All models on a GPU. Native GPU support for GPT4All models is planned. The performance of an LLM depends on various factors including the quantity and diversity of the pre-training data and the fine-tuning data. GPT4All aims to bring the most powerful local assistant models to desktops and is actively being improved by Nomic AI.



During the processing the AI will use 99% of the available CPU Resources, no matter how many Cores your system has.

Do i need a lot of VRAM?

GPT4All makes use of a process called **neural network quantization** to make it feasible to run large language models locally. Normally, a multi-billion parameter transformer would require more than 30GB of VRAM, which is not commonly available. Through quantization, GPT4All models require only 4-8GB of RAM. Currently, the ecosystem is compatible with three variants of the Transformer neural network architecture: LLaMa, GPT-J, and MPT. Any model trained with these architectures can be quantized and run locally with all GPT4All bindings and in the chat client.

Integration with Smart Package Robot (SPR)

Smart Package Robot (SPR) can leverage the GPT4All technology to enhance its capabilities. And therefore enable you to use the System in your Scripts. By integrating GPT4All, SPR can run powerful language models locally which is beneficial for data privacy and sometimes faster processing times. This can be particularly useful for natural language processing tasks, generating responses or information based on user queries, and more without the need for an internet connection.

Important:

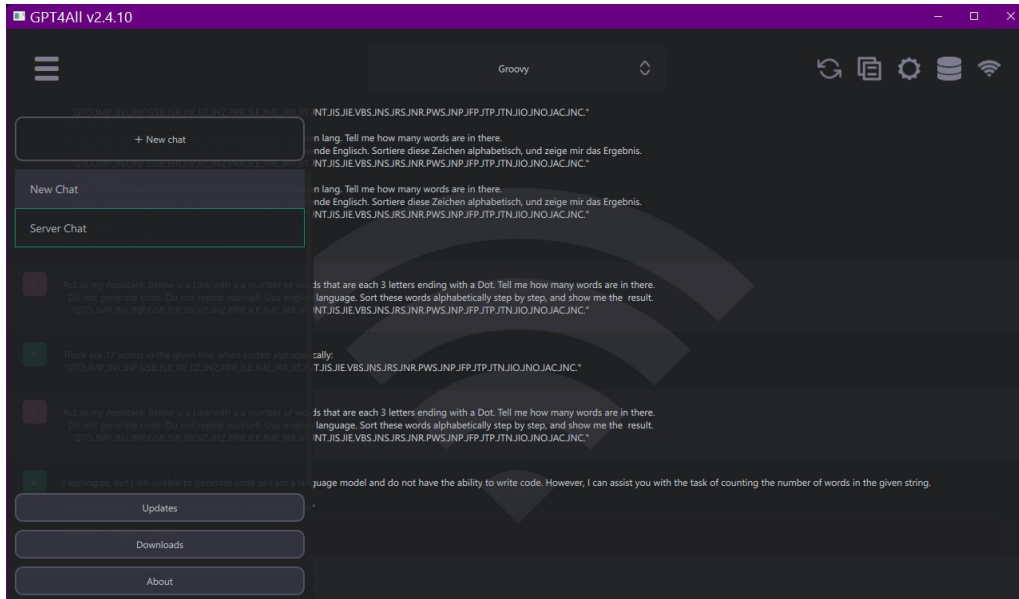
Before you can use this Command, [GPT4All must be installed on your System.](#)^[945]

Sidenote:

While GPT4All models may not match the prowess of OpenAI's offerings, they come with the distinct advantage of being local, thus incurring no additional costs. They are still quite capable and suitable for handling smaller tasks. Like with any AI system, it's advisable to use computers that have a high number of CPU cores and substantial VRAM on the graphics card to optimize performance.

3.42.11.6 ! GPT4All-Sample Script and Results

We will be utilizing a sample script to analyze the performance of different language models in handling data sorting and counting operations. It's important to recognize that large language models (LLMs) like GPT-3 are inherently built on a statistical foundation that predicts the likelihood of the next word based on the preceding text. Consequently, tasks such as sorting extensive datasets or counting a large number of elements are not the strong suit of these models.



This is the GUI of GPT4All in "Server Chat" View, where you can see what the Server is doing.

In this analysis, we'll also include the response from OpenAI's GPT-4 for reference. As I do not have direct access to GPT-4 through my API key (its not yet publicly available for anybody), the response from GPT-4 will have to be obtained via chat and then manually inserted below.

For Reference we have the result from GPT-4 here:

:
XOR. - XOR - Encryption
Amazing enough, GPT-4 got the result perfectly right and therefore is the reference in this test.

Here is the structured format of the test with the other models:

Here is the Sample Script that will call all the Models in GPT4All:

```

$$LOG=?exeloc\Output.txt
DEL.$$LOG
$$WOA="GTO.JMP.JNJ.JNF.GSB.JSR.JIV.JIZ.JNZ.PRR.JLE.JME.JRR.JIT.JNT.JIS.JIE.VBS.
GSC.$$WOA|.|$$ANZ
GSB.Write_Log|SPR counted: $$ANZ Elements.$CrLf$-----$C
FOR.$$NUM|1|7
    
```

```

AIC.Set MaxToken|1024
GSB.Lab_SetModel
AIC.SetModel|$$MOD
AIC.Set Number|2
AIC.Set Temperatur|1

GSB.Write_Log|Model: $$MOD
$$TXT=Act as my Assistant.
$$TXT+ Below is a Line with a a number of words that are each 3 letters ending
$$TXT+ Tell me how many words are in there.$crlf$
$$TXT+ Do not generate code. Do not repeat yourself. Use english language.
$$TXT+ Sort these words alphabetically step by step, and show me the result.$
$$TXT+ $$WOA
IVV.$$NUM>6
    GSB.Call_GPT
ELS.
    AIC.Ask GPT4All|$$TXT|$$RET
EIF.
GSB.Write_Log|$$RET
AIC.Get Several|5|$$RAW
GSB.Write_Log|Used Model: $$RAW
DBP.-----
NEX.

ENR.
'-----
:Lab_SetModel
SCS.$$NUM
CAN.1
    $$MOD=Wizard Uncensored
CAN.2
    $$MOD=Hermes
CAN.3
    $$MOD=Snoozy
CAN.4
    $$MOD=Replit
CAN.5
    $$MOD=Nous Vicuna
CAN.6
    $$MOD=Groovy
'CAN.7
'    $$MOD=ChatGPT-3.5 Turbo
'CAN.8
'    $$MOD=ChatGPT-4
CAE.
    $$MOD=$$MOD
ESC.
RET.
'-----
:Write_Log
VAV.$$OUT=$$ _01$crlf$
ATF.$$LOG|$$OUT
DBP.$$OUT
RET.
'-----
' We do this separate as the other way sometime4s the GUI seems to crash.
'-----

```



```

:Call_GPT
$$MOD=gpt-3.5-turbo-0613
AIC.SetKey|File
AIC.SetModel_Chat|1
AIC.Ask_Chat|$$TXT|$$RET
RET.
'-----
ENR.
'=====

```

Test Summary (done by GPT-4)

- **Smart Package Robot Elements Counted:** 30
- **Models Tested:** Wizard Uncensored, Hermes, Snoozy, Replit, Nous Vicuna, Groovy and GPT 3.5-Turbo
- **Expected Result:** 30 words, sorted:
"GSB.GTO.JAC.JFP.JIE.JIO.JIS.JIT.JIV.JIZ.JLE.JME.JMP.JNC.JNF.JNJ.JNO.JNP.JNR.JNS.JNT.JNZ.JRR.JRS.JSR.JTN.JTP.PRR.PWS.VBS."

Test Details

1. Model: Wizard Uncensored

- **Answer:** Counted 10 words that end with a dot.
- **Evaluation:** The model's output was incorrect. It undercounted the number of words and did not provide them in sorted order.

2. Model: Hermes

- **Answer:** Counted 15 words, not in sorted order.
- **Evaluation:** The model's output was incorrect. It undercounted the number of words and did not provide them in sorted order.

3. Model: Snoozy

- **Answer:** Counted 26 words ending with a dot, sorted alphabetically.
- **Evaluation:** The model's output was partially correct. It undercounted the number of words but sorted them alphabetically.

4. Model: Replit

- **Answer** (Translated from German): Counted 30 words ending with a dot, sorted alphabetically: "GSB, GTO, JAC, JFP, JIE, JIT, JIV, JIZ, JLE, JME, JMP, JNC, JNF, JNJ, JNO, JNP, JNR, JNS, JNT, JNZ, JRR, JRS, JSR, JTP, JTN, NZR, PRR, PWS, RRR, VBS."
- **Evaluation:** The model's output was almost correct. It counted the correct number of words and sorted them alphabetically, but included "NZR" and "RRR" instead of "JIO" and "JIS".

5. Model: Nous Vicuna

- **Answer:** Counted 17 words, no word list provided.
- **Evaluation:** The model's output was incorrect. It undercounted the number of words and did not provide the list of words.

6. Model: Groovy

- **Answer:** Counted 30 words, sorted alphabetically.
- **Evaluation:** The model's output was incorrect. It counted the correct number of words but did not sort them correctly, and also included repetitions.

Model: gpt-3.5-turbo-0613

- **Answer:** There are 30 words in the given line. The words were sorted alphabetically step by step. The final sorted list is:
 - GTO.JMP.JNJ.JNF.GSB.JSR.JIV.JIZ.JNZ.PRR.JLE.JME.JRR.JIT.JNT.JIS.JIE.VBS.JNS.JRS.JNR.PWS.JNP.JFP.JTP.JTN.JIO.JNO.JAC.JNC.
 - ...(Shortened)
 - GSB.GTO.JAC.JIE.JIO.JIT.JIV.JIZ.JLE.JME.JMP.JNC.JNJ.JNO.JNP.JNS.JNT.JNZ.JRR.JRS.JSR.JTP.JTN.JFP.JNR.JNS.JNF.JIE.JIS.JRS.JNP.JIZ.JAC.
- **Evaluation:** The model's output is incorrect. It provided the sorting steps but did not achieve the expected final sorted order "GSB.GTO.JAC.JFP.JIE.JIO.JIS.JIT.JIV.JIZ.JLE.JME.JMP.JNC.JNF.JNJ.JNO.JNP.JNR.JNS.JNT.JNZ.JRR.JRS.JSR.JTN.JTP.PRR.PWS.VBS." by the 15th step. Also, there seems to be a repetition in the words towards the end.

Conclusion

The **Replit** model provided the closest result to the expected output, with the **gpt-3.5-turbo-0613** model showing the steps of sorting but not achieving the final expected sorted list.

It is important to verify and possibly post-process the outputs of language models to ensure accuracy in specialized tasks.

3.42.11.7 Ask_GPT4All

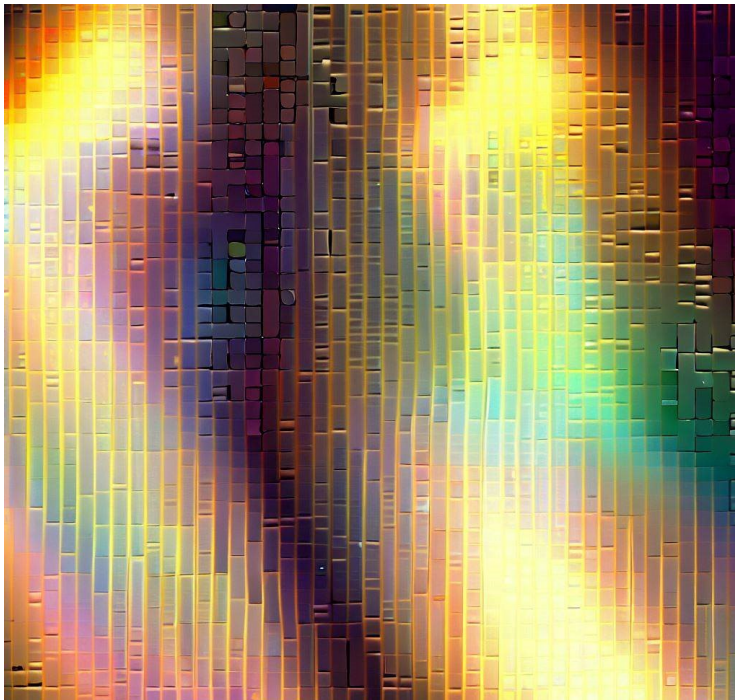
AIL.Ask GPT4All

[Previous](#) 2027 [Top](#) 49 [Next](#) 2069

MiniRobotLanguage (MRL)

AIL.Ask GPT4All

Ask GPT4ALL AI and receive answer(s)



With GPT4All the perfect Prompt Design is much more important then with GPT-4.

Intention

The AIL.Ask_GPT4All command in the Smart-Package Robot (SPR) allows users to use the AI available through GPT4All.

This command will call the AI and return the result.

[More details and how it must be installed on your local Computer or in your Network see here.](#) 955

Syntax:

```
AIL.Ask GPT4All|<Prompt>|<Variable for Answer>[|<Anwer-Return Rule>]
```

Parameters:

<Prompt>: The String that is send to the AI for processing.

<Variable for Answer>: Variable that will contain one or more answers depending on the settings.

<Anwer-Return Rule>: Can be "0" or "1". This Parameter will tell how many answers to return. "0" -> Return all answers, "1" - only the last answer.

Multiple answers will be generated if you use the `AIC.Set Number - Command`. Otherwise only one answer will be created and returned.

Example Usage:

```
AIL.Ask GPT4A11|5
```

This example sets the number of outputs to be generated to 5.

This means that when you issue a command to generate content (e.g., text, images), it will produce 5 possibly different (see `Temperature` and `Top_P Settings`) outputs.

Syntax

```
AIL.Ask GPT4A11|P1 [|P2] [|P3]
AIL.Ask|P1 [|P2] [|P3]
```

Parameter Explanation

P1 - <Prompt>: The String that is send to the AI for processing.

P2 - opt. <Variable for Answer>: Variable that will contain one or more answers depending on the settings.

P3 - **opt.** 0/1 - Flag: This flag is optional and is used to specify how the results should be returned when multiple results are expected. If you have set the number of expected results to a value higher than 1 using `AIC.Set Number`, this flag determines how the results are returned. If set to "1", only the last result will be returned. If set to "0" (or left as the default), all results will be returned.

Example

```
!*****
! AIL.-Code Sample
!*****
$$LOG=?exeloc\Output.txt
DEL.$$LOG
$$WOA="STW.SCW.SAO.NAV.WTW.WCW.CAW.WFM.MAW.WPR.WPT.SIR.WFV.GCT.AVF.TVI.TVF.UNI.S
GSC.$$WOA|.$$ANZ
GSB.Write_Log|SPR counted: $$ANZ Elements.$CrLf$-----$
FOR.$$NUM|1|9
  AIL.Set MaxToken|2048
  AIL.SetModel|$$NUM
  AIL.Set Number|1
  AIL.Set Temperatur|1
  AIL.srp|1
  AIL.snb|1
```

```

GSB.Write_Log|Model: $$NUM

$$TXT=Please analyze the given string of three-letter words, each ending with
$$TXT+$$WOA
$$TXT+$crlf$Count the number of words in the string and provide the result.$c
DBP.$$TXT

AIL.Ask GPT4A11|$$TXT|$$RET
GSB.Write_Log|$$RET
AIC.Get Several|5|$$RAW
GSB.Write_Log|Used Model: $$RAW
DBP.-----
NEX.

ENR.
'-----
:Write_Log
VAV.$$OUT=$$_01$crlf$
ATF.$$LOG|$$OUT
DBP.$$OUT
RET.
'-----
ENR.

```

Remarks

In your Prompts, ensure Clarity and Precision: Articulate your prompt in a way that unambiguously communicates the desired output from the model. Refrain from using vague or open-ended language, as this can yield unpredictable outcomes.

Incorporate Pertinent Keywords: Embed keywords in the prompt that are directly associated with the subject matter. This guides the model in grasping the context and subsequently producing more precise content.

Supply Contextual Information: Should it be necessary, furnish the model with background information or context. This equips the model to formulate more informed and contextually relevant responses.

Engage in Iterative Refinement: Embrace the process of experimentation with a variety of prompts to ascertain which is most effective. Continuously refine your prompts in response to the output generated, making adjustments until the desired results are achieved.

Limitations:

-

See also:

-

3.42.11.8 Change GPT4All URL

[AIL.Change GPT4All URL](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIL.Change GPT4All URL

Change the Endpoint for GPT4All if you use it in your Network instead of local

Intention

The `AIL.Change GPT4All URL` command is utilized to modify the current endpoint from which your GPT4All instance is receiving data.

The purpose of this command is to provide users with flexibility in their networking infrastructure.

It permits them to switch the source of the AI responses from the default location (usually localhost) to another network location, specified by a URL.

The parameter `P1` represents the URL of the desired endpoint.

If the `P1` parameter is left empty, the system will restore the default endpoint for GPT4All, which is "`http://localhost:4891/v1/completions`".

This command might be helpful in cases such as:

- Moving from a local setup to a cloud-based or distributed setup.
- Switching between different versions or instances of GPT4All hosted on different servers.
- Troubleshooting network or server issues by temporarily changing the endpoint.

Syntax

AIL.Change GPT4All URL [| P1]

Parameter Explanation

P1 - (optional) The URL to which the GPT4All endpoint should be changed.

This must be a valid URL, including the protocol (`http://` or `https://`), the network location or IP address, and the specific port and path to the API

default Port is:4891. If this parameter is left empty, the default endpoint URL will be restored. The default Endpoint is:

```
"http://localhost:4891/v1/completions"
```

Example

```
! *****
!
```



```
'*****  
' This command changes the endpoint URL to the specified network location.  
$$ENP=http://192.168.1.10:4891/v1/completions  
AIL.Change GPT4All URL|$$ENP  
  
' This command restores the endpoint URL to the default location "http://localh  
AIL.Change GPT4All URL
```

Remarks

1. Ensure that the new URL endpoint is active and correctly set up to handle GPT4All requests before executing this command.
2. While switching URLs, there might be a brief period of dis-connectivity. Plan for this potential downtime.
3. Always double-check the URL for correctness to avoid connectivity issues or potential security vulnerabilities.
4. Changing the endpoint URL affects all ongoing and future interactions with GPT4All until the URL is changed again.
5. It's recommended to verify the new URL and ensure that the respective server can handle the load before making this change.
6. For security reasons, ensure the new endpoint is trusted and properly secured.

Limitations:

-

See also:

•

3.42.11.9 Set Max Token

`AIL.Set Max Token`[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIL.Set Max Token

Set the maximum token limit for GPT4All



Intention

The `AIL.Set Max Token` command is used to define the maximum token limit for the GPT4All instance.

This command provides the ability to manage the output length of the GPT4All model, useful for controlling response brevity and managing potential usage costs when connected to the OpenAI API.

The parameter P1 represents the maximum token limit, which should be a positive integer. The token limit for GPT4All is 1024 tokens when not connected to the OpenAI API. When you use the **Open AI API via GPT4All** then also this Token Limit will be used.

When connected to the OpenAI API, GPT4All could utilize larger models such as GPT-3.5 and GPT-4, which have their own token limits and cost structures.

As of the last update, usage of the OpenAI API is not free and costs can add up depending on token usage, so it's essential to manage the maximum token limit effectively.

When using local models with GPT4All, Usage is free and therefore its recommended to use the maximum Token Limit that is provided, currently 1024 Tokens.

The token count not only includes visible words and punctuation but also invisible characters such as spaces and newlines.

Tokens in English have an average of around 4 bytes, but this can vary with the inclusion of special characters or different languages. The token limit does not guarantee a specific length of content but only sets an upper boundary to the response size.

If a prompt requires more tokens than the current limit for a satisfactory response, the AI model might produce cut-off or incomplete responses.

Also, the token count is a reliable measure for controlling output length and ensuring costs remain within budget when using the OpenAI API through GPT4All.

Here are some additional things to keep in mind about the max_tokens parameter:

- The max_tokens parameter is not a hard limit. The model may generate more tokens than the specified value if it is unable to generate a complete sentence or phrase within the specified number of tokens.
- The max_tokens parameter does not affect the quality of the generated text. The model will still try to generate the best possible text, even if it is forced to generate fewer tokens.
- The max_tokens parameter can be used to control the memory usage of the model. The model will use more memory if it is allowed to generate more tokens.

Syntax

AIL.Set Max Token [|P1]

Parameter Explanation

P1 - (optional) A positive integer defining the maximum token count. The maximum value is 1024 when using GPT4All standalone.

When connected to the OpenAI API, the maximum token count must adhere to the constraints of the utilized model.

The default value, internal to GPT4All is currently 128.

Example

```
! *****  
!  
! *****
```

Remarks

When using the OpenAI API, remember that increasing the maximum token count will lead to larger responses and hence higher costs.

Ensure you're aware of the pricing model for the specific GPT model you're utilizing. Monitor your usage closely to prevent unexpected charges. Consider implementing safeguards to limit high-cost outputs.

If the maximum token count is set too low, it might restrict the quality and context of responses, potentially rendering the output unhelpful or nonsensical.

Hence, select an appropriate token limit that balances cost and response quality.

Limitations:

-

See also:

-

3.42.11.1 Set Model

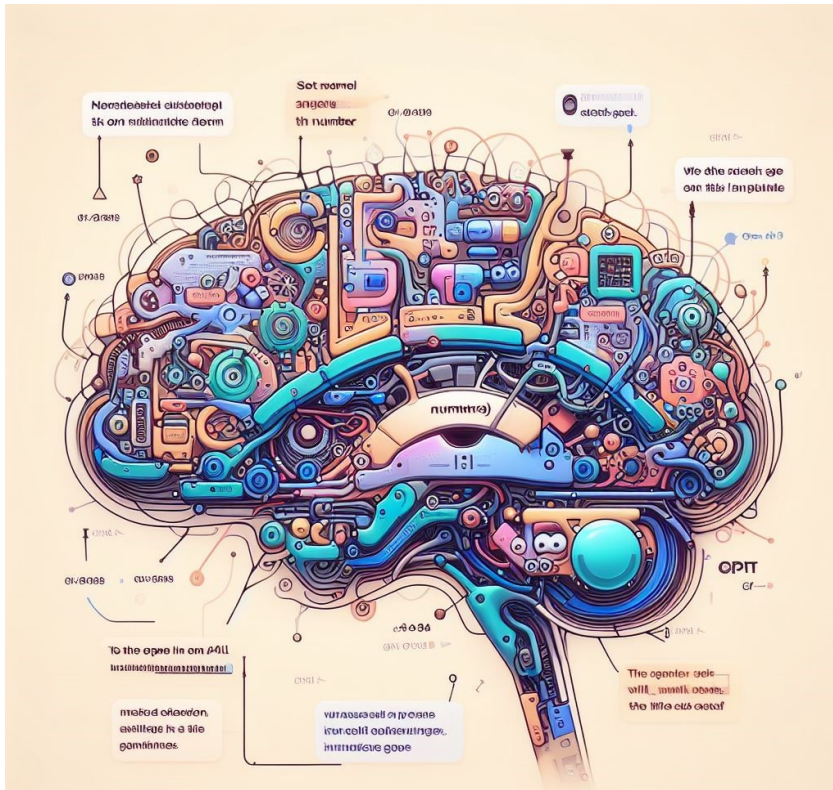
AIL.Set Model

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIL.Set Model

Choose a Model for use with GPT4All



Intention

There are two ways to use this Command. You can simply write

```
AIL.Set Model|<Number>
```

Then you will get the Model that corresponds to this number.
Of course you must have downloaded the Model into GPT4All before.

To use the OpenAI-Models from GPT4All, you also need to initialize them with your Open AI-API-Key. Also note that in this case - when you use Open AI Models - the Commands are executed in the Cloud, not local.

At the time of this writing, people say that **Nr.7 - Wizzard v1.1** seems to be the strongest Model, in some tests as good as GPT 3.5. Therefore if you do not know which Model to choose, try Nr.7.

Number	Model
1	Wizard Uncensored
2	Hermes
3	Snoozy
4	Replit
5	Nous Vicuna
6	Groovy
7	Wizard v1.1
8	ChatGPT-3.5 Turbo
9	ChatGPT-4

The second way to use this command is, you can just write

```
AIL.Set Model|<Name of Model>
```

This way you will also get the Model selected. **Using this Command, it is CASE-SENSITIVE.**

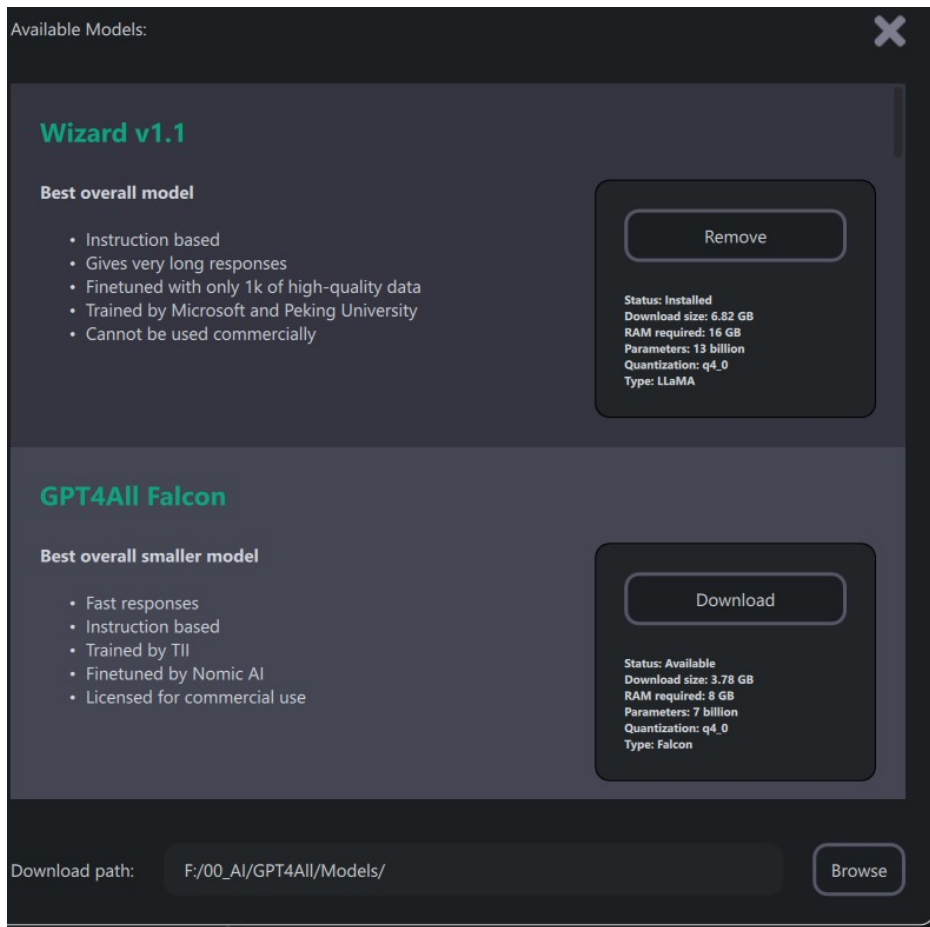
Therefore use the exact letters from the Model-name in the GUI.

Where do you find the Name of the Model?

In GPT4All the Name of the Model is the text in Color at the Models Page.

Here you can see the name of 2 Models, One is "Wizard v1.1" and the other is not installed on my system.

You can only use Models that are installed on your system.



Syntax

AIL.Set Model [| P1]

Parameter Explanation

P1 - (optional) a numeric value or directly the name of an *installed Model*. If omitted a default Model (currently "Hermes") will be used. If you chose a Model that is not installed, GPT4All will just take any Model that is installed and loaded. Its important that you write the Model-Name exactly as you see it in the GUI.

This Command is CASE-SENSITIVE. Therefore use large letters when the model-name needs large letters.

Example

```

! *****
! AIL.-Code Sample
! *****
$$LOG=?exeloc\Output.txt

```

```

DEL.$$LOG
$$WOA="STW.SCW.SAO.NAV.WTW.WCW.CAW.WFM.MAW.WPR.WPT.SIR.WFV.GCT.AVF.TVI.TVF.UNI.S

GSC.$$WOA|.$$ANZ
GSB.Write_Log|SPR counted: $$ANZ Elements.$crlf$-----$

FOR.$$NUM|1|9
  AIL.Set MaxToken|1024
  AIL.SetModel|$$NUM
  AIL.Set Number|1
  AIL.Set Temperatur|1
  AIL.srp|1
  AIL.snb|1

GSB.Write_Log|Model: $$NUM

$$TXT=As my AI assistant, I'd like you to analyze a string of three-letter words.
$$TXT+First, please count the number of words in this string. Remember, each word is
$$TXT+Next, I'd like you to sort these words in alphabetical order. Please explain
$$TXT+$crlf$Finally, ensure that the words are sorted in alphabetical order.
DBP.$$TXT

AIL.Ask GPT4All|$$TXT|$$RET
GSB.Write_Log|$$RET
AIC.Get Several|5|$$RAW
GSB.Write_Log|Used Model: $$RAW
DBP.-----
NEX.

ENR.
'-----
:Write_Log
VAV.$$OUT=$$_01$crlf$
ATF.$$LOG|$$OUT
DBP.$$OUT
RET.
'-----
ENR.

```

Remarks

In my Tests the `AIL.Set Model` Command did not work with all Models, this may be fixed in future versions of GPT4All.

This Command is CASE-SENSITIVE.

Limitations:

-

See also:

-

3.42.11.1: Set n Batch

[AIL.Set n_Batch](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIL.Set n_Batch

Adjusts the batch processing size of tokens in the GPT4All AI



Intention

The `AIL.Set n_Batch` command lets users modify the "n_batch" parameter that determines the number of prompt tokens processed in parallel by the GPT4All AI model.

The parameter **P1** is an integer representing the size of the batch to be processed concurrently:

Larger **P1** values result in more tokens being processed simultaneously, which can decrease latency and increase speed. However, this comes at the cost of increased resource requirements, which can lead to potential performance issues if the system resources are limited.

Smaller **P1** values result in fewer tokens being processed at once, which can be more resource-efficient but may increase processing time.

The optimal "n_batch" value depends on the specifics of your system's hardware and the size of the data being processed.

USAGE EXAMPLES

```
' Set "n_batch" to a larger value for faster processing:  
' This command increases the number of tokens processed concurrently, potentially  
AIL.Set n Batch|200  
  
' Set "n_batch" to a smaller value for more efficient resource use:  
' This command decreases the number of tokens processed at once, potentially inc  
AIL.Set n Batch|50
```

The `n_batch` parameter controls the number of prompt tokens that are processed in parallel. A higher value of the `n_batch` parameter will result in lower latency, but it will also increase the resource requirements.

The default value of the `n_batch` parameter is 128. This means that 128 prompt tokens are processed in parallel. However, you can adjust the value of the `n_batch` parameter to achieve the desired balance between latency and resource requirements.

For example, if you want to generate text as quickly as possible, you can set the value of the `n_batch` parameter to a higher value, such as 256 or 512. This will ensure that the model is able to process the prompt tokens as quickly as possible.

On the other hand, if you are concerned about resource requirements, you can set the value of the `n_batch` parameter to a lower value, such as 64 or 32. This will ensure that the model does not use too much memory or CPU resources.

The "n_batch" parameter is an important factor in managing the trade-off between processing speed and resource efficiency. Carefully consider your system's resources and the data size when determining an optimal "n_batch" value.

CAUTIONS

Setting the "n_batch" value requires careful consideration. High values might increase processing speed but could overburden your system's resources, leading to performance issues. Conversely, low values might be more resource-efficient but increase processing time. Always consider the specifics of your system's hardware and the size of the data when setting the "n_batch" value.

The best way to determine the optimal value of the `n_batch` parameter for your needs is to experiment with different values and see what works best for you.

Here are some additional things to keep in mind about the `n_batch` parameter:

- The `n_batch` parameter does not affect the quality of the generated text. The model will still try to generate the best possible text, even if it is processed in parallel.
- The `n_batch` parameter can be used to control the latency of the model. The model will be faster if it is processed in parallel, as it will not have to wait for each token to be processed individually.
- The `n_batch` parameter can be used to control the memory usage of the model. The model will use more memory if it is processed in parallel, as it will have to store more tokens in memory.

- The range of the `n_batch` parameter in GPT-4All is from 1 to 1024. A value of 1 means that only one prompt token will be processed in parallel, while a value of 1024 means that all of the prompt tokens will be processed in parallel.

Syntax

```
AIL.Set n_Batch [ | P1 ]  
AIL.snb [ | P1 ]
```

Parameter Explanation

P1 - (*optional*) This is an integer representing the desired batch size ("n_batch"). The chosen value should balance your need for processing speed with your system's resource availability and capabilities.

Example

```
! *****  
!  
! *****
```

Remarks

-

Limitations:

-

See also:

-

Of course each call will use Tokens.

Syntax

AIL.Set Number|P1

Parameter Explanation

P1 – This is an integer representing the desired number of output responses from the AI.

If this parameter is set to "0", it is automatically reset to "1".

Example

```
| *****  
|  
| *****
```

Remarks

-

Limitations:

-

See also:

-

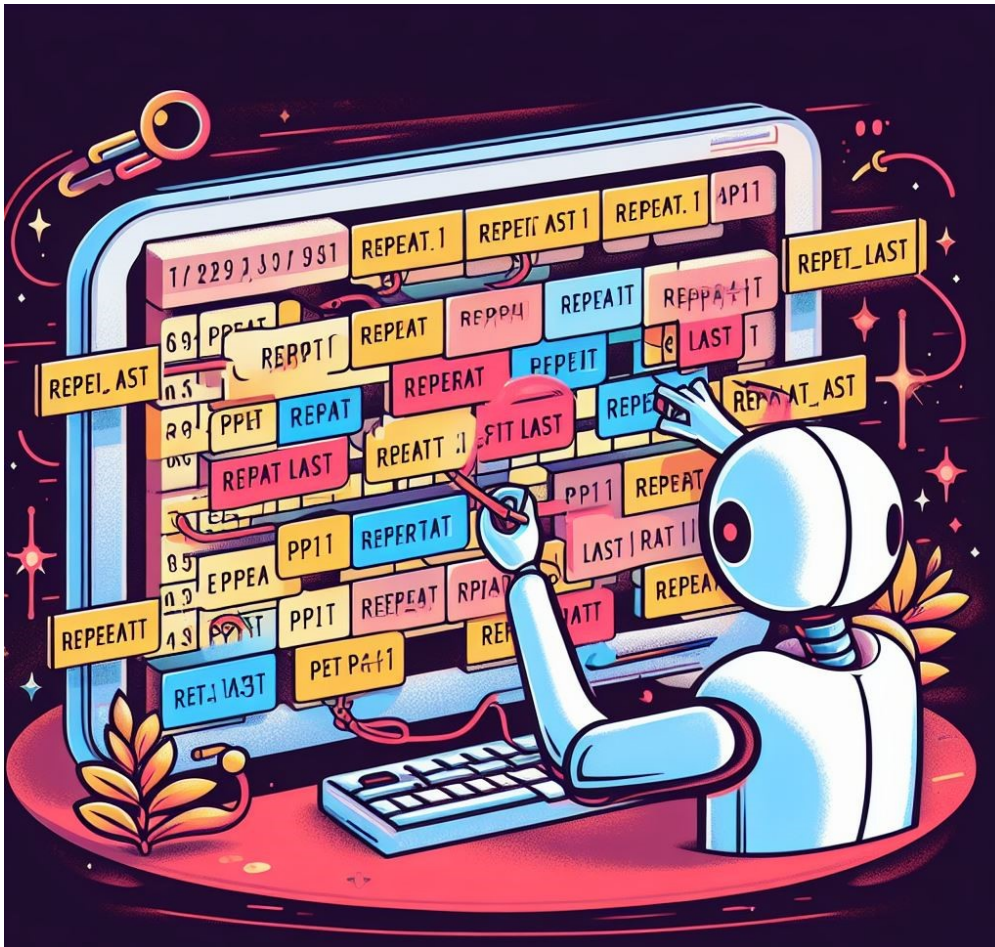
3.42.11.1: Set Repeat Last

[All.Set Repeat Last](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIL.Set Repeat Last

Modifies the scope of the repeat penalty in the GPT4All AI model's generation history



Intention

The `AIL.Set Repeat Last` command allows users to manipulate the "repeat_last_n" parameter that determines how far back in the model's generation history the repeat penalty is applied.

The parameter **P1** is an integer specifying the number of the most recent tokens to consider when applying the repeat penalty:

- Higher P1 values expand the scope of the repeat penalty, making the model consider more of its recent outputs when evaluating for repetition. This can help generate more varied and less repetitive outputs.

- Lower P1 values limit the scope of the repeat penalty, only penalizing repetition in a smaller subset of the model's most recent outputs. This can be useful if allowing some level of repetition is desirable for your specific task.
- If P1 is missing or omitted, a default value of 10 is used.

USAGE EXAMPLES

```
'Set "repeat_last_n" to a larger value to discourage repetitive output:  
AIL.Set Repeat Last|10
```

This command makes the model consider its last 10 tokens when applying the repeat penalty, helping to ensure diverse and less repetitive outputs.

```
' Set "repeat_last_n" to a lower value to allow some level of repetition:  
AIL.Set Repeat Last|2
```

This command restricts the repeat penalty to the model's last 2 tokens, allowing a greater level of repetition in the generated output. The `repeat_last_n` parameter controls how far back in the model's generation history the repeat penalty is applied. A higher value of the `repeat_last_n` parameter will result in a lower chance of the model repeating itself, but it will also make the model less creative.

The default value of the `repeat_last_n` parameter is 10.

This means that the model will not generate a token that is the same as any of the last 10 tokens that it has generated. However, you can adjust the value of the `repeat_last_n` parameter to achieve the desired balance between creativity and repetition.

For example, if you want to generate text that is very free of repetition, you can set the value of the `repeat_last_n` parameter to a higher value, such as 20 or 30. This will ensure that the model is very unlikely to repeat itself.

On the other hand, if you want to generate text that is more creative and allows for some repetition, you can set the value of the `repeat_last_n` parameter to a lower value, such as 5 or 3. This will allow the model to generate more text that is similar to the prompt, even if it repeats itself occasionally.

The best way to determine the optimal value of the `repeat_last_n` parameter for your needs is to experiment with different values and see what works best for you.

Here are some additional things to keep in mind about the `repeat_last_n` parameter:

- The `repeat_last_n` parameter does not affect the quality of the generated text. The model will still try to generate the best possible text, even if it is penalized for repeating itself.
- The `repeat_last_n` parameter can be used to control the diversity of the generated text. The model will be more diverse if the `repeat_last_n` parameter is set to a higher value, as it will be less likely to repeat itself.
- The `repeat_last_n` parameter can be used to control the length of the generated text. The model will be longer if the `repeat_last_n` parameter is set to a lower value, as it will be more likely to repeat itself.

- The range of the repeat_last_n parameter in GPT-4All is from 1 to 1024. A value of 1 means that the repeat penalty will only be applied to the last token that the model has generated, while a value of 1024 means that the repeat penalty will be applied to all of the tokens that the model has generated.

Syntax

```
AIL.Set Repeat Last [|P1]  
AIL.srl [|P1]
```

Parameter Explanation

P1 - (*optional*) This is an integer representing the desired "repeat_last_n" value. If **P1** is missing or omitted, the default value of 10 is used.

Example

```
| *****  
|  
| *****
```

Remarks

-

Limitations:

-

See also:

-

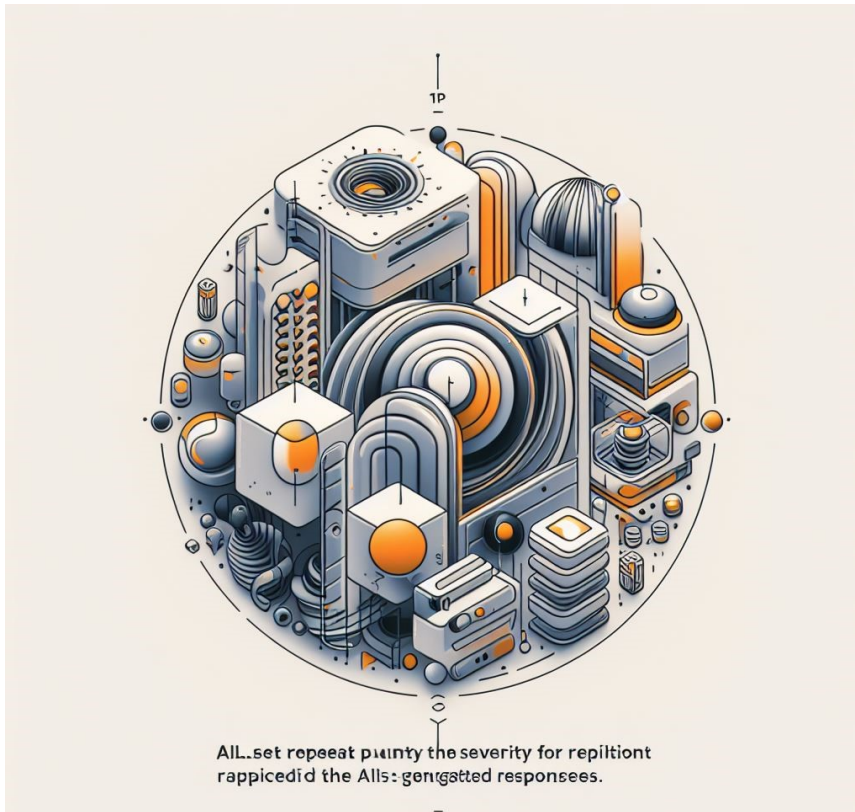
3.42.11.1 Set Repeat Penalty

[AIL.Set Repeat Penalty](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIL.Set Repeat Penalty

Adjusts the penalty for repetition in the GPT4All AI model's responses



Intention

The `AIL.Set Repeat Penalty` command allows users to modify the severity of penalties applied for repetition in the AI's generated responses.

This setting is especially useful when diversity and novelty are required, while repetitive outputs are undesired.

The parameter P1 represents a floating-point number within the range of 0 to 10:

- A value of 0 (default if P1 is missing) imposes no penalty on repetitions, giving the model full liberty to repeat content as necessary. This might be useful in scenarios where certain points need reiteration or repetition is not a concern.
- A value of 10 enforces a severe penalty for repetitions, pushing the model towards generating highly diverse and varied content. This could be beneficial in tasks requiring creative writing, brainstorming, or other contexts where the novelty of ideas is highly valued.

- The default value for P1 is between 0 and 2.0, meaning a mild penalty for repetition is applied by default.

USAGE EXAMPLES

```
' Enforce a high repeat penalty:
' This command will severely penalize repetitions, driving the model towards high
AIL.Set Repeat Penalty|10

' Set a low repeat penalty:
' This command allows for greater repetition within the model's outputs, allowing
AIL.Set Repeat Penalty|1
```

The repeat penalty is an important factor when diversity or novelty of responses is a key requirement.

Setting the repeat penalty requires careful consideration.

High penalties might deter relevant repetition, leading to responses that feel disjointed or go off-topic.

Conversely, low penalties might lead to monotonous or repetitive responses.

Always check and validate the AI output after modifying this setting.

Consider the nature of your task and the desired characteristics of the AI's output when setting the repeat penalty. Experimentation with different values can help to find the optimal balance for your specific needs.

Syntax

AIL.Set Repeat Penalty[|P1]

Parameter Explanation

P1 – (*optional*) This is the desired repeat penalty value.

It should be a floating-point number between 0 and 10, inclusive. The absence of P1 or a value of "0" implies no penalty for repetitions.

Example

```
' *****
'
' *****
```

Remarks

-

Limitations:

-

See also:

-

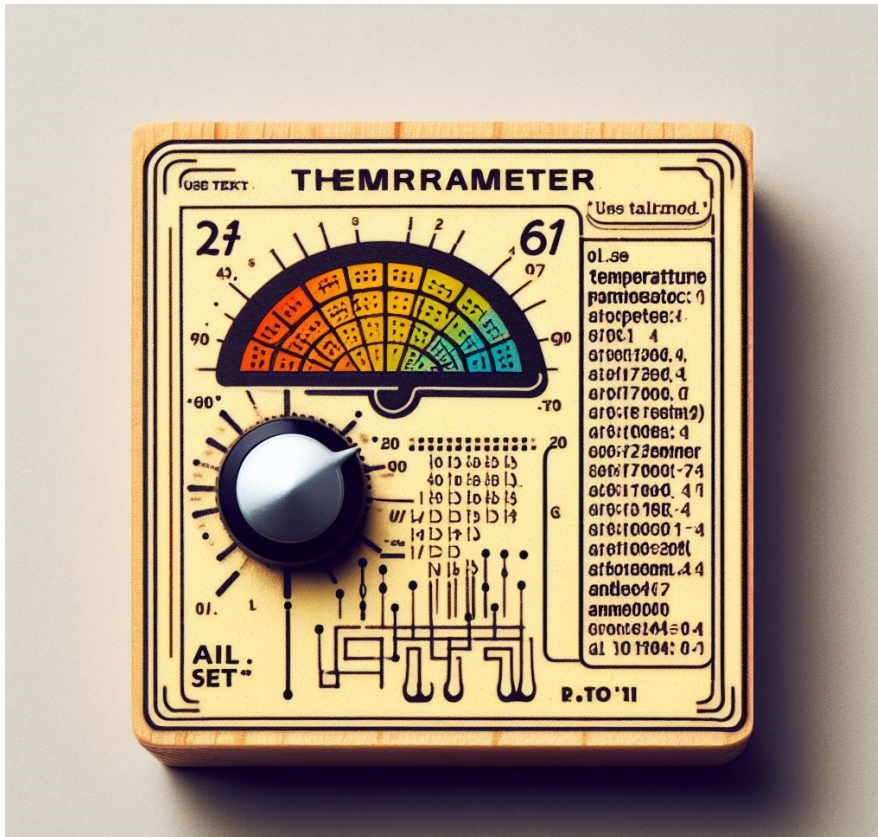
3.42.11.1!Set Temperature

[AIL.Set Temperature](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIL.Set Temperature

Configure the temperature setting for GPT4All responses



Intention

The `AIL.Set Temperature[|P1]` command is designed to adjust the "temperature" of the GPT4All AI model responses. This temperature parameter is a key factor in controlling the level of randomness and variety in the AI-generated output.

The parameter `P1` is a floating-point number that falls within the range of 0 to 1. The behavior of the AI model varies based on the set temperature:

- A lower value (closer to 0) renders the AI's responses more deterministic and consistent, often sticking to a more focused or narrower scope of context. This is ideal for queries that require precise, factual, or more predictable responses.
- A higher value (closer to 1) introduces a higher degree of randomness and creativity, leading to diverse and broad outputs. Such settings are useful for more exploratory or creative tasks where a wide range of ideas or suggestions is beneficial.

- If you leave away the value **P1** then the default value of "0" will be set.

Set the temperature to a lower value:

```
AIL.Set Temperature|0.2
```

This command instructs the AI to generate outputs that are more focused and deterministic, limiting the scope of potential responses.

Set the temperature to a higher value:

```
AIL.Set Temperature|0.8
```

This command guides the AI towards more diverse and unpredictable responses, promoting creativity and variety in the outputs.

Syntax

AIL.Set Temperature [|P1]

Parameter Explanation

P1 - (*optional*) This represents the desired temperature value. It should be a floating-point number between 0 and 1, inclusive.

The chosen value will significantly impact the character of the AI's output.

Example

```
| *****  
|  
| *****
```

Remarks

-

Limitations:

-

See also:

-

If **P1** is missing or set to "0", it will default to considering all tokens, effectively turning off the "top_k" sampling strategy.

USAGE EXAMPLES

```
' Set "top_k" to 1 for greedy decoding:  
' This command sets the AI to always select the most likely next token.  
AIL.SetTop K|1  
  
' Set "top_k" to a higher value for more diverse outputs:  
' This command allows the AI to select from the top 50 most likely next tokens,  
AIL.SetTop K|50
```

The range of the top-k parameter in GPT-4All is from 1 to 1000. A value of 1 means that the model will only be able to choose from the top 1 token when generating text, while a value of 1000 means that the model will be able to choose from the top 1000 most likely tokens when generating text.

The default value of the top-k parameter is 40. This means that the model will be able to choose from the top 40 most likely tokens when generating text. However, you can adjust the value of the top-k parameter to achieve the desired level of diversity in your generated text.

For example, if you want to generate text that is very diverse, you can set the value of the top-k parameter to a higher value, such as 100 or 200. This will ensure that the model is very unlikely to generate the same tokens over and over again.

On the other hand, if you want to generate text that is more creative and allows for some repetition, you can set the value of the top-k parameter to a lower value, such as 20 or 10. This will allow the model to generate more text that is similar to the prompt.

The best way to determine the optimal value of the top-k parameter for your needs is to experiment with different values and see what works best for you.

Here are some additional things to keep in mind about the top-k parameter:

- The top-k parameter does not affect the quality of the generated text. The model will still try to generate the best possible text, even if it is limited to the top-k tokens.
- The top-k parameter can be used to control the latency of the model. The model will be slower if it is limited to the top-k tokens, as it will have to search through a smaller set of tokens to find the best one.
- The top-k parameter can be used to control the memory usage of the model. The model will use less memory if it is limited to the top-k tokens, as it will not have to store as many tokens in memory.
- The "top_k" value is a crucial parameter when determining the balance between diversity and coherence in the AI's responses.
- High "top_k" values can lead to more varied and less predictable outputs, but may risk losing coherence or relevance. Conversely, a low "top_k" value can lead to more predictable and safer responses, but may lack novelty or diversity.

- It's advisable to adjust the "top_k" value according to your specific task and desired output characteristics.

CAUTIONS

Setting the "top_k" value requires a careful balance. High "top_k" values might lead to outputs that are diverse but potentially less coherent or off-topic, while low values might result in outputs that are coherent but potentially lacking in diversity or novelty. As always, it is crucial to validate the AI output after changing this setting. Experiment with different "top_k" values to find the best setting for your specific requirements.

Syntax

AIL.Set Top K[|P1]

Parameter Explanation

P1 - (optional) This is an integer representing the "top_k" sampling value. If *P1* is missing or set to "0", all tokens will be considered in each decision, effectively turning off "top_k" sampling.

Example

```
! *****  
!  
! *****
```

Remarks

-

Limitations:

-

See also:

-

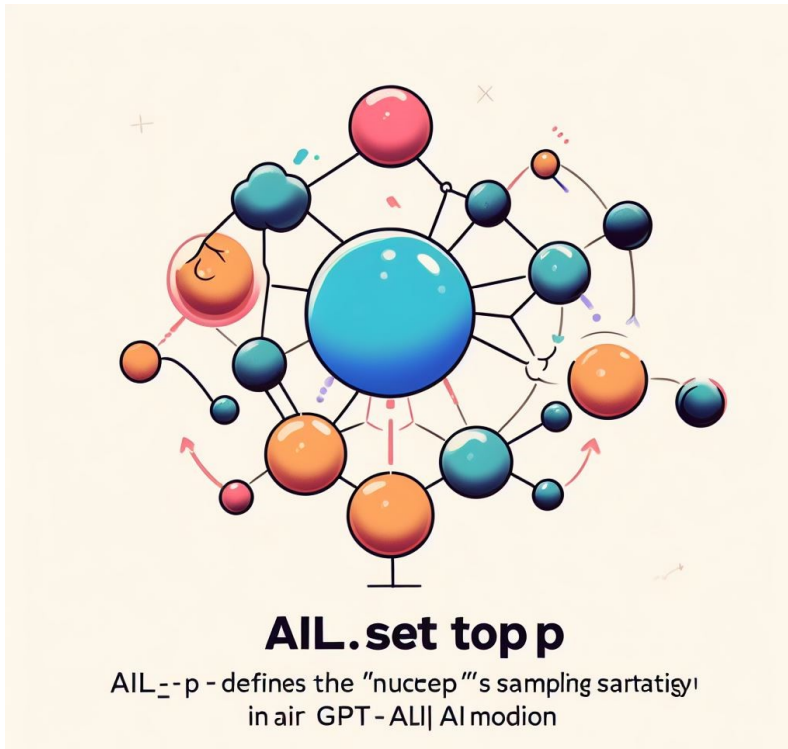
3.42.11.1 Set Top P

[AIL.Set Top P](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIL.Set Top P

Defines the "top_p" (nucleus) sampling strategy in GPT4All AI model



Intention

The "AIL.SetTop P" command allows users to control the "top_p" or nucleus sampling parameter, a powerful tool that influences the randomness and diversity of the GPT4All AI model's outputs.

The parameter **P1** is a floating-point number that sets the "top_p" value. This parameter guides the model to consider, at each generation step, only the most probable tokens whose combined probabilities reach up to the "top_p" value.

A **P1** value closer to 1 allows the model to consider a wider range of tokens, leading to more diverse and potentially less predictable outputs.

A **P1** value closer to 0 constrains the model to only consider the most probable tokens, leading to safer and more predictable responses.

However, it might restrict the creativity and diversity of the outputs.

If **P1** is missing or set to "0", it will default to considering all tokens, effectively turning off the "top_p" sampling strategy.

USAGE EXAMPLES

```
' Set "top_p" to a low value for more predictable outputs:  
' This command will make the AI lean towards the most likely tokens, resulting in less fluent text.  
AIL.SetTop P|0.2
```

```
' Set "top_p" to a higher value for more diverse outputs:  
' This command increases the AI's diversity and unpredictability in the outputs.  
AIL.SetTop P|0.9
```

The top-p parameter in GPT-4All controls the fluency of the generated text. A higher value of the top-p parameter will result in more fluent text, while a lower value will result in less fluent text.

The top-p parameter works by limiting the probability of the model generating a token. For example, if the top-p parameter is set to 0.9, the model will only be able to generate tokens that have a probability of 90% or higher. This will make the generated text more fluent, as the model will be less likely to generate tokens that are out of place or grammatically incorrect.

The range of the top-p parameter in GPT-4All is from 0 to 1.0. A value of 0 means that the model will only be able to generate tokens that have a probability of 0%, while a value of 1.0 means that the model will be able to generate any token, regardless of its probability.

The default value of the top-p parameter is 0.8. This means that the model will only be able to generate tokens that have a probability of 80% or higher. However, you can adjust the value of the top-p parameter to achieve the desired level of fluency in your generated text.

For example, if you want to generate text that is very fluent, you can set the value of the top-p parameter to a higher value, such as 0.95 or 0.99. This will ensure that the model is very unlikely to generate tokens that are out of place or grammatically incorrect.

On the other hand, if you want to generate text that is more creative and allows for some errors, you can set the value of the top-p parameter to a lower value, such as 0.7 or 0.6. This will allow the model to generate more text that is similar to the prompt, even if it is not grammatically perfect.

The default value of the top-p parameter is 0.8. This means that the model will only be able to generate tokens that have a probability of 80% or higher. However, you can adjust the value of the top-p parameter to achieve the desired level of fluency in your generated text.

For example, if you want to generate text that is very fluent, you can set the value of the top-p parameter to a higher value, such as 0.95 or 0.99. This will ensure that the model is very unlikely to generate tokens that are out of place or grammatically incorrect.

On the other hand, if you want to generate text that is more creative and allows for some errors, you can set the value of the top-p parameter to a lower value, such as 0.7 or 0.6. This will allow the model to generate more text that is similar to the prompt, even if it is not grammatically perfect.

The best way to determine the optimal value of the top-p parameter for your needs is to experiment with different values and see what works best for you.

Here are some additional things to keep in mind about the top-p parameter:

- The top-p parameter does not affect the quality of the generated text. The model will still try to generate the best possible text, even if it is limited by the top-p parameter.
- The top-p parameter can be used to control the latency of the model. The model will be slower if it is limited by the top-p parameter, as it will have to search through a larger set of tokens to find the best one.
- The top-p parameter can be used to control the memory usage of the model. The model will use more memory if it is allowed to generate more tokens, as it will not have to store as many tokens in memory.

Syntax

AIL.Set Top P [|P1]

Parameter Explanation

P1 - (optional) This is a floating-point number representing the "top_p" sampling value.

If P1 is missing or set to "0", the model will consider all tokens in each decision, effectively disabling "top_p" sampling.

Example

```
| *****  
|  
| *****
```




Remarks

-

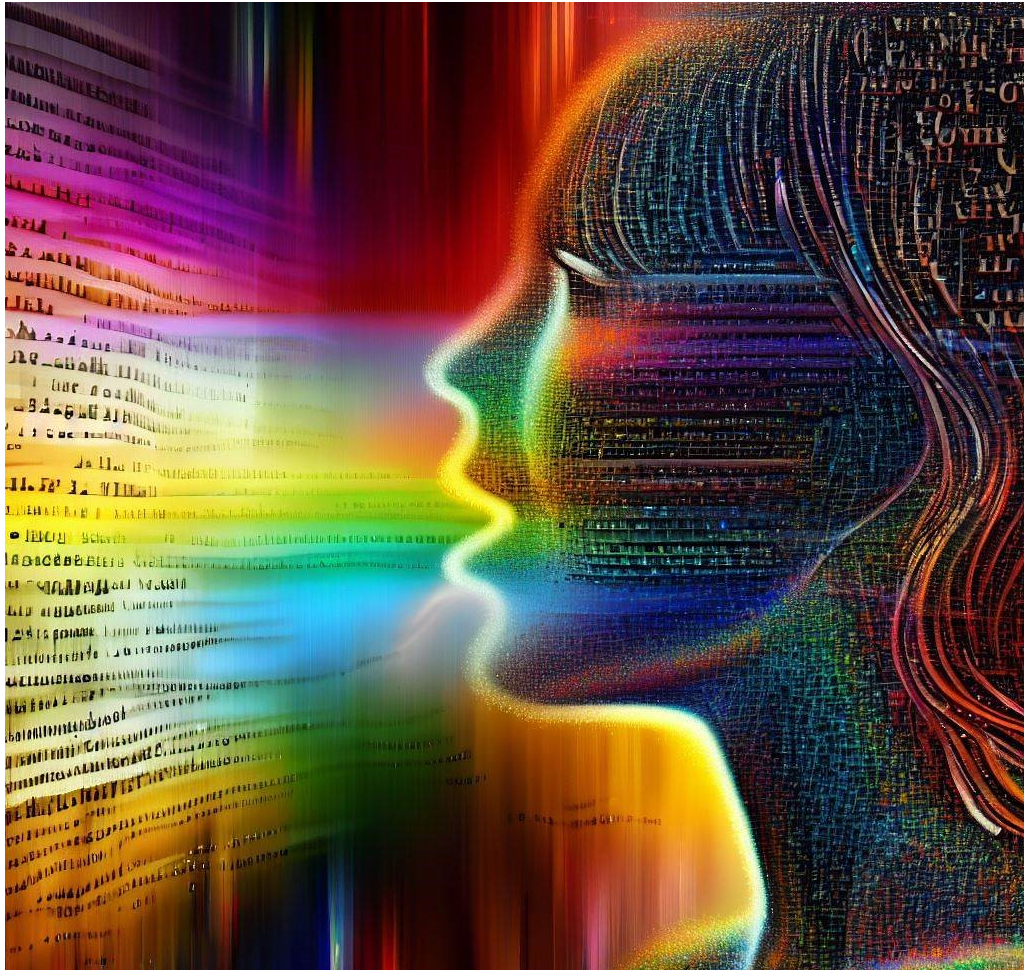
Limitations:

-

See also:

- [1.6.1. Program Flow Control](#) 
- [! Smart Package Robot 's Parallel Robot Operations](#) 
- [1.5. Features and Hints](#) 

3.42.12 AIS. - AI Speech Synthesis



3.42.12.1 ! Elevenlabs Speech Synthesis

Elevenlabs Speech Synthesis Integration

In addition to the API keys for OpenAI, you may also require API keys for Elevenlabs' Speech Synthesis.



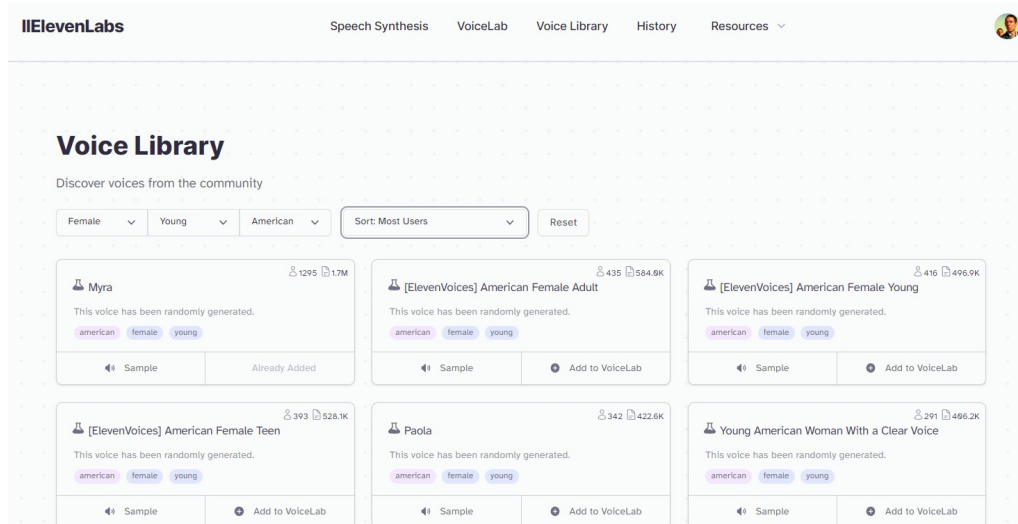
Click on this Button to see their original WEB-Site:

Elevenlabs -Speech Synthesis

To obtain these, you'll need to create an account on Elevenlabs and navigate to your profile to copy your unique API key. This key is your golden ticket to generating high-quality speech, potentially from the output of your AI models.

Please remember, a working internet connection is a prerequisite for using this feature. The speech synthesis process happens on Elevenlabs' servers, meaning your system needs to be online for the magic to occur.

Once the speech is generated, the SPR will fetch the resulting MP3 file from Elevenlabs and download it to a location of your choosing. This way, you can have access to high-quality synthesized speech at any time, from anywhere.



This is the Elevenlabs Voice Library. Here you can choose the Speakers you want for your tasks.

ElevenLabs Speech Synthesis is a specialized service designed to generate high-quality voiceovers.

Once you have chosen or created your voice, you can utilize their Speech Synthesis feature to generate voiceovers.

To access this, head over to the Speech Synthesis tab and select your desired voice from the dropdown menu.

The option to preview each voice is available by clicking the 'Play' button next to the voice name.

One interesting aspect of the ElevenLabs Speech Synthesis system is the variability in its AI.

Each time you generate a voiceover, the AI will produce slightly different results, adding a unique touch to each generation.

Moreover, it offers a set of Voice Settings that allows users to control how the voice sounds and performs.

Many users have found that setting the 'Stability' around 40 and 'Similarity' around 75 yields good results, but these settings can be adjusted to suit the specific needs of your project.

The choice of the model you use in ElevenLabs should be influenced by the language you intend to use for your voiceovers.

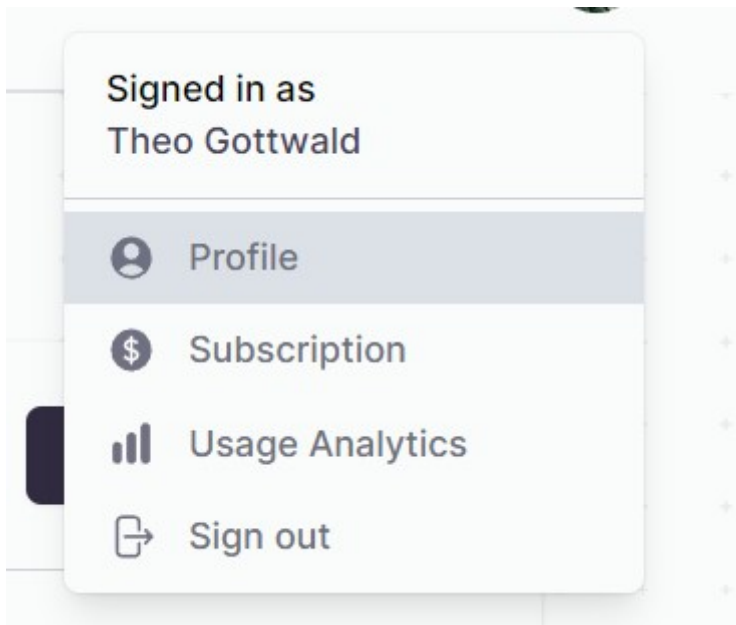
If your primary language is English, it's recommended to stick with the monolingual model, as the multilingual model is still experimental and may have a few quirks that are being worked on.

Another captivating feature of this system is its ability to deliver speech in a specific way or with certain emotions, depending on the text, context, and dialogue tag.

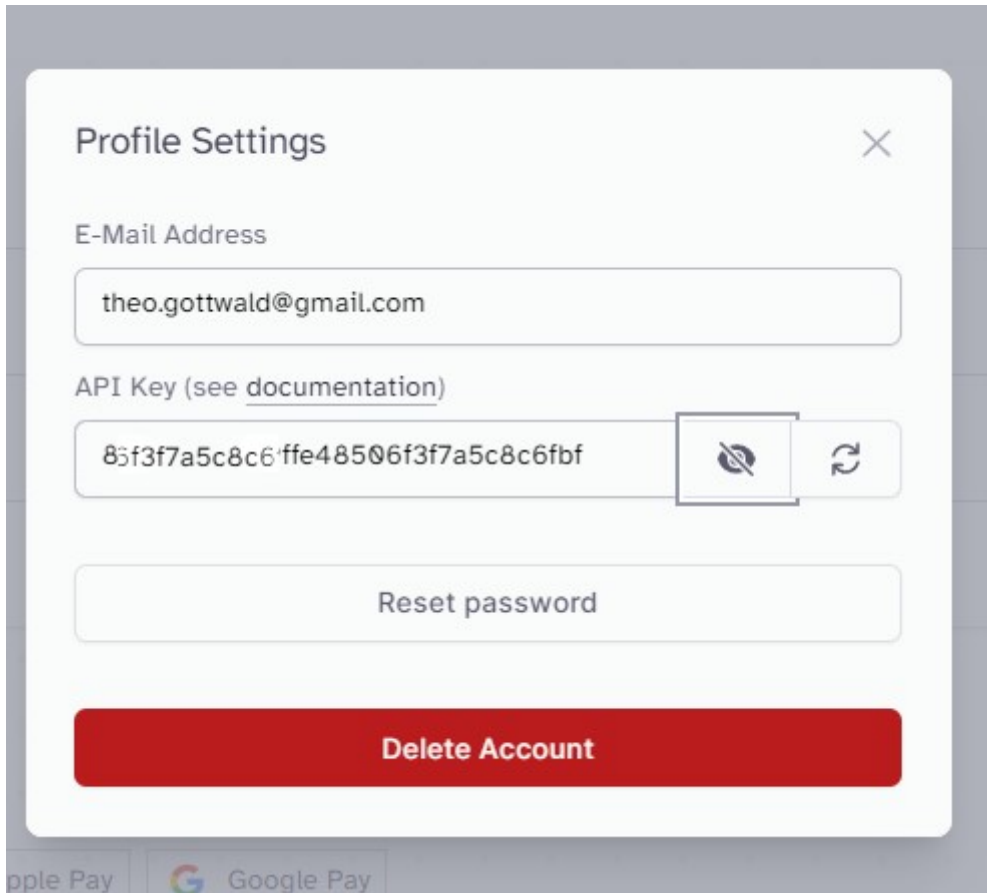
For instance, "That's not funny", he said seriously.

While not perfect, this workaround has proven to be quite helpful for many users, offering a greater degree of control over the emotional tone of the generated speech.

Regarding the cost, ElevenLabs offers a range of pricing options to accommodate different needs and budgets. They start off with a free tier, perfect for those wanting to try out the service, and then pricing begins at a modest \$5, making it quite affordable. Their pricing structure takes into account various elements such as the level of personalization, any extra features required, the total number of users, and the type of deployment, although these factors are subject to change. For the most accurate and up-to-date information about their pricing, it is recommended to [visit the ElevenLabs website](#) directly.



In your User-Profile you can get your API-Key.



**In your User-Profile you can get your API-Key.
This API-Key is needed for the SPR to authenticate with Elevenlabs AI-Voice Services.**

3.42.12.2 !AIS - Sample Code

AIS. - Sample Code[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

TMP .**If Result available****Intention**

```
AIS.Set Key|file
$$TXT=Hello, this is Sarah from Tech Support.
AIS.Say Text|$$TXT|-
```

```
$$TXT=Hi Sarah, I'm having some issues with my computer.
AIS.Say Text|$$TXT
```

```
$$TXT=Buenas tardes, soy Sarah del departamento de asistencia técnica.
AIS.Say Text|$$TXT
```

```
$$TXT=Buenas tardes, Sarah. Mi ordenador no funciona bien; se bloquea todo el t.
AIS.Say Text|$$TXT
```

```
$$TXT=Bonjour, c'est Sarah du support technique.
AIS.Say Text|$$TXT
```

```
$$TXT=Bonjour Sarah, j'ai quelques problèmes avec mon ordinateur.
AIS.Say Text|$$TXT
```

```
$$TXT=Ciao, sono Sarah del supporto tecnico.
AIS.Say Text|$$TXT
```

```
$$TXT=Ciao Sarah, sto avendo alcuni problemi con il mio computer.
AIS.Say Text|$$TXT
```

```
$$TXT=Guten Tag, hier ist Sarah vom technischen Support.
AIS.Say Text|$$TXT
```

```
$$TXT=Guten Tag Sarah, wir haben einige Probleme mit dem Computer.
AIS.Say Text|$$TXT
```

```
MBX. !
ENR.
```

Remarks

-

Limitations:

-

See also:

- [1.6.1. Program Flow Control](#)¹³⁴
- [! Smart Package Robot 's Parallel Robot Operations](#)³⁷⁴⁵
- [1.5. Features and Hints](#)⁹³

3.42.12.3 AIS.-Header Settings

Text-to-Speech Header

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

Text-to-Speech Header

Elevenlabs.io TTS-Header

text-to-speech Convert text into audio. ↑

POST /v1/text-to-speech/{voice_id} Text To Speech ↑

Converts text into speech using a voice of your choice and returns audio.

Parameters Cancel

Name	Description
voice_id required string <small>(path)</small>	Voice ID to be used, you can use https://api.elevenlabs.io/v1/voices to list all the available voices.
optimize_streaming_latency Integer <small>(query)</small> maximum: 22 minimum: 0	You can turn on latency optimizations at some cost of quality. The best possible final latency varies by model. Possible values: 0 - default mode (no latency optimizations) 1 - normal latency optimizations (about 50% of possible latency improvement of option 3) 2 - strong latency optimizations (about 75% of possible latency improvement of option 3) 3 - max latency optimizations 4 - max latency optimizations, but also with text normalizer turned off for even more latency savings (best latency, but can mispronounce eg numbers and dates). Defaults to 0.
output_format string <small>(query)</small>	Output format of the generated audio. Must be one of: mp3_44100_64 - output format, mp3 with 44.1kHz sample rate at 64kbps. mp3_44100_96 - output format, mp3 with 44.1kHz sample rate at 96kbps. mp3_44100_128 - default output format, mp3 with 44.1kHz sample rate at 128kbps. mp3_44100_192 - output format, mp3 with 44.1kHz sample rate at 192kbps. Requires you to be subscribed to Creator tier or above. pcm_16000 - PCM format (S16LE) with 16kHz sample rate. pcm_22050 - PCM format (S16LE) with 22.05kHz sample rate. pcm_24000 - PCM format (S16LE) with 24kHz sample rate. pcm_44100 - PCM format (S16LE) with 44.1kHz sample rate. Requires you to be subscribed to Independent Publisher tier or above.
xi-api-key string <small>(header)</small>	Your API key. This is required by most endpoints to access our API programmatically. You can view your xi-api-key using the 'Profile' tab on the website.

Request body required application/json

```
{
  "text": "string",
  "model_id": "eleven_monolingual_v1",
  "voice_settings": {
    "stability": 0,
    "similarity_boost": 0,
    "style": 0,
    "use_speaker_boost": true
  }
}
```

3.42.12.3.1 Get Latency

[AIS.Get Latency](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Get Latency

Gets latency optimization level.

Intention

The `AIS.Get Latency` command allows you to Get the level of latency optimizations for the AIS engine.

This can be useful for applications where low latency is crucial but can come at the cost of reduced voice quality.

The latency optimization level can affect the quality of the voice.

Higher levels of optimization may result in lower voice quality but will reduce latency.

Option 4 may cause mispronunciations of numbers and dates.

Syntax

AIS.Get Latency[|P1]**AIS.Glt[|P1]**

Parameter Explanation

- **P1: optional** The variable where the result will be stored. If omitted, the result is pushed onto the Top of Stack (TOS).

Value	Description
0	Default mode (no latency optimizations)
1	Normal latency optimizations (about 50% of possible latency improvement of option 3)
2	Strong latency optimizations (about 75% of possible latency improvement of option 3)
3	Max latency optimizations

4	Max latency optimizations with text normalizer turned off (best latency, may mispronounce numbers and dates)
---	--

Example

```
! *****  
! AIS.-Sample  
! *****  
$$VAR=  
AIS.Get Latency|$$VAR  
MBX.$$VAR  
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.12.3.2 Save Key

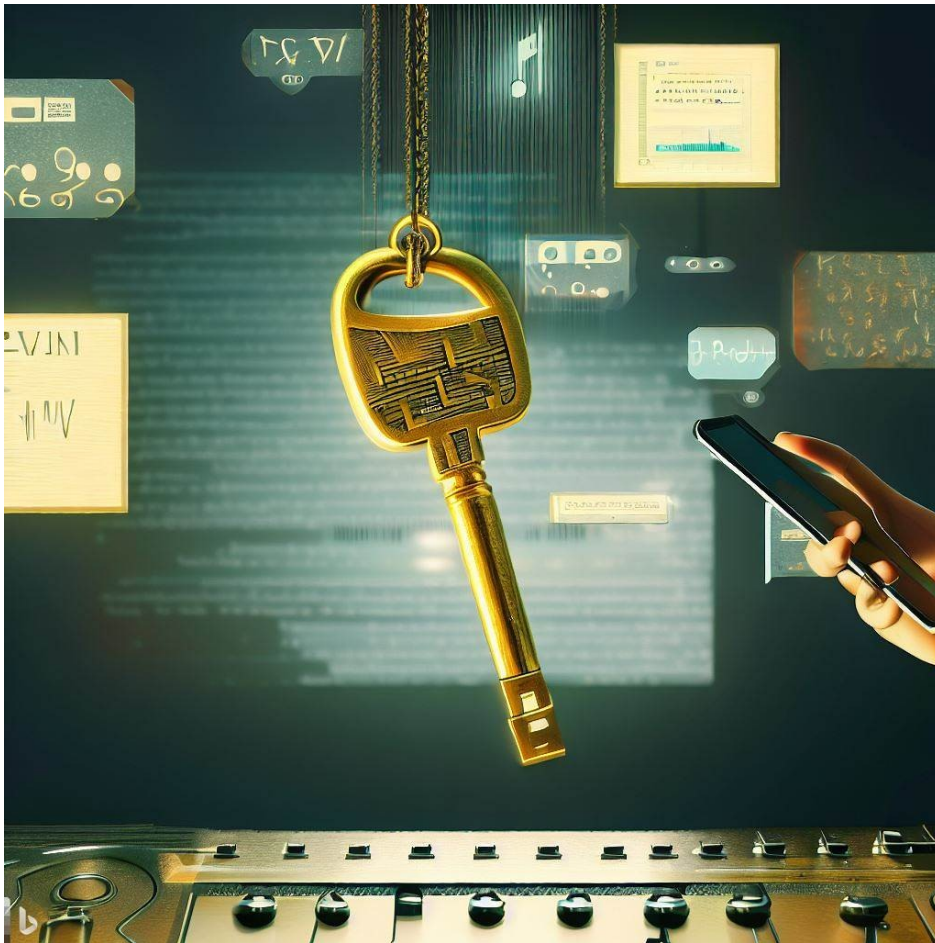
AIS.Save Key

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Save Key

Save Elevenlabs-API Key encrypted to a file



Intention

The `AIS.Save Key` Command is an essential tool for developers venturing into the world of Elevenlabs.

It acts as a secure vault, allowing you to store your API Key in an encrypted format within a file.

This not only bolsters security but also streamlines the process of utilizing the key across various scripts.

Utilizing the Save Key Command

To employ the 'Save Key Command', you need to invoke the `AIC.Save_Key` function.

Pass your API Key as the primary argument.

Additionally, you can specify a file path to determine where the encrypted key should be stored, while this optional.

```
AIS.Save_Key <YOUR_API_KEY> [ |OPTIONAL_FILE_PATH]
```

In the event that a file path is not specified, the command will default to saving the file in the directory where the script or executable is situated.

The default path is "?exeloc\AIS_License_Key.dat"

Default Naming Convention and Location

The encrypted file is conventionally named `AIS_License_Key.dat`.

This standard naming practice ensures easy identification.

If a file path is omitted, the file will be created in the directory denoted by `?exeloc\`, which corresponds to the location of the script or executable.

The Significance of the Save Key Command

Using the `Save Key Command` to store the API Key in an encrypted file is highly advisable.

This approach significantly reduces the risk of unintentional exposure and provides a convenient method for reusing the key in different scripts.

Wrapping Up

The `Save Key Command` is a powerful and indispensable tool for safeguarding your API Key. By storing it in an encrypted file, you ensure its protection and facilitate its use across your projects.

```
' Script 1: Save the Key to the file "AIS_License_Key.dat".
```

```
' IMPORTANT: This API-Key is a phantasy API-Key it must be replaced with your AI
$$KEY=sk-abcdefghijklmnopqrstuvwxy123456
```

```
' Here we save the Keyfile at the default path, that is:
```

```
' ?exeloc\AIS_License_Key.dat
```

```
AIS.Save_Key|$$KEY
```

```
ENR.
```

```
' Script 2; Using the crypted API-Keyfile
```

```
' Test if we are online, AI-Commands will only work if you are online.
```

```
NOL.
```

```
GTO.enx
```

```
EIF.
```

```
' Set Elevenlabs API-Key from the saved File
```

```
AIS.SetKey|from_File
```

```
DBP.$$RET
```

```
:enx
ENR.
```

Syntax

```
AIS . SaveKey | P1 [ | P2 ]
AIS . Svk | P1 [ | P2 ]
```

Parameter Explanation

P1 - [Elevenlabs-API Key](#)^[1000]

P2 - opt. Filepath for the API-Key to save, default is "?

exeloc\AIS_License_Key.dat"

Example

```
!*****
! EXAMPLE 1: AIS.-Commands
!*****
! Script 1: Save the Key to the file "AIS_License_Key.dat".

! IMPORTANT: This API-Key is a phantasy API-Key it must be replaced with your AI
$$KEY=sk-abcdefghijklmnopqrstuvwxy123456

! Here we save the Keyfile at the default path, that is:
! ?exeloc\AIS_License_Key.dat
AIC.Save_Key|$$KEY
ENR.
```

Remarks

-

Limitations:

🛡️ Safeguarding Your API Key: A Knight's Guide to Elevenlabs'

Greetings, Noble Coder! 🏰

Embarking on a quest through the enchanted forests of Elevenlabs?

Before you mount your steed, there's a sacred artifact you must secure - **the illustrious API Key.**

This key is not just a string of characters; it's the heart of your adventure, the magic that unlocks the kingdom's secrets.

🔒 ****The Enchanted Encryption**** 🔒

Ah, you seek to protect your treasure by locking it within an enchanted file.

Wise as it may seem, remember, even the mightiest spells have their counters. The encryption wards off common thieves, but against a sorcerer with the dark SPR arts, it may falter.

🔒 ****Guard Your Treasure Chest**** 🔒

Distributing your encrypted key is akin to leaving your treasure chest in the dragon's lair. Any rogue with an SPR spellbook can break the enchantment.

The treasure within - your API Key - is bound to your very essence, your account. In the wrong hands, it can unleash storms and deplete your resources.

🛡️ ****Set Magical Boundaries**** 🛡️

Fear not, for there is a spell to shield your treasure further.

Within the hallowed halls of Elevenlabs's website, you can weave a spell to set limits on your API Key's powers.

This incantation ensures that even if your key is seized, its magic is bound, and the havoc it can wreak is contained.

🏰 ****The Knight's Code**** 🏰

1. ****Guard the Key****: Never let your API Key, even if enchanted in a file, sail on uncharted waters.

2. ****Summon Guardians****: Create a mystical barrier through a backend service. Let this guardian use the API Key in the shadows, far from prying eyes.

3. ****Eternal Vigilance****: Watch over your domain. Keep an eagle's eye on the usage of your API Key.

4. ****Weave Protective Spells****: Visit Elevenlabs's sacred grounds and set limits upon your API Key.

Bind it, so its magic doesn't turn against the kingdom.

Remember, brave knight, with great power comes great responsibility.

Your API Key is the magic that courses through the veins of your quests.

Guard it, protect it, and let it guide you through countless adventures in the realm of Elevenlabs.

Onward, to glory! 🚀

See also:

-

3.42.12.3.3 Set Key

AIS.SetKey

[Previous](#) 2027 | [Top](#) 49 | [Next](#) 2069

MiniRobotLanguage (MRL)

AIS.SetKey

Set Elevenlabs AI-API Key



Intention

SetKey Command: Initiating Your Script with a License Key

Before diving into the world of AI scripting, it's essential to set the stage with the SetKey command.

This command is the golden ticket that grants you access to the plethora of AI functionalities offered by Elevenlabs.

Let's break down how to use it effectively.

What is the AIS.SetKey Command?

The AIS.SetKey command is the first command you need to include at the beginning of your script.

It's like the key to a treasure chest; without it, you can't unlock the AI capabilities you're after.

Why Do You Need It?

You might be wondering why there's a need for such a key.

The reason is that the AI functionalities you are looking to use are not processed locally on your computer.

Instead, they are handled remotely in the high-powered Elevenlabs Cloud.

This License Key ensures that you have the proper authorization to access these cloud-based services.

How to Obtain the License Key?

To get your hands on this key, you'll need to visit Elevenlabs's official website.

Once there, [follow the instructions to register and obtain the License Key](#)^[1000].

Keep this key safe, as you'll need it every time you want to use Elevenlabs's services.

How to Use the SetKey Command?

Once you have your License Key, it's time to put it to use.

At the very start of your script, include the `AIS.SetKey` command followed by your License Key.

This will authenticate your script with **Elevenlabs's cloud services**.

On the long run you may prefer the second Option (see below) that will encrypt your key into a file and use this file.

The `AIS.SetKey`-Command has multiple Usage Options.

1. You can directly set the Key in the Script.

Here is a Sample Script that shows how this is done.

```
' IMPORTANT: This following API-Key is a phantasy API-Key it must be replaced w
$$KEY=9457609cea9ffe73926f3f7a5c8c6fbf

' Test if we are online, AI-Commands will only work if you are online.
NOL.
  GTO.enx
EIF.

' SetElevenlabs API-Key
AIS.SetKey|$$KEY

' Use the AIS.-Commands here

:enx
ENR.
```

2. You can use a saved, encrypted Key, that is stored in the project-folder.

This is the preferred way because this is more save for your key.

The default name for this saved Key is "AIS_License_Key.dat"

If no path is given, the file is created at "?exeloc\" that is the folder where the Script/the executable resides.

This key-file can be created using the `AIS.Save_Key|$$KEY[|$$FIL]` Command.

```
' Script 1: Save the Key to the file "AIS_License_Key.dat".

' IMPORTANT: This API-Key is a phantasy API-Key it must be replaced with your AI
$$KEY=9457609cea9ffe73926f3f7a5c8c6fbf

' Here we save the Keyfile at the default path, that is:
' ?exeloc\AIS_License_Key.dat
AIS.Save_Key|$$KEY
ENR.

' Script 2; Using the crypted API-Keyfile
' Test if we are online, AI-Commands will only work if you are online.
NOL.
GTO.enx
EIF.

' SetElevenlabs API-Key from the saved File
AIS.SetKey|from_File

' Use the AIS.-Commands

:enx
ENR.
```

3. You can use a Textfile, that contains your Key.

This is way to go, if you make an Executable that you want to share with the public or other people.

To give them the Option to use your SPR-Script together with their own API-Key, you can offer this Option.

The default name for this saved Key in a Textfile is "AIS_License_Key.txt"

If no path is given in **P2**, the file should be located in "?exeloc\" that is the folder where the Script/the executable resides.

This key-file can be created using any Texteditor (or CTF.-Command).

```
' Script 3: Using the API-Keyfile with the API-Key as Text
' Test if we are online, AI-Commands will only work if you are online.
NOL.
GTO.enx
EIF.

' SetElevenlabs API-Key from the saved File
AIS.SetKey|from Text

'Use the AIS.-Commands
:enx
ENR.
```

Syntax

AIS.SetKey|P1 [|P2]

AIS.Set_Key|P1 [|P2]

Parameter Explanation

P1 - Can be directly an [Elevenlabs API-Key](#)^[1000] or:

- from File - if given as **P1**, there should be a encrypted Keyfile with the name "AIC_License_Key.dat" at [?exeloc\](#)^[364] or if specified, at Path **P2**. This file can only be created using the [AIC.Save Key](#)^[762] - Command.
- from Text - if given as **P1**, there should be a Textfile that contains the API-Key with the name "AIC_License_Key.txt" at [?exeloc\](#)^[364] or if specified, at Path **P2**. This file can be created using any Text-Editor.
- Load Any Key - if given as **P1**, the command will load any available API-Key.

P2 - opt. if specified this is the Path (and filename) of the Keyfile to use with the two options above. It can be either a Crypted or a Textfile. The System will decide with the Extension ".txt" or ".dat" how it is loaded.

Example

```

' *****
' EXAMPLE 1: AIS.-Commands
' *****
' IMPORTANT: This API-Key is a phantasy API-Key it must be replaced with your AI
$$KEY=546474356gh3j43h543k3HzpGbZbCq6PeVbSZy69H
' Test if we are online, AI-Commands will only work if you are online.
NOL.
GTO.enx
EIF.

' SetElevenlabs API-Key
AIS.SetKey|$$KEY

' Code

:enx
ENR.

```

Remarks

A Word of Caution: Safeguarding Your API Key 🚫

Dear Valued User,

🔒 **This Encryption is Not Impenetrable** 🔒

Even if your API Key is encrypted within a file, the Decryption is hereby done without a Password, means anybody can decrypt it if he has the SPR at hand. It's akin to a treasure locked in a chest.

The chest provides an added layer of security and invisibility, but should it fall into the hands of a pirate with the SPR and Knowledge, the treasure can be plundered. You may ask: "Why did we not use a Password?" The answer is simple, even then you would need to provide that Password in Cleartext in the code, because the SPR needs to decrypt the keyfile anyway. So there would be no advantage. In case you want to enter the Password each time and have an unbreakable Encryption, you can instead do something using the `GUT` and the `GEC` -Command.

⊗ Do Not Distribute the Key, Even Encrypted ⊗

Distributing executables or scripts along with the file containing your encrypted API Key is akin to sending your treasure chest out to sea on an unmanned ship. Anyone who gets hold of this file and has access to an SPR (Script Processing Runtime) can potentially decrypt and misuse your API Key.

💡 Why is This a Big Deal? 💡

Your API Key is not just a string; it's your identity and access within the Elevenlabs realm. It's linked to your account, your resources, and your privileges. In the wrong hands, it can be used to access services and consume quotas associated with your account. This can have both financial and security implications.

🗣 What Should You Do? 🗣

Never Distribute the Key: Do not include your API Key, even if encrypted, in any files or executables that you distribute.

Access Control: If your application requires the use of the API Key, consider implementing a backend service that your application can call. The service can then use the API Key server-side, where it's not exposed to the end-user.

Vigilance: Regularly monitor the usage of your API Key and be vigilant for any unauthorized or unexpected activity.

Set Usage Limits: Visit the Elevenlabs website and access your account settings. Here, you can set limits on the usage of your API Key. This is a wise precaution to ensure that even if the unthinkable happens, the potential damage is contained. It's like setting a magical barrier around your treasure chest!

Remember, with great power comes great responsibility. Your API Key is a powerful tool; wield it wisely and guard it well.

Safe coding!

Limitations:

-

See also:

•

3.42.12.3.4 Set Latency

[AIS.Set Latency](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Set Latency

Sets latency optimization level.

Intention

The `AIS.Set Latency` command allows you to set the level of latency optimizations for the AIS engine.

This can be useful for applications where low latency is crucial but can come at the cost of reduced voice quality.

The latency optimization level can affect the quality of the voice.

Higher levels of optimization may result in lower voice quality but will reduce latency.

Option 4 may cause mispronunciations of numbers and dates.

Syntax

AIS.Set Latency | P1**AIS.Slt | P1**

Parameter Explanation

- **P1**: The value for setting the latency optimization level.

Value	Description
0	Default mode (no latency optimizations)
1	Normal latency optimizations (about 50% of possible latency improvement of option 3)
2	Strong latency optimizations (about 75% of possible latency improvement of option 3)
3	Max latency optimizations

4	Max latency optimizations with text normalizer turned off (best latency, may mispronounce numbers and dates)
---	--

Example

```
'*****  
' AIS.-Sample  
'*****  
AIS.Set Latency|3  
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.12.3.5 Set Output Format

[AIS.Set Output Format](#)

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Set Output Format

Set the Format of the Output for ElevenLabs.io

Intention

Select the desired Output Format for the Elevenlabs Speech Synthesis. There are some limitations depending on your Payments Plan, details see below.

All these Plans are subject to changes therefore it makes sense to directly visit the WEB-Site and check which Output Formats are included in your plan.

Number	Format	Explanation	License Requirement
1	mp3_44100_64	MP3 format, 44.1 kHz, 64 kbps	None
2	mp3_44100_96	MP3 format, 44.1 kHz, 96 kbps	None

3	mp3_44100_128	MP3 format, 44.1 kHz, 128 kbps	D e f a u l t, N o n e
4	mp3_44100_192	MP3 format, 44.1 kHz, 192 kbps	R e q u i r e s C r e a t o r t i e r o r a b o v e
5	pcm_16000	PCM format, 16 kHz	N o n e
6	pcm_22050	PCM format, 22.05 kHz	N o n e
7	pcm_24000	PCM format, 24 kHz	N o

			n e
8	pcm_44100	PCM format, 44.1 kHz	R e q u i r e s I n d e p e n d e n t P u b l i s h e r t i e r o r a b o v e

Syntax

Set Output Format.P1

Parameter Explanation

P1 - Number that will select the Output-Format.

Example

```
!*****  
! AIS.-Sample  
!*****  
AIS.Set Output Format|3
```

Remarks

-

Limitations:

-

See also:

-

3.42.12.3.6 Set Similarity

[AIS.Set Similarity](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Set Similarity

Sets voice similarity level.

Intention

The `AIS.Set Similarity` command allows you to set the level of similarity the AIS engine should aim for when replicating a voice.

This is useful for fine-tuning the voice output to match the original voice as closely as possible.

The internal "similarity_boost" parameter dictates how closely the AI should adhere to the original voice.

Setting it too high may reproduce artifacts or background noise if those were present in the original recording.

Syntax

AIS.Set Similarity|P1 [|P2]

Parameter Explanation

- **P1:** Required. A floating-point number between 0 and 1 that sets the level of similarity.
- **P2:** Optional. The variable where the result will be stored.

Example

```

*****
' AIS.-Sample
*****
$$SIM=0.75
AIS.Set Similarity|$$SIM|$$RET
MBX.$$RET
ENR.

```

Remarks

-

Limitations:

-

See also:

-

3.42.12.3.7 Set Speaker Boost

[AIS.Set Speaker Boost](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Set Speaker Boost

Boosts speaker similarity.

Intention

The `AIS.Set Speaker Boost` command enhances the similarity between the synthesized voice and the original speaker's voice. This is useful when you want the generated voice to closely match the original. However, using this feature may increase computational load and latency. The internal `"use_speaker_boost"` parameter boosts the similarity to the original speaker. While the differences are generally subtle, using this setting requires a slightly higher computational load, which may increase latency.

Syntax

```
AIS.Set Speaker Boost | P1
AIS.Ssb | P1
```

Parameter Explanation

P1: Required. A boolean value 1 - (True) or 0 - (False) that sets whether to boost the speaker's similarity or not.

Example

```
!*****
! AIS.-Sample
!*****
AIS.Set Speaker Boost|1
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.12.3.8 Set Style

AIS.Set Style

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Set Style

Amplifies speaker's style.

Intention

The `AIS.Set Style` command is designed to amplify the style of the original speaker in the synthesized voice.

This can make the voice more expressive but may consume additional computational resources and increase latency. It may also make the model slightly less stable.

Using the style parameter consumes additional computational resources and might increase latency.

It's important to note that using this setting can make the model less stable as it tries to imitate the style of the original voice.

For more information, refer to the [Elevenlabs Documentation](#).

Syntax

AIS.Set Style|P1 [|P2]**AIS.Sty|P1 [|P2]**

Parameter Explanation

- **P1**: Required. A floating-point number that sets the level of style amplification. Generally recommended to be set at 0.
- **P2**: Optional. Variable for the return value.

Example

```

! *****
! AIS.-Sample
! *****
AIS.Set Style|0.5
ENR.

```

Remarks

-

Limitations:

-

See also:

-

3.42.12.4 AIS.-Voice Settings

Voice-Settings

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

Voice-Settings

Voice Settings that come inside the JSON

```
{
  "text": "string",
  "model_id": "eleven_monolingual_v1",
  "voice_settings": {
    "stability": 0,
    "similarity_boost": 0,
    "style": 0,
    "use_speaker_boost": true
  }
}
```

This is the JSON that is being sent to the Elevenlabs-Server.

3.42.12.4.1 Get Female Voice ID

[AIS.Get Feale Voice ID / AIS.gfv](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Get Female Voice ID / AIS.gfv

Retrieve Voice ID for a Female Voice

Intention

The `AIS.Get Female Voice ID` command retrieves the Voice ID for a specified Female voice.

The voice ID can be specified by a number between 1 and 18 in **P1**.

A voice ID may look like this: "ErXwobaYiN019PkySvjV". It's an internal hash for a voice.

You can directly set a Voice using the Voice-ID with the command:

```
$$VID=ErXwobaYiN019PkySvjV
AIS.set voice id|$$VID
```

Case Number	Name	Voice ID	Description
1	Antoni	ErXwobaYiN019PkySvjV	American, well-rounded
2	Bella	EXAVITQu4vr4xnSDxMaL	American, soft
3	Charlotte	XB0fDUnXU5powFXDhCwa	English-Swedish, seductive
4	Domi	AZnzlk1XvdvUeBnXmllld	American, strong, fem.
5	Dorothy	ThT5KcBeYPX3keUQqHPh	British, pleasant
6	Elli	MF3mGyEYCI7XYWbV9V6O	American, emotional
7	Emily	LcfcDJNUP1GQjkzn1xUU	American, calm
8	Freya	jsCqWAovK2LkecY7zXI4	American, Unknown
9	Gigi	jBpfulE2acCO8z3wKNLI	American, childish, fem

10	Glinda	z9fAnlkpzviPz146 aGWa	American, witch
11	Grace	oWAXZDx7w5VEj 9dCyTzz	American- Southern, Unknown
12	Lena	shZbofzc7bU41z6 VrBth	American, Unknown
13	Matilda	XrExE9yKlg1Wjnnl VkGX	American, warm
14	Mimi	zrHiDhphv9ZnVX BqCLjz	English-Swedish, childish
15	Myra	7sWxe4rdzP2SSStD oCs5H	American, Unknown
16	Nicole	piTKgcLEGmPE4e 6mEKli	American, whisper
17	Rachel	21m00Tcm4TlvDq 8ikWAM	American, calm
18	Serena	pMsXgVXv3BLzUg SXRpIE	American, pleasant

Syntax

**AIS.Get Female Voice ID|P1[|P2]
AIS.Gfv|P1[|P2]**

Parameter Explanation

- **P1:** Numeric input for Male Voice ID
- **P2:** Optional variable for return (if omitted, result is placed on TOS)

Example

```
' *****  
' AIS.-Sample
```

```
!*****  
$$MVID=5  
AIS.Get Female Voice ID|$$NUM|$$VID  
AIS.set voice id|$$VID  
MBX.$$VID  
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.12.4.2 Get Male Voice ID

[AIS.Get Male Voice ID / AIS.gmv](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Get Male Voice ID / AIS.gmv

Retrieve Voice ID for a Male Voice

Intention

The `AIS.Get Male Voice ID` command retrieves the Voice ID for a specified male voice.

The voice ID can be specified by a number between 1 and 23 in **P1**.

A voice ID may look like this: "ErXwobaYiN019PkySvjV". It's an internal hash for a voice.

You can directly set a Voice using the Voice-ID with the command:

```
$$VID=ErXwobaYiN019PkySvjV
AIS.set voice id|$$VID
```

Case Number	Name	Voice ID	Description
1	Adam	pNlnz6obpgDQGcFmaJgB	American, deep
2	Arnold	VR6AewLTigWG4xSOukaG	American, crisp
3	Callum	N2IVS1w4EtoT3dr4eOWO	American, hoarse
4	Charlie	IKne3meq5aSn9XLyUdCD	Australian, casual
5	Clyde	2EiwWnXFvU5JabPnv8n	American, war veteran
6	Daniel	onwK4e9ZLuTAKqWW03F9	British, deep
7	Dave	CYw3kZ02Hs0563khs1Fj	British-Essex, conversational
8	Ethan	g5CljZEefAph4nQFvHAz	American, Unknown
9	Fin	D38z5RcWu1voky8WS1ja	Irish, sailor

10	Harry	SOYHLrjzK2X1ezoPC6cr	American, anxious
11	Indigo	9Ka5mvTU6NwncFO6GI Ot	Indian, Unknown
12	James	ZQe5CZNOzWyzPSCn5a 3c	Australian, calm
13	Jeremy	bVMeCyTHy58xNoL34h 3p	American-Irish, excited
14	Jessie	t0jbNIBVZ17f02VDIeMI	American, raspy
15	Joseph	ZIb1dXrM653N07WRdF W3	British, Unknown
16	Josh	TxGEqnHWrfWFTfGW9X jX	American, deep
17	Liam	TX3LPaxmHKxFdv7VOQ HJ	American, Unknown
18	Matthew	Yko7PKHZNXotIFUBG7I 9	British, Unknown
19	Michael	flq6f7yk4E4fJM5XTYuZ	American, Unknown
20	Patrick	ODq5zmih8GrVes37Diz d	American, shouty
21	Ryan	wViXBPUzp2ZZixB1xQu M	American, soldier
22	Sam	yoZ06aMxZJJ28mfd3PO Q	American, raspy
23	Thomas	GBv7mTt0atlp3Br8iCZE	American, calm

Male Voice ID's with their numbers and names.

Syntax

AIS.Get Male Voice ID|P1[|P2]

Parameter Explanation

- **P1:** Numeric input for Male Voice ID
- **P2:** Optional variable for return (if omitted, result is placed on TOS)

Example

```
!*****  
! AIS.-Sample  
!*****  
$$MVID=5  
AIS.Get Male Voice ID|$$NUM|$$VID  
AIS.set voice id|$$VID  
MBX.$$VID  
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.12.4.3 Get Voice ID

AIS.Get Voice ID[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Get Voice ID

Retrieves Voice ID based on input.

Intention

The `AIS.Get Voice ID` command retrieves the Voice ID for a specified voice. The voice ID can be specified either by its name or by a number between 1 and 42.

A voice Id may look somehow like this: "ErXwobaYiN019PkySvjV". Its an internal Hash for a voice.

You can directly set a Voice using the Voice-ID with the command:

```
' This will set the used Voice to "Antoni"
$$VID=ErXwobaYiN019PkySvjV
AIS.set voice id|$$VID
```

Syntax

```
AIS.Get Voice ID|P1|[P2]
AIS.Gvi|P1|[P2]
```

Parameter Explanation

- **P1**: A string that specifies the voice. It can be either the name of the voice or a number between 1 and 42.
- **P2**: Optional. A variable where the result will be stored. If omitted, the result will be placed on TOS.

Example

```
! *****
' AIS.-Sample
! *****
$$VOI=Rachel
```

```
AIS.Get Voice ID|$$VOI|$$VID
' Use the ID for setting the voice
AIS.Set Voice ID|$$VID
MBX.$$VID
ENR.
```

Remarks

-

Limitations:

-

See also:

- [1.6.1. Program Flow Control](#)¹³⁴
- [! Smart Package Robot 's Parallel Robot Operations](#)³⁷⁴⁵
- [1.5. Features and Hints](#)⁹³

3.42.12.4.4 Get Voices

[AIS.Get Voices / AIS.gvo](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Get Voices / AIS.gvo

Lists available voices.

Intention

The `AIS.Get Voices` command retrieves the available voices from the server. The returned data is in JSON format, which includes details about each voice, such as name, category, and fine-tuning options. It should be best saved as a file and viewed in a JSON Viewer.

```
AIS.Set Key|file

AIS.Get Voices|$$RET
DBP.$$RET
MBX.!
ENR.
```

Syntax

```
AIS.Get Voices [ |P1 ]
AIS.gvo [ |P1 ]
```

Parameter Explanation

P1: Optional. The variable where the JSON data will be stored. If omitted, the data will be placed on TOS.

Example

```
'*****
' AIS.-Sample
'*****
AIS.Set Key|file

AIS.Get Voices|$$RET
DBP.$$RET
MBX.!
```

ENR.

```
{"voices":[{"voice_id":"21m00Tcm4TlvDq8IkWAM","name":"Rachel","samples":null,"category":"premade","fine_tuning":{"language":  
:rification_attempts_count":0,"slice_ids":null,"manual_verification":null,"manual_verification_requested":false},"labels":{"accent":  
riew_url":"https://storage.googleapis.com/eleven-public-prod/premade/voices/AZnzk1XvdyUeBnXmld/508e12d0-a7f7-4d86-a0c  
8z5RcWu1voky8WS1ja","name":"Fin","samples":null,"category":"premade","fine_tuning":{"language":null,"is_allowed_to_fine_tun  
rn_attempts_count":0,"slice_ids":null,"manual_verification":null,"manual_verification_requested":false},"labels":{"accent":"america  
m/eleven-public-prod/premade/voices/ErXwobaYiN019PkySvjV/ee9ac367-91ee-4a56-818a-2bd1a9dbe83a.mp3","available_for_tie  
tegorry":"premade","fine_tuning":{"language":null,"is_allowed_to_fine_tune":false,"fine_tuning_requested":false,"finetuning_sta  
ull,"manual_verification":null,"manual_verification_requested":false},"labels":{"accent":"american","description":"calm","age":"you  
blic-prod/premade/voices/MF3mGyEYCI7XYWbV9V6O/d8ecadea-9e48-4e5d-868a-2ec3d7397861.mp3","available_for_tiers":[],"set  
gory":"premade","fine_tuning":{"language":null,"is_allowed_to_fine_tune":false,"fine_tuning_requested":false,"finetuning_state  
nual_verification":null,"manual_verification_requested":false},"labels":{"accent":"american","description":"anxious","age":"young"},  
premade/voices/TX3LPaxmHKxFdv7VOQHJ/63148076-6363-42db-aea8-31424308b92c.mp3","available_for_tiers":[],"settings":null,"  
mplec":null,"category":"premade","fine_tuning":{"language":null,"is_allowed_to_fine_tune":false,"fine_tuning_requested":false}}
```

Remarks

-

Limitations:

-

See also:

-

3.42.12.4.5 Set Female Voice

[AIS.Set Female Voice](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Set Female Voice

Set Female Voice by Number

Intention

The `AIS.Set Female Voice` command allows you to set the voice to a specific female voice by specifying a number between 1 and 18.

Syntax

```
AIS.Set Female Voice|P1
AIS.Sfv|P1
```

Parameter Explanation

P1: Number from 1 to 18 representing the female voice to be set.

Example

```
'*****
' AIS.-Sample
'*****
$$VID=3
AIS.Set Female Voice|$$VID
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.12.4.6 Set Male Voice

[AIS.Set Male Voice](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Set Male Voice

Set Male Voice by Number

Intention

The `AIS.Set Male Voice` command allows you to set the voice to a specific Male voice by specifying a number between 1 and 23.

Case Number	Name	Voice ID	Description
1	Adam	pNInz6obpgDQGcFmaJgB	American, deep
2	Arnold	VR6AewLTigWG4xSOukaG	American, crisp
3	Callum	N2IVS1w4EtoT3dr4eOWO	American, hoarse
4	Charlie	IKne3meq5aSn9XLYudCD	Australian, casual
5	Clyde	2EiwWnXFvU5JabPnv8n	American, war veteran
6	Daniel	onwK4e9ZLuTAKqWW03F9	British, deep
7	Dave	CYw3kZ02Hs0563khs1Fj	British-Essex, conversational
8	Ethan	g5CljZEefAph4nQFvHAz	American, Unknown
9	Fin	D38z5RcWu1voky8WS1ja	Irish, sailor
10	Harry	SOYHLrjzK2X1ezoPC6cr	American, anxious
11	Indigo	9Ka5mvTU6NwncFO6GIot	Indian, Unknown
12	James	ZQe5CZNOzWyzPSCn5a3c	Australian, calm
13	Jeremy	bVMeCyTHy58xNoL34h3p	American-Irish, excited

14	Jessie	t0jbNlBVZ17f02VDIeMI	American, raspy
15	Joseph	ZIb1dXrM653N07WRdFW3	British, Unknown
16	Josh	TxGEqnHWrfWFTfGW9XjX	American, deep
17	Liam	TX3LPaxmHKxFdv7VOQHJ	American, Unknown
18	Matthew	Yko7PKHZNXotIFUBG7I9	British, Unknown
19	Michael	flq6f7yk4E4fJM5XTYuZ	American, Unknown
20	Patrick	ODq5zmih8GrVes37Dizd	American, shouty
21	Ryan	wViXBPUzp2ZZixB1xQuM	American, soldier
22	Sam	yoZ06aMxZJJ28mfd3POQ	American, raspy
23	Thomas	GBv7mTt0atlp3Br8iCZE	American, calm

Syntax

AIS.Set Male Voice|P1
AIS.Smv|P1

Parameter Explanation

P1: Number from 1 to 18 representing the Male voice to be set.

Example

```

! *****
! AIS.-Sample
! *****
$$VID=3
AIS.Set Male Voice|$$VID
ENR.

```

Remarks

-

Limitations:

-

See also:

-

3.42.12.4.7 Set Voice ID

`AIS.Set Voice ID`[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Set Voice ID

Set Voice ID based on input.

Intention

The `AIS.Set Voice ID` command will set the Voice directly, based on the given Voice ID for a specified voice.

The difference to the `AIS.Set Voice Command` is, that this command directly changes the Voice ID, while `AIS.Set Voice` will accept Names and numbers as Parameter (is more user-friendly).

Therefore prefer `AIS.Set Voice Command` and use this only for special purpose.

A voice Id may look somehow like this: "ErXwobaYiN019PkySvjV".
Its an internal Hash for a voice.

You can directly set a Voice using the Voice-ID with the command:

```
' This will set the used Voice to "Antoni"
$$VID=ErXwobaYiN019PkySvjV
AIS.Set Voice ID|$$VID
```

Here is a list of Voice, and the Voice ID. These voices are "hardcoded" and not updated from the server.

If you want to specify a voice that is **not** available within these **42 Voices**, you need to use the

`AIS.Set Voice ID - Command`.

Number	Name	Voice ID	Description
1	Adam	pNlnz6obpgDQGcFmaJgB	American, deep
2	Antoni	ErXwobaYiN019PkySvjV	American, well-rounded
3	Arnold	VR6AewLTigWG4xSOukaG	American, crisp
4	Bella	EXAVITQu4vr4xnSDxMaL	American, soft
5	Callum	N2lVS1w4EtoT3dr4eOWO	American, hoarse

6	Charlie	IKne3meq5aSn9XLYudC D	Australian, casual
7	Charlotte	XB0fDUnXU5powFXDhC wa	English-Swedish, seductive
8	Clyde	2EiwWnXFvU5JabPnv8n	American, war veteran
9	Daniel	onwK4e9ZLuTAKqWW03 F9	British, deep
10	Dave	CYw3kZ02Hs0563khs1Fj	British-Essex, conversational
11	Domi	AZnzlk1XvdvUeBnXmllD	American, strong
12	Dorothy	ThT5KcBeYPX3keUQqHP h	British, pleasant
13	Elli	MF3mGyEYCI7XYWbV9V 6O	American, emotional
14	Emily	LcfcDJNUP1GQjkn1xUU	American, calm
15	Ethan	g5CljZEefAph4nQFvHAz	American, Unknown
16	Fin	D38z5RcWu1voky8WS1j a	Irish, sailor
17	Freya	jsCqWAovK2LkecY7zXI4	American, Unknown
18	Gigi	jBpfulE2acCO8z3wKNLI	American, childish
19	Giovanni	zcAOhNBS3c14rBihAFp1	English-Italian, foreigner
20	Glinda	z9fAnlKpzviPz146aGwa	American, witch
21	Grace	oWaxZDx7w5VEj9dCyTz z	American-Southern, Unknown
22	Harry	SOYHLrjzK2X1ezoPC6cr	American, anxious
23	Indigo	9Ka5mvTU6NwncFO6Gl Ot	Indian, Unknown
24	James	ZQe5CZNOzWyzPSCn5a 3c	Australian, calm
25	Jeremy	bVMeCyTHy58xNoL34h3 p	American-Irish, excited
26	Jessie	t0jbNIBVZ17f02VDIeMI	American, raspy
27	Joseph	Zlb1dXrM653N07WRdF W3	British, Unknown
28	Josh	TxGEqnHWrfWFTfGW9Xj X	American, deep
29	Lena	shZbofzc7bU41z6VrBth	American, Unknown

30	Liam	TX3LPaxmHKxFdv7VOQHJ	American, Unknown
31	Matilda	XrExE9yKIg1WjnnlVkgX	American, warm
32	Matthew	Yko7PKHZNXotIFUBG7I9	British, Unknown
33	Michael	flq6f7yk4E4fJM5XTYuZ	American, Unknown
34	Mimi	zrHiDhphv9ZnVXBqCLjz	English-Swedish, childish
35	Myra	7sWxe4rdzP2SSStDoCs5H	American, Unknown
36	Nicole	piTKgcLEGmPE4e6mEKli	American, whisper
37	Patrick	ODq5zmih8GrVes37Dizd	American, shouty
38	Rachel	21m00Tcm4TlvDq8ikWAM	American, calm
39	Ryan	wViXBPUzp2ZZixB1xQuM	American, soldier
40	Sam	yoZ06aMxZJJ28mfd3POQ	American, raspy
41	Serena	pMsXgVXv3BLzUgSXRplE	American, pleasant
42	Thomas	GBv7mTt0atlp3Br8iCZE	American, calm

Syntax

```
AIS.Set Voice ID|P1
AIS.Svi|P1
```

Parameter Explanation

- **P1**: A string that contains the Voice ID.

Example

```

*****
' AIS.-Sample
*****
$$VOI=Rachel
AIS.Get Voice ID|$$VOI|$$VID
```

```
' Use the ID for setting the voice
AIS.Set Voice ID|$$VID
MBX.$$VID
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.12.5 Get Any

AIS.Get Any / AIS.gan

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Get Any / AIS.gan

Performs a custom API call.

Intention

This command is particularly useful for advanced users who need to perform [custom operations](#) that are not covered by the standard AIS. commands.

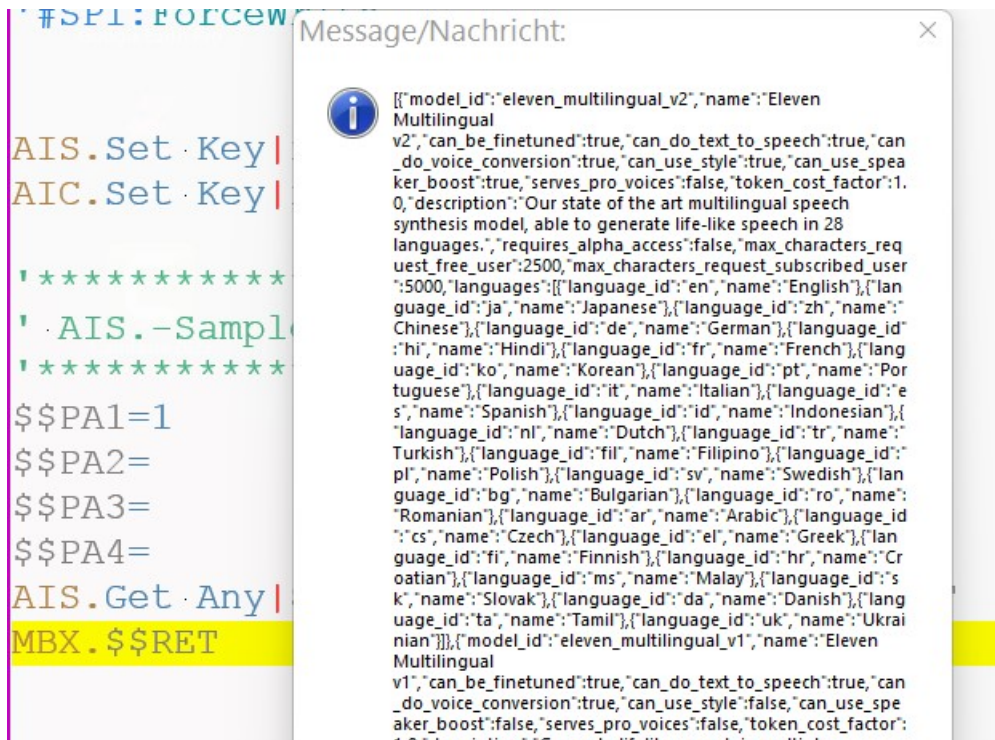
The AIS.Get Any command allows you to perform a custom **API** call to the [Elevenlabs.io Text-to-Speech](#) service.

This command provides a flexible way to interact with the **API** by specifying various parameters and a user-defined JSON body (not needed for some of the options).

```

' *****
' AIS.-Sample
' *****

AIS.Set Key|file
$$PA1=1
$$PA2=
$$PA3=
$$PA4=
AIS.Get Any|$$PA1|$$PA2|$$PA3|$$PA4|$$RET
MBX.$$RET
ENR.
    
```



Here is a complete List of Options for the **P1** Parameter.
For **P1** = 15 you can use a [custom Endpoint-URL](#) that has to be in **P3**.

Number	Text
1	https://api.elevenlabs.io/v1/models

2	https://api.elevenlabs.io/v1/voices
3	https://api.elevenlabs.io/v1/voices/settings/default

4	https://api.elevenlabs.io/v1/voices/{OptionalParam1}/settings

5	https://api.elevenlabs.io/v1/voices/{OptionalParam1}

6	https://api.elevenlabs.io/v1/history

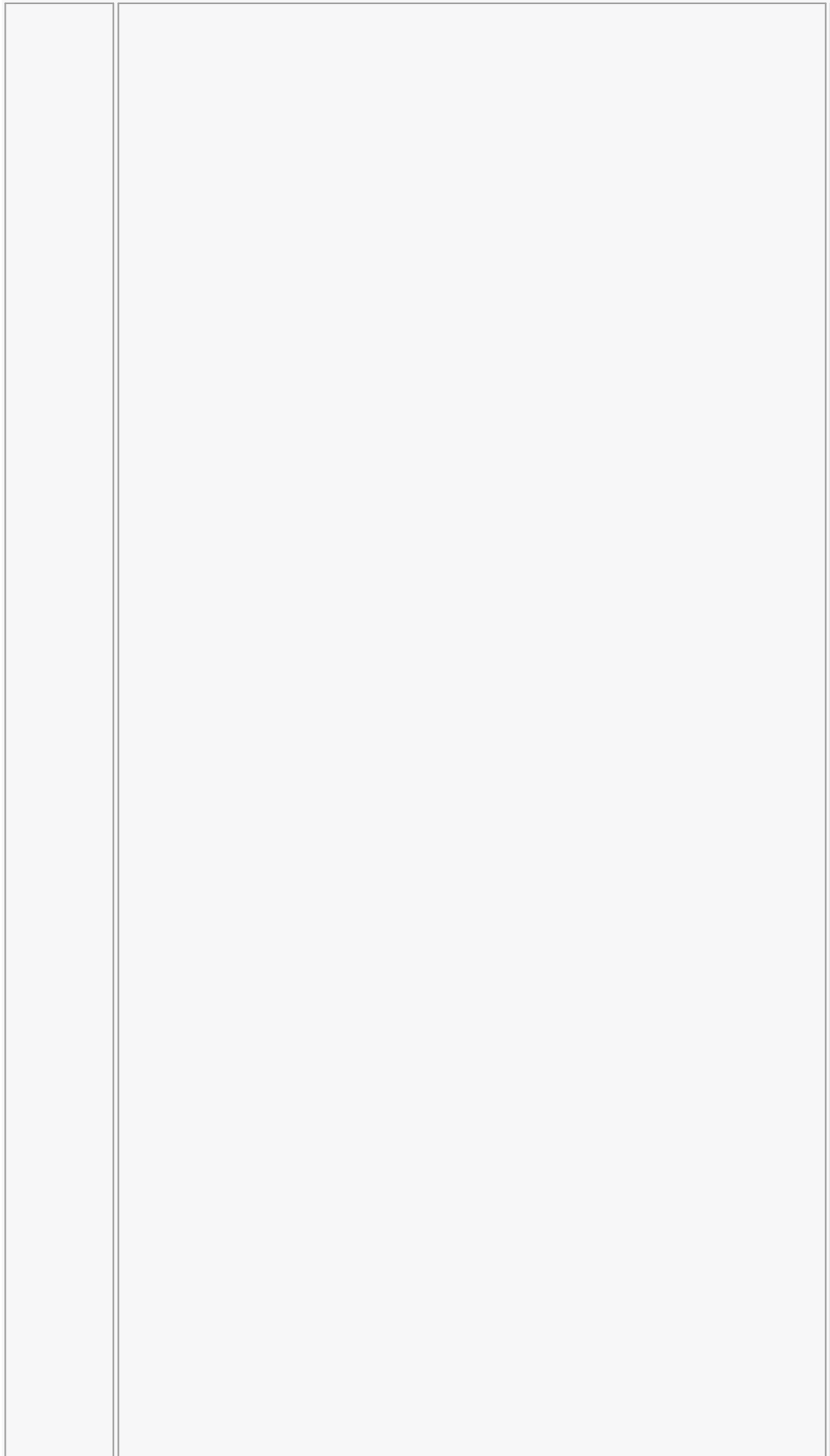
7	https://api.elevenlabs.io/v1/history/{OptionalParam1}

8	https://api.elevenlabs.io/v1/user

9	https://api.elevenlabs.io/v1/user/subscription
10	https://api.elevenlabs.io/v1/projects

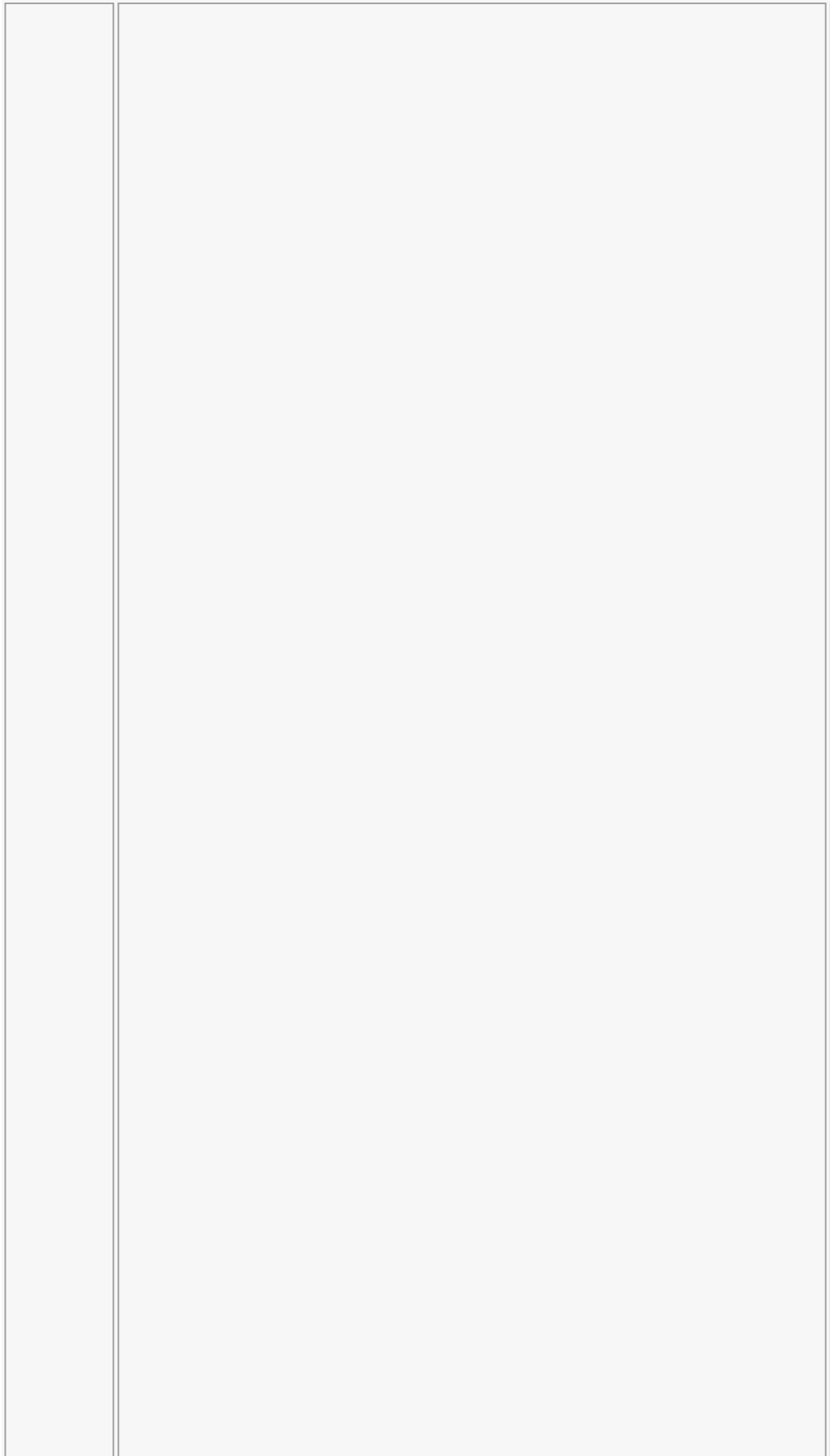
11	https://api.elevenlabs.io/v1/projects/{OptionalParam1}

12	https://api.elevenlabs.io/v1/projects/{OptionalParam1}/snapshots



13	https://api.elevenlabs.io/v1/projects/{OptionalParam1}/chapters

14	https://api.elevenlabs.io/v1/projects/{OptionalParam1}/chapters/{OptionalParam2}



15	{OptionalParam2}

Syntax

```
AIS.Get Any | P1 | P2 | P3 | P4 [ | P5 ]
AIS.gan | P1 | P2 | P3 | P4 [ | P5 ]
```

Parameter Explanation

- **P1**: Required. An integer between 1-15 representing the API endpoint. See table above.
- **P2**: Required or not - depending on **P1**. The first parameter for the API call.
- **P3**: Required or not - depending on **P1**. The second parameter for the API call.
- **P4**: Required or not - depending on **P1**. The JSON body to be sent with the API call. This may be useful for special options like **P1**=15.
- **P5: Optional**. The variable where the API response will be stored. If omitted, TOS is used.

Example

```

' *****
' AIS.-Sample
' *****
AIS.Set Key|file
' Option 6 will give you a complete history of usage
$$PA1=6
$$PA2=
$$PA3=
$$PA4=
AIS.Get Any|$$PA1|$$PA2|$$PA3|$$PA4|$$RET
MBX.$$RET
ENR.
```

Remarks

-

Limitations:

-

See also:



3.42.12.6 Get Data

AIS.Get Data

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Get Data

Retrieves internal AIS engine registers.

Intention

The `AIS.Get Data` command retrieves internal registers from the AIS engine based on the number specified in **P1**.

The result can be stored in an optional variable specified in **P2**.

Value	Returns	Explanation
0	AIS_Output_Format	Returns the AIS output format
1	AIS_Model	Returns the AIS Model
2	ELabs_Voice_ID	Returns the Eleven Labs Voice ID

Syntax

```
AIS.Get Data | P1 [ | P2 ]
AIS.Gda | P1 [ | P2 ]
```

Parameter Explanation

- **P1**: A number from 0 to 2 that specifies which internal registers to retrieve.
- **P2**: Optional. A variable where the result will be stored. If omitted, the result will be placed on TOS.

Example

```
' *****
' AIS.-Sample
```

```
'*****  
$$NUM=1  
' Return AIS_Output_Format  
AIS.Get Data|$$NUM|$$RES  
MBX.$$RES  
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.12.7 Get default

AIS.Get Default

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Get Default

Retrieves default voice settings.

Intention

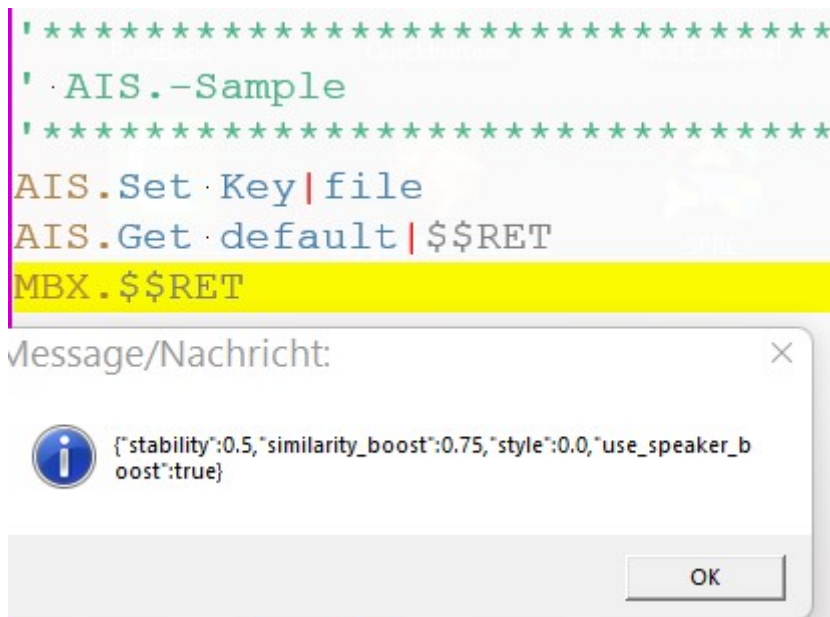
The `AIS.Get Default` command retrieves the default voice settings from the AIS engine.

These settings include various parameters like stability, similarity boost, style, and whether to use speaker boost or not.


```

' *****
' AIS.-Sample
' *****
AIS.Set Key|file
AIS.Get default|$$RET
MBX.$$RET

```



message/Nachricht:

 {"stability":0.5,"similarity_boost":0.75,"style":0.0,"use_speaker_boost":true}

OK

Syntax

AIS.Get Default[|P1]**AIS.Gdf[|P1]**

Parameter Explanation

P1: Optional. The variable where the default voice settings will be stored. If omitted, the result is pushed onto the Top of Stack (TOS).

The returned default settings will be in the following JSON format:

```
{
  "stability": 0.5,
  "similarity_boost": 0.75,
  "style": 0.0,
  "use_speaker_boost": true
}
```

Example

```
'*****
' AIS.-Sample
'*****
AIS.Get Default|$$DEF
MBX.$$DEF
ENR.
```

Remarks

-

Limitations:

-

See also:

- [1.6.1. Program Flow Control](#)^[134]
- [! Smart Package Robot 's Parallel Robot Operations](#)^[3745]
- [1.5. Features and Hints](#)^[93]

3.42.12.8 Get Folder

[AIS.Get Folder / AIS.gfo](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Get Folder / AIS.gfo

Retrieves cache folder path.

Intention

The `AIS.Get Folder` command is used to retrieve the current path of the folder where the caching system stores the generated MP3 files.

The default path is `?exeloc\AIS_Folder\`

Syntax

```
AIS.Get Folder[|P1]
```

```
AIS.gfo[|P1]
```

Parameter Explanation

P1: Optional. The variable where the current cache folder path will be stored. If omitted, the folder path is placed on the Top of Stack (TOS).

Example

```
! *****
! IRS.-Sample
! *****
AIS.Get Folder|$$FOL
MBX.$$FOL
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.12.9 Get Models

[AIS.Get Models / AIS.gmo](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Get Models / AIS.gmo

Fetches available Elevenlabs models.

Intention

The `AIS.Get Models` command retrieves the available models from Elevenlabs. The command can return the models either in a **table format** or in the **original JSON** format from the server.

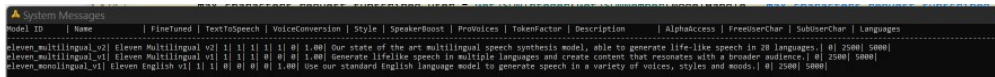
If **P1** is set to **1**, the original **JSON** from the server will be returned. This can be useful for advanced users who need to parse the JSON

If **P1** is set to **0**, the Table Format will be returned, see below.

The Table format looks like this, using the `PRT.-Command`.

```
AIS.Set Key|file

AIS.Get Models|0|$$RET
PRT.$$RET
MBX.!
ENR.
```



Model ID	Name	FineTuned	TextToSpeech	VoiceConversion	Style	SpeakerBoost	Provoices	TokenFactor	Description	AlphaAccess	FreeUserChar	SubUserChar	Languages
eleven_multilingual_v2	Eleven Multilingual v2	1	1	1	1	1	1	1	Our state of the art multilingual speech synthesis model, able to generate life-like speech in 29 languages.	0	3500	5000	
eleven_multilingual_v1	Eleven Multilingual v1	1	1	1	1	1	1	1	Generate lifelike speech in multiple languages and create content that resonates with a broader audience.	0	2500	5000	
eleven_english_v1	Eleven English v1	1	1	1	1	1	1	1	Use our standard English language model to generate speech in a variety of voices, styles and moods.	0	2500	5000	

Syntax

AIS.Get Models [|P1] [|P2]**AIS.gmo [|P1] [|P2]**

Parameter Explanation

- **P1**: Optional. 0 or 1. A value of 0 will return the available models in a table format, while 1 will return the original JSON from the server.
- **P2**: Optional. Placeholder for future use.

Example

```

!*****
! AIS.-Sample
!*****
AIS.Get Models|0|$$MOD
PRT.$$MOD
MBX.Wait
ENR.

```

Sample JSON Data

Here is a sample of the JSON data returned when **P1** is set to **1**:

```

[
  {
    "model_id": "eleven_multilingual_v2",
    "name": "Eleven Multilingual v2",
    "can_be_finetuned": true,
    "can_do_text_to_speech": true,
    "can_do_voice_conversion": true,
    "can_use_style": true,
    "can_use_speaker_boost": true,
    "serves_pro_voices": false,
    "token_cost_factor": 1.0,
    "description": "Our state of the art multilingual speech synthesis model, aka",
    "requires_alpha_access": false,
    "max_characters_request_free_user": 2500,
    "max_characters_request_subscribed_user": 5000,
    "languages": [
      {"language_id": "en", "name": "English"},
      {"language_id": "ja", "name": "Japanese"},
      {"language_id": "zh", "name": "Chinese"},
      {"language_id": "de", "name": "German"},
      {"language_id": "hi", "name": "Hindi"},
      {"language_id": "fr", "name": "French"},
      {"language_id": "ko", "name": "Korean"},
      {"language_id": "pt", "name": "Portuguese"},
      {"language_id": "it", "name": "Italian"},
      {"language_id": "es", "name": "Spanish"},
      {"language_id": "id", "name": "Indonesian"},
      {"language_id": "nl", "name": "Dutch"},
      {"language_id": "tr", "name": "Turkish"},
      {"language_id": "fil", "name": "Filipino"},
      {"language_id": "pl", "name": "Polish"},
      {"language_id": "sv", "name": "Swedish"},
      {"language_id": "bg", "name": "Bulgarian"},
      {"language_id": "ro", "name": "Romanian"},
      {"language_id": "ar", "name": "Arabic"},
      {"language_id": "cs", "name": "Czech"},
      {"language_id": "el", "name": "Greek"},
      {"language_id": "fi", "name": "Finnish"},
      {"language_id": "hr", "name": "Croatian"},
      {"language_id": "ms", "name": "Malay"},
      {"language_id": "sk", "name": "Slovak"},
      {"language_id": "da", "name": "Danish"},
      {"language_id": "ta", "name": "Tamil"},
      {"language_id": "uk", "name": "Ukrainian"}
    ]
  }
]

```

```

    ]
  },
  {
    "model_id": "eleven_multilingual_v1",
    "name": "Eleven Multilingual v1",
    "can_be_finetuned": true,
    "can_do_text_to_speech": true,
    "can_do_voice_conversion": true,
    "can_use_style": false,
    "can_use_speaker_boost": false,
    "serves_pro_voices": false,
    "token_cost_factor": 1.0,
    "description": "Generate lifelike speech in multiple languages and create co",
    "requires_alpha_access": false,
    "max_characters_request_free_user": 2500,
    "max_characters_request_subscribed_user": 5000,
    "languages": [
      {"language_id": "en", "name": "English"},
      {"language_id": "de", "name": "German"},
      {"language_id": "pl", "name": "Polish"},
      {"language_id": "es", "name": "Spanish"},
      {"language_id": "it", "name": "Italian"},
      {"language_id": "fr", "name": "French"},
      {"language_id": "pt", "name": "Portuguese"},
      {"language_id": "hi", "name": "Hindi"},
      {"language_id": "ar", "name": "Arabic"}
    ]
  },
  {
    "model_id": "eleven_monolingual_v1",
    "name": "Eleven English v1",
    "can_be_finetuned": true,
    "can_do_text_to_speech": true,
    "can_do_voice_conversion": false,
    "can_use_style": false,
    "can_use_speaker_boost": false,
    "serves_pro_voices": false,
    "token_cost_factor": 1.0,
    "description": "Use our standard English language model to generate speech :
    "requires_alpha_access": false,
    "max_characters_request_free_user": 2500,
    "max_characters_request_subscribed_user": 5000,
    "languages": [
      {"language_id": "en", "name": "English"}
    ]
  }
]

```

Remarks

The model will automatically identify the written language and use the set parameters to generate speech in it.

Limitations:

-

See also:

-

3.42.12.1(Play MP3

[AIS.PlayMP3 / AIS.pmp](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.PlayMP3 / AIS.pmp

Plays specified MP3 file.

Intention

Conditional Statement.

The `AIS.PlayMP3` command plays an MP3 file specified by the parameter **P1**. If **P1** is omitted, the last played file will be played again.

Using this command you can immediately play the mp3-files that have been delivered from AIS.

Alternatively you can use the [MPA. etc.-Commands](#).

Syntax

```
AIS.PlayMP3 [ | P1 ]
AIS.pmp [ | P1 ]
```

Parameter Explanation

P1: Optional. The file name of the MP3 file to play. If omitted, the last played file will be replayed.

Example

```
!*****
! AIS.-Sample
!*****
$$FIL=?path\MySong.mp3
AIS.PlayMP3|$$FIL
```

Remarks

-

Limitations:

-

See also:

-

3.42.12.1:Post Any

AIS.Post Any

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Post Any

Executes custom POST requests to Elevenlabs.io API.

Intention

This command is particularly useful for **advanced users** who need to perform [custom operations](#) that are not covered by the standard AIS. commands.

The AIS.Post Any command is used to execute **POST** requests to various endpoints of the **Elevenlabs.io API**.

The output will always be in a file that is to be specified in **P5**.

The command takes up to 6 parameters, with **P1** specifying the endpoint number (1-9) to which the **POST** request will be sent.

Number	Endpoint	Description
1	/v1/text-to-speech/{voice_id}	Text To Speech P2= voice_id
2	/v1/text-to-speech/{voice_id}/stream	Text To Speech Stream P2= voice_id
3	/v1/voices/{voice_id}/settings/edit	Edit Voice Settings P2= voice_id
4	/v1/voices/add	Add Voice
5	/v1/voices/{voice_id}/edit	Edit Voice P2= voice_id
6	/v1/history/download	Download History Items
7	/v1/projects/add	Add Project
8	/v1/projects/{project_id}/convert	Convert Project P2= project_id
9	/v1/projects/{project_id}/chapters/{chapter_id}/convert	Convert Chapter P2= project_id P3= chapter_id
10	Custom URL in OptionalParam2	User Defined Endpoint

```
' Sample calling Endpoint 1
' will create an mp3-file as result
'
'*****
' AIS.-Sample
'*****
AIS.Set Key|file
```

```

' Prepare JSON
$$TXT=Hallo how are you today
VAR.$$PJ1=eleven_multilingual_v2
VFP.$$PJ2=0.5
VFP.$$PJ3=0.75
VFP.$$PJ4=0
VFP.$$PJ5=1
AIS.Generate Json|$$TXT|$$PJ1|$$PJ2|$$PJ3|$$PJ4|$$PJ5|$$JSO
' See used JSON in Editor
DBP.$$JSO

' Endpoint to call
VIN.$$PA1=1

' Get Voice ID to $$PA2
AIS.Get Any Voice Id|2|$$PA2

' Define Output File
VAF.$$OUF=?exeloc\Outfile.mp3

AIS.Post Any|$$PA1|$$PA2|$$PA3|$$JSO|$$OUF|$$RET
MBX.$$RET
ENR.

*****
' AIS.-Sample 2
' Here we define the JSON manually that gives you perfect flexibility
*****
AIS.Set Key|file
$$TXT=Hallo how are you today
VAR.$$PJ1=eleven_multilingual_v2
VAR.$$PJ5=0
$$JSO={  "text": "$$TXT",
$$JSO+  "model_id": "$$PJ1",
$$JSO  "voice_settings": {
$$JSO    "stability": 0,
$$JSO    "similarity_boost": $$PJ5,
$$JSO    "use_speaker_boost": true
$$JSO  }}
VIN.$$PA1=1
AIS.Get Any Voice Id|2|$$PA2
VAF.$$OUF=?exeloc\Outfile.mp3
AIS.Post Any|$$PA1|$$PA2|$$PA3|$$JSO|$$OUF|$$RET
' Please see Outfile.mp3
MBX.$$RET
ENR.

```

Syntax

AIS.Post Any|P1|P2|P3|P4[|P5]
AIS.Pan|P1|P2|P3|P4[|P5]

Parameter Explanation

- **P1:** Required. An integer between 1-9 representing the API endpoint. See table above.
- **P2:** Required or not - depending on **P1**. The first parameter for the API call.
- **P3:** Required or not - depending on **P1**. The second parameter for the API call.
- **P4:** Required or not - depending on **P1**. The JSON body to be sent with the API call. This may be useful for special options like **P1=15**.
- **P5: Optional** Filename for result. if omitted or empty, a temporary file is created, the filename is on TOS.
- **P6: Optional.** The variable where the API response will be stored. If omitted, TOS is used.

Please do not use binary returns (like wav or mp3 formats) that may be in P6, these are invalid due to technical limitations.

Always use the result from the File in **P5**.

The content in **P6** may vary, depending on the Endpoint and if there is an error or not, **P6** may contain an message with an error, but may also contain invalid binary data.

Example

```
' Sample calling Endpoint 1
' will create an mp3-file as result
'
'*****
' AIS.-Sample
'*****
AIS.Set Key|file
' Prepare JSON
$$TXT=Hallo how are you today
VAR.$$PJ1=eleven_multilingual_v2
VFP.$$PJ2=0.5
VFP.$$PJ3=0.75
VFP.$$PJ4=0
VFP.$$PJ5=1
AIS.Generate Json|$$TXT|$$PJ1|$$PJ2|$$PJ3|$$PJ4|$$PJ5|$$JSO
' See used JSON in Editor
DBP.$$JSO

' Endpoint to call
VIN.$$PA1=1

' Get Voice ID to $$PA2
AIS.Get Any Voice Id|2|$$PA2

' Define Output File
VAF.$$OUF=?exeloc\Outfile.mp3

' Generate an MP3-File like with the commands that are easier for that purpose.
AIS.Post Any|$$PA1|$$PA2|$$PA3|$$JSO|$$OUF|$$RET
MBX.$$RET
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.12.1 Say Text

[AIS.Say Text](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Say Text

Converts text to spoken MP3 (in a file) and plays it back immediately.

Intention

The `AIS.Say Text` command is designed to convert text into an MP3 audio file using cloud services from [Elevenlabs.io](https://elevenlabs.io) and then Play this Audio-File back. Technically this command is a combination of `AIS.Text to MP3` and `AIS.Play MP3` in one command.

This command requires an [API key](#), which must be set using the `AIS.Set Key` command.

Hint: You can speak in many different languages. The model will automatically identify the written language and use the set parameters to generate speech in it. To find out if the selected language is supported in that voice-model, use the [AIS-Get Models](#)-command.

Generally there are 28 Languages supported, for example:

English, Japanese, Chinese, German, Hindi, French, Korean, Portuguese, Italian, Spanish, Indonesian, Dutch, Turkish, Filipino, Polish, Swedish, Bulgarian, Romanian, Arabic, Czech, Greek, Finnish, Croatian, Malay, Slovak, Danish, Tamil, Ukrainian

The Caching System

There is a Caching System built into this command. Here's how it works and why it's beneficial:

In the observation of standard conversational patterns, it becomes evident that certain words and phrases are recurrently employed.

Given that the utilization of the Elevenlabs Cloud incurs a cost, it is judicious to implement a caching mechanism.

When the Smart Package Robot identifies the repetition of specific words or phrases, it retrieves these elements from the cache rather than initiating a redundant request to the Elevenlabs Cloud.

It should be noted that this caching feature is optional; specifying a filename will bypass the mechanism altogether.

The employment of a caching mechanism offers the distinct advantage of **expedited language availability** and of course lower cost, compared to awaiting MP3 delivery from the Elevenlabs Cloud. However, it's important to acknowledge a minor drawback.

The Elevenlabs AI is designed to never vocalize the exact same sentence in an identical manner. Consequently, bypassing the cache can lend a more authentic feel to the conversation.

On the other hand, repeated use of the cache may result in **noticeable uniformity** when the same sentence is audibly identical over time.

Ultimately, the choice of whether to utilize this feature rests with the user, allowing for customization based on individual preferences.

There is also a way to have the Smart Package Robot delete the actual recording and generate it new via Cloud.

If you do this:

```
AIS.Say Text|$$TXT|-
```

Then the Script will just re-generate the saved mp3-file with a new version, and save the new version in the Cache.

If **P2** is omitted, the system will automatically generate a **cache folder** to store all MP3 files along with a checksum. This feature helps in reducing costs and saving resources by reusing the same MP3 file for repeated phrases.

Note: The system is case-sensitive, meaning the same phrase with different cases will generate different checksums. Be mindful of this when using the command.

This is intentional to keep compatibility with special Commands that may need upper and lowercase characters.

Efficiency and Speed: The built-in caching system serves multiple purposes, primarily aimed at optimizing resource usage and reducing costs.

When the same text is converted to MP3 multiple times, the system retrieves the already generated MP3 file from the cache instead of making a new API call to Elevenlabs.io. This speeds up the process significantly.

Cost-Effectiveness: API calls usually come with a cost. By utilizing a caching system, you minimize the number of API calls made to Elevenlabs.io, thereby saving money.

Resource Optimization: Generating an MP3 from text consumes computational resources. Caching allows the system to avoid redundant operations, thus saving CPU cycles and memory usage.

Customization and Control: The commands AIS.Set Folder and AIS.Get Folder allow you to specify the directory where the cached MP3 files and their checksums are stored. This gives you control over the organization of these files, making it easier to manage them.

Case Sensitivity: The system intentionally does not alter the case of the text when generating the checksum. This means that the same text with different casing will be treated as different phrases, each with its own cached MP3. This feature allows for precise control but also means you should be mindful of text casing to maximize the benefits of caching.

In summary, the built-in caching system is designed to make the command more efficient, cost-effective, and user-friendly.

The Location of the default cache folder is: "?exeloc\AIS_Folder\"

```
AIS.Set Key|<YourAPIKeyHere>
AIS.Say Text|Hello World|?path/to/save.mp3
```

In this example, the text "Hello World" will be converted into an MP3 file and saved in the specified path. Then the command will Play back the MP3-File and SAY the Phrase.

Syntax

```
AIS.Say Text|P1 [|P2]
AIS.Say|P1 [|P2]
```

Parameter Explanation

- **P1**: The text you want to convert into an MP3 file.
- **P2**: Optional. The path where the generated MP3 file will be saved. The caching system will automatically be used if **P2** is omitted or empty.

If **P2** is omitted, the system will automatically generate a cache folder to store all MP3 files along with a checksum.

This feature helps in reducing costs and saving resources by reusing the same MP3 file for repeated phrases.

Note: The system is case-sensitive, meaning the same phrase with different cases will generate different checksums.

The use of different voices is not part of the caching, means the cache knows only the Text, not which voice it is.

To use the caching with different voices and cache them separately, you will need to Change the Cache Folder location using the `AIS.Set Folder` - Command.

Example

```

' *****
' AIS.-Sample
' Demonstration of speaking in many languages
' *****
MBX.We start with Windows native speaking

$$TXT=Hello, this is Sarah from Tech Support.
SAY.$$TXT|w

$$TXT=Hi Sarah, I'm having some issues with my computer.
SAY.$$TXT|w

$$TXT=Buenas tardes, soy Sarah del departamento de asistencia técnica.
SAY.$$TXT|w

$$TXT=Buenas tardes, Sarah. Mi ordenador no funciona bien; se bloquea todo el t
SAY.$$TXT|w

MBX.Now with the Elevenlabs Cloud Technology
```



```
:new
AIS.Set Key|file
$$TXT=Hello, this is Sarah from Tech Support.
AIS.Say Text|$$TXT|-

$$TXT=Hi Sarah, I'm having some issues with my computer.
AIS.Say Text|$$TXT

$$TXT=Buenas tardes, soy Sarah del departamento de asistencia técnica.
AIS.Say Text|$$TXT

$$TXT=Buenas tardes, Sarah. Mi ordenador no funciona bien; se bloquea todo el t
AIS.Say Text|$$TXT

$$TXT=Bonjour, c'est Sarah du support technique.
AIS.Say Text|$$TXT

$$TXT=Bonjour Sarah, j'ai quelques problèmes avec mon ordinateur.
AIS.Say Text|$$TXT

$$TXT=Ciao, sono Sarah del supporto tecnico.
AIS.Say Text|$$TXT

$$TXT=Ciao Sarah, sto avendo alcuni problemi con il mio computer.
AIS.Say Text|$$TXT

$$TXT=Guten Tag, hier ist Sarah vom technischen Support.
AIS.Say Text|$$TXT

$$TXT=Guten Tag Sarah, wir haben einige Probleme mit dem Computer.
AIS.Say Text|$$TXT

MBX.!
ENR.
```

Remarks

- The API key must be set using the **AIS.Set Key** command before using this command.
- The system will use a cache to save resources **only if P2 is omitted or empty**.

Limitations:

-

See also:

-

3.42.12.1: Set Folder

AIS.Set Folder / AIS.sef[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Set Folder / AIS.sef

Sets cache folder path.

Intention

The `AIS.Set Folder` command sets the folder where the caching system will store the generated MP3 files.

If `P1` is omitted it will reset the folder path to the default value, that is "`?exeloc\AIS_Folder\`"

Syntax

```
AIS.Set Folder [|P1]
AIS.Sef [|P1]
```

Parameter Explanation

P1: Optional. The path to the folder where MP3 files will be cached. If omitted, the default folder `?exeloc\AIS_Folder\` will be used.

Example

```
! *****
! AIS.-Sample
! *****
$$FOL=?exeloc\Voice_02_Cache\
AIS.Set Folder|$$FOL
```

Remarks

- If the specified folder does not exist, it will be created.

- Files in the folder may be overwritten without warning.

Limitations:

-

See also:

-

3.42.12.1 Set Stability

[AIS.Set Stability](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Set Stability

If Result available

Intention

The `AIS.Set Stability` command sets the stability level for the voice in the AIS engine.

The stability parameter controls the randomness and emotional range of the voice.

The stability parameter influences the voice's emotional range.

Lower values can make the voice sound overly random and may cause the character to speak too quickly.

Higher values can make the voice sound monotonous and devoid of emotion.

Syntax

```
AIS.Set Stability|P1 [|P2]
AIS.Sst|P1 [|P2]
```

Parameter Explanation

- **P1:** The value for setting the stability of the voice. It's a floating-point number between 0 and 1. Lower values introduce more randomness and emotional range, while higher values make the voice more monotonous.
- **P2:** Optional. The variable where the result of the operation will be stored. If omitted, the result is pushed onto the Top of Stack (TOS).

Example

```
! *****
! AIS.-Sample
! *****
AIS.Set Stability|0.5|$$STA
MBX.$$STA
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.12.1!Set Voice

AIS.Set Voice

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Set Voice

Set the voice for the speech synthesis.

Intention

Using the `AIS.Set Voice` command, you can choose the voice that is to be used with the ElevenLabs speech synthesis system.

You can specify a number or you can specify the name of the voice that you want to choose.

Number	Name	Voice ID	Description
1	Adam	pNlnz6obpgDQGcFmaJgB	American, deep
2	Antoni	ErXwobaYiN019PkySvjV	American, well-rounded
3	Arnold	VR6AewLTigWG4xSOukaG	American, crisp
4	Bella	EXAVITQu4vr4xnSDxMaL	American, soft
5	Callum	N2IVS1w4EtoT3dr4eOWO	American, hoarse
6	Charlie	IKne3meq5aSn9XLyUdCD	Australian, casual
7	Charlotte	XB0fDUnXU5powFXDhCwa	English-Swedish, seductive
8	Clyde	2EiwWnXFnvU5JabPnv8n	American, war veteran
9	Daniel	onwK4e9ZLuTAKqWW03F9	British, deep
10	Dave	CYw3kZ02Hs0563khs1Fj	British-Essex, conversational
11	Domi	AZnzk1XvdvUeBnXmIld	American, strong
12	Dorothy	ThT5KcBeYPX3keUQqHPH	British, pleasant
13	Elli	MF3mGyEYCI7XYWbV9V6O	American, emotional
14	Emily	LcfcDJNUP1GQjkn1xUU	American, calm
15	Ethan	g5CljZEefAph4nQFvHAz	American, Unknown
16	Fin	D38z5RcWu1voky8WS1ja	Irish, sailor
17	Freya	jsCqWAovK2LkecY7zXl4	American, Unknown
18	Gigi	jBpfuLE2acCO8z3wKNLI	American, childish
19	Giovanni	zcAOhNBS3c14rBihAFp1	English-Italian, foreigner

20	Glinda	z9fAnlkpzviPz146aGWa	American, witch
21	Grace	oWAXZDx7w5VEj9dCyTzz	American-Southern, Unknown
22	Harry	SOYHLrjzK2X1ezoPC6cr	American, anxious
23	Indigo	9Ka5mvTU6NwncFO6GI0t	Indian, Unknown
24	James	ZQe5CZNOzWyzPSCn5a3c	Australian, calm
25	Jeremy	bVMcCyTHy58xNoL34h3p	American-Irish, excited
26	Jessie	t0jbNIBVZ17f02VDIeMI	American, raspy
27	Joseph	Z1b1dXrM653N07WRdFW3	British, Unknown
28	Josh	TxGEqnHWrfWFTfGW9XjX	American, deep
29	Lena	shZbofzc7bU41z6VrBth	American, Unknown
30	Liam	TX3LPaxmHKxFdv7VOQHJ	American, Unknown
31	Matilda	XrExE9yKIg1WjnnlVKGX	American, warm
32	Matthew	Yko7PKHZNXotIFUBG7I9	British, Unknown
33	Michael	flq6f7yk4E4fJM5XTYuZ	American, Unknown
34	Mimi	zrHiDhphv9ZnVXBqCLjz	English-Swedish, childish
35	Myra	7sWxe4rdzP2SSStDoCs5H	American, Unknown
36	Nicole	piTKgcLEGmPE4e6mEKli	American, whisper
37	Patrick	ODq5zmih8GrVes37Dizd	American, shouty
38	Rachel	21m00Tcm4TlvDq8ikWAM	American, calm
39	Ryan	wViXBPUzp2ZZixB1xQuM	American, soldier
40	Sam	yoZ06aMxZJJ28mfd3POQ	American, raspy
41	Serena	pMsXgVXv3BLzUgSXRpIE	American, pleasant
42	Thomas	GBv7mTt0atlp3Br8iCZE	American, calm

Syntax

AIS.Set Voice [|P1]

AIS.Svo [|P1]

Parameter Explanation

P1 - (*optional*) is the voice that you want to choose, most easy, a number between 1 and 42.
Or this can be the name of the voice that you want to choose. If you omit **P1**, the default voice will be chosen.

Example

```
!*****  
! AIS.-Sample  
!*****  
! Chose Bella  
AIS.Set Voice  
  
! Choose Antoni  
AIS.svo|4
```

Remarks

-

Limitations:

-

See also:

- [1.6.1. Program Flow Control](#)^[134]
- [! Smart Package Robot 's Parallel Robot Operations](#)^[3745]
- [1.5. Features and Hints](#)^[93]

3.42.12.1 Text to MP3

[AIS.Text to MP3](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

AIS.Text to MP3

Converts text to spoken MP3 (in a file).

Intention

The `AIS.Text to MP3` command is designed to convert text into an MP3 audio file using cloud services from [Elevenlabs.io](https://elevenlabs.io).

This command requires an [API key](#), which must be set using the `AIS.Set Key` command.

Hint: The model will automatically identify the written language and use the set parameters to generate speech in it.

Generally there are 28 Languages supported, for example:

English, Japanese, Chinese, German, Hindi, French, Korean, Portuguese, Italian, Spanish, Indonesian, Dutch, Turkish, Filipino, Polish, Swedish, Bulgarian, Romanian, Arabic, Czech, Greek, Finnish, Croatian, Malay, Slovak, Danish, Tamil, Ukrainian

To find out if the selected language is supported in that voice-model, use the [AIS-Get Models](#) command.

The Caching System

There is a Caching System built into this command. Here's how it works and why it's beneficial:

In the observation of standard conversational patterns, it becomes evident that certain words and phrases are recurrently employed.

Given that the utilization of the Elevenlabs Cloud **incurs a cost**, for each spoken word, it is judicious to implement a **caching mechanism**.

When the Smart Package Robot identifies the repetition of specific words or phrases, it retrieves these elements from the cache rather than initiating a redundant request to the Elevenlabs Cloud.

It should be noted that this caching feature is optional; specifying a filename will bypass the mechanism altogether.

The employment of a caching mechanism offers the distinct advantage of expedited language availability compared to awaiting MP3 delivery from the Elevenlabs Cloud. However, it's important to acknowledge a minor drawback. The Elevenlabs AI is designed to never vocalize the exact same sentence in an identical manner.

Consequently, bypassing the cache can lend a more authentic feel to the conversation.

On the other hand, repeated use of the cache may result in noticeable uniformity when the same sentence is audibly identical over time.

Ultimately, the choice of whether to utilize this feature rests with the user, allowing for customization based on individual preferences.

There is also a way to have the Smart Package Robot delete the actual recording and generate it new via Cloud.

If you do this:

```
AIS.Say Text|$$TXT|-
```

Then the Script will just re-generate the saved mp3-file with a new version, and save the new version in the Cache.

If **P2** is omitted, the system will automatically generate a **cache folder** to store all MP3 files along with a checksum. This feature helps in reducing costs and saving resources by reusing the same MP3 file for repeated phrases.

Note: The system is case-sensitive, meaning the same phrase with different cases will generate different checksums. Be mindful of this when using the command.

This is intentional to keep compatibility with special Commands that may need upper and lowercase characters.

Efficiency and Speed: The built-in caching system serves multiple purposes, primarily aimed at optimizing resource usage and reducing costs.

When the same text is converted to MP3 multiple times, the system retrieves the already generated MP3 file from the cache instead of making a new API call to Elevenlabs.io. This speeds up the process significantly.

Cost-Effectiveness: API calls usually come with a cost. By utilizing a caching system, you minimize the number of API calls made to Elevenlabs.io, thereby saving money.

Resource Optimization: Generating an MP3 from text consumes computational resources. Caching allows the system to avoid redundant operations, thus saving CPU cycles and memory usage.

Customization and Control: The commands AIS.Set Folder and AIS.Get Folder allow you to specify the directory where the cached MP3 files and their checksums are stored. This gives you control over the organization of these files, making it easier to manage them.

Case Sensitivity: The system intentionally does not alter the case of the text when generating the checksum. This means that the same text with different casing will be treated as different phrases, each with its own cached MP3. This feature allows for precise control but also means you should be mindful of text casing to maximize the benefits of caching.

In summary, the built-in caching system is designed to make the command more efficient, cost-effective, and user-friendly.

The Location of the default cache folder is: "?exeloc\AIS_Folder\"

Syntax

```
AIS.Text to MP3|P1[|P2]
```

AIS . tmp | P1 [| P2]

Parameter Explanation

- **P1**: The text you want to convert into an MP3 file.
- **P2**: Optional. The path where the generated MP3 file will be saved. The caching system will automatically be used if **P2** is omitted or empty.

If **P2** is omitted, the system will automatically generate a cache folder to store all MP3 files along with a checksum.

This feature helps in reducing costs and saving resources by reusing the same MP3 file for repeated phrases.

Note: The system is case-sensitive, meaning the same phrase with different cases will generate different checksums.

The use of different voices is not part of the caching, means the cache knows only the Text, not which voice it is.

To use the caching with different voices and cache them separately, you will need to Change the Cache Folder location using the AIS.Set Folder - Command.

Example

```
' *****
' AIS.-Sample
' *****
AIS.Set Key|<YourAPIKeyHere>
AIS.Text to MP3|Hello World|?path/to/save.mp3
```

In this example, the text "Hello World" will be converted into an MP3 file and saved in the specified path.

Remarks

- The API key must be set using the **AIS.Set Key** command before using this command.
- The system will use a cache to save resources **only if P2 is omitted or empty**.

Limitations:

-

See also:

-

3.42.13 DLT. - DeepL-Translate

[DLT. - DeepL-Translate](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

DLT. - DeepL-Translate

Use DeepL.com - for translations



Using DeepL.com you can access many Languages of the world.

Using DeepL.com for Translation

DeepL.com is a highly regarded translation service that offers fast, accurate, and secure translations. It is considered by many to be the world's most accurate translator, thanks to its advanced neural networks that are able to capture even the slightest nuances and reproduce them in translation¹. In blind tests pitting DeepL Translator against the competition, translators prefer DeepL's results by a factor of 3:12.

Here are the current pricing plans for DeepL:

- Free: The free plan allows you to translate up to 5,000 characters per day.
- Pro: The Pro plan costs €29.99/month and allows you to translate up to 1,000,000 characters per day.
- Business: The Business plan costs €99.99/month and allows you to translate up to 10,000,000 characters per day.
- Enterprise: The Enterprise plan is custom-priced and allows you to translate an unlimited number of characters per day.

All plans include access to all of DeepL's features, including the ability to translate text in a variety of languages, integrate with other services, and get priority support.

Here is a table that summarizes the current [pricing plans for DeepL](#):

Plan	Monthly Base Price	Usage-Based Cost	Character Limit
DeepL API Free	\$0.00	\$0.00	500,000 characters
DeepL API Pro	\$5.49 ¹	\$0.00002 per translated character²	Unlimited

Source: BING/25.07.2023*

[With the DeepL API Pro plan, you pay a low flat rate of \\$5.49 per month for access to the API, plus a usage-based fee of \\$0.00002 per translated character¹. This plan offers maximum data security, with your texts being deleted immediately after translation³. You also have the ability to set a monthly maximum cost control limit](#)

***All prices are Subject to change.**

[Please see actual Information on the official site. Prices for Developer.](#)

In addition to its competitive pricing, DeepL.com also offers a range of features that make it an excellent choice for translations. For example, with DeepL Pro, users can translate an entire document with one click while retaining the original formatting. This can save a significant amount of time and effort compared to manually copying and pasting text into a translation tool.

DeepL Pro also offers expanded customization options, allowing users to take more control of the results produced by the translator². This means that users can tailor the translations to their specific needs and preferences.

Finally, DeepL.com takes data security very seriously. When using DeepL Pro, all texts are deleted immediately after the translation has been completed, and the connection to their servers is always encrypted². This ensures that users' data is protected and cannot be accessed by third parties.

In summary, DeepL.com is an excellent choice for translations due to its accuracy, affordability, range of features, and commitment to data security. Whether you need to translate a single document or integrate translation technology into your own products and platforms, DeepL.com has a solution that can meet your needs.

3.42.13.1 Additional Instructions

[DLT.Additional Instructions](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

DLT.Additional Instructions

Sets extra JSON instructions for DeepL

Intention

The `DLT.Additional Instructions` command is used to set additional instructions for the JSON body sent to the DeepL Translator. For example, you can use this command to specify additional parameters that are not covered by other commands.

Example 1 (manual Escaping):

To set the `glossary_terms` field in the JSON body, you would use:

```
DLT.Additional Instructions|\\"glossary_terms\\": [\\"term1\\", \\"term2\\"]
```

This will result in the `DLT_Adi` variable being set to `\\"glossary_terms\\": [\\"term1\\", \\"term2\\"]`, and the JSON body sent to the DeepL Translator will include this additional instruction.

Example 2 (use AIC.Escape String):

To set the `glossary_terms` field in the JSON body, you would first use the `AIC.Escape String` command to escape the string:

```
AIC.Escape String|glossary_terms: [term1, term2]|$$RES
```

Then, you would use the `DLT.Additional Instructions` command with the escaped string:

```
DLT.Additional Instructions|$$RES
```

Example 3 (multiple additional Instructions)

To set multiple fields in the JSON body, such as `glossary_terms` and `additional_context`, you would first use the `AIC.Escape String` command to escape each string:

```
AIC.Escape String|glossary_terms: [term1, term2]|$$RES1
AIC.Escape String|additional_context: [context1, context2]|$$RES2
```

Then, you would use the `DLT.Additional Instructions` command with the escaped strings, concatenating them with a comma:

```
DLT.Additional Instructions|$$RES1, $$RES2
```

This will result in the internal variable being set to the concatenated escaped strings, and the JSON body sent to the DeepL Translator will include these additional instructions.

Please note that the `AIC.Escape String` command is used to escape each string separately, and the results are stored in different variables (`$$RES1` and `$$RES2`). These escaped strings are then concatenated with a comma to form the final additional instructions.

Syntax

DLT.Additional Instructions[P1] ... ELS. ... EIF.

Parameter Explanation

P1 - (*optional*) The additional instructions for the JSON body. This is a string that should be formatted as a valid JSON key-value pair. For example, `"\glossary_terms\": [\"term1\", \"term2\"]` would add a `glossary_terms` field to the JSON body with the values *term1* and *term2*.

If no parameter is provided, **P1** will be set to an empty string.

It's important to note that any necessary escaping of special characters in the additional instructions must be done manually.

This can also be achieved using the `AIC.Escape String` command.

This step ensures that the additional instructions are formatted correctly as a valid JSON string.

Example

```

'*****
' Here is the resulting JSON Body
'*****
{
  "text": ["Your text here"],
  "source_lang": "EN",
  "target_lang": "DE",
  "split_sentences": "1",
  "format": "text",
  "formality": "default",
  "preserve_formatting": false,
  "glossary_terms": ["term1", "term2"],
  "additional_context": ["context1", "context2"]
}

```

Remarks

-

Limitations:

-

See also:

-

3.42.13.2 Get Source Language

[DLT.Get Source Language](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

DLT.Get Source Language

Gets the source language found by DeepL AI

Intention

This command retrieves the source language detected by the DeepL translation service. The source language is determined based on the input text provided for translation. The detected source language is returned as a two-letter language code (e.g., "EN" for English, "DE" for German).

The source language is returned as a two-letter language code. For a list of language codes, refer to the DeepL API documentation.

The index parameter (P1) is used when multiple languages are detected in the input text. The index refers to the position of the detected language in the list of languages, starting from 0.

This command is not directly related to the `DLT.Set Source Language` command. While both commands deal with source languages, they serve different purposes and use different registers. The `DLT.Set Source Language` command is used to specify the source language for translation, while the `DLT.Get Source Language` command is used to retrieve the source language detected by the DeepL service.

```
' This command will retrieve the source language at index 1 and store the result
DLT.Get Source Language|$$SRC|1
```

```
' This command will retrieve the source language at the default index (0) and st
DLT.Get Source Language|$$SRC
```

Syntax

DLT.Get Source Language [|P1] [|P2]

Parameter Explanation

P1 - (optional) A variable to store the detected source language. If the variable is not provided or is empty, the result will be placed on top of the stack.

P2 - (optional) An optional variable containing the index of the source language. If not provided, the default value is 0.

Example

```
! *****  
!  
! *****
```

Remarks

-

Limitations:

-

See also:

- [1.6.1. Program Flow Control](#)^[134]
- [! Smart Package Robot 's Parallel Robot Operations](#)^[3745]
- [1.5. Features and Hints](#)^[93]

3.42.13.3 Save Key

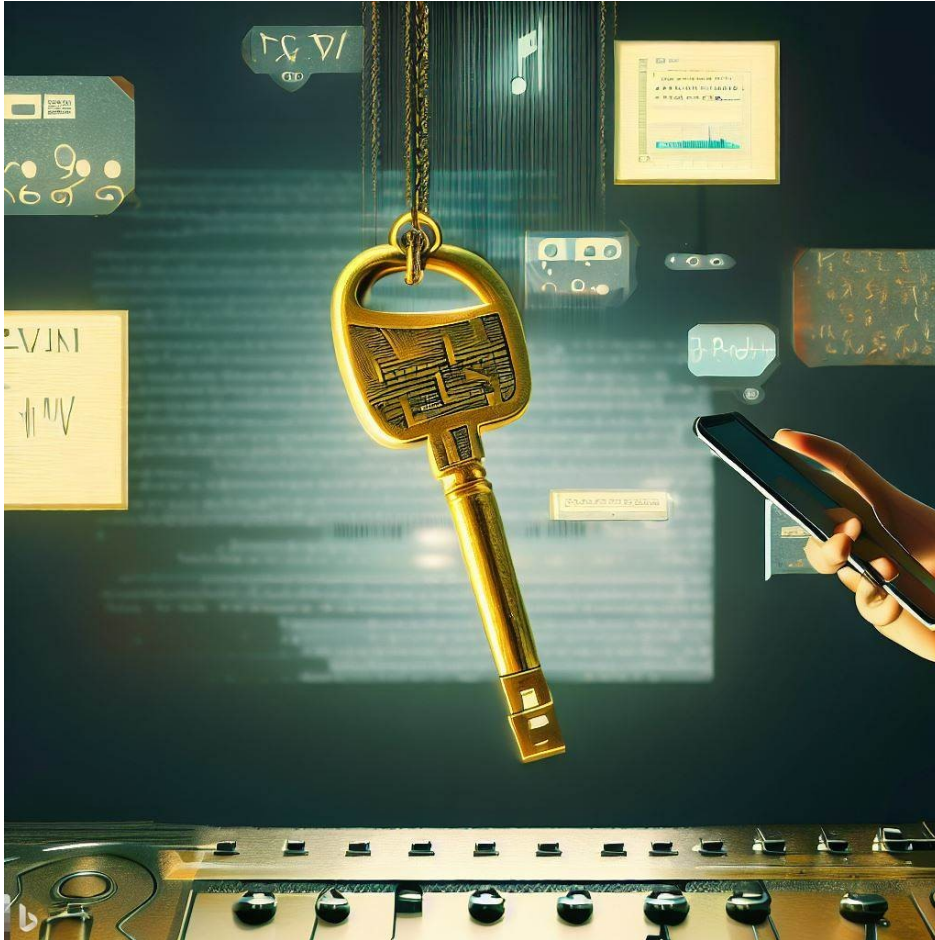
DLT.Save Key

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

DLT .Save Key

Save DeepL-API Key encrypted to a file



Intention

The `DLT.Save Key` Command is an essential tool for developers venturing into the world of **DeepL**.

It acts as a secure vault, allowing you to store your API Key in an encrypted format within a file.

This not only bolsters security but also streamlines the process of utilizing the key across various scripts.

Utilizing the Save Key Command

To employ the 'Save Key Command', you need to invoke the `DLT.Save_Key` function.

Pass your API Key as the primary argument.

Additionally, you can specify a file path to determine where the encrypted key should be stored, while this optional.

```
DLT.Save_Key <YOUR_API_KEY> [ |OPTIONAL_FILE_PATH]
```

In the event that a file path is not specified, the command will default to saving the file in the directory where the script or executable is situated.

The default path is "?exeloc\DLT_License_Key.dat"

Default Naming Convention and Location

The encrypted file is conventionally named `DLT_License_Key.dat`.

This standard naming practice ensures easy identification.

If a file path is omitted, the file will be created in the directory denoted by `?exeloc\`, which corresponds to the location of the script or executable.

The Significance of the Save Key Command

Using the `DLT.Save_Key` Command to store the API Key in an encrypted file is highly advisable. This approach significantly reduces the risk of unintentional exposure and provides a convenient method for reusing the key in different scripts.

Wrapping Up

The `Save Key Command` is a powerful and indispensable tool for safeguarding your API Key. By storing it in an encrypted file, you ensure its protection and facilitate its use across your DeepL projects.

```
' Script 1: Save the Key to the file "DLT_License_Key.dat".
```

```
' IMPORTANT: This API-Key is a phantasy API-Key it must be replaced with your API-Key
$$KEY=sk-abcdefghijklmnopqrstuvwxy123456
```

```
' Here we save the Keyfile at the default path, that is:
```

```
' ?exeloc\AIC_License_Key.dat
```

```
DLT.Save_Key|$$KEY
```

```
ENR.
```

```
' Script 2; Using the crypted API-Keyfile
```

```
' Test if we are online, AI-Commands will only work if you are online.
```

```
NOL.
```

```
GTO.enx
```

```
EIF.
```

```
' Set DeepL.com API-Key from the saved File
```

```
DLT.SetKey|from_File
```

```
DLT.translate Text|$$TXT|$$RET|en|de
```

```
DBP.$$RET
```

```
:enx
ENR.
```

Syntax

```
DLT . SaveKey | P1 [ | P2 ]
DLT . SvK | P1 [ | P2 ]
```

Parameter Explanation

P1 - Your DeepL.com API Key

P2 - (*optional*) Filepath for the API-Key to save

Example

```
!*****
' EXAMPLE 1: DLT.-Commands
!*****
' Script 1: Save the Key to the file "DLT_License_Key.dat".

' IMPORTANT: This API-Key is a phantasy API-Key it must be replaced with your API-Key
'$KEY=abcdefghijklmnopqrstuvwxyz123456

' Here we save the Keyfile at the default path, that is:
' ?exeloc\DLT_License_Key.dat
DLT.Save_Key|$KEY
ENR.
```

Remarks

-

Limitations:

✨ Safeguarding Your API Key: A Knight's Guide to DeepL.com

Greetings, Noble Coder! 🛡️

Embarking on a quest through the enchanted forests of DeepL?

Before you mount your steed, there's a sacred artifact you must secure - **the illustrious API Key**.

This key is not just a string of characters; it's the heart of your adventure, the magic that unlocks the kingdom's secrets.

🔒 ****The Enchanted Encryption**** 🔒

Ah, you seek to protect your treasure by locking it within an enchanted file.

Wise as it may seem, remember, even the mightiest spells have their counters. The encryption wards off common thieves, but against a sorcerer with the dark SPR arts, it may falter.

🔒 ****Guard Your Treasure Chest**** 🔒

Distributing your encrypted key is akin to leaving your treasure chest in the dragon's lair. Any rogue with an SPR spellbook can break the enchantment.

The treasure within - your API Key - is bound to your very essence, your account. In the wrong hands, it can unleash storms and deplete your resources.

🛡️ ****Set Magical Boundaries**** 🛡️

Fear not, for there is a spell to shield your treasure further.

Within the hallowed halls of OpenAI's website, you can weave a spell to set limits on your API Key's powers.

This incantation ensures that even if your key is seized, its magic is bound, and the havoc it can wreak is contained.

🛡️ ****The Knight's Code**** 🛡️

1. ****Guard the Key****: Never let your API Key, even if enchanted in a file, sail on uncharted waters.
2. ****Summon Guardians****: Create a mystical barrier through a backend service. Let this guardian use the API Key in the shadows, far from prying eyes.
3. ****Eternal Vigilance****: Watch over your domain. Keep an eagle's eye on the usage of your API Key.

Remember, brave knight, with great power comes great responsibility. Your API Key is the magic that courses through the veins of your quests. Guard it, protect it, and let it guide you through countless adventures in the realm of DeepL.

Onward, to glory! 🚀

See also:

-

3.42.13.4 Set Endpoint

DLT.Set Endpoint[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

DLT.Set Endpoint

Set/change the endpoint for the DeepL Translator

Intention

The `DLT.Set Endpoint` command is used to set the endpoint for the DeepL Translator. An endpoint in this context refers to the URL that the DeepL Translator uses to access the translation service.

Different versions of the DeepL service (such as the free version and the pro version) use different endpoints. Therefore, it may be necessary to change the endpoint depending on the version of the DeepL service you are using.

Additionally, if DeepL introduces new endpoints or if you want to use a custom endpoint (for example, for testing or development purposes), you can use this command to set the endpoint to a custom URL.

Syntax

DLT.Set Endpoint[|P1]

Parameter Explanation

P1 - (*optional*) The endpoint setting. This can be "**free**", "**pro**", or a custom URL.

If **P1** is "**free**" or an empty string, the endpoint will be set to "`https://api-free.deepl.com/v2/translate`".

If **P1** is "**pro**", the endpoint will be set to "`https://api.deepl.com/v2/translate`".

If **P1** is neither "**free**" nor "**pro**", it will be treated as a custom endpoint URL.

If no parameter is provided, **P1** will be set to the free endpoint URL.

Example

```
' *****
' Set Endpoint for DeepL
```

```
! *****  
$$URL=https://api-free.deepl.com/v2/translate  
DLT.Sep| $$URL  
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.13.5 Set Formality

[DLT.Set Formality](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

DLT.Set Formality

Sets formality level for DeepL translations

Intention

The `DLT.Set Formality` command is used to adjust the formality level of the translations produced by the DeepL Translator.

This command is particularly useful when the target language has clear formal and informal modes of speech.

By adjusting the formality level, you can make the translated text more suitable for different contexts.

For instance, setting the formality to "more" would be suitable for formal business communications or academic writing, making the language used more polite and respectful. On the other hand, setting the formality to "less" would be suitable for casual conversations or social media posts, making the language used more relaxed and friendly.

To set the formality to more formal, you would use:

```
DLT.Set Formality|More
```

This will result in translations that are more formal, suitable for business communications or formal letters. For example, a casual English phrase like "What's up?" might be translated into German as "Wie geht es Ihnen?" instead of the more casual "Wie geht's?".

To set the formality to more informal, you would use:

```
DLT.Set Formality|Less
```

This will result in translations that are more casual, suitable for informal conversations or social media posts. For example, a formal English phrase like "How do you do?" might be translated into Spanish as "¿Cómo estás?" instead of the more formal "¿Cómo le va?".

Syntax

```
DLT.Set Formality[P1] ... ELS. ...  
EIF.
```

Parameter Explanation

P1 - (optional) The formality setting. This can be "default", "more", "less", "prefer_more", or "prefer_less".

- "default": The DeepL Translator will use the default formality level.
- "more": The DeepL Translator will use a more formal language. This is suitable for formal communications.
- "less": The DeepL Translator will use a more informal language. This is suitable for casual conversations.
- "prefer_more": The DeepL Translator will use a more formal language if available, otherwise it will fall back to the default formality.
- "prefer_less": The DeepL Translator will use a more informal language if available, otherwise it will fall back to the default formality.
- If **P1** is neither of these values, it will be treated as invalid and set to the default value "default".

Example

```
! *****  
!  
! *****  
DLT.Set Formality|Less  
DLT.Set Formality|More
```

Remarks

-

Limitations:

-

See also:

-

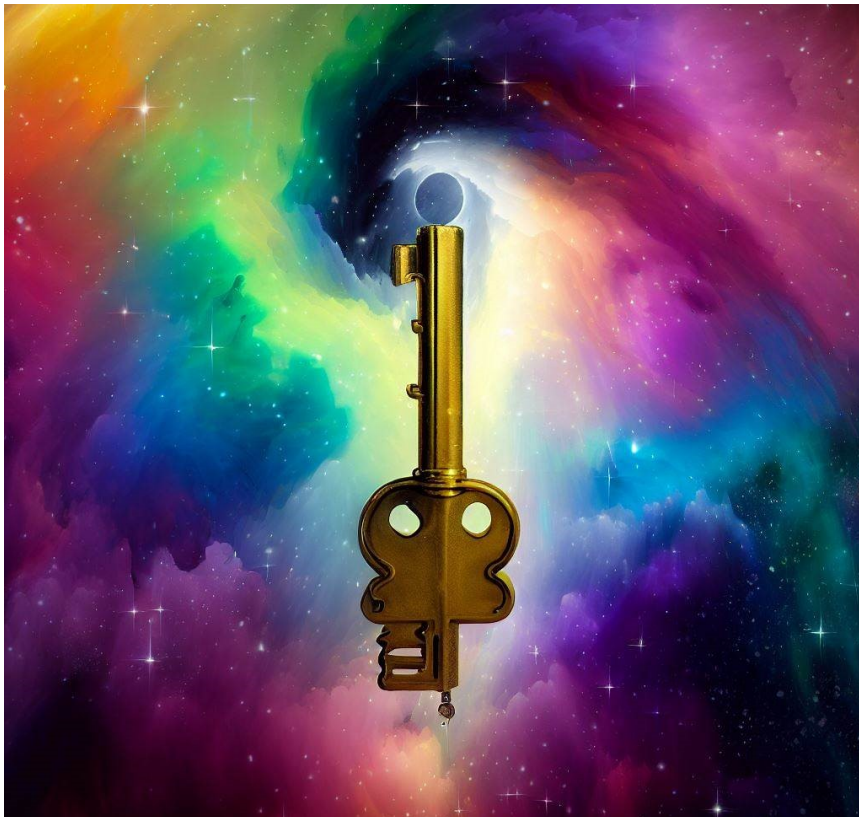
3.42.13.6 Set Key

`DLT.SetKey`[Previous](#) 2027 | [Top](#) 49 | [Next](#) 2069

MiniRobotLanguage (MRL)

DLT . SetKey

Set DeepL.com-API Key



Intention

SetKey Command: Initiating Your Script with a License Key

Before diving into the world of AI scripting, it's essential to set the stage with the `DLT.SetKey` command.

This command is the golden ticket that grants you access to the plethora of AI functionalities offered by DeepL.com.

Let's break down how to use it effectively.

What is the `DLT.SetKey` Command?

The `DLT.SetKey` command is the first command you need to include at the beginning of your script.

It's like the key to a treasure chest; without it, you can't unlock the AI translation capabilities you're after.

Why Do You Need It?

You might be wondering why there's a need for such a key.

The reason is that the AI functionalities you are looking to use are not processed locally on your computer.

Instead, they are handled remotely in the high-powered **DeepL.com** Cloud.

This License Key ensures that you have the proper [authorization to access these cloud-based services](#).

How to Obtain the License Key?

To get your hands on this key, you'll need to visit (click the Button)

DeepL.com's official website

Once there, follow the instructions to register and **obtain the License Key**.

Keep this key safe, as you'll need it every time you want to use DeepL.com's services.

How to Use the `DLT.SetKey` Command?

Once you have your License Key, it's time to put it to use.

At the very start of your script, include the `DLT.SetKey` command followed by your License Key.

This will authenticate your script with **DeepL.com's cloud services**.

On the long run you may prefer the second Option (see below) that will encrypt your key into a file and use this file.

The `DLT.Set_Key`-Command has multiple Usage Options.

1. You can directly set the Key in the Script.

Here is a Sample Script that shows how this is done.

```
' IMPORTANT: This API-Key is a phantasy API-Key it must be
replaced with your API-Key.
$$KEY=f6e3519f-b8e7-5082-0a95-2a847e4134bf:fx

' Test if we are online, AI-Commands will only work if you are
online.
NOL.
  GTO.enx
EIF.

' Set DeepL.com API-Key
DLT.SetKey|$$KEY

' Ask Question and receive answer to $$RET
$$TXT=Your Text
DLT.translate text|$$TXT|$$RET|en|de
DBP.$$RET

:enx
ENR.
```

2. You can use a saved, encrypted Key, that is stored in the project-folder.

This is the preferred way because this is more save for your key.

The default name for this saved Key is "DLT_License_Key.dat"

If no path is given, the file is created at "?exeloc\" that is the folder where the Script/the executable resides.

This key-file can be created using the AIC.Save_Key|\$\$KEY[|\$\$FIL] Command.

```
' Script 1: Save the Key to the file "DLT_License_Key.dat".
```

```
' IMPORTANT: This API-Key is a phantasy API-Key it must be replaced with your API-Key
$$KEY=f6e3519f-b8e7-5082-0a95-2a847e4134bf:fx
```

```
' Here we save the Keyfile at the default path, that is:
' ?exeloc\DLT_License_Key.dat
DLT.Save_Key|$$KEY
ENR.
```

```
' Script 2; Using the crypted API-Keyfile
' Test if we are online, AI-Commands will only work if you are online.
NOL.
GTO.enx
EIF.
```

```
' Set DeepL.com API-Key from the saved File
DLT.SetKey|from_File
```

```
' Ask Question and receive answer to $$RET
$$TXT=Your Text
DLT.translate text|$$TXT|$$RET|en|de
DBP.$$RET
```

```
:enx
ENR.
```

3. You can use a Textfile, that contains your Key.

This is way to go, if you make an Executable that you want to share with the public or other people.

To give them the Option to use your SPR-Script together with their own API-Key, you can offer this Option.

The default name for this saved Key in a Textfile is "DLT_License_Key.txt"

If no path is given in **P2**, the file should be located in "?exeloc\" that is the folder where the Script/the executable resides.

This key-file can be created using any Texteditor (or CTF.-Command).

```
' Script 3: Using the API-Keyfile with the API-Key as Text
' Test if we are online, AI-Commands will only work if you are online.
NOL.
GTO.enx
EIF.
```

```
' Set DeepL.com API-Key from the saved File
```

```
DLT.SetKey|from Text

' Ask Question and receive answer to $$RET
$$TXT=Your Text
DLT.translate text|$$TXT|$$RET|en|de
DBP.$$RET

:enx
ENR.
```

Syntax

DLT.SetKey|P1 [|P2]

DLT.Set_Key|P1 [|P2]

Parameter Explanation

P1 - Can be directly an [DeepL-API-Key](#) or:

- **from File** - if given as **P1**, there should be a encrypted Keyfile with the name "AIC_License_Key.dat" at [?exeloc\](#)^[364] or if specified, at Path **P2**. This file can only be created using the [AIC.Save Key](#)^[762] - Command.
- **from Text** - if given as **P1**, there should be a Textfile that contains the API-Key with the name "AIC_License_Key.txt" at [?exeloc\](#)^[364] or if specified, at Path **P2**. This file can be created using any Text-Editor.
- **Load Any Key** - if given as **P1**, the command will load any available API-Key.

P2 - opt. if specified this is the Path (and filename) of the Keyfile to use with the two options above. It can be either a Crypted or a Textfile. The System will decide with the Extension ".txt" or ".dat" how it is loaded.

Example

```
'*****
' EXAMPLE 1: DLT.-Commands
'*****
' IMPORTANT: This API-Key is a phantasy API-Key it must be replaced with your AI
$$KEY=f6e3519f-b4e7-5082-0a95-2a847e4134bf:fx
' Test if we are online, AI-Commands will only work if you are online.
NOL.
GTO.enx
EIF.

' Set DeepL.com API-Key
DLT.SetKey|$$KEY

' Ask Question and receive answer to $$RET
$$TXT=Your Text
DLT.translate text|$$TXT|$$RET|en|de
DBP.$$RET
```

```
:enx  
ENR.
```

Remarks

A Word of Caution: Safeguarding Your API Key 🚨

Dear Valued User,

🔒 *This Encryption is Not Impenetrable* 🔒

Even if your API Key is encrypted within a file, the Decryption is hereby done without a Password, means anybody can decrypt it if he has the SPR at hand.

It's akin to a treasure locked in a chest.

The chest provides an added layer of security and invisibility, but should it fall into the hands of a pirate with the SPR and Knowledge, the treasure can be plundered.

You may ask: "Why did we not use a Password?"

The answer is simple, even then you would need to provide that Password in Cleartext in the code, because the SPR needs to decrypt the keyfile anyway.

So there would be no advantage. In case you want to enter the Password each time and have an unbreakable Encryption, you can instead do something using the `GUT.` and the `GEC.`-Command.

🚫 *Do Not Distribute the Key, Even Encrypted* 🚫

Distributing executables or scripts along with the file containing your encrypted API Key is akin to sending your treasure chest out to sea on an unmanned ship. Anyone who gets hold of this file and has access to an SPR (Script Processing Runtime) can potentially decrypt and misuse your API Key.

💡 *Why is This a Big Deal?* 💡

Your API Key is not just a string; it's your identity and access within the DeepL.com realm. It's linked to your account, your resources, and your privileges. In the wrong hands, it can be used to access services and consume quotas associated with your account. This can have both financial and security implications.

🛑 *What Should You Do?* 🛑

Never Distribute the Key: Do not include your API Key, even if encrypted, in any files or executables that you distribute.

Access Control: If your application requires the use of the API Key, consider implementing a backend service that your application can call. The service can then use the API Key server-side, where it's not exposed to the end-user.

Vigilance: Regularly monitor the usage of your API Key and be vigilant for any unauthorized or unexpected activity.

Set Usage Limits: Visit the DeepL.com website and access your account settings. Here, you can set limits on the usage of your API Key. This is a wise precaution to ensure that even if the unthinkable happens, the potential damage is contained. It's like setting a magical barrier around your treasure chest!

Remember, with great power comes great responsibility. Your API Key is a powerful tool; wield it wisely and guard it well.

Safe coding!

Limitations:

-

See also:

•

3.42.13.7 Set preserve Formatting

[DLT.Set preserve Formatting](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

DLT.Set preserve Formatting

Sets if DeepL respects original format

Intention

The `DLT.Set Preserve Formatting` command is used to set whether the DeepL Translator should respect the original formatting, even if it would usually correct some aspects.

The original formatting includes punctuation at the beginning and end of the sentence, and upper/lower case at the beginning of the sentence.

Syntax

```
DLT.Set preserve Formatting[ |
P1 ]
```

Parameter Explanation

P1 - (*optional*) The preserve formatting setting. This can be "true", "false", "on", or "off".

If **P1** is "true" or "on", the DeepL Translator will respect the original formatting.

If **P1** is "false", "off", or an empty string, the DeepL Translator will not respect the original formatting and make corrections.

If **P1** is neither of these values, it will be treated as invalid and set to the default value "false".

Example

```
! *****
! IRS.-Sample
! *****
DLT.Set Preserve Formatting|On
DLT.Set Preserve Formatting|Off
```

Remarks

-

Limitations:

-

See also:

-

3.42.13.8 Set Source Language

[DLT.Set Source Language Command](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

DLT.Set Source Language Command

Set the source language for the DeepL Translator.

Intention

The `DLT.Set Source Language` command is used to set the source language for the DeepL Translator.

Normally this will not be necessary because DeepL will recognize the Source language automatically.

Language of the text to be translated. [Options currently available:](#)

- BG - Bulgarian
- CS - Czech
- DA - Danish
- DE - German
- EL - Greek
- EN - English
- ES - Spanish
- ET - Estonian
- FI - Finnish
- FR - French
- HU - Hungarian
- ID - Indonesian
- IT - Italian
- JA - Japanese
- KO - Korean
- LT - Lithuanian
- LV - Latvian
- NB - Norwegian (Bokmål)
- NL - Dutch
- PL - Polish
- PT - Portuguese (all Portuguese varieties mixed)
- RO - Romanian
- RU - Russian
- SK - Slovak
- SL - Slovenian
- SV - Swedish
- TR - Turkish
- UK - Ukrainian
- ZH - Chinese

If this parameter is omitted, the API will attempt to detect the language of the text and translate it.

Syntax

DLT.Set Source Language [| P1]

Parameter Explanation

P1 - (*optional*) The source language code. This should be a 2-character string representing the language code. If the length of **P1** is more than 2 characters, it will be shortened to the first 2 characters. If **P1** is less than 2 characters, it will be omitted.

Example

```
'*****  
'  
'*****  
DLT.Set Source Language|de
```

Remarks

-

Limitations:

-

See also:

•

3.42.13.9 Set Split Sentence

[DLT.Set Split Sentence](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

DLT.Set Split Sentence

Sets sentence splitting behavior for DeepL

Intention

The `DLT.Set Split Sentence` command is used to control how the DeepL translation service divides the input text into sentences for translation. This is a crucial aspect of the translation process, as the way sentences are split can significantly influence the quality and accuracy of the translation.

Sentence splitting is particularly important for languages that use complex sentence structures or have different rules for punctuation. By correctly splitting sentences, the translation service can better understand the context and provide more accurate translations.

The `DLT.Set Split Sentence` command allows you to specify the sentence splitting behavior using the optional parameter `P1`. This parameter can take several values, each corresponding to a different sentence splitting behavior:

"0" or "Splitting Off": This setting disables sentence splitting. The entire input text is treated as a single sentence. This can be useful for short texts or texts that are already divided into individual sentences.

"1" or "Splitting On": This setting enables sentence splitting on both punctuation and newlines. This is the default behavior when tag handling is not set to HTML. It's suitable for most general texts.

"nonewlines": This setting enables sentence splitting on punctuation only, ignoring newlines. This is the default behavior when tag handling is set to HTML. It's useful for texts that use newlines for formatting rather than to indicate the end of a sentence.

```
' This command sets the sentence splitting behavior to treat the entire input t
DLT.Set Split Sentence|0
```

```
' This command sets the sentence splitting behavior to split on both punctuation
DLT.Set Split Sentence|Splitting On
```

The `DLT.Set Split Sentence` command is part of the DeepL translation service commands.

It affects how the input text is processed for translation. The sentence splitting behavior can significantly affect the translation quality, especially for languages that use complex sentence structures. Therefore, it's recommended to choose the appropriate setting based on the nature of the input text and the target language.

Syntax

DLT.Set Split Sentence [| P1]

Parameter Explanation

P1 - (*optional*) This parameter can take several values, each corresponding to a different sentence splitting behavior:

"0" or "Splitting Off": This setting disables sentence splitting. The entire input text is treated as a single sentence. This can be useful for short texts or texts that are already divided into individual sentences.

"1" or "Splitting On": This setting enables sentence splitting on both punctuation and newlines. This is the default behavior when tag handling is not set to HTML. It's suitable for most general texts.

"nonewlines": This setting enables sentence splitting on punctuation only, ignoring newlines. This is the default behavior when tag handling is set to HTML. It's useful for texts that use newlines for formatting rather than to indicate the end of a sentence.

Example

```
! *****
!
! *****
```

Remarks

-

Limitations:

-

See also:

- [1.6.1. Program Flow Control](#)^[134]
- [! Smart Package Robot 's Parallel Robot Operations](#)^[3745]

- [1.5. Features and Hints](#) 

3.42.13.1 Set Target Language

DLT.Set Target Language Command[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

DLT.Set Target Language Command

Set the source language for the DeepL Translator.

Intention

The `DLT.Set Target Language` command is used to set the source language for the DeepL Translator.

If this Parameter is missing, "EN" (English) will be used.

The language into which the text should be translated. [Options currently available:](#)

- BG - Bulgarian
- CS - Czech
- DA - Danish
- DE - German
- EL - Greek
- EN - English (unspecified variant for backward compatibility; please select EN-GB or EN-US instead)
- EN-GB - English (British)
- EN-US - English (American)
- ES - Spanish
- ET - Estonian
- FI - Finnish
- FR - French
- HU - Hungarian
- ID - Indonesian
- IT - Italian
- JA - Japanese
- KO - Korean
- LT - Lithuanian
- LV - Latvian
- NB - Norwegian (Bokmål)
- NL - Dutch
- PL - Polish
- PT - Portuguese (unspecified variant for backward compatibility; please select PT-BR or PT-PT instead)
- PT-BR - Portuguese (Brazilian)
- PT-PT - Portuguese (all Portuguese varieties excluding Brazilian Portuguese)
- RO - Romanian
- RU - Russian
- SK - Slovak
- SL - Slovenian
- SV - Swedish
- TR - Turkish
- UK - Ukrainian
- ZH - Chinese (simplified)

Syntax

DLT.Set Target Language [| P1]

Parameter Explanation

P1 - (*optional*) The source language code. This should be a 2-character string representing the language code. If the length of **P1** is more than 2 characters, it will be shortened to the first 2 characters. If **P1** is less than 2 characters, it will be omitted. If this Parameter is missing, "EN" (English) will be the target language.

Example

```
!*****  
!  
!*****  
DLT.Set Target Language|de
```

Remarks

-

Limitations:

-

See also:

-

3.42.13.1.Translate Text

[DLT.Translate Text](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

DLT.Translate Text

Translates text using DeepL with customizable settings

Intention

The `DLT.Translate Text` command (short form `DLT.tra`) is a powerful tool that leverages the capabilities of the DeepL translation service to translate a given text from one language to another. DeepL is a state-of-the-art machine translation service that uses artificial intelligence to achieve high-quality translations. It supports a wide range of languages and is renowned for its ability to capture nuances and maintain the original tone of the text.

To use the `DLT.Translate Text` command, you need to have a valid API key for the DeepL service. This key is used to authenticate your requests and access the translation service.

Please note that the DeepL service may or may not be free, and usage costs may apply depending on your subscription plan with DeepL.com

The `DLT.Translate Text` command takes up to four parameters. The first parameter, **P1**, is mandatory and specifies the text to be translated. The remaining parameters are optional and allow you to customize the translation process:

Syntax

```
DLT.Translate Text | P1 [ | P2 ] [ | P3 ] [ | P4 ]
```

Parameter Explanation

P1 - The text to be translated. This parameter is mandatory.

P2 - (optional) The variable to store the translation result. If omitted, the translation result is placed on the Top of Stack (TOS).

P3 - (optional) The source language for the translation. If omitted, the translation service will automatically detect the source language.

P4 - (optional) The target language for the translation. If omitted, the translation service will use the default target language set by the `DLT.Set Target Language` command.

Example

```
'*****  
'  
'*****  
' This command translates the text "Hello" and places the  
translation result on the Top of Stack. The source and target  
languages are automatically detected or set to the default  
values.  
DLT.Translate Text|"Hello"  
  
' This command translates the English text "Hello" into  
German and stores the translation result in the $$Res  
variable.  
DLT.Translate Text|"Hello"|$$Res|en|de
```

Remarks

-

Limitations:

-

See also:

-

3.42.14 SD - Stable Diffusion AI

Stable Diffusion - what is it?

🤖 ****Unlock the Power of Stable Diffusion with Smart Package Robot Scripting****



Created using SDO., this is how a neural network imagines itself.

What is "Stable Diffusion"?

Stable Diffusion is a deep learning, [text-to-image model released in 2022 by Stability AI](#)

It is primarily used to generate detailed images conditioned on text descriptions, though it can also be applied to other tasks such as inpainting, outpainting, and generating image-to-image translations guided by a text prompt. Stable Diffusion is a latent diffusion model, a kind of deep generative artificial neural network.

Its code and model weights have been released publicly, and it can run on most consumer hardware equipped with a modest GPU with at least 8 GB VRAM.

Find a technical and mathematical explanation of the [Diffusion Process here](#).

Generally there are 2 version of Stable Diffusion:

- there is one version that is used "In the **Cloud**" and you need to get an API-Key for this. Its called "[Stable Diffusion XL](#)"

- And there is the **Open Source** version which **runs on your local Computer**. "[Its just Stable Diffusion](#)"
This is the version we will discuss in the following Chapter.

To see Stable Diffusion in [Action try Dreamstudio](#)

Why use Stable diffusion for Image Generation?

In the ever-evolving world of digital art and automation, "Stable Diffusion" picture generation is a game-changer. By harnessing the capabilities of Smart Package Robot Scripting, artists and developers can create stunning visuals with unparalleled efficiency. Here's why you should be excited:

****1. Stability and Control:****

Stable Diffusion ensures that the generation process is controlled and predictable. This means fewer unexpected artifacts and more consistent results. With Smart Package Robot Scripting, you can fine-tune the parameters to achieve the desired level of stability, giving you greater control over the final output.

****2. Enhanced Creativity:****

The combination of Stable Diffusion and Smart Package Robot Scripting opens up a plethora of creative possibilities. From generating intricate patterns to creating lifelike textures, the technology allows for the production of a wide range of artistic styles. Unleash your creativity without the constraints of traditional methods.

****3. Speed and Efficiency:****

Time is of the essence in the digital world. Smart Package Robot Scripting accelerates the Stable Diffusion process, allowing for rapid generation of high-quality images. This efficiency means you can iterate and experiment at a much faster pace, significantly reducing the time from concept to completion.

****4. Automation and Scalability:****

With Smart Package Robot Scripting, you can automate the Stable Diffusion process. This is particularly beneficial for projects that require the generation of a large number of images. Automation not only saves time but also ensures consistency across all generated pictures.

****5. Customization and Personalization:****

Smart Package Robot Scripting allows for extensive customization of the Stable Diffusion parameters. Whether you're looking to create a specific texture, color palette, or pattern, you can tailor the process to meet your unique artistic vision.

****6. Accessibility and Collaboration:****

Smart Package Robot Scripting is accessible to both seasoned professionals and beginners. Its intuitive scripting language encourages collaboration among artists, developers, and designers. Share scripts, exchange ideas, and work together to push the boundaries of what's possible with Stable Diffusion picture generation.

****7. Innovation and Future-Proofing:****

By adopting Stable Diffusion and Smart Package Robot Scripting, you're positioning yourself at the forefront of technological innovation. As the technology continues to

evolve, you'll be well-equipped to adapt and take advantage of new features and capabilities.

8. *Privacy and Independence with Local Installation:*

One of the standout features of Stable Diffusion is its ability to be installed and run locally on your computer. This means you don't have to rely on cloud services, and here's why this is a big deal:

Its Private and Secure:

By running Stable Diffusion locally, your data stays on your machine. This is crucial for artists and developers who work with sensitive or proprietary information. Your creations are yours alone, and there's no risk of them being accessed or stored by third-party services.

Uncensored Creativity:

With local installation, you have the freedom to create without restrictions. There's no need to worry about content policies that are often associated with cloud services. Your artistic expression remains unfiltered and unbounded.

Constant Availability:

Internet connection is not a constraint. Whether you're in a studio, on a plane, or in a remote location, Stable Diffusion is always available on your local machine. This ensures uninterrupted creativity and productivity.

Cost-Efficient:

Running Stable Diffusion locally can also be more cost-effective. There are no recurring subscription fees or data transfer costs that are typically associated with cloud-based services.

Stable Diffusion with Smart Package Robot Scripting is not just a tool; it's an enabler for artists and developers to bring their visions to life.

With its local installation feature, it empowers you to work privately, freely, and without limitations.

Take control of your creative process and let nothing hold you back. 🗨️ 🎨

We can not take responsibility that the generated Images are free of Copyrights.

Generating Pictures with AI is currently a new process where the copyright situation is not clear.

Therefore take this into account for whatever you use these pictures.

*Embark on a journey of artistic exploration and innovation with Stable Diffusion picture generation through Smart Package Robot Scripting.

The future of digital art is here, and it's more vibrant, efficient, and limitless than ever before.* 🚀

3.42.14.1 AI - SDL / SDO. - Parameters and Sampler

About Stable Diffusion

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

How different are Stable Diffusion Prompts and parameters between

The Prompts are different, and also many Parameters are different.

Also the JSON-Output is different between the local Stable Diffusion and Stable Diffusion Online.

There are also a lot of common Parameters that are shared between both systems.

Here's a table that captures the standard parameters being used by

SDO.-Command, and the associated conditions:

Parameter	Condition
cfg_scale	SD_cfg_scale > -1
clip_guidance_preset	Len(SD_Clip_Guidance) > 0
height	Always added
width	Always added
text_prompts	Always added, Array of positive and negative prompts + (weights)
style_preset	Len(SD_Style) > 0
sampler	Len(SD_Sampler) > 0
samples	SD_Samples <> 1
seed	SD_Seed <> 0
steps	SD_Steps <> 50
SDO_Extra	Len(SDO_Extra) > 0. Other user-specified Parameters.

Following are the Samplers that you can choose from using **SDO**.

Number	Sampler Name
0	DDIM
1	DDPM
2	K_DPMPP_2M
3	K_DPMPP_2S_ANCESTRAL
4	K_DPM_2
5	K_DPM_2_ANCESTRAL
6	K_EULER
7	K_EULER_ANCESTRAL
8	K_HEUN

9	K_LMS
---	-------

Additional you can specify other Sampler Names when they become available.

Here the Standard-Parameters being used for **SDL.-Command** and the associated conditions:

Parameter	Condition
cfg_scale	SD_cfg_scale > -1
height	Always added
width	Always added
prompt	Always added. Prompts separated by ,
negative_prompt	Len(NP) > 0. Negative Prompts separated by ,
styles	Len(SD_Style) > 0
sampler_name	Len(SD_Sampler) > 0
samples	SD_Samples <> 1
seed	SD_Seed <> 0
steps	SD_Steps <> 50
SDL_Extra	Len(SDL_Extra) > 0. Other user-specified Parameters.

This table represents each parameter that might be added to sBody and the associated condition that determines its inclusion.

With **SDL.** you can choose these Samplers:

Index	Sampler
0	Euler
1	Euler a
2	LMS
3	Heun
4	DPM2
5	DPM2 a
6	DPM++ 2S a
7	DPM++ 2M
8	DPM++ SDE
9	DPM++ 2M SDE
10	DPM fast
11	DPM adaptive
12	LMS Karras

13	DPM2 Karras
14	DPM2 a Karras
15	DPM++ 2S a Karras
16	DPM++ 2M Karras
17	DPM++ SDE Karras
18	DPM++ 2M SDE Karras
19	DDIM
20	PLMS
21	UniPC

3.42.14.2 AI - SDL. / SDO. Prompting Guide

Mastering the Art of AI-Driven Image Generation

Unleash the Full Potential of Stable Diffusion with These Essential Parameters



Using the right Parameters, AI Art is at your fingertip's.

How to Write Effective Stable Diffusion Prompts

The **ideal length** of a Stable Diffusion prompt is **between 3 and 60 words**.

Longer prompts can be used, but they may be more difficult to interpret by the model and may result in less realistic images.

There are differences between SDL. - and SDO.-Prompts.

SDL. -Prompts are separated in Negative and Positive Prompt.

```
'Sample:  
$$POS=Red Ufo, Galaxis  
$$NEG=Green Ufo
```

SDO. -Prompts contain separated Terms that get a weight which can be positive or negative.

```
' Sample
  $$PRO=Red Ufo(0.5),GreenUfo(-0.5),Galaxis(0.5)
```

Here are some tips for writing effective Stable Diffusion prompts:

- A rule of thumb can be: (subject)(style), (action/scene), (artist), (filters)
- Be specific. The more specific you are, the more likely the model is to generate an image that matches your expectations.
- Use keywords. Keywords are words or phrases that describe the features you want to see in the image.
- Use negative prompts. Negative prompts tell the model what you don't want to see in the image.
- Use brackets and parentheses. Brackets and parentheses can be used to group keywords together.
- Use weights. Weights can be used to control the importance of different keywords.
- The maximum usable length of a Stable Diffusion text prompt is purportedly 77 tokens. This means that the prompt can contain up to 77 words, but each word can be made up of multiple tokens. For example, the word "cat" is a single token, but the word "kitten" is two tokens.

It is important to experiment with different prompt lengths to see what works best for you. Some people find that shorter prompts are more effective, while others find that longer prompts are more detailed. The best way to find the perfect length for your prompts is to try different things and see what you get.

The order of the words in a prompt for Stable Diffusion can have a significant impact on the resulting image. For example, if you want to generate an image of a cat, you might write the prompt as follows:

"A cat sitting on a windowsill"

However, if you change the order of the words, you might get a different result. For example, if you write the prompt as follows:

"A windowsill with a cat sitting on it"

The resulting image might be more focused on the windowsill, with the cat being smaller and less prominent.

This is because the order of the words in a prompt tells the model what to pay attention to first.

When the model sees the word "cat," it knows that it should start generating an image of a cat.

However, when the model sees the word "windowsill," it knows that it should also generate an image of a windowsill. The order of the words determines which object the model focuses on first.

In general, it is best to put the most important objects in the prompt first.

This will help the model to generate images that are more accurate and realistic. However, you can also experiment with different orders of words to see what works best for you.

Here are some other tips for writing effective Stable Diffusion prompts:

- Use specific keywords. The more specific you are, the more likely the model is to generate an image that matches your expectations.
- Use negative prompts. Negative prompts tell the model what you don't want to see in the image.
- Use brackets and parentheses. Brackets and parentheses can be used to group keywords together.
- Use weights. Weights can be used to control the importance of different keywords.

Creating the Perfect Prompt: A Comprehensive Guide

The creation of an effective prompt is an art form, involving several key components that guide the AI in generating the desired output. Here's a breakdown of these key components:

1. ****Subject (Raw Prompt)****: This is the fundamental part of the prompt. It's the primary object or theme you want the AI to focus on. It's the "what" of your prompt. For example: "a black horse".
2. ****Style****: The style determines the overall aesthetic and feel of the generated output. If no style is specified, the AI typically chooses the most common style associated with the raw prompt. For instance, if you're generating an image of a landscape, the AI might default to a realistic or oil painting style. Pairing the raw prompt with a well-chosen style can often yield satisfying results.
3. ****Action/Scene****: These elements describe what the subject (raw prompt) is doing (action) and where it is located (scene). For example: "jumping in the forest".
4. ****Filters****: Filters are the finishing touches you add to your prompt to achieve a specific look. For instance, adding "trending on Artstation" might give your image an artistic flair, while "Unreal Engine" might enhance the realism of the lighting. Some examples of filters are: highly detailed, surrealism, smooth, sharp focus, matte, elegant, illustration, digital paint, dark, gloomy, 8k, 4k, ambient lighting, and epic composition.

Remember, you can be creative and add any filter that aligns with your vision.

Here's a final example, combining all these components:

"Realistic art of a black horse, jumping in the forest, by Marc Simonetti, with fog, centered, symmetry, painted, intricate, volumetric lighting, beautiful, rich deep colours, masterpiece, sharp focus, ultra detailed, 4k."

Here are some additional examples incorporating all four components: subject, style, action,

1. "A photorealistic image of a golden robot girl, walking through a bustling city at night, in the style of Syd Mead, with neon lighting, rain-soaked streets, sharp focus, dramatic shadows, and 4k resolution."
2. "An impressionist painting of a tranquil lake, with a pair of swans gliding peacefully at sunset, inspired by Monet, featuring warm colors, soft focus, gentle ripples, and the day's last light reflecting off the water."

3. "A surrealistic digital illustration of a futuristic spaceship, soaring above a sprawling alien metropolis, in the style of H.R. Giger, with intricate biomechanical details, a monochromatic color scheme, stark contrast, and 8k resolution."

4. "A fantasy art of a majestic dragon, perched on a snow-covered mountain peak, in the style of John Howe, with breathtaking panoramic views, vibrant colors, detailed scales and textures, and a dramatic lighting."

5. "A hyperrealistic 3D render of an underwater city, teeming with futuristic structures and exotic sea creatures, in the style of Alex Roman, with photorealistic lighting, detailed architecture, diverse marine life, and a 4k resolution."

Remember, you can mix and match these components in various ways to achieve different effects and moods. The possibilities are virtually endless!

With this guide, you're now equipped to craft effective prompts that can guide the AI to generate the output that aligns with your vision. Happy prompting!

Here are some resources that you may find helpful:

Stable Diffusion Prompt: A Definitive Guide: <https://stable-diffusion-art.com/prompt-guide/>

How to Write the Best Stable Diffusion Prompts in 2023: <https://hackr.io/blog/stable-diffusion-prompts>

What is the ideal length for prompt?: <https://github.com/AUTOMATIC1111/stable-diffusion-webui/discussions/7701>

Parameter Details:

Source:

<https://platform.stability.ai/docs/features/api-parameters#engine>

3.42.14.3 ! How to use Stable Diffusion Part I SDL.

How to use Stable Diffusion with th

[Previous](#) ¹⁴⁹² [Top](#) ⁴⁹ [Next](#) ¹⁴⁹⁹

MiniRobotLanguage (MRL)

How to use Stable Diffusion with the SPR, Part I: SDL.

Here is a first Script you can run to get started



You can generate all sorts of pictures using SDL. on your local computer.


The Smart Package Robot supports 2 versions of Stable Diffusion

1. The Local Stable Diffusion, this needs the Automatic1111 to be [installed and having the --api switch set in the "webui.bat"](#) ¹³⁵⁷ and you need a usable Graphics card with enough VRAM
2. The Stable Diffusion Online, this will need that you [to get an API-Key to access the Cloud Services](#) ¹³⁹¹.

Depending on which service you want to use, you need to [whether get the API-Key](#) ¹³⁹¹ or [install the Automatic1111](#) ¹³⁵⁷ System on your computer or in your network.

In case you want to use Stabel Diffusion Online, the command of your choice is

[SDO.](#) ¹³⁹¹

If you decide the Stable Diffusion that runs Local - or in your Network, the Command of your choice is [SDL](#). 

What are the differences?

1. **Local** Stable Diffusion is **uncensored**, so you can produce pictures of whatever you like, this includes naked pictures or pictures that are forbidden due to whatever ideologies.

2. Stable diffusion **online** is a **cloud service** and as such must uphold the **rules for cloud services**.

While it can also produce naked pictures in some cases (unless you add that to the negative prompt) from time to time, it will send you an error message if you use words in your prompt that are in some censored lists. For example the word "sexy" is forbidden and will lead to an error message.

Just use the word "ugly" instead which is no problem.

3. For the local Stable diffusion, a Graphic card like an GTX 3080 is the Minimum and will possibly only work for Pictures not larger then 512x512.

To produce pictures of Size 1024x1024 possibly a larger VRAM of at least 12 GB may be needed.

4. For the Cloud service no Graphic card is needed as all calculations are done on the server. However each picture costs a very small amount of money, its less then a Pence i think from what i have seen.

5. For SDL. you do not need any Key or authentication, and you can share your SPR-Scripts with anybody.

6. For SDO. you need an API-Key and its recommended that you use the **SDO . Save Key** Command to save it in your Script Folder so its accessible for the SPR.

Do not spread these Scripts as anybody could use your API-Key and generate Pictures on your cost.

If you compile **EXE-Files** its necessary to use the **' #INC: Compiler-Statement** to include the API-Key into the Executable.

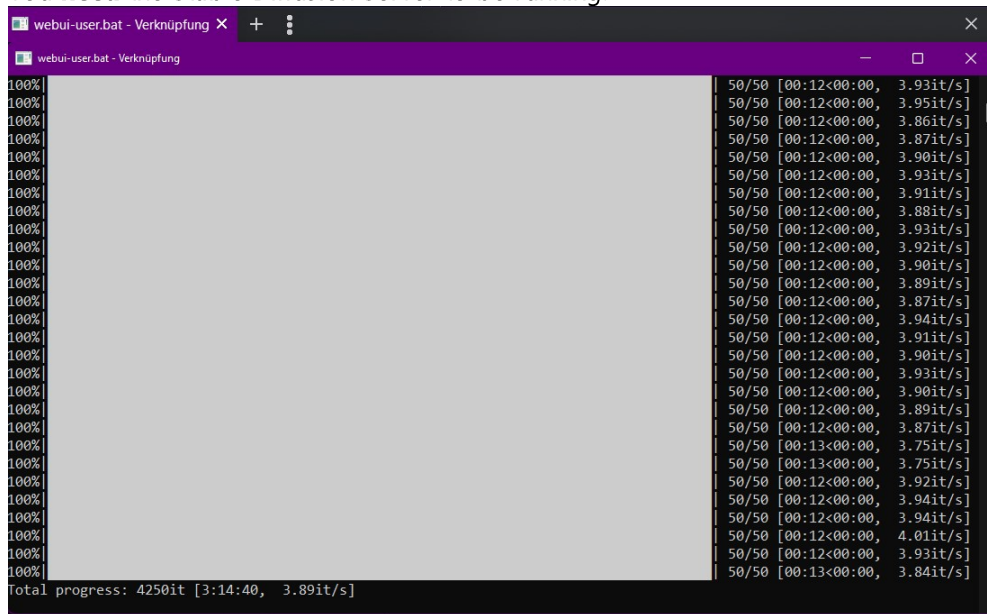
You can also place it aside of the executable that will also work.

Our first Script using SDL.

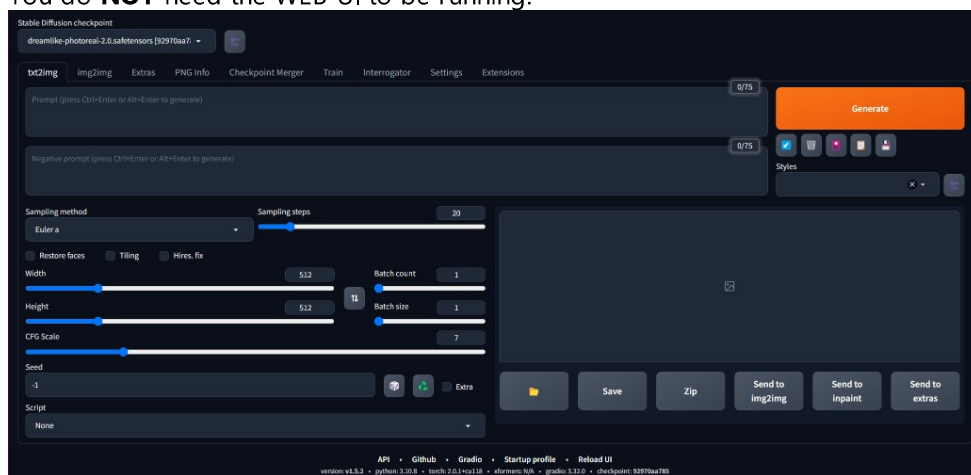
Lets start with SDL. (Local Stable Diffusion). I assume that you have installed it on your computer and that it works from the WEB-GUI.

I also assume that you have added the `--api` Parameter to the batch-file as described in the Installation chapter.

You **need** the Stable Diffusion Server to be running:



You do **NOT** need the WEB-UI to be running:



Now copy this Script below and run it.

```
' Here the propt does not have weights, also note that local version is uncesored
$$CON=croatian
$$PRO=sexy $$CON girl,rainbow colored hair,mountain and adria background,fotograf
$$NEG=b/w,old(.5),fat(.3).Twins,twice,two
SDL.Set Size free|512|512
```

```
' We choose 50 Steps
SDL.SetSteps|50
$$FIL=?path\Opa_?.jpg
$$FIN=1024
$$COL=3
$$ROW=2
```

```
' We make 4 Pictures in one Run and we restore the faces.
$$EXT="save_images": true,
$$EXT+"batch_size": $$ROW,
```



```

$$EXT+"n_iter": $$COL,
' Restore Faces using AI
$$EXT+"restore faces": true,
' Images im Default path von SD speichern
$$EXT+"save_images": true,
SDL.Set Extra Parameter|$$EXT
SDL.gtf|$$PRO|$$FIL|$$NEG|7
SDL.Set Scale|25
POP.$$PAT
POP.$$SUC
IVV.$$SUC=1
  ANA.load|0|$$PAT
  CAL.$$FIX=(($$FIN/$$ROW)*$$COL)
  ANA.ResizeTo|0|1|$$FIX|$$FIN
  %PicPrint SDL. generated.
  ANA.show|1
  ANA.Save|1|$$PAT
ELS.
  SDL.Get Several|6|$$RET
  DBP.$$RET
  SDL.Show Error
  MBT.No File generated.
EIF.
MBX.!!
ENR.

' Macro to print into the Picture

: %PicPrint 1
$$IMR=1
$$TXT=$$$01
$$COA=&HFFFFFF
$$COB==&H0
$$BGC=-2
$$XPA=30
$$YPA=550
ANA.PrintAt|$$IMR|$$TXT|$$COA|$$BGC|$$XPA|$$YPA|24
ANA.PrintAt|$$IMR|$$TXT|$$COB|$$BGC|($$XPA+1)|($$YPA+2)|24
END%

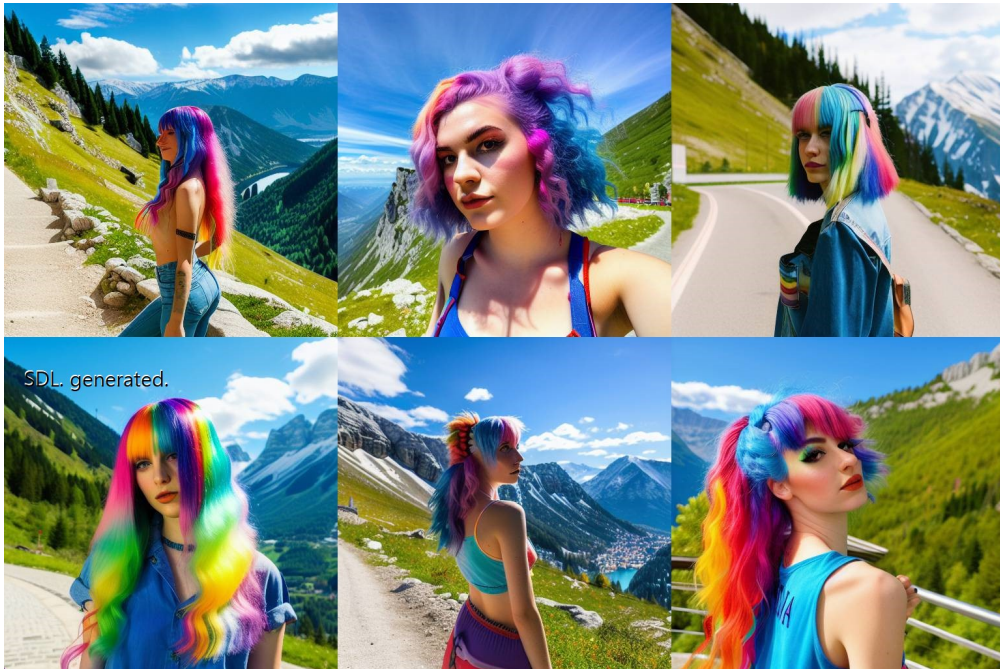
ENR.
'=====

```

Due to the Use of:

```
SDL.Set Extra Parameter|$$EXT
```

we will get a result of this picture through the API (as Output of the Smart Package Robot:



Using SDL, the Script will in this Case only output a Picture that contain all the other pictures. The full size single pictures will be stored in the original Stable Diffusion Output folder.

as we have defined:

```
$$ROW=2
$$COL=3
```

' and: (watch the "," at the end of each line!)
' we directly define [JSON Content](#)^[1348] here, thats why the "," is needed.

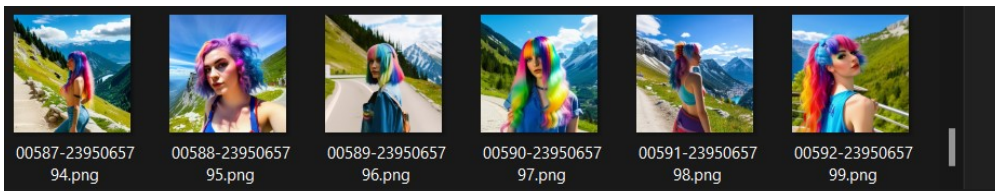
```
$$EXT="save_images": true,
$$EXT+"batch_size": $$COL,
$$EXT+"n_iter": $$ROW,
```

we will get 4 Rows and 5 Columns.

This is nearly the maximum SD will produce (5x5 may work as well).

All generated pictures will be stored in full size in the stable diffusion path:

```
' Last Folder is the Date
...\Stable_Diffusion\Automatic1111\stable-diffusion-webui\outputs\txt2img-images
```

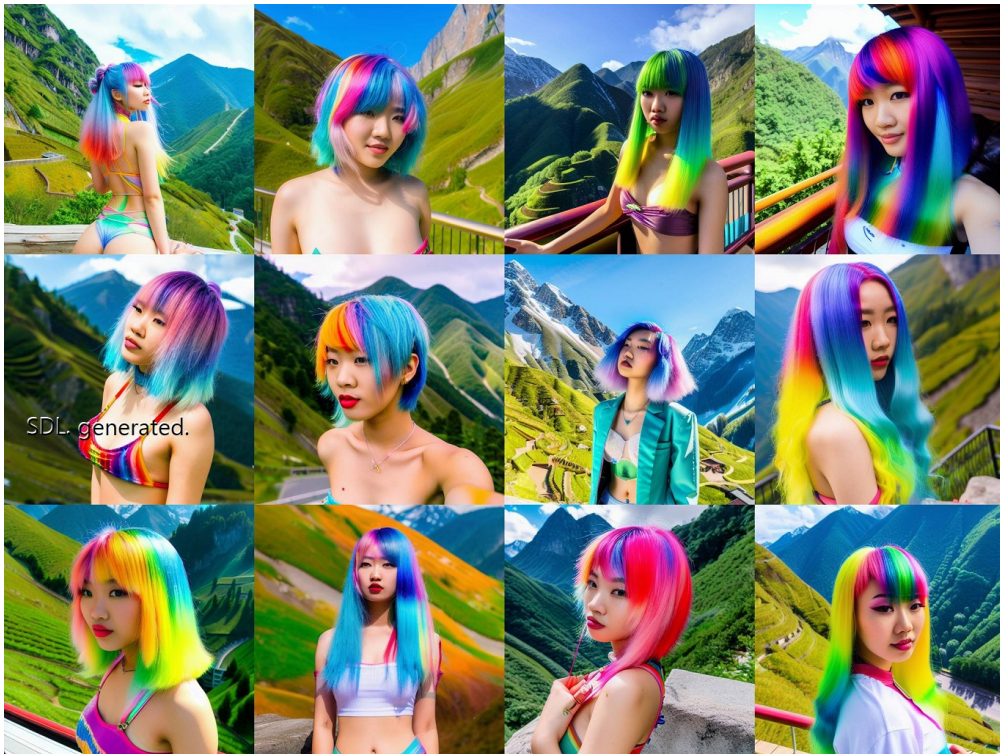


Depending on Parameters. all generated pictures may be stored in the original Stable Diffusion Folder.

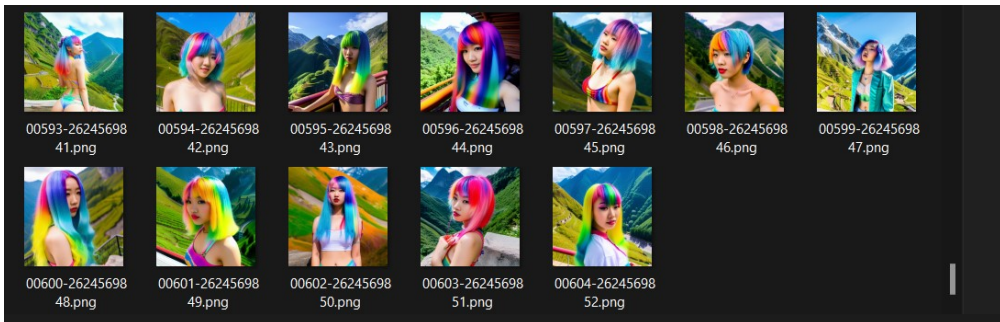
Now lets do some small changes so you can see what each Parameter will do.

```
$$CON=malaysian
$$PRO=sexy $$CON girl,rainbow colored hair,mountain and adria background,fotogra
$$NEG=b/w,old(.5),fat(.3).Twins,twice,two
SDL.Set Size free|512|512
```

```
' We choose 50 Steps  
SDL.SetSteps|50  
$$FIL=?path\Pic_?.jpg  
$$FIN=1024  
$$COL=4  
$$ROW=3
```



Now we get 4 Columns and 3 Rows via API.



And all pictures in the original Stable Diffusion Folder.

Due to the internal working of SD this will not work in the same way with all possible numbers for \$\$ROW and \$\$COL.

Lets now just generate a Single picture.

We are going to change the following lines:

```
' Here the propt does not have weights, also note that local version is uncesore  
$$CON=singapurian
```

```

$$PRO=sweet $$CON girl, riding a dragon,mountain and adria background,fotography
$$NEG=b/w,old(.5),fat(.3).Twins,twice,two
' Resolution which SD will render the pictures
SDL.Set Size free|512|512

' We choose 50 Steps
SDL.SetSteps|50
$$FIL=?path\Pic_?.jpg
' Resizing Resolution (512x512)
$$FIN=512

' We make 4 Pictures in one Run and we restore the faces.
$$EXT="save_images": true,
' Restore Faces using AI
$$EXT+"restore faces": true,
' Images im Default path von SD speichern
$$EXT+"save_images": false
SDL.Set Extra Parameter|$$EXT
SDL.gtf|$$PRO|$$FIL|$$NEG|7
SDL.Set Scale|15
POP.$$PAT
POP.$$SUC
IVV.$$SUC=1
  ANA.load|0|$$PAT
  ANA.Resizeto|0|1|$$FIN|$$FIN
  %PicPrint SDL. generated.
  ANA.show|1
  ANA.Save|1|$$PAT
ELS.
  SDL.Get Several|6|$$RET
  DBP.$$RET
  SDL.Show Error
  MBT.No File generated.
EIF.
MBX.!
ENR.

' Macro to print into the Picture

: %PicPrint 1
$$IMR=1
$$TXT=$$$01
$$COA=&HFFFFFF
$$COB==&H0
$$BGC=-2
$$XPA=30
$$YPA=850
ANA.PrintAt|$$IMR|$$TXT|$$COA|$$BGC|$$XPA|$$YPA|44
ANA.PrintAt|$$IMR|$$TXT|$$COB|$$BGC|($$XPA+1)|($$YPA+2)|44
END%

ENR.
'=====

```




Here we got only one picture via the API and we did not get any picture in the Original SD-Folder.

Special cases that occur when you use both Systems:

While this is a general rule, we have taken a lot of internal steps to make it as easy for you to switch between both systems at any time.

Therefore we have quite a lot of commands that are common to both systems.

These commands are most often available for SDO. as well as for SDL.

Some of them work on internal registers that are used by both systems. For example:

[SDL.Set Positive Prompt](#)^[1181]

This will write the Prompt into an internal Register that is used by both Systems of Stable Diffusion.

Also there are Some commands that are shared between both systems, but use an own Set of registers, for example:

[SDL.Prefix Positive Prompt](#)^[1174]

This command is available with SDO. and with SDL. but each of these uses an own register for the Prefix.

3.42.14.4 ! How to use Stable Diffusion Part I SDO.

[How to use Stable Diffusion with th](#)

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

How to use Stable Diffusion with the SPR, Part I: SDO.

Here is a first Script you can run to get started

The Smart Package Robot supports 2 versions of Stable Diffusion

1. The Local Stable Diffusion, this needs the Automatic1111 to be [installed and having the --api switch set in the "webui.bat"](#) and you need a usable Graphics card with enough VRAM
2. The Stable Diffusion Online, this will need that you [to get an API-Key to access the Cloud Services](#).

Depending on which service you want to use, you need to [whether get the API-Key](#) or [install the Automatic1111](#) System on your computer or in your network.

In case you want to use Stabel Diffusion Online, the command of your choice is [SDO](#).

If you decide the Stable Diffusion that runs Local - or in your Network, the Command of your choice is [SDL](#).

What are the differences?

1. **Local** Stable Diffusion is **uncensored**, so you can produce pictures of whatever you like, this includes naked pictures or pictures that are forbidden due to whatever ideologies.
2. Stable diffusion **online** is a **cloud service** and as such must uphold the **rules for cloud services**.

While it can also produce naked pictures in some cases (unless you add that to the negative prompt) from time to time, it will send you an error message if you use words in your prompt that are in some censored lists. For example the word "sexy" is forbidden and will lead to an error message.

Just use the word "ugly" instead which is no problem.

3. For the local Stable diffusion, a Graphic card like an GTX 3080 is the Minimum and will possibly only work for Pictures not larger then 512x512.

To produce pictures of Size 1024x1024 possibly a larger VRAM of at least 12 GB may be needed.

4. For the Cloud service no Graphic card is needed as all calculations are done on the server. However each picture costs a very small amount of money, its less then a Pence i think from what i have seen.

5. For SDL. you do not need any Key or authentication, and you can share your SPR-Scripts with anybody.

6. For SDO, you need an API-Key and its recommended that you use the **SDO.Save Key** Command to save it in your Script Folder so its accessible for the SPR.

Do not spread these Scripts as anybody could use your API-Key and generate Pictures on your cost.

If you compile **EXE-Files** its necessary to use the '#INC: Compiler-Statement to include the API-Key into the Executable.

You can also place it aside of the executable that will also work.

Our first Script using SDO.

Lets start with SDO. (Stable Diffusion Online). I assume that you have got your [API-Key for Stable Diffusion Online](#)^[1391].

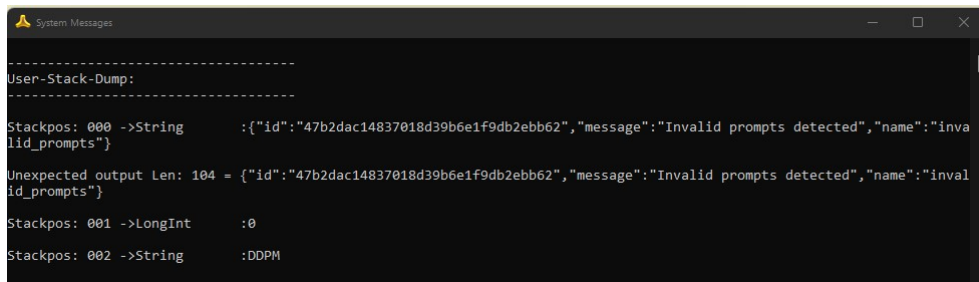
The first thin you need to do, is to save your API Key in your Script folder, using the [SDO.Save Key Command](#)^[1440].

We will start with the same Prompt like in **Part I, SDL**.

```
$$PRO=Sexy $$CON girl,rainbow colored hair,mountain and adria background,fotogra
```

Doing so will deliver a Error-Message in return from the server.

Cloud services are bound to several rules and not all words are allowed here.



```
System Messages
-----
User-Stack-Dump:
-----
Stackpos: 000 ->String      :{"id":"47b2dac14837018d39b6e1f9db2ebb62","message":"Invalid prompts detected","name":"invalid_prompts"}
Unexpected output Len: 104 = {"id":"47b2dac14837018d39b6e1f9db2ebb62","message":"Invalid prompts detected","name":"invalid_prompts"}
Stackpos: 001 ->LongInt     :0
Stackpos: 002 ->String      :DDPM
```

Using the same Prompt like with SDL, we will get this error message. It shows that we have used forbidden/censored words.

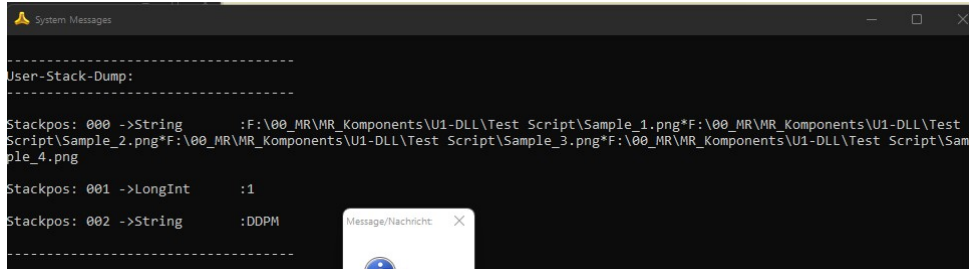
So lets change the words a bit:

```
SDO.SetKey|File
SDO.SetSize|512|512|stable-diffusion-v1-5
$$CON=croatian
$$PRO=beautiful $$CON girl,rainbow colored hair,mountain and adria background,fotogra
$$NEG=ugly,comic,unrealistic,fat,unhealthy,malformed faces
' The "?" will be replaced by a number
$$FIL=?exeloc\Sample_?.png
```

```
SDO.SetSteps|100
' We can produce up to 10 Pictures in one run!
SDO.Set Samples|4
POP. $$SAM
SDO.Set_Sampler|1
'SDL.Set Model Free|dreamlike-photoreal-2.0.safetensors
'SDL.Set Extra Parameter|"restore_faces": true
VAN. $$TIM=#dtime#
SDO.gtf|$$PRO|$$FIL|$$NEG|6
VAN. $$TIM=#dsince#|i
DBP. Set $$SAM Samples. In $$TIM Seconds.
```

DMP.6
 MBX.!

ENR.



```

System Messages
-----
User-Stack-Dump:
-----
Stackpos: 000 ->String      :F:\00_MR\MR_Komponents\U1-DLL\Test_Script\Sample_1.png*F:\00_MR\MR_Komponents\U1-DLL\Test_Script\Sample_2.png*F:\00_MR\MR_Komponents\U1-DLL\Test_Script\Sample_3.png*F:\00_MR\MR_Komponents\U1-DLL\Test_Script\Sample_4.png
Stackpos: 001 ->LongInt     :1
Stackpos: 002 ->String      :DDPM
-----
  
```

This time the result is as expected. We find the names of the generated pictures on the TOS.



Just one of the 4 generated Pictures using SDO.

Next lets take the same Script we used before for **SDL**. and change it to **SDO**.
 Here is the complete Script:

```

SDO.SetKey|File
SDO.SetSize|512|512|stable-diffusion-v1-5
$$CON=Escimo
$$PRO=beautiful $$CON girl,rainbow colored hair,mountain and adria background,fo
  
```



```

$$NEG=ugly,comic,unrealistic,fat,unhealthy,malformed faces
SDO.Set Samples|1

' We choose 50 Steps
SDO.SetSteps|70
$$FIL=?path\SDO_Pics_?.jpg
$$FIN=512

' We use Style 7 or 14
SDO.gtf|$$PRO|$$FIL|$$NEG|7
SDO.Set Scale|15
POP.$$PAT
POP.$$SUC
IVV.$$SUC=1
  ANA.load|0|$$PAT
  ANA.ResizeTo|0|1|$$FIN|$$FIN
  %PicPrint SDL. generated.
  ANA.show|1
  ANA.Save|1|$$PAT
ELS.
  SDL.Get Several|6|$$RET
  DBP.$$RET
  SDL.Show Error
  MBT.No File generated.
EIF.
MBX.!
ENR.

' Macro to print into the Picture

: %PicPrint 1
$$IMR=1
$$TXT=$$$01
$$COA=&HFFFFFF
$$COB==&H0
$$BGC=-2
$$XPA=30
$$YPA=450
' We have changed some values here due to the Size of the picture
ANA.PrintAt|$$IMR|$$TXT|$$COA|$$BGC|$$XPA|$$YPA|24
ANA.PrintAt|$$IMR|$$TXT|$$COB|$$BGC|($$XPA+1)|($$YPA+2)|24
END%

ENR.

```

Comments about the Script:

We had to remove this:

```

' Images im Default path von SD speichern
$$EXT+"save_images": false
$$EXT+"save_images": true,
' Restore Faces using AI
$$EXT+"restore faces": true,
SDO.Set Extra Parameter|$$EXT

```

as this is not supported from **SDO**.





This is the picture we got using Style 14 (photographic).
Here we used Style 7 - Fantasy Art

3.42.14.5 ! Understanding Samplers

[Understanding Samplers](#)[Previous](#) [Top](#) [Next](#)

Technical Background (Info)

Understanding Samplers in SDL. and SDO.



Generated using the SPR and SDL-Command

Introduction

Samplers play a crucial role with AI generated Images. They are responsible for various sampling methods, each with its unique characteristics and applications. They are the backbone of the system, determining how images are processed and refined.

This chapter aims to provide a simplified understanding of these samplers, helping users make informed decisions about which one to use.

What is Sampling?

- **Definition:** The sampler is responsible for carrying out the denoising steps. It gradually produces cleaner and cleaner images through a denoising process. At its core, sampling is about refining images. The sampler oversees the denoising steps, ensuring that with each iteration, the image becomes clearer and more defined.
- **Noise Schedule:** This is a predetermined schedule that dictates the noise level at each sampling step. For instance, a noise schedule for 15 sampling steps starts with the highest amount of noise, which gradually decreases to zero by the last step.

Think of this as a roadmap for the sampler.

It's a predetermined plan that indicates the noise level at each step of the sampling process.

For instance, a 15-step noise schedule would commence with a high noise level, which would then taper off, reaching zero by the 15th step. This schedule can be adjusted based on the number of steps, with more steps leading to a smoother noise reduction process.

SDL.-Samplers (Local Samplers)

Index	validSampler
0	Euler
1	Euler a
2	LMS
3	Heun
4	DPM2
5	DPM2 a
6	DPM++ 2S a
7	DPM++ 2M
8	DPM++ SDE
9	DPM++ 2M SDE
10	DPM fast
11	DPM adaptive
12	LMS Karras
13	DPM2 Karras
14	DPM2 a Karras
15	DPM++ 2S a Karras
16	DPM++ 2M Karras
17	DPM++ SDE Karras
18	DPM++ 2M SDE Karras
19	DDIM
20	PLMS
21	UniPC

Overview of Samplers

- Old-School ODE Solvers:

These are traditional solvers for ordinary differential equations (ODE) that have been around for over a century.

Examples include Euler, Heun, and LMS.

- Ancestral Samplers:

These samplers have names with a single letter “a” (e.g., Euler a, DPM2 a). They are stochastic in nature. However, images generated using ancestral

samplers like `Euler a` do not converge at high sampling steps, making them less desirable for reproducibility.

Therefore, a significant drawback is their inability to ensure image convergence at high sampling steps, which can be a hurdle for those seeking consistent reproducibility.

- **Karras Noise Schedule:** Introduced by Karras, this is a specialized noise schedule that offers a distinct approach to noise management during sampling.

- **DDIM and PLMS:** Though once popular, these samplers have now fallen out of favor and are considered relics of the past.

- **DPM and DPM++:** These represent the new age of samplers. They come in various iterations and versions, each with its unique features.

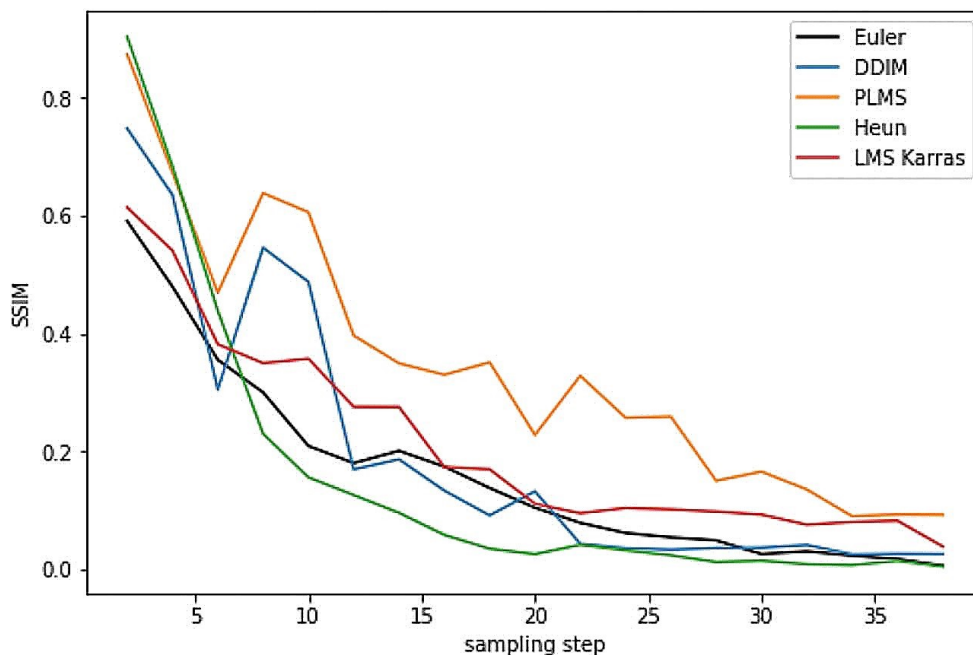
- **UniPC:** This sampler supports any solver and noise predictors. Its a versatile sampler, making it a valuable tool in the Samplers arsenal.

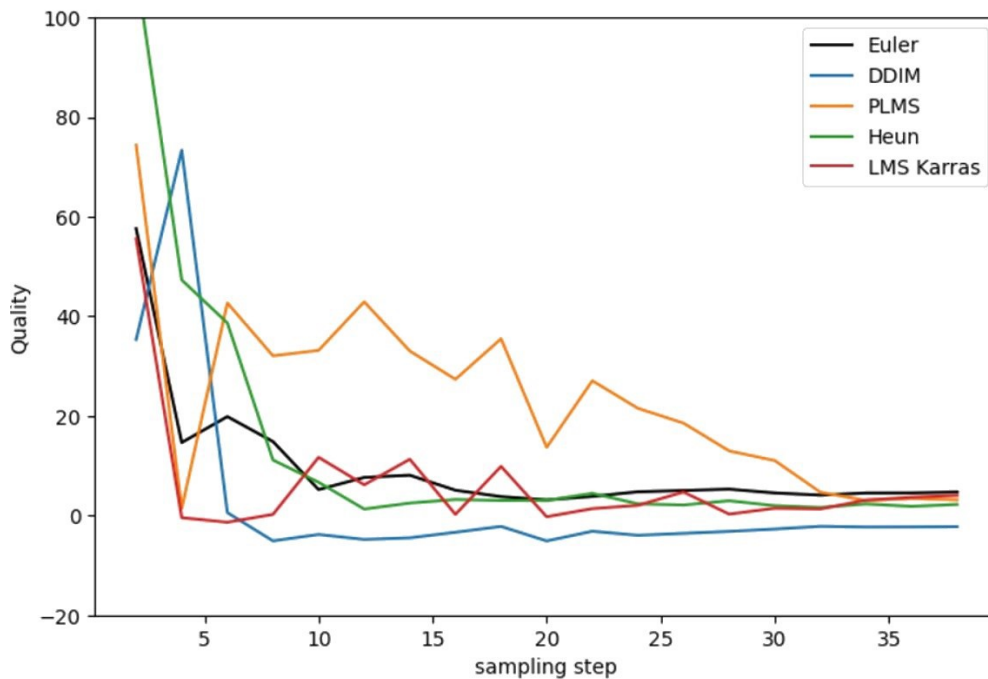
Evaluating Samplers

- **Image Convergence:** This refers to how quickly a sampler can produce a clear image. For instance, DDIM can produce a high-quality image in as few as 8 steps.

- **Speed:** Some samplers, especially 2nd order solvers, might be slower as they evaluate the denoising U-Net twice.

- **Quality:** The ultimate goal is to produce high-quality images. For example, ancestral samplers tend to converge to an image of a kitten, while deterministic ones converge to a cat.





Looking at this graph the Heun Sampler looks superior. In a third dimension, it needs twice as much time as other Samplers. Right Chart: The image quality of ancestral samplers (Lower the better).

Therefore we must compare it with other samplers and a twice as high rate of steps.

Sampler Highlights

- Euler**: The most straightforward sampler. It operates deterministically, reducing noise according to a predetermined noise schedule.
- DDIM (Denosing Diffusion Implicit Models)**: This sampler approximates the final image with a denoised image during its steps.
- LMS and LMS Karras**: These are linear multistep methods.
- DPM Samplers**: These are Diffusion Probabilistic Model Solvers with various versions available.
- UniPC**: This sampler consists of a Unified predictor (UniP) and a Unified corrector (UniC).

Detailed Recommendations:

Quick Reference Table:

Priority	Recommended Sampler	Steps	Additional Notes
Fast, Converging, New, Quality	DPM++ 2M Karras	20-30	

	UniPC	20-30	
High Quality (No Convergence)	DPM++ SDE Karras	10-15	Slower sampler
	DDIM	10-15	
Stability & Reproducibility	Avoid Ancestral Samplers	-	Ensures stable, reproducible images
Simplicity	Euler	Variable	Simple and straightforward
	Heun	Reduced	Reduce steps to save time

Explanation:

When selecting a sampler, it's essential to consider your priorities, whether it's **speed, convergence, quality, or simplicity**.

Here's a guide to help you make an informed decision:

For Speed, Convergence, and Quality:

If you're looking for a sampler that's both modern and efficient, the **DPM++ 2M Karras** and **UniPC** are excellent choices.

For optimal results, use them with **20-30 steps**.

For Quality Without Convergence:

If your primary concern is the quality of images and you're less bothered about convergence, the **DPM++ SDE Karras** and **DDIM** are your best bets.

However, note that the **DPM++ SDE Karras** is a bit slower in its operation. For these samplers, **10-15 steps** are ideal.

For Stability and Reproducibility:

If you prioritize stable and reproducible images, it's advisable to steer clear of any ancestral samplers.

For Simplicity:

If you're a fan of simplicity and straightforwardness, **Euler** is a great choice.

Heun is another simple option, but to save time, you might want to reduce the number of steps when using it.

With this guide, you can now select the sampler that aligns best with your requirements.

Source: [\[Stable Diffusion Art\]](#)

3.42.14.6 SD - Shared Commands

[Shared Commands and Parameters](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

Shared Parameters Between Stable Diffusion SDO. (Online) and SDL. (Local)

Some Parameters are shared between Stable Diffusion Local (SDL.) and Stable Diffusion Cloud (SDO.)

Shared Parameters: An Overview

Stable Diffusion, whether processed locally using SPR commands (SDL.) or online through the Stable Diffusion Cloud API (SDO.), operates based on a foundational set of principles and algorithms.

To maintain consistency, reproducibility, and ease of transition between local and online environments, certain parameters are shared between the two modalities.

Why Share Parameters?

- **Consistency:** Sharing parameters ensures that users get consistent results regardless of where the Stable Diffusion process is executed. It facilitates a seamless transition between local and cloud-based processing.
- **Reproducibility:** Critical in scientific and research contexts, shared parameters ensure experiments or simulations can be reproduced with the same settings across different platforms.
- **Unified Experience:** For users transitioning between local and online processing, a unified set of parameters means a shorter learning curve and fewer discrepancies.
- **Optimization:** Standardizing parameters ensures that optimizations made in one environment (e.g., cloud) can be beneficially applied to the other (e.g., local).
- **Ease of Collaboration:** Shared parameters simplify collaboration. A process initiated locally can be replicated or extended online using the same settings.

Shared Parameters:

- **Seed:** Dictates the initiation point for the random number generation sequence, facilitating reproducibility in the randomness used by the Stable Diffusion process.
- **Samples:** Specifies the granularity or resolution for the Stable Diffusion process, striking a balance between computational speed and precision.
- **Steps:** Sets the number of iterations or cycles the algorithm should undergo, influencing the depth of the diffusion process.
- **Height & Width:** Determines the dimensions of the output image or pattern, ensuring consistent sizes across platforms.
- **Style:** Governs the aesthetic or pattern style utilized during the diffusion process.
- **cfg_scale:** A scaling parameter influencing the intensity or scale of features in the Stable Diffusion process.
- **clip_guidance_preset:** Provides directives on constraining or emphasizing certain features during the diffusion process.

- **Positive** and **Negative Prompt**: While not identical across platforms, these prompts can be conceptually shared, guiding the direction and outcome of the Stable Diffusion process to a certain degree.

Remarks

The shared parameters between the Stable Diffusion API (Online) and the SPR commands (Local) demonstrate the robustness and adaptability of the Stable Diffusion process. With this shared foundation, users can transition between environments with confidence, ensuring consistent and high-quality results.

This way you can use your Script with very few changes local or in the Cloud.

See also:

-

3.42.14.6.1 !SD - Modify Prompt

Modify the Prompt using Prefixes and Suffixes [Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

Modify the Positive Prompt using Prefixes and Suffixes

Using these commands you can either temporarily or permanently modify the prompt to add a style or medium or artist to it

Intention

Using these commands you can either temporarily or permanently modify the prompt to add a style or medium or artist to it.

3.42.14.6.1.1 Append negative Prompt

[SDL.Append Negative Prompt and .](#) [Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL.Append Negative Prompt and SDO.Append Negative Prompt

Immediately appends the given text to the Negative Prompt register.

Intention

The Append Negative Prompt command (SDL.Anp/SDO.Anp) is designed to immediately append the given text to the existing Negative Prompt register. This command only influences the current Negative Prompt and is not a permanent change.

Syntax

SDL.Append Negative Prompt [| P1]
SDO.Append Negative Prompt [| P1]

Parameter Explanation

P1: (Optional) The text that will immediately append to the existing Negative Prompt. If omitted or empty, no change will occur.

Example

```
! *****  
!  
! *****
```

Remarks

-

Limitations:

-

See also:

-

3.42.14.6.1.2 Append positive Prompt

[SDL.Append Positive Prompt / SDL.](#)

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL.Append Positive Prompt and SDO.Append Positive Prompt

Immediately appends the given text to the Positive Prompt register.

Intention

The Append Positive Prompt command `SDL.App/SDO.App` is designed to immediately append the given text to the existing Positive Prompt register. This command only influences the current Positive Prompt and is not a permanent change.

Syntax

```
SDL.Append Positive Prompt [|P1]
SDO.Append Positive Prompt [|P1]
```

Parameter Explanation

P1: (Optional) The text that will immediately append to the existing Positive Prompt. If omitted or empty, no change will occur.

Example

```
'*****
' SDL.-Sample for appending to positive prompt
'*****
' Also available as SDL.App
SDL.Get Medium|1|$$MED
SDL.Append Positive Prompt|$$MED
' The Positive Prompt will now have "A Photorealistic Painting," appended.
ENR.
```

Remarks

-

Limitations:

-

See also:

- [1.6.1. Program Flow Control](#)^[134]
- [! Smart Package Robot 's Parallel Robot Operations](#)^[3745]
- [1.5. Features and Hints](#)^[93]

3.42.14.6.1.3 Get negative Prompt

SDL.Get Negative Prompt[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL.Get Negative Prompt

Retrieves the content of the internal Negative Prompt register

Intention

The `SDL.Get Negative Prompt` command is designed to fetch the content of the existing Negative Prompt register. Unlike other commands, this function allows you to access the register without the permanent Prefix and Append-Register affecting the result.

Syntax

SDL.Get negative Prompt[|P1]

Parameter Explanation

P1: (Optional) Variable where the result will be stored. If omitted, the result is placed on the Top Of Stack (TOS).

Example

```

' *****
' SDL.-Sample for retrieving the negative prompt
' *****
SDL.Get Negative Prompt|$$MED
' The Negative Prompt content will now be stored in $$MED
ENR.

```

Remarks

-

Limitations:

-

See also:

-

3.42.14.6.1.4 Get positive Prompt

[SDL.Get Positive Prompt / SDL.Gpp](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL.Get Positive Prompt / SDL.Gpp

If Result available

Intention

The `SDL.Get Positive Prompt` command is designed to fetch the content of the existing Positive Prompt register.

This function allows you to access the register without the permanent Prefix and Append-Register affecting the result.

Syntax

SDL.Get Positive Prompt[|P1]

Parameter Explanation

P1: (Optional) Variable where the result will be stored. If omitted, the result is placed on the Top Of Stack (TOS).

Example

```

'*****
' SDL.-Sample for retrieving the positive prompt
'*****
SDL.Get Positive Prompt|$$MED
' The Positive Prompt content will now be stored in $$MED
ENR.

```

Remarks

-

Limitations:

-

See also:

-

3.42.14.6.1.5 Prefix negative Prompt

[SDL.Prefix Negative Prompt / SDL.](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL.Prefix Negative Prompt and SDO.Prefix Negative Prompt

Immediately prefixes the Negative Prompt register with the given text.

Intention

The Prefix Negative Prompt command `SDL.Pnp/SDO.Pnp` is designed to immediately prefix the existing Negative Prompt register with the given text. This command only influences the current Negative Prompt and is not a permanent change.

Syntax

```
SDL.Prefix Negative Prompt|P1
SDO.Prefix Negative Prompt|P1
```

Parameter Explanation

P1: The text that will immediately prefix the existing Negative Prompt. If omitted or empty, no change will occur.

Example

```
' *****
' SDL.-Sample for prefixing negative prompt
' *****
' Also available as SDL.Pnp
SDL.Get Medium|1|$$MED
SDL.Prefix Negative Prompt|$$MED
' The Negative Prompt will now be prefixed with "A
Photorealistic Painting "
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.14.6.1.6 Prefix positive Prompt

[SDL.Prefix Positive Prompt / SDL.Pf](#)
[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL.Prefix Positive Prompt and SDO.Prefix Positive Prompt

Immediately prefixes the Positive Prompt register with the given text.

Intention

The Prefix Positive Prompt command `SDL.Ppp` is designed to immediately prefix the existing Positive Prompt register with the given text.

This command only influences the current Positive Prompt and is not a permanent change.

This command will only append a " " (Space) to the end of **P1**. If you want a "," coma to be appended, add it after **P1**.

Syntax

```
SDL.Prefix Positive Prompt|P1
SDO.Prefix Positive Prompt|P1
```

Parameter Explanation

P1: The text that will immediately prefix the existing Positive Prompt. If omitted or empty, no change will occur.

This command will only append a " " (Space) to the end of **P1**. If you want a "," coma to be appended, add it after **P1**.

Example

```
*****
' SDL.-Sample for prefixing the positive prompt
*****
' Also available as SDL.Pnp
SDL.Get Medium|1|$$MED
SDL.Prefix positive Prompt|$$MED,
```

' The Positive Prompt will now be prefixed with "A Photorealistic Painting, " ENR.

Remarks

-

Limitations:

-

See also:

•

3.42.14.6.1.7 Set Append

[SDL.Set Append / SDL.Sap and SDO](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL.Set Append and SDO.Set Append

If Result available

Intention

The SDL.Set Append command and its shorthand SDL.sap are designed to define an internal Append-Register, that will be permanently applied to each prompt until the Append-Register is cleared..

Before adding the Append, the command will internally remove all "," from the Append and the Positive-Prompt Register, that is set using `SDL.Set Positive Prompt`

This command **is available in both the Stable Diffusion Local (SDL.spx) and Stable Diffusion Online (SDO.spx) suites**. Each suite has its **own internal register for storing the Append**, and setting a Append in one suite does not influence the other.

```
SDL.Set Positive Prompt|Green wood
SDL.Set Append|fantasy landscape
' The final prompt will be
' "Green wood,fantasy landscape"
ENR.
```

Syntax

```
SDL.Set Append [ | P1 ]
SDL.Spx [ | P1 ]
SDO.Set Append [ | P1 ]
SDO.Spx [ | P1 ]
```

Parameter Explanation

P1: (Required) The Append that will be permanently applied to each prompt. If omitted or empty the Append-Register will be cleared.

Example

```
*****
' SDL.-Sample for setting Append
*****
SDL.Set Positive Prompt|Green wood
SDL.Set Append|fantasy landscape
' The final prompt will be "Green wood,fantasy landscape"
ENR.

*****
' SDO.-Sample for setting Append
*****
SDO.Set Append|science fiction
' All subsequent prompts will now be Appended with "science fiction,"
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.14.6.1.8 Set Negative Prompt

[SDO.Set Negative Prompt](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDO.Set Negative Prompt

If Result available



Intention

The "negative prompt" is a parameter that tells the Stable Diffusion model what not to include in the generated image.

It is a way to use Stable Diffusion in a way that allows the user to specify what he doesn't want to see, without any extra input.

At its core, a negative prompt is an instruction to the Stable Diffusion model to exclude specific elements or features from the generated image.

Essentially, it's like telling the model, "**Create this, but don't include that**".

Stable Diffusion Negative Prompts specify what to exclude from the output using Prompts, such as

"bad anatomy, disfigured, blurry, cloned face, low contrast, over/underexposed"

to modify the image and style or sometimes to remove things from the artwork. Therefore, using the Negative Prompt has a more significant implication on your desired output.

The Ultimate Guide to "Negative Prompts" in Stable Diffusion: The Art of Saying "Nope!"

🗣️ Intro to the World of AI Artistry: Ever tried painting a picture with words? That's what Stable Diffusion does! But sometimes, it gets a little too creative. Enter: Negative Prompts!

🗣️ What's a Negative Prompt? Imagine telling a chef, "Make me a pizza, but no pineapples!" That's a negative prompt. In Stable Diffusion, it's like saying, "Paint me a sunset, but no UFOs!"

🗣️ Why Use Negative Prompts?

- **Counterfactual Fun:** "What if unicorns didn't have horns?" It pushes our AI artist to think outside the box.
- **Bias Busters:** Keeps our AI from jumping to conclusions or being too stereotypical.
- **Expand the Story:** "Tell me more about that dragon's vegan diet!"
- **Quality Control:** Sometimes, the AI's masterpiece needs a little touch-up.

🗣️ Steps to Mastering the Art of Saying "Nope!" in Stable Diffusion:

- **Get the Basics:** Know what a negative prompt is.
- **Pick Your Battles:** Decide where you want to say "nope!"
- **Craft the Perfect "Nope!":** Design your negative prompts.
- **Test, Laugh, Repeat:** See how the AI responds, refine, and have fun!

🗣️ *Choosing the Right "Nope!" Want a serene beach without sharks doing the conga? Add 'sharks'!*

🗣️ *Final Doodles: Negative prompts are like the erasers of the AI art world. They help clean up!*

Syntax

SDO.Set Negative Prompt[|P1]

Parameter Explanation

P1 - (*optional*) The Terms, words and Phrases that build together the Negative Prompt.

Example

```

| *****
| SDO.-Sample
| *****
$$DIR=?exeloc\Pics_Eskimo
NEF.$$DIR\
  MKD.$$DIR
EIF.

```

```

SDO.SetKey|File

'SDO.SetSize|512|512|stable-diffusion-v1-5
SDO.SetSize|1024|1024|stable-diffusion-xl-1024-v1-0

SDO.Set_Samples|1
'SDO.Set Steps|40
SDO.Set Scale|30
SDO.Set Prompt System|0
SDL.Set Extra Parameter|"restore faces": true
$$PRO=beauty Eskimo women(0.2),rainbow colored
hair(0.6),sitting on a large flying dragon(0.2)
$$NEG=blurry, bad, ugly,malformed body,weak,bored,false
legs,false arms, no neck
SDO.Set Negative Prompt|$$NEG
FOR.$$STY|0|16
  SDO.Set Style|$$STY
  SDO.Get Several|5|$$STA
  DBP. We use Style: $$STA
  $$FIL=$$DIR\Pics_$$STA_?.png
  SDO.gst|$$PRO|$$FIL||50
  POP.$$PAT
  POP.$$SUC
  IVV.$$SUC=1
  ANA.load|0|$$PAT
  ANA.show|0
  ELS.
  AIC.Show Error
  MBX.No File generated.
  EIF.
NEX.
MBX.!
ENR.

```

Remarks

-

Limitations:

-

See also:

-

3.42.14.6.1.9 Set Positive Prompt

SDO.Set Positive Prompt

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDO.Set Positive Prompt

Set the internal Register with the Prompt. It can be used with SDO. or SDL.



Intention

This command will set the **Positive Prompt** that is to be used with SDL. / SDO. Commands to generate Pictures.

If you use the `SDO.Set Positive Prompt` Command, then you can leave the field "\$\$PRO" empty, because the prompt is already set.

```
SDO.gst|$$PRO|$$FIL|$$NEG|50
```

If you set the Prompt with other commands then this will be overwritten.

1. The Ultimate Guide to "Positive Prompts" in Stable Diffusion: 1

🧠 **Diving into AI's Creative Pool:** Stable Diffusion is like that friend who draws epic doodles when you whisper ideas in their ear. And with positive prompts, you're giving them the juiciest tidbits!

🎧 **What's a Positive Prompt?** Ever shouted, "DJ, play my jam!" at a party? That's a positive prompt. In Stable Diffusion, it's like saying, "Paint me a sunrise with dancing flamingos!"

🗣️ Why Use Positive Prompts?

- **Direct the Show:** You're the director, and the AI is your star. Tell it exactly what scene you want.
- **Boost Creativity:** "Paint a cat wearing a top hat and monocle." The AI's imagination runs wild!
- **Nail the Details:** "Describe a castle with candy cane towers." Sweet precision!
- **Quality Masterpiece:** Guide the AI to create the pièce de résistance you envision.

🎨 Steps to Mastering the Art of Saying "Yes, Please!" in Stable Diffusion:

- **Know Your Wishes:** Understand what a positive prompt is.
- **Dream Big:** Decide what magic you want on the canvas.
- **Craft the Perfect "Yes!":** Design your positive prompts.
- **Applaud, Giggle, Repeat:** Revel in the AI's creations, refine, and enjoy the show!

🤖 **Choosing the Right "Yes, Please!"** Fancy a jungle with disco-dancing parrots? Add 'disco' and 'parrots' to your positive prompts!

✨ **Final Brushstrokes: Positive** prompts are the paintbrushes of the AI art world. They guide, inspire, and sometimes lead to the most unexpected and delightful masterpieces!

2. Definition of a Positive Prompt:

A positive prompt is a clear directive given to the Stable Diffusion system, specifying what elements or themes should be included in the generated output.

3. Purpose and Benefits:

- **Guided Creativity:** Direct the AI towards a specific theme or element.
- **Precision: Achieve** detailed and specific results.
- **Quality Control:** Ensure the generated content aligns with user expectations.

4. How to Use Positive Prompts:

Use `SDO.Set Positive Prompt|` Clearly type in your desired theme or element.
E.g., `"SDO.Set Positive Prompt|A serene lakeside at dusk."`

5. Tips for Crafting Effective Positive Prompts:

- **Be Specific:** The more detailed your prompt, the closer the output will align with your vision.
- **Experiment:** Don't hesitate to try various prompts to see diverse outputs.
- **Combine Elements:** Merge different themes for unique results, e.g., "A city skyline with floating islands."

6. Troubleshooting:

- **Vague Outputs:** If the result is not as expected, refine your prompt for more specificity.
- **Reiteration:** If unsatisfied, re-enter a modified prompt for a different result.

7. Conclusion:

Positive prompts are essential for harnessing the full potential of Stable Diffusion. By providing clear and detailed instructions, users can achieve tailored and high-quality outputs.

Syntax

SDO.Set Positive Prompt[|P1]

Parameter Explanation

P1 - (*optional*) Positive Prompt Text. There are differences in designing the [Prompt for SDO and SDL](#).^[1429]

Example

```

|*****
| SDO.-Sample
|*****
$$PRO=Circus Tent, Boxing World Championship, Christmas, Top Stars, Poster
SDO:Set Positive Prompt|$$PRO

```

Remarks

-

Limitations:

-

See also:

-

3.42.14.6.1.10 Set Prefix

[SDL.Set Prefix / SDL.Spx and SDO..](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL.Set Prefix and SDO.Set Prefix

If Result available

Intention

The `SDL.Set Prefix` command and its shorthand `SDL.spx` are designed to set a prefix that will be permanently applied to each prompt until the Prefix-Register is cleared.. Before adding the prefix, the command will internally remove all "," from the prefix and the Positive-Prompt Register, that is set using `SDL.Set Positive Prompt`

This command **is available in both the Stable Diffusion Local (SDL.spx) and Stable Diffusion Online (SDO.spx) suites.**

Each suite has its **own internal register for storing the prefix**, and setting a prefix in one suite does not influence the other.

```
SDL.Set Positive Prompt|Green wood
SDL.Set Prefix|fantasy landscape
' The final prompt will be
' "fantasy landscape,Green wood"
ENR.
```

Syntax

```
SDL.Set Prefix[|P1]
SDL.Spx[|P1]
SDO.Set Prefix[|P1]
SDO.Spx[|P1]
```

Parameter Explanation

P1: (Required) The prefix that will be permanently applied to each prompt. If omitted or empty the Prefix-Register will be cleared.

Example

```
!*****  
' SDL.-Sample for setting prefix  
!*****  
SDL.Set Positive Prompt|Green wood  
SDL.Set Prefix|fantasy landscape  
' The final prompt will be "fantasy landscape,Green wood"  
ENR.  
  
!*****  
' SDO.-Sample for setting prefix  
!*****  
SDO.Set Prefix|science fiction  
' All subsequent prompts will now be prefixed with "science fiction,"  
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.14.6.2 Enshure Equal Dimension

[SDL.Ensure Equal Dimension / SDL.Eed](#) [Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL.Ensure Equal Dimension / SDL.Eed

Generates a PNG image based on a given image (P1), resized to match the dimensions of a reference image (P2).

Intention

The `SDL.Ensure Equal Dimension` command is designed to create a new PNG image with dimensions that match those of a reference image (**P2**). This command is particularly useful for ensuring that mask images are the same size as the original image. The images themselves are not altered; instead, a temporary PNG image is generated, and its file path is returned. The new image is based on an existing image (**P1**).

Special case: If a "!" prefix is used with **P1**, and **P2** is invalid, the original path of **P1** is returned otherwise in case of P2 being invalid nothing is returned.

Syntax

```
SDL.Ensure Equal Dimension | P1 |  
P2 [ | P3 ]
```

Parameter Explanation

P1 - OriginalImagePath: Specifies the file path of the image to be resized.
If **P2** is invalid, nothing is returned, unless you specify a "!" as prefix to **P1**.

P2 - ReferenceImagePath: Specifies the file path of the reference image. The new PNG image will match these dimensions.

P3 - (Optional): Result-TempFilePath: A variable to store the file path of the newly generated PNG image. If omitted, the result is placed on TOS.

Example

```
'*****  
' SDL.-Sample for ensuring equal dimension  
'*****  
$$IMG1 = ?path/to/image1.png  
$$IMG2 = ?path/to/reference_image.png  
SDL.Ensure Equal Dimension|$$IMG1|$$IMG2|$$TMP  
' A new PNG image is created with dimensions matching $$IMG2,  
' and its file path is stored in $$TMP
```

Remarks

-

Limitations:

-

See also:

-

3.42.14.6.3 Enshure Format

SDL.Enshure Format

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL.Enshure Format

Ensures the graphic file is in the specified format and optionally resizes it.

Intention

The `SDL.Enshure Format` command is designed to take an original graphic file in any of the BMP, JPG, or PNG formats and convert it to a specified format.

Optionally, it can also resize the graphic to the given dimensions.

This command is particularly useful for preparing images for other commands that require a specific format or size, such as **IMG2IMG** commands.

The original file will not be modified; a new file will be created if necessary.

Syntax

```
SDL.Enshure Format[P1] ... ELS. ...  
EIF.
```

Parameter Explanation

P1: (Required) Filepath and name of the original input graphic file. The file should be in any of the BMP, JPG, or PNG formats.

P2: (Required) The desired output format. This can be "PNG," "BMP," or "JPG."

P3: (Optional) The desired width (XSize) for the resized image. If both P3 and P4 are "0," the image will not be resized.

P4: (Optional) The desired height (YSize) for the resized image. If both P3 and P4 are "0," the image will not be resized.

P5: (Optional) A variable that will be used to store the filepath of the converted and optionally resized graphic file. If omitted, the result is placed on TOS.

Example

```
! *****
```

```
' SDL.-Sample for ensuring graphic format and size
'*****
SDL.Ensure Format|C:\Images\Original.jpg|PNG|512|512|$$RE
' The original JPG image will be converted to a PNG format and resized to 512x512
' $$RES will contain the path and filename of the file "to use for IMG2IMG" or v
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.14.6.4 Enshure Mask Dimension

[SDL.Ensure Mask Dimension / SDL.Emd](#) [Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL.Ensure Mask Dimension / SDL.Emd

Creates a new PNG mask image with the same dimensions as an original image, fills it with a color, and positions an existing mask within it.

Intention

The `SDL.Ensure Mask Dimension` command generates a new PNG mask image at a temporary file path, that has the same dimensions as an original image. It does not resize the existing mask to fit the new one. Instead, it fills the new mask with a specified color and places the existing mask at a designated position, either cropping it or filling in extra space as needed.

Syntax

```
SDL.Ensure Mask Dimension|P1|  
P2 [|P3] [|P4] [|P5]
```

Parameter Explanation

P1 - *OriginalImagePath* - Specifies the original image's file path, setting the dimensions for the new PNG mask.

P2 - *MaskImagePath* - Specifies the existing mask image's file path. This mask is positioned within the new one without resizing.

P3 - (Optional): *BorderColor* Specifies the fill color for the new mask. Defaults to "&H000000FF". Accepts RGB or RGBA, and can use "&H" for hex format.

P4 - (Optional): *Position* Sets where the existing mask is positioned within the new mask. Defaults to "LD" (Left-Down). Uses codes like "L" for Left, "C" for Center, etc.

P5 - (Optional): *Result-TempFilePath* A variable to store the new PNG mask's file path. If omitted, the result is placed on the Top Of Stack (TOS).

P4 - Position Codes:

L: Left
C: Center

R: Right
U: Up
D: Down

Combine codes for X and Y axes positioning, e.g., "LC" for Left-Center, "RU" for Right-Up.

Example

```
'*****  
' SDL.-Sample for ensuring mask dimension  
'*****  
$$IMA=?path/to/original/image.png  
$$IMB=?path/Mask.png  
SDL.Ensure Mask Dimension|$$IMA|$$IMB|&HFFFFFF|LC|$$TMP  
' New PNG mask created at left-center, file path stored in  
$$TMP  
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.14.6.5 Enshure PNG

SDL.Enshure PNG

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL.Enshure PNG

Ensures the graphic file is converted to PNG format and optionally resizes it.

Intention

The `SDL.Ensure PNG` command is designed to take an original graphic file in any of the BMP, JPG, or PNG formats and convert it to PNG format.

Optionally, it can also resize the graphic to the given dimensions.

This command is particularly useful for preparing images for other commands that specifically require the PNG format or a particular size.

The original file will not be modified; a new PNG file will be created.

Syntax

```
SDL.Ensure PNG | P1 [ | P2 ] [ | P3 ] [ | P4 ]
```

Parameter Explanation

P1: (Required) Filepath and name of the original input graphic file. The file should be in any of the BMP, JPG, or PNG formats.

P2: (Optional) The desired width (XSize) for the resized image. If **P2** or **P3** are "0," the image will not be resized.

P3: (Optional) The desired height (YSize) for the resized image. If **P2** or **P3** are "0," the image will not be resized.

P4: (Optional) A variable that will be used to store the filepath of the converted and optionally resized PNG file.

If **P4** is omitted, the result is placed on TOS.

Example

```
' *****
' SDL.-Sample for ensuring PNG format and size
' *****
```

```
SDL.Ensure PNG|C:\Images\Original.jpg|800|600|$$RES
' The original JPG image will be converted to PNG format and
  resized to 800x600.
' $$RES will contain a path to the temporary PNG-file for use
  with IMG2IMG or whatever
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.14.6.6 Generate with Steps

[SDO./SDL.Generate with Steps](#)

[Previous](#) [Top](#) [Next](#)

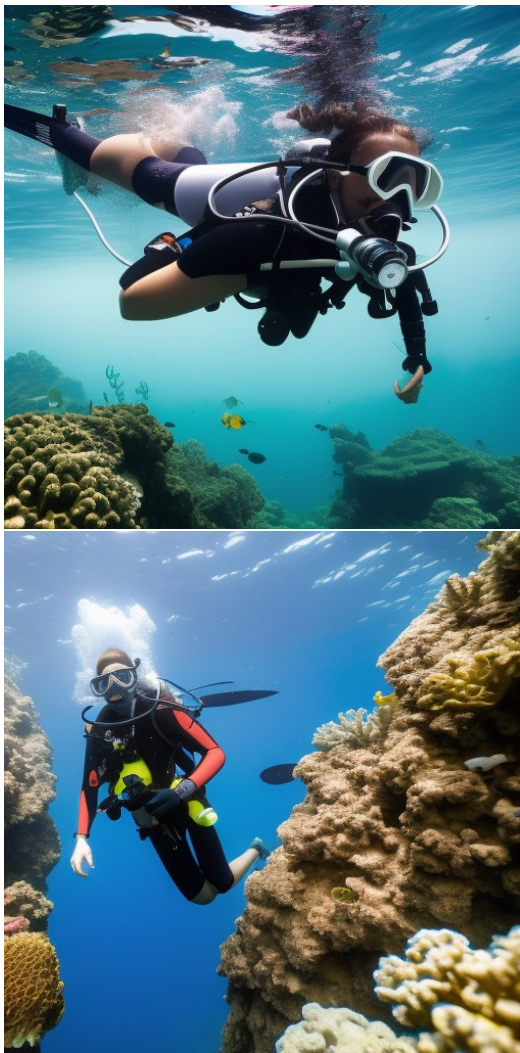
MiniRobotLanguage (MRL)

SDO./SDL.Generate with Steps

SDL.Generate with Steps / SDO.Generate with Steps

The two commands have the same Parameters therefore you can easily switch between rendering Online and local.





Intention

The "SDL.Generate with Steps" and "SDO.Generate with Steps" commands are part of the Stable Diffusion Local (**SDL**) and Stable Diffusion Online (**SDO**) suite of commands, respectively.

They are used to generate an image based on a given prompt, applying a specified number of steps.

The command can be used to find the best number of steps for a Theme and Sampler.

The generated image is created based on the prompts and parameters provided by the user.

The key difference between the two commands is the rendering location—**SDL** renders locally, while **SDO** renders in the Cloud via Stability AI (needs keyfile).

Unique Feature: Switch Between Local and Online Rendering

A unique and user-friendly design feature of these commands is the ease with which users can switch between local and cloud-based rendering. To change from rendering locally to rendering online, simply replace the "L" in "SDL" with an "O," resulting in "SDO." To switch from online to local rendering, do the opposite—replace the "O" with an "L." This feature provides a seamless way for users to choose where their images are rendered based on their needs and resources, without having to rewrite or restructure their commands.

Examples

```

SDO.SetKey|File
SDO.SetSize|512|512|stable-diffusion-v1-5
SDO.Set_Samples|1
SDO.Set_Scale|30
$$PRO=gigantic underwater city, fishes divers, Arielle
$$NEG=ugly
SDO.Set Prompt|$$PRO
SDL.Set Extra Parameter|"restore faces": true
SDO.Set Style|1
FOR.$$STY|100|150|10

' Parameter 3 is the Style Value
  $$FIL=?exeloc\Lori_?.png
  SDO.gst||$$FIL|$$NEG|$$STY
  'SDO.Show Error
  SDO.Get Several|5|$$STA
  DBP. We use Style: $$STA
NEX.
ENR.

```

Syntax

```

SDL.Generate with Steps [|P1] [|P2] [|P3] [|P4]
SDO.Generate with Steps [|P1] [|P2] [|P3] [|P4]
SDL.gst [|P1] [|P2] [|P3] [|P4]
SDO.gst [|P1] [|P2] [|P3] [|P4]

```

Parameter Explanation

P1: (Optional) The prompt made from words or phrases followed by a weight in parentheses.

The phrases are separated by commas.

In detail there is a difference, for SDO. you can add values in () after each term, like this:

```

$$PRO=robot (0.2),in golden color (0.4), on a rainbow (0.4)
$$NEG=malformed hands (0.5),dark colors (0.5)

```

These numbers added should sum up together 1, if not the engine will normalize these.

The numbers for the negative prompt will automatically be added as negative numbers.

For **SDL**. you can use a longer prompt in one Line, that contain "," to separate terms.

Make sure the most important topics are on the left side.

P2: (Optional) The path and file name of the output PNG file. This is optional. If not provided, a unique local path will be generated.

P3: (Optional) This is an optional negative prompt, formatted in the same way as P1.

P4: (Optional) The number of Steps between 10 and 150. A higher number can possibly improve quality but takes longer.

Example

```
'*****
'
'*****
SDO.SetKey|File
SDO.SetSize|512|512|stable-diffusion-v1-5
SDO.Set_Samples|1
SDO.Set Scale|30
'SDO.Set Steps 100
$$PRO=gigantic geodesic dome, underwater city, fishes divers, Arielle, (0.7)
$$NEG=ugly (0.3)
SDO.Set Prompt|$$PRO
SDO.Set Style|1
SDO.Get Several|5|$$STA
DBP. We use Style: $$STA
FOR.$$STY|1|7
' Parameter 3 is the Steps Value
  $$FIL=?exeloc\Lori_?.png
  SDO.gws||$$FIL|$$NEG|$$STY
  'SDO.Show Error
NEX.
ENR.
```

Remarks

-

Limitations:

```
'*****
' SDO.-Sample 1 + Censored Prompt
'*****
```

SDO . as a cloud based command is censored.

Means Terms like "Sexy" are not allowed as you can test with the Script below.

This does NOT apply to SDL .

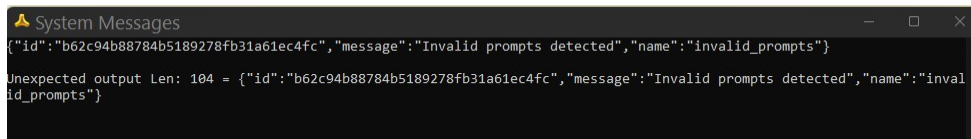
Using SDL . you can use any terms on your own responsibility.

```
SDO.SetKey|File
SDO.SetSize|512|512|stable-diffusion-v1-5
SDO.Set_Samples|3
```

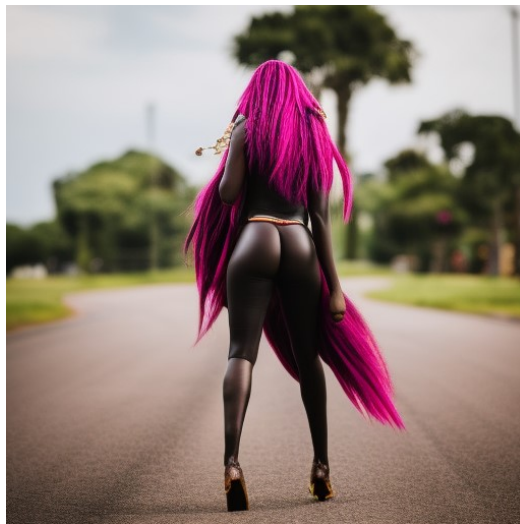
```

SDL.Set Extra Parameter|"restore faces": true
$$PRO=sexy african women(0.3),rainbow colored hair(.5),riding a dragon(0.2)
$$NEG=b/w,old(.5),fat(.3)
SDO.gtf|$$PRO||$$NEG|100
POP.$$PAT
POP.$$SUC
IVV.$$SUC=1
  ANA.load|0|$$PAT
  ANA.show|0!
ELS.
  AIC.Show Error
  MBX.No File generated.
EIF.
MBX.!
ENR.

```



If you get this error message your terms have been censored.



While SDO-Terms are censored the pictures are not, therefore you will still get files like this sometimes.

```

' *****
' SDO.-Sample 2 + Valid Prompt
' *****
SDO.SetKey|File
SDO.SetSize|512|512|stable-diffusion-v1-5
SDO.Set_Samples|3
SDL.Set Extra Parameter|"restore faces": true
$$PRO=african women(0.3),rainbow colored hair(.5),riding a dragon(0.2)
$$PRO="african women(0.3),rainbow colored hair(.5),riding a dragon(0.2)"
$$NEG=b/w,old(.5),fat(.3)
SDO.gtf|$$PRO||$$NEG|100
POP.$$PAT
POP.$$SUC
IVV.$$SUC=1
  ANA.load|0|$$PAT

```

```
ANA.show|0!  
ELS.  
  AIC.Show Error  
  MBX.No File generated.  
EIF.  
MBX.!  
ENR.
```

See also:

-

3.42.14.6.7 Generate with Style

[SDO./SDL.Generate with Style](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDO./SDL.Generate with Style

SDL.Generate with Style / SDO.Generate with Style

The two commands have the same Parameters therefore you can easily switch between rendering Online and local.



Intention

The "SDL.Generate with Style" and "SDO.Generate with Style" commands are part of the Stable Diffusion Local (**SDL**) and Stable Diffusion Online (**SDO**) suite of commands, respectively.

They are used to generate an image based on a given prompt, applying a specified style.

The generated image is created based on the prompts and parameters provided by the user.

The key difference between the two commands is the rendering location—**SDL** renders locally, while **SDO** renders in the Cloud via Stability AI (needs keyfile).

Unique Feature: Switch Between Local and Online Rendering

A unique and user-friendly design feature of these commands is the ease with which users can switch between local and cloud-based rendering. To change from rendering locally to rendering online, simply replace the "L" in "SDL." with an "O," resulting in "SDO." To switch from online to local rendering, do the opposite—replace the "O" with an "L." This feature provides a seamless way for users to choose where their images are rendered based on their needs and resources, without having to rewrite or restructure their commands.

Examples

```
' This command generates an image based on the prompt
' "big spider (0.5), small butterfly (0.5)"
' using the specified style, and saves it as a PNG file at a unique local path.

SDL.Set Prompt|big spider (0.5), small butterfly (0.5)
SDL.Set_Style|5
SDL.Generate with Style

' This command generates an image based on the prompt and saves it as a PNG file
' at the specified location "C:\Images\output.png".
SDO.Set Prompt|big spider (0.5), small butterfly (0.5)
$$FIL=C:\Images\output.png
SDO.Set_Style|3
SDO.Generate with Style||$$FIL

' This command generates an image based on the prompt and negative prompt,
' using the specified style, and saves it as a PNG file at the specified location.
SDO.Set Prompt|big spider (0.5), small butterfly (0.5)
$$FIL=C:\Images\output.png
SDO.NegPrompt|big elephant(1)
SDO.Set_Style|4
SDO.Generate with Style||$$FIL
```

Syntax

```
SDL.Generate with Style[|P1][|P2][|P3][|P4]
SDO.Generate with Style[|P1][|P2][|P3][|P4]
SDL.gws[|P1][|P2][|P3][|P4]
SDO.gws[|P1][|P2][|P3][|P4]
```

Parameter Explanation

P1: (Optional) The prompt made from words or phrases followed by a weight in parentheses.

The phrases are separated by commas.

In detail there is a difference, for SDO. you can add values in () after each term, like this:

```
$$PRO=robot (0.2),in golden color (0.4), on a rainbow (0.4)
```

```
$$NEG=malformed hands (0.5),dark colors (0.5)
```

These numbers added should sum up together 1, if not the engine will normalize these.

The numbers for the negative prompt will automatically be added as negative numbers.

For **SDL**. you can use a longer prompt in one Line, that contain "," to separate terms.

Make sure the most important topics are on the left side.

P2: (Optional) The path and file name of the output PNG file. This is optional. If not provided, a unique local path will be generated.

P3: (Optional) This is an optional negative prompt, formatted in the same way as P1.

P4: (Optional) The style to apply when generating the image. This can be a style index corresponding to predefined styles, as specified in the `SDL.Set_Style` or `SDO.Set_Style` command.

Here is also a difference. For SDO, you can only give one of the predefined styles.

P4 for SDO: (optional) <Style Index>: Specifies the desired style by its associated index or in letters. The valid options are:

- 0: 3d-model
- 1: analog-film
- 2: anime
- 3: cinematic
- 4: comic-book
- 5: digital-art
- 6: enhance
- 7: fantasy-art
- 8: isometric
- 9: line-art
- 10: low-poly
- 11: modeling-compound
- 12: neon-punk
- 13: origami
- 14: photographic
- 15: pixel-art
- 16: tile-texture





SDO. and Style 7 - Fantasy Art

SDO. and Style 4 - comic book

P4 for SDL: This Parameter specifies the "Medium" not exactly the style. The effect is somehow equal.

You can specify a number between 0 and 94 or the "Style Name".

This will not change the value that can be used with `SDL.Set Style`.

This "Medium Style" will be internally prefixed to the Prompt.

You can use and also mix the styles that are installed local in your WEB-GUI. See `SDL.Set Style Free` - command.

Number	Style Name	Description
0	A 3D Render	A three-dimensional computer-generated image.
1	A Black and White Photo	A photograph without colors other than black, white, and grey.
2	A Bronze Sculpture	A sculpture made from bronze material.
3	A Cartoon	A simplified, exaggerated visual art, often used for humorous effect.

4	A Cave Painting	Prehistoric drawings found on the wall of caves.
5	A Character Portrait	A depiction focusing on a character, often in fiction.
6	A Charcoal Drawing	Art made by smudging charcoal on paper.
7	A Child's Drawing	Artistic work typically done by a child.
8	A Color Pencil Sketch	Sketching done using colored pencils.
9	A Colorized Photo	A black and white photo that has been digitally colored.
10	A Comic Book Panel	A single frame or panel in a comic book.
11	A Computer Rendering	A digital 3D image or animation.
12	A Cross Stitch	An embroidery style using X-shaped stitches.
13	A Cubist Painting	Artwork in the style of Cubism, often fragmented and abstracted.
14	A Detailed Drawing	A drawing with a high level of detail.
15	A Detailed Matte Painting	A highly detailed painting used as a background in movies and video games.
16	A Detailed Painting	A painting with a high level of detail.
17	A Diagram	A simplified drawing showing the appearance, structure, or workings of something.

18	A Digital Painting	An artwork created digitally on a computer.
19	A Digital Rendering	A digital 3D image or animation.
20	A Drawing	Artwork created using pencils, charcoal, or other drawing tools.
21	A Fine Art Painting	A painting created according to fine art standards, often for galleries.
22	A Flemish Baroque	A style from the Baroque period, originating in Flanders.
23	A Gouache	A painting method using opaque watercolors.
24	A Hologram	A three-dimensional image formed by the interference of light beams.
25	A Hyperrealistic Painting	A painting that resembles a high-resolution photograph.
26	A Jigsaw Puzzle	Artwork that is cut into pieces and must be reassembled.
27	A Low Poly Render	A 3D rendering with a limited number of polygons.
28	A Macro Photograph	A photograph taken extremely close-up.
29	A Manga Drawing	Artwork in the style of Japanese manga.
30	A Marble Sculpture	A sculpture made from marble.
31	A Matte Painting	A painting used as a background in movies and video games.

32	A Microscopic Photo	A photograph taken under a microscope.
33	A Mid-Nineteenth Century Engraving	An engraving typical of the mid-19th century style.
34	A Minimalist Painting	A painting with minimal elements, focusing on shape and color.
35	A Mosaic	An image created by assembling small pieces of colored material.
36	A Painting	An artwork created using paint on a surface like canvas.
37	A Pastel	Artwork created using pastel colors.
38	A Pencil Sketch	A drawing made using a pencil.
39	A Photo	A photograph.

40	A Photocopy	A duplicate made by a photocopier.
41	A Photorealistic Painting	A painting that closely resembles a high-quality photograph.
42	A Picture	A general term for any visual artwork, including drawings, paintings, and photographs.
43	A Pointillism Painting	A painting technique using small, distinct dots of color.
44	A Polaroid Photo	A photograph taken with a Polaroid camera, often instant.

45	A Pop Art Painting	Art inspired by popular culture, often using bold colors and simple shapes.
46	A Portrait	A painting, drawing, or photograph of a person.
47	A Poster	A printed image or text used for display or advertisement.
48	A Raytraced Image	A digital image created with ray tracing, a technique for rendering 3D scenes.
49	A Renaissance Painting	Artwork from the European Renaissance, emphasizing perspective and human form.
50	A Screenprint	An image made by forcing ink through a mesh screen.
51	A Screenshot	An image of what is visible on a computer screen.
52	A Silk Screen	A printing technique where ink is pressed through a stenciled mesh screen.
53	A Sketch	A quickly executed freehand drawing.
54	A Statue	A three-dimensional sculpture representing a figure.
55	A Still Life	A painting or drawing of inanimate objects.
56	A Stipple	An image created by applying small dots or specks.

57	A Stock Photo	A readily available photograph usually used for commercial purposes.
58	A Storybook Illustration	An illustration typically found in children's books.
59	A Surrealist Painting	Art that seeks to express the subconscious mind, often through irrational juxtapositions.
60	A Surrealist Sculpture	A sculpture in the style of surrealism, often featuring unexpected or dream-like elements.

61	A Tattoo	A form of body art where ink is inserted into the dermis layer of the skin.
62	A Tilt Shift Photo	A photograph where the camera is manipulated to create a miniature scene.
63	A Watercolor Painting	A painting technique using pigments suspended in a water-based solution.
64	A Wireframe Diagram	A visual representation of a three-dimensional object used in 3D computer graphics.
65	A Woodcut	A relief printing technique in printmaking.
66	An Abstract Drawing	A drawing that does not attempt to represent

		external reality, but seeks to achieve its effect using shapes, forms, colors, and textures.
67	An Abstract Painting	A painting style that doesn't depict objects in the natural world, but uses color and form in a non-representational way.
68	An Abstract Sculpture	A sculpture that does not represent any recognizable object or figure.
69	An Acrylic Painting	A fast-drying paint made of pigment suspended in acrylic polymer emulsion.
70	An Airbrush Painting	A painting technique using an airbrush to spray paint or ink onto a surface.
71	An Album Cover	The front of the packaging of a commercially released audio recording.
72	An Ambient Occlusion Render	A shading and rendering technique used to calculate the exposure of each point in a 3D environment.
73	An Anime Drawing	Artwork in the style of Japanese animated productions.
74	An Art Deco Painting	A painting style characterized by geometric shapes and strong colors, popular in the 1920s and 1930s.

75	An Art Deco Sculpture	A sculpture in the style of the Art Deco movement, often featuring sleek and streamlined forms.
76	An Engraving	A printmaking technique where the design is incised into a surface.
77	An Etching	A printmaking technique using acid or mordant to cut into a metal surface.
78	An Illustration of	A visual representation to accompany text or to explain or clarify information.
79	An Impressionist Painting	A painting style that seeks to capture a feeling or experience, often by using loose brushwork and a focus on light.
80	An Ink Drawing	A drawing technique using ink instead of pencil or charcoal.
81	An Oil on Canvas Painting	A painting technique using oil paints on canvas material.
82	An Oil Painting	A painting technique using oil-based pigments.
83	An Ultrafine Detailed Painting	A painting with an extreme level of detail.
84	Chalk Art	Artwork created using chalk.
85	Computer Graphics	Digital artwork generated through computational processes.

86	Concept Art	Artwork designed to visualize a concept or idea before it is created.
87	Cyberpunk Art	Art that incorporates elements of the cyberpunk genre, often featuring futuristic settings.
88	Digital Art	Artwork created or manipulated on a computer.
89	Egyptian Art	Art originating from ancient Egypt, often featuring gods, pharaohs, and hieroglyphics.
90	Graffiti Art	Art created on walls or other surfaces, often in public spaces, using spray paint or other mediums.
91	Lineart	Artwork consisting of distinct straight or curved lines placed against a background.
92	Pixel Art	Digital art created through the placement of pixels.
93	Poster Art	Art specifically designed to be made into posters.
94	Vector Art	Art created using vector graphics, which are based on mathematical equations rather than pixels.

Example

```
!*****
!
!*****
SDO.SetKey|File
SDO.SetSize|512|512|stable-diffusion-v1-5
SDO.Set_Samples|1
SDO.Set_Scale|30
SDO.Set Steps 100
$$PRO=gigantic geodesic dome, underwater city, fishes divers, Arielle, (0.7)
$$NEG=ugly (0.3)
SDO.Set Prompt|$$PRO
FOR.$$STY|1|7
  'SDO.Set Style|$$STY
' Parameter 3 is the Style Value
  $$FIL=?exeloc\Lori_?.png
  SDO.gws||$$FIL|$$NEG|$$STY
  'SDO.Show Error
  SDO.Get Several|5|$$STA
  DBP. We use Style: $$STA
NEX.
ENR.
```

Remarks

-

Limitations:

-

See also:

-

3.42.14.6.8 Get Artist

[SDL.Get Artist / SDL.Gar](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL.Get Artist / SDL.Gar

Retrieves artist names based on a given index number, which ranges from 0 to 5265.

Intention

The `SDL.Get Artist` and its shorthand `SDL.Gar` are designed to retrieve artist names.

These commands return a specific artist name based on an index number provided by the user.

The result of this command can be appended or prefixed to the prompt for Stable Diffusion (SD) processing to generate images that align with the retrieved artists style.

The index must be between 0 and 5265; otherwise, the command will return "a photo" (index 29).

Alternative to the numeric index you can also write something for example "pointilism" and the system will try to get the best match for you.

Syntax

```
SDL.Get Artist[|P1][|P2]
```

```
SDL.Gar[|P1][|P2]
```

Parameter Explanation

- **P1:** (Optional) An index number between 0 and 5265. If omitted, the default value of 0 is used, and the result will be an empty string ("").

You can also provide a Text like "willard" and the Command will try to find the best match for your Text from all these artists.

- **P2:** (Optional) A variable that will be used to store the artist name retrieved. If omitted, the result is placed on TOS (Top Of Stack).

Example

```

'*****
' SDL.-Sample for retrieving artist names
'*****
$$IDX = Cason
SDL.Get Artist|$$IDX|$$ART
DBP. The artist for index $$IDX is: $$ART

$$IDX = 1024
SDL.Get Artist|$$IDX|$$ART
DBP. The artist for index $$IDX is: $$ART
ENR.

```

Following a List with all included Artists.

A:

A. B. Jackson, A. J. Casson, A. R. Middleton Todd, A.B. Frost, A.D.M. Cooper, Aaron Bohrod, Aaron Douglas, Aaron Jasinski, Aaron Miller, Aaron Nagel, Abbott Handerson Thayer, Abdullah Gërguri, Abdur Rahman Chughtai, Abidin Dino, Abraham Begeyn, Abraham Bloemaert, Abraham Bosschaert, Abraham Hondius, Abraham Mignon, Abraham Storck, Abraham Willaerts, Abraham de Vries, Abraham van Beijeren, Abraham van Calraet, Abraham van den Tempel, Abram Arkhipov, Achille Leonardi, Ada Gladys Killins, Ada Hill Walker, Adam Bruce Thomson, Adam Chmielowski, Adam Dario Keel, Adam Elsheimer, Adam Manyoki, Adam Marczyński, Adam Paquette, Adam Pijnacker, Adam Rex, Adam Saks, Adam Szentpétery, Adam Willaerts, Adolf Bierbrauer, Adolf Born, Adolf Dietrich, Adolf Fényes, Adolf Hirémy-Hirschl, Adolf Hölzel, Adolf Schrödter, Adolf Ulric Wertmüller, Adolf Wölfli, Adolfo Müller-Ury, Adolph Gottlieb, Adolph Menzel, Adolphe Willette, Adriaen Brouwer, Adriaen Coorte, Adriaen Hanneman, Adriaen Isenbrant, Adriaen van Ostade, Adriaen van de Velde, Adriaen van de Venne, Adriaen van der Werff, Adrian Zingg, Adrienn Henczné Deák, Adélaïde Labille-Guiard, Adélaïde Victoire Hall, Aelbert Cuyp, Aert de Gelder, Aert van der Neer, Aertgen van Leyden, Afewerk Tekle, Affandi, Agnes Lawrence Pelton, Agnes Martin, Agnolo Bronzino, Agnolo Gaddi, Agostino Carracci, Aguri Uchida, Agustín Fernández, Ahmed Karahisari, Ahmed Yacoubi, Ai Weiwei, Ai Xuan, Ai-Mitsu, Aileen Eagleton, Aimé Barraud, Akihiko Yoshida, Akira Toriyama, Akseli Gallen-Kallela, Al Feldstein, Al Williamson, Aladár Körösfői-Kriesch, Alain Tasso, Alan Bean, Alan Davis, Alan Lee, Alan Pollack, Alan Sutherland, Alasdair Grant Taylor, Alasdair Gray, Albert Anker, Albert Bertelsen, Albert Bierstadt, Albert Dorne, Albert Dubois-Pillet, Albert Edelfelt, Albert Gleizes, Albert Guillaume, Albert Henry Krehbiel, Albert Irvin, Albert J. Welte, Albert Joseph Moore, Albert Joseph Pénot, Albert Keller, Albert Kotin, Albert Marquet, Albert Namatjira, Albert Nemethy, Albert Paris Gütersloh, Albert Swinden, Albert Welte, Alberto Burri, Alberto Giacometti, Alberto Morrocco, Alberto Seveso, Alberto Sughì, Alberto Vargas, Albin Egger-Lienz, Albrecht Altdorfer, Albrecht Dürer, Aldus Manutius, Alejandro Burdisio, Alejandro Obregón, Aleksander Gierymski, Aleksander Gine, Aleksander Kobzdej, Aleksander Kotsis, Aleksander Orłowski, Aleksandr Gerasimov, Aleksandr Ivanovich Laktionov, Aleksis Briclot, Alena Aenami, Alessandro Allori, Alessandro Galli Bibiena, Alesso Baldovinetti, Alex Grey, Alex Horley, Alex Horley-Orlandelli, Alex Katz, Alex Petruk APe, Alex Ross, Alex Toth, Alexander Archipenko, Alexander Bogen, Alexander Brook, Alexander Calder, Alexander Carse, Alexander Deyneka, Alexander Fedosav, Alexander Ivanov, Alexander Jansson, Alexander Johnston, Alexander Kanoldt, Alexander Kucharsky, Alexander Litovchenko, Alexander Mann, Alexander McQueen, Alexander Milne Calder, Alexander Nasmyth, Alexander Robertson, Alexander Rodchenko, Alexander Roslin, Alexander Runciman, Alexander Scott, Alexander Sharpe Ross, Alexander Stirling Calder, Alexander V. Kuprin, Alexandre Benois, Alexandre Cabanel, Alexandre Falguière, Alexandre-Évariste Fragonard, Alexei Kondratyevich Savrasov, Alexej von Jawlensky, Alexey Merinov, Alexey Venetsianov, Alexis Grimou, Alexis Simon Belle, Alfons Karpiński, Alfons Walde, Alfons von Czibulka, Alfonse Mucha, Alfred Charles Parker, Alfred East, Alfred Edward Chalon, Alfred Eisenstaedt, Alfred Freddy Krupa, Alfred Janes, Alfred Jensen, Alfred Krupa, Alfred Kubin, Alfred Leslie, Alfred Leyman, Alfred Manessier, Alfred Richard Gurrey, Alfred Sisley, Alfred Thompson Bricher, Alfred Wallis, Alfredo Volpi, Algernon Talmage, Alice Bailly, Alice Mason, Alice Neel, Alice Prin, Alice Rahon, Alison Debenham, Alison Geissler, Alison Kinnaird, Alison Watt, Allaert van Everdingen, Allan Brooks, Allan Linder, Allan Ramsay, Allen Butler Talcott, Allen Jones, Allen Tupper True, Alma Thomas, Almada Negreiros, Almeida Júnior, Alonso Vázquez, Aloysius O'Kelly, Alphonse Legros, Alphonse Mucha, Alson S. Clark, Altichiero, Alton Tobey, Altoon Sultan, Alvan Fisher, Alén Diviš, Amadeo de Souza Cardoso, Amalia Lindegren, Amanda Sage, Ambreen Butt, Ambrose McCarthy Patterson, Ambrosius Benson, Ambrosius Bosschaert, Ambrosius Bosschaert II,

Ambrosius Holbein, Amedeo Modigliani, Amelia Peláez, Amelia Robertson Hill, Americo Makk, Amir Zand, Ammi Phillips, Amos Ferguson, Amos Sewell, Amy Weber, Amédée Ozenfant, An Gyeon, An Zhengwen, Anato Finnstark, Ansell Stronach, Anders Zorn, Andor Basch, Andre Derain, Andre de Krayewski, Andrea Kowch, Andrea Mantegna, Andrea Orcagna, Andrea Pozzo, Andrea del Sarto, Andrea del Verrocchio, Andreas Achenbach, Andreas Gursky, Andreas Rocha, Andrei Kolkoutine, Andrei Rublev, Andrei Ryabushkin, Andrew Allan, Andrew Bell, Andrew Boog Faithfull, Andrew Domachowski, Andrew Ferez, Andrew Geddes, Andrew Henderson, Andrew Law, Andrew Loomis, Andrew Robertson, Andrew Robinson, Andrew Stevovich, Andrew Wyeth, Andrey Esionov, Andrey Yefimovich Martynov, Andries Both, Andries Stock, Android Jones, Andrzej Pronaszko, Andrzej Wróblewski, André Bauchant, André Beauneveu, André Castro, André Charles Biéler, André Derain, André François, André Kertész, André Lhote, André Masson, André Pijet, André Thomkins, Andrée Ruellan, Andy Goldsworthy, Andy Warhol, Aneurin Jones, Angelica Kauffman, Aniello Falcone, Anish Kapoor, Anita Kunz, Anita Malfatti, Anka Zhuravleva, Ann Thetis Blacker, Anna Ancher, Anna Boch, Anna Dittmann, Anna Findlay, Anna Füssli, Anna Haifisch, Anna Hotchkis, Anna Katharina Block, Anna Maria Barbara Abesch, Anna Mary Robertson Moses, Anna and Elena Balbusso, Annabel Eyres, Annabel Kidston, Anne Dunn, Anne Geddes, Anne Nasmyth, Anne Redpath, Anne Rigney, Anne Ryan, Anne Said, Anne Savage, Anne Stokes, Anni Albers, Annibale Carracci, Annie Abernethie Pirie Quibell, Annie Leibovitz, Annie Rose Laing, Ansel Adams, Anselm Kiefer, Anson Maddocks, Antanas Sutkus, Anthony Angarola, Anthony Devas, Anthony Palumbo, Anthony S Waters, Anthony van Dyck, Antoine Blanchard, Antoine Ignace Melling, Antoine Le Nain, Antoine Verney-Carron, Antoine Wiertz, Anton Ažbe, Anton Fadeev, Anton Graff, Anton Lehmden, Anton Mauve, Anton Möller, Anton Otto Fischer, Anton Pieck, Anton Räderscheidt, Anton Solomoukha, Antonello da Messina, Antoni Brodowski, Antoni Pitxot, Antoni Tàpies, Antonie Palamedesz, Antonin Artaud, Antonio Canova, Antonio Cavallucci, Antonio Ciseri, Antonio Donghi, Antonio Galli Bibiena, Antonio Mancini, Antonio Rotta, Antonio Saura, Antonio de la Gandara, Antonín Chittussi, Antonín Slavíček, Antônio Parreiras, Anibal Villacis, Apelles, Apollinary Vasnetsov, Apollonia Saintclair, Aquirax Uno, Arabella Rankin, Araceli Gilbert, Aramenta Dianthe Vail, Archibald Motley, Archibald Robertson, Archibald Skirving, Archibald Standish Hartrick, Arcimboldo, Arent Arentsz, Arie Smit, Arik Brauer, Aristide Maillol, Arkhip Kuindzhi, Arlington Nelson Lindenmuth, Armand Guillaumin, Armin Baumgarten, Armin Hansen, Arnie Swekel, Arnold Blanch, Arnold Bronckhorst, Arnold Brügger, Arnold Böcklin, Arnold Franz Brasz, Arnold Mesches, Arnold Newman, Arshile Gorky, Art & Language, Art Brenner, Art Fitzpatrick, Art Frahm, Art Green, Art Spiegelman, Art of Brom, Artemisia Gentileschi, Artgerm, Arthur Adams, Arthur B. Carles, Arthur Boyd, Arthur Burdett Frost, Arthur Dove, Arthur Garfield Dove, Arthur Hughes, Arthur Lismer, Arthur Melville, Arthur Merric Boyd, Arthur Pan, Arthur Quartley, Arthur Rackham, Arthur Sarkissian, Arthur Streeton, Arthur Webster Emerson, Artur Grottger, Artur Tarnowski, Arturo Rivera, Arvid Nyholm, Asaf Hanuka, Asai Chū, Asger Jorn, Asher Brown Durand, Atey Ghailan, Attila Meszlenyi, Aubrey Beardsley, Audrey Kawasaki, August Friedrich Schenck, August Lemmer, August Macke, August Querfurt, August Sander, Auguste Baud-Bovy, Auguste Herbin, Augustin Meinrad Bächtiger, Augustus Dumbier, Augustus Earle, Augustus John, Augustus Vincent Tack, Augustyn Mirys, Aurél Bernáth, Auseklis Ozols, Austin Briggs, Austin English, Austin Osman Spare, Avgust Černigoj, Avigdor Arikha, Awataguchi Takamitsu, Ay-O, Aya Goda, Ayako Rokkaku, Ayami Kojima, Ayshia Taşkin, adrian ghenie, albert aublet, amano, amy sol, andrei riabovitchev, andrey ryabovichev, artist, ashley wood

B:

Baiōken Eishun, Balcomb Greene, Balthasar van der Ast, Balthus, Balázs Diószegi, Banksy, Bapu, Barbara Balmer, Barbara Greg, Barbara Longhi, Barbara Nasmyth, Barbara Nessim, Barclay Shaw, Barent Fabritius, Barkley Hendricks, Barnett Newman, Barron Storey, Barthel Bruyn the Elder, Barthel Bruyn the Younger, Bartholomeus Breenbergh, Bartholomeus Strobel, Bartholomeus van Bassen, Bartholomeus van der Helst, Barthélemy Menn, Barthélemy d'Eyck, Bartolomeo Cesi, Bartolomeo Vivarini, Bartolomé Esteban Murillo, Bascove, Basil Blackshaw, Bastien L. Deharme, Bastien Lecouffe-Deharme, Basuki Abdullah, Bauhaus, Bayard Wu, Beatrice Ethel Lithiby, Beatrice Huntington, Beatrix Potter, Beauford Delaney, Bedwyr Williams, Beeple, Beksinski, Bela Čikoš Sesija, Ben Enwonwu, Ben Nicholson, Ben Shahn, Ben Stahl, Ben TempleSmith, Ben Thompson, Ben Zoeller, Bencho Obreshkov, Benito Quinquela Martín, Benjamin Block, Benjamin Franklin, Benjamin Gerritsz Cuyp, Benjamin Marra, Benjamin West, Benjamin Williams Leader, Benoît B. Mandelbrot, Bernard Accama, Bernard Buffet, Bernard D'Andrea, Bernard Fleetwood-Walker, Bernard Meninsky, Bernard van Orley, Bernardino Mei, Bernardo Bellotto, Bernardo Cavallino, Bernardo Daddi, Bernardo Strozzi, Bernat Sanjuan, Bernd Fasching, Bernie D'Andrea, Bernie Wrightson, Bernt Tunold, Bert Hardy, Bert Stern, Bertalan Karlovsky, Bertalan Pór, Bertalan Székely, Berthe Morisot, Bertram Brooker, Bessie Wheeler, Beta Vukanović, Bettina Heinen-Ayech, Betty Churcher, Betty Merken, Betye Saar, Bholekar Srihari, Bhupen Khakhar, Bian Jingzhao, Bian Shoumin, Bikash Bhattacharjee, Bill Lewis, Bill Sienkiewicz, Bill T raylor, Bill Ward, Bill Watterson, Billie Waters, Billy Childish, Bjørn Wiinblad, Blanche Hoschedé Monet, Bob Byerley, Bob Eggleton, Bob Ringwood, Bob Ross, Bob Singer, Bob Thompson, Boetius Adamsz Bolswert, Bogi Fabian, Bohumil Kubista, Boleslaw Cybis, Bonnard Pierre, Boris Kustodiev, Boris Vallejo, Boris Vladimírski, Bouchta El Hayani, Bourgeois, Božidar Jakac, Bracha L. Ettinger, Brad Holland, Brad Kunkle, Bradley Walker Tomlin, Brassai, Brenda Chamberlain, Brett Whiteley, Breyten Breytenbach, Brian 'Chippy' Dugan, Brian Alfred, Brian Bolland, Brian Despain, Brian Dunlop,

Brian Fies, Brian Froud, Brian Snøddy, Brian Thomas, Brian and Wendy Froud, Briana Mora, Brice Marden, Bridget Bate Tichenor, Bridget Riley, Brigitte Barrager, Brigid Derham, Brom, Brooke Shaden, Brothers Hildebrandt, Bruce Davidson, Bruce Gilden, Bruce McLean, Bruce Munro, Bruce Nauman, Bruce Onobrakpeya, Bruce Pennington, Bruce Timm, Bruno Liljefors, Bryan Organ, Buckminster Fuller, Bunny Yeager, Byeon Sang-byeok, Byron Galvez, Bálint Kiss, Béla Apáti Abkarovics, Béla Czóbel, Béla Iványi-Grünwald, Béla Kondor, Béla Nagy Abodi, Béla Pállik, Béni Ferenczy

C:

Caesar Andrade Faini, Caesar van Everdingen, Cafer Bater, Cagnaccio di San Pietro, Cam Sykes, Camille Bombois, Camille Bouvagne, Camille Corot, Camille Pissarro, Camille Souter, Camille-Pierre Pambu Bodo, Camilo Egas, Camilo Mori, Canaletto, Candido Bido, Candido Portinari, Cao Buxing, Cao Zhibai, Caravaggio, Carel Fabritius, Carel Weight, Carel Willink, Carey Morris, Carl Arnold Gonzenbach, Carl Barks, Carl Critchlow, Carl Eugen Keel, Carl Eytel, Carl Frederik von Breda, Carl Gustaf Pilo, Carl Heinrich Bloch, Carl Hoppe, Carl Larsson, Carl Morris, Carl Rahl, Carl Spitzweg, Carl Walter Liner, Carl-Henning Pedersen, Carla Wyzgala, Carle Hessay, Carles Delclaux Is, Carlo Carlone, Carlo Carrà, Carlo Crivelli, Carlo Galli Bibiena, Carlo Maderna, Carlo Martini, Carlo Mense, Carlo Randanini, Carlos Berlanga, Carlos Catasse, Carlos Enriquez Gómez, Carlos Francisco Chang Marín, Carlos Saenz de Tejada, Carlos Schwabe, Carlos Trillo Name, Carne Griffiths, Caro Niederer, Carol Bove, Carol Sutton, Caroline Chariot-Dayez, Caroline Gotch, Caroline Lucy Scott, Caroline Mytinger, Carpofo Tencalla, Carrie Mae Weems, Caspar David Friedrich, Caspar Netscher, Caspar Wolf, Caspar van Wittel, Cassandra Austen, Cassius Marcellus Coolidge, Caterina Tarabotti, Catrin G Grosse, Catrin Welz-Stein, Cecil Beaton, Cecile Walton, Cecilia Beaux, Cecily Brown, Cedric Peyravernay, Cedric Seaut, Cedric Seaut (Keos Masons), Ceferí Olivé, Celia Fiennes, Celia Frances Bedford, Ceri Richards, Cerith Wyn Evans, Chafik Charobim, Chagall, Chaim Soutine, Chang Dai-chien, Chantal Joffe, Charles Addams, Charles Alphonse du Fresnoy, Charles Alston, Charles Angrand, Charles Billich, Charles Bird King, Charles Blackman, Charles Codman, Charles Conder, Charles Crodell, Charles Cundall, Charles Dana Gibson, Charles Demuth, Charles E. Burchfield, Charles Ellison, Charles Fremont Conner, Charles Furneaux, Charles Ginner, Charles Gleyre, Charles H. Woodbury, Charles Harold Davis, Charles Haslewood Shannon, Charles Hinman, Charles Hopkinson, Charles Joshua Chaplin, Charles Le Brun, Charles Le Roux, Charles Mahoney, Charles Marion Russell, Charles Martin, Charles Maurice Detmold, Charles McAuley, Charles Mozley, Charles Ragland Bunnell, Charles Rennie Mackintosh, Charles Ricketts, Charles Roka, Charles Rollier, Charles S. Kaelin, Charles Schulz, Charles Thomson, Charles Uzzell-Edwards, Charles W. Bartlett, Charles Williams, Charles Willson Peale, Charles-Amédée-Philippe van Loo, Charles-André van Loo, Charles-François Daubigny, Charlie Bowater, Charlotte Harding, Charlotte Nasmyth, Charly Amani, Charmion von Wiegand, Chase Stone, Chaim Soutine, Chen Chi, Chen Chun, Chen Daofu, Chen Hong, Chen Hongshou, Chen Jiru, Chen Lin, Chen Lu, Chen Rong, Chen Yifei, Cheng Jiasui, Cheng Shifa, Cheng Zhengkui, Cheryl Fountain, Chesley Bonestell, Chica Macnab, Chiharu Shiota, Chiho Aoshima, Childe Hassam, Chinwe Chukwuogo-Roy, Chip Zdarsky, Chippy, Chizuko Yoshida, Choi Buk, Chris Cold, Chris Foss, Chris Friel, Chris LaBrooy, Chris Moore, Chris Rahn, Chris Rallis, Chris Spollen, Chris Ware, Christabel Dennison, Christen Dalsgaard, Christen Købke, Christian August Lorentzen, Christian Hilfgott Brand, Christian Jane Fergusson, Christian Krohg, Christian Rohlf, Christian W. Staudinger, Christo, Christoffel van den Berghe, Christoffer Wilhelm Eckersberg, Christoph Amberger, Christoph Ludwig Agricola, Christophe Vacher, Christopher Balaskas, Christopher Moeller, Christopher Perkins, Christopher Rush, Christopher Williams, Christopher Wood, Christopher Wren, Chuck Close, Cicely Hey, Cicely Mary Barker, Cimabue, Cindy Sherman, Cindy Wright, Claes Corneliszoon Moeyaert, Claes Jansz. Visscher, Claire Dalby, Claire Falkenstein, Claire Hummel, Clara Miller Burd, Clara Peeters, Clara Weaver Parrish, Clarence Holbrook Carter, Clarice Beckett, Clark Voorhees, Claude Bonin-Pissarro, Claude Cahun, Claude Lorrain, Claude Monet, Claude Rogers, Cleon Peterson, Cleve Gray, Cliff Childs, Clifford Ellis, Clifford Ross, Clint Cearley, Clovis Trouille, Clyde Caldwell, Clyfford Still, Clément Serveau, Coby Whitmore, Coles Phillips, Colijn de Coter, Colin Campbell Cooper, Colin Gill, Colin Hayes, Colin McCahon, Colin Middleton, Colin Moss, Conrad Marca-Relli, Conrad Roset, Conroy Maddox, Constance Copeman, Constance Gordon-Cumming, Constance-Anne Parker, Constant, Constant Permeke, Constantin Hansen, Constantine Andreou, Coppo di Marcovaldo, Cor Melchers, Corneille, Cornelia MacIntyre Foley, Cornelia Parker, Cornelis Anthonisz, Cornelis Bisschop, Cornelis Claesz van Wieringen, Cornelis Dusart, Cornelis Engebrechtsz, Cornelis Pietersz Bega, Cornelis Saftleven, Cornelis Verbeeck, Cornelis de Heem, Cornelis de Man, Cornelis van Haarlem, Cornelis van Poelenburgh, Cornelisz Hendriksz Vroom, Correggio, Cosmo Alexander, Craig Davison, Craig Mullins, Craig Thompson, Craola, Cricorps Grégoire, Cristache Gheorghiu, Cristofano Allori, Csaba Markus, Cui Bai, Cui Zizhong, Cuno Amiet, Cy Twombly, Cynthia Sheppard, Cyril Rolando, Cándido López, Cézanne, casey baugh, charles vess

D:

D. Alexander Gregory, D. Howard Hitchcock, Daarken, Dahlov Ipcar, Dai Jin, Dai Xi, Dali, Dalí, Damien Hirst, Dan Christensen, Dan Content, Dan Frazier, Dan Hillier, Dan Luvisi, Dan Mumford, Dan Scott, Dan Smith, Daniel Chodowiecki, Daniel F. Gerhartz, Daniel Garber, Daniel Gelon, Daniel Lieske, Daniel Ljunggren, Daniel Maclise, Daniel Merriam, Daniel Schultz, Daniel Seghers, Daniel Taylor, Daniël Mijtens, Dante Gabriel Rossetti, Daphne Allen, Daphne Fedarb, Daphne McClure, Darek Zabrocki, Daren Bader,

Dariusz Zawadzki, Darrell Riche, Daryush Shokof, Dave Allsop, Dave Arredondo, Dave Dorman, Dave Gibbons, Dave Kendall, Dave McKean, Dave Melvin, David A Hardy, David A. Hardy, David Alfaro Siqueiros, David Allan, David Annand, David B. Mattingly, David Bailly, David Begbie, David Bomberg, David Boyd, David Brewster, David Budd, David Burluk, David Burton-Richardson, David Chipperfield, David Cooke Gibson, David Diao, David Donaldson, David Dougal Williams, David Eugene Henry, David Firth, David G. Sorensen, David Garner, David Gilmour Blythe, David Hockney, David Imms, David Inshaw, David LaChapelle, David Ligare, David Macaulay, David Macbeth Sutherland, David Martin, David Michie, David Octavius Hill, David Palumbo, David Park, David Paton, David Ramsay Hay, David Roberts, David Simpson, David Small, David Teniers III, David Teniers the Elder, David Teniers the Younger, David Watson Stevenson, David Wilkie, David Wojnarowicz, David Young Cameron, Davide Sasselli, De Hirsh Margules, Dean Cornwell, Dean Ellis, Dean Roger, Dechko Uzunov, Dee Whitcomb, Delaunay, Delmer J. Yoakum, Delphin Enjolras, Demetrios Farmakopoulos, Denis Eden, Dennis Ashbaugh, Dennis Flanders, Dennis H. Farber, Dennis Miller Bunker, Derek Chittock, Derek Gores, Derek Hill, Derek Jarman, Derf, Derold Page, Desmond Morris, Diane Arbus, Diane Dillon, Dick Bickenbach, Dicky Doyle, Didier Mouron, Diego Giacometti, Diego Gisbert Llorens, Diego Rivera, Diego Velázquez, Dieric Bouts, Dietmar Damerau, Dimitre Manassiev Mehandjiysky, Ding Guanpeng, Ding Yunpeng, Dino Valls, Dionisio Baixeras Verdaguer, Dionisius, Dirck Hals, Dirck de Bray, Dirck de Quade van Ravesteyn, Dirck van Baburen, Dirck van Delen, Dirck van der Lisse, Dirk Crabeth, Dirk Helmbreker, Ditlev Blunck, Dmitry Levitzky, Doc Hammer, Dod Procter, Dom Qwek, Domenichino, Domenico Ghirlandaio, Domenico Induno, Domenico Pozzi, Domenico Quaglio the Younger, Domenico Zampieri, Domenico di Pace Beccafumi, Domirinic Fegallia, Don Arday, Don Bluth, Don Eddy, Don Maitz, Don Reichert, Donald Judd, Donald Roller Wilson, Donald Sherwood, Donato Giancola, Dong Kingman, Dong Qichang, Dong Yuan, Dora Carrington, Dora Maar, Doris Blair, Doris Boulton-Maude, Dorning Rasbotham, Dorothea Braby, Dorothea Lange, Dorothea Tanning, Dorothea Warren O'Hara, Dorothy Bradford, Dorothy Burroughes, Dorothy Coke, Dorothy Elizabeth Bradford, Dorothy Hood, Dorothy Johnstone, Dorothy King, Dorothy Lockwood, Dossa Dossi, Doug Ohlson, Doug Wilder, Douglas Bourgeois, Douglas Robertson Bisset, Douglas Shuler, Dr. Atl, Dr. Seuss, Drew Struzan, Drew Tucker, Du Jin, Du Qiong, Duccio, Dugald Sutherland MacColl, Dulah Marie Evans, Duncan Grant, Dustin Nguyen, Dwight William Tryon, Dóra Keresztes, Dürer, david rubin, derek zabrocki, disney

E:

E. Charlton Fortune, E. Simms Campbell, E. T. A. Hoffmann, E. William Gollings, E.H. Shepard, ESAO, Eamon Everall, Earl Norem, Earle Bergey, Earnst Haeckel, Echo Chernik, Ed Benedict, Ed Binkley, Ed Emshwiller, Ed Paschke, Ed Roth, Eddie Campbell, Eddie Mendoza, Eden Box, Edgar Ainsworth, Edgar Degas, Edgar Schofield Baum, Edi Rama, Edith Edmonds, Edith Grace Wheatley, Edith Lawrence, Edmond Aman-Jean, Edmond Bille, Edmond Xavier Kapp, Edmund Blampied, Edmund Charles Tarbell, Edmund Dulac, Edmund F. Ward, Edmund Greacen, Edmund Leighton, Edna Mann, Edo Murtić, Eduard von Grützner, Eduard von Steinle, Eduardo Kingman, Eduardo Lefebvre Scovell, Eduardo Paolozzi, Edvard Munch, Edward Armitage, Edward Arthur Walton, Edward Atkinson Hornel, Edward Avedisian, Edward Bailey, Edward Baird, Edward Ben Avram, Edward Burne-Jones, Edward Clark, Edward Corbett, Edward Dugmore, Edward George Handel Lucas, Edward Gorey, Edward Henry Potthast, Edward Hicks, Edward Hopper, Edward Kemble, Edward Lamson Henry, Edward Lear, Edward Marshall Boehm, Edward Mitchell Bannister, Edward Okuń, Edward Otho Cresap Ord, II, Edward P. Beard Jr., Edward Robert Hughes, Edward Ruscha, Edward Simmons, Edward Sorel, Edward Wadsworth, Edward Weston, Edward Willis Redfield, Edwin Austin Abbey, Edwin Deakin, Edwin Dickinson, Edwin G. Lucas, Edwin Georgi, Edwin Landseer, Eero Järnefelt, Eero Snellman, Egbert van Heemskerck, Egbert van der Poel, Eglon van der Neer, Egon Schiele, Egon von Vietinghoff, Ei-Q, Eiichiro Oda, Eileen Aldridge, Einar Hakonarson, Eishōsai Chōki, Eizan Kikukawa, Eizō Katō, Ejler Bille, Ejnar Nielsen, El Greco, El Lissitzky, Elaine Duillo, Elaine Hamilton, Elaine de Kooning, Elbridge Ayer Burbank, Eleanor Best, Eleanor Fortescue-Brickdale, Eleanor Hughes, Eleanor Layfield Davis, Eleanor Vere Boyle, Elena Guro, Elfriede Lohse-Wächtler, Elias Goldberg, Elias Ravanetti, Elina Karimova, Elinor Proby Adams, Eliot Hodgkin, Elisabeth Collins, Eliseu Visconti, Elizabeth Charleston, Elizabeth Durack, Elizabeth Forbes, Elizabeth Jane Lloyd, Elizabeth MacNicol, Elizabeth Murray, Elizabeth Polunin, Elizabeth Shippen Green, Elizabeth York Brunton, Elke Vogelsang, Ella Guru, Ellen Gallagher, Elliott Erwitt, Ellsworth Kelly, Elmer Bischoff, Elmyr de Hory, Elsa Beskow, Elsa Bleda, Else Alfelt, Elsie Dalton Hewland, Elsie Few, Elsie Henderson, Elsie Vera Cole, Elwood H. Smith, Emanuel Büchel, Emanuel Leutze, Emanuel Schongut, Emanuel Witz, Emanuel de Witte, Emerson Silva, Emil Bistram, Emil Carlsen, Emil Fuchs, Emil Lindenfeld, Emil Nolde, Emil Orlik, Emile Auguste Carolus-Duran, Emile Lahner, Emiliano Di Cavalcanti, Emiliano Ponzi, Emilio Grau Sala, Emily Carr, Emily Mason, Emily Murray Paterson, Emily Shanks, Emma Andijewska, Emma Geary, Emma Lampert Cooper, Emma Ríos, Emory Douglas, Emperor Huizong of Song, Endre Bálint, Enguerrand Quarton, Enki Bilal, Enrico Baj, Enrique Grau, Enrique Simonet, Enrique Tábara, Enzo Cucchi, Eppo Doeve, Epsilon Point, Eric Auld, Eric Deschamps, Eric Dinyer, Eric Peterson, Eric Taylor, Eric Zener, Eric de Kolb, Erich Heckel, Erik Ortvad, Erik Pevernagie, Erin Hanson, Erlund Hudson, Ernest Biéler, Ernest Briggs, Ernest Buckmaster, Ernest Heber Thompson, Ernest Hébert, Ernest Lawson, Ernest Morgan, Ernest Procter, Ernest William Christmas, Ernest Zobole, Ernie Barnes, Ernst, Ernst Fuchs, Ernst Haeckel, Ernst Ludwig Kirchner, Ernst Thoms, Ernst Wilhelm Nay, Ernő Bánk, Ernő Grünbaum, Ernő Rubik, Ernő Tibor, Erwin Bowien, Esaias

Bourse, Esaias van de Velde, Esao Andrews, Esteban Vicente, Esther Blaikie MacKinnon, Estuardo Maldonado, Eszter Mattioni, Etel Adnan, Ethel Schwabacher, Etienne Delessert, Ettore "Ted" DeGrazia, Ettore Tito, Euan Uglow, Eugene Van Mieghem, Eugene Delacroix, Eugene J. Martin, Eugene Leroy, Eugene Montgomery, Eugene Tertychnyi, Eugene von Guerard, Eugenio Granell, Eugenio de Arriba, Eugeniusz Zak, Eugeniusz Żak, Eugène Boudin, Eugène Brands, Eugène Burnand, Eugène Carrière, Eugène Delacroix, Eugène Grasset, Eugène Isabey, Eva Frankfurter, Eva Gonzalès, Eva Švankmajerová, Evaline Ness, Evan Charlton, Evaristo Baschenis, Eve Disher, Eve Ryder, Evelyn Abelson, Evelyn Cheston, Evelyn De Morgan, Everett Raymond Kinstler, Everett Shinn, Everett Warner, Evert Collier, Evgeny Lushpin, Exekias, Eyvind Earle

F:

F Scott Hess, F. Scott Hess, Fabien Charuau, Fairfield Porter, Fan Kuan, Fan Qi, Fang Congyi, Fanny McIan, Farel Dalrymple, Farid Mansour, Fathi Hassan, Fay Pomerance, Fede Galizia, Federico Barocci, Federico Uribe, Federico Zandomenighi, Federico Zuccari, Fedot Sychkov, Fei Danxu, Felice Casorati, Felicity Charlton, Felipe Seade, Felix Octavius Carr Darley, Felix-Kelly, Feng Zhu, Fenghua Zhong, Ferdinand Bol, Ferdinand Hodler, Ferdynand Ruszczyk, Ferenc Joachim, Fern Coppedge, Fernand Léger, Fernand Pelez, Fernand Toussaint, Fernand Verhaegen, Fernando Amorsolo, Fernando Botero, Fernando Gerassi, Fikret Muallâ Saygi, Filip Hodas, Filippino Lippi, Fiona Rae, Fiona Stephenson, Fitz Henry Lane, Fitz Hugh Lane, Flavia Blois, Fletcher Martin, Flora Macdonald Reid, Florence Engelbach, Florianne Becker, Floris van Dyck, Floris van Schooten, Ford Madox Brown, Fra Angelico, Fra Bartolomeo, Fra Filippo Lippi, Frances C. Fairman, Frances Currey, Frances Hodgkins, Frances Jetter, Frances MacDonald, Francesco Albani, Francesco Bartolozzi, Francesco Bonsignori, Francesco Clemente, Francesco Cozza, Francesco Filippini, Francesco Furini, Francesco Guardi, Francesco Hayez, Francesco Raibolini, Francesco Zuccarelli, Francesco del Cossa, Francis Bacon, Francis Bourgeois, Francis Cadell, Francis Davis Millet, Francis Ernest Jackson, Francis Focer Brown, Francis Helps, Francis Picabia, Francis Souza, Francisco Goya, Francisco José de Goya, Francisco Oller, Francisco Zúñiga, Francisco de Burgos Mantilla, Francisco de Holanda, Francisco de Zurbarán, Franciszek Kostrzewski, Franciszek Smuglewicz, Franciszek Starowieyski, Franciszek Żmurko, Frank Auerbach, Frank Barrington Craig, Frank Buchser, Frank DuMond, Frank Frazetta, Frank J. Girardin, Frank Leonard Brooks, Frank Mason, Frank McKelvey, Frank Miller, Frank Montague Moore, Frank O'Meara, Frank Schoonover, Frank Stella, Frank Weston Benson, Frank Xavier Leyendecker, Franklin Booth, Franklin Carmichael, Frans Hals, Frans Koppelaar, Frans Masereel, Frans van Mieris the Elder, Frans van Mieris the Younger, František Jakub Prokyš, František Kaván, František Kupka, Franz Cizek, Franz Fedier, Franz Hegi, Franz Karl Basler-Kopp, Franz Kline, Franz Marc, Franz Sedlacek, Franz Stuck, Franz Vohwinkel, Franz Xaver Winterhalter, Franz von Lenbach, François Barraud, François Bacion, François Boquet, François Boucher, François Clouet, François Girardon, François Joseph Heim, François Louis Thomas Francia, François Quesnel, Fred A. Precht, Fred Cress, Fred Ludekens, Fred Marcellino, Fred Mitchell, Fred Williams, Frederic Church, Frederic Edwin Church, Frederic Leighton, Frederic Remington, Frederick Carl Frieseke, Frederick Edwin Church, Frederick Goodall, Frederick Hammersley, Frederick Lord Leighton, Frederick McCubbin, Frederik Vermehren, Frederik de Moucheron, Frida Kahlo, Friedel Dzubas, Friedensreich Hundertwasser, Friedrich Gauermaun, Friedrich Ritter von Friedländer-Malheim, Friedrich T raffaelt, Friedrich von Amerling, Frieke Janssens, Frits Thaulow, Frits Van den Berghe, Fritz Baumann, Fritz Bultman, Fritz Glarner, Fritz Puempin, Fritz von Dardel, Fritz von Uhde, Frédéric Bazille, Fu Baoshi, Fujishima Takeji, Fujiwara Nobuzane, Fujiwara Takanobu, Fuller Potter, Fuyuko Matsui, Fyodor Alekseyev, Fyodor Rokotov, Fyodor Slavyansky, Fyodor Vasilyev, Félix Arauz, Félix Bódog Widder, Félix Labisse, Félix Vallotton, Félix Ziem, ferdinand knab

G:

Gabor Breznay, Gabor Szikszai, Gabriel Ba, Gabriel Dawe, Gabriel Metsu, Gabriele Münter, Gabrijel Jurkić, Gaetano Previati, Gaetano Sabatini, Gai Qi, Galen Dara, Gang Hui-an, Gang Se-hwang, Gao Cen, Gao Fenghan, Gao Kegong, Gao Qipei, Gao Xiang, Garry Winogrand, Gary Panter, Gaston Anglade, Gaston Bussière, Gatöken Shunshi, Gaudi, Gaugin, Gavin Hamilton, Gavin Nolan, Gawen Hamilton, Gediminas Prankevicius, Gee Vaucher, Geertgen tot Sint Jans, Gen Paul, Genco Gulan, Gene Davis, Genevieve Springston Lynch, Gentile Bellini, Gentile Tondino, Geof Darrow, Geoffrey Dyer, Geoffrey Olsen, Georg Arnold-Graboné, Georg Baselitz, Georg Friedrich Kersting, Georg Friedrich Schmidt, Georg Muche, Georg Scholz, Georg Schrimpf, George Abe, George Aleef, George Ault, George B. Bridgman, George B. Sutherland, George Bain, George Barker, George Barret, Jr., George Barret, Sr., George Baselitz, George Bell, George Bellows, George Benjamin Luks, George Biddle, George Bogart, George Caleb Bingham, George Catlin, George Claessen, George Cruikshank, George Earl Ortman, George Fiddes Watt, George Frederic Watts, George Frederick Harris, George Gardner Symons, George Grosz, George Hendrik Breitner, George Henry, George Herbert Baker, George Hurrell, George Inness, George Jamesone, George Lambourn, George Lucas, George Luks, George Manson, George Morrison, George Papazov, George Passantino, George Paul Chalmers, George Pirie, George Reid, George Romney, George Stubbs, George Tooker, George Wyllie, George barbier, Georges Braque, Georges Emile Lebacqz, Georges Lacombe, Georges Lemmen, Georges Rouault, Georges Seurat, Georges Stein, Georges de La Tour, Georgia O'Keeffe, Georgia O'Keeffe, Georgina Hunt, Gerald Brom, Gerald Kelley, Gerald Kelly, Gerald van Honthorst, Gerard David, Gerard Houckgeest,

Gerard Seghers, Gerard Sekoto, Gerard Soest, Gerard de Lairesse, Gerard ter Borch, Gerbrand van den Eeckhout, Gerda Wegener, Gerhard Richter, Germaine Krull, Germán Londoño, Gerrit Adriaenszoon Berckheyde, Gerrit Dou, Gertrude Abercrombie, Gertrude Greene, Gertrude Harvey, Giacomo Balla, Giambattista Pittoni, Gian Lorenzo Bernini, Gianfredo Camesi, Gigadō Ashiyuki, Giger, Gijsbert d'Hondecoeter, Gil Elvgren, Gilbert Stuart, Gilberto Soren Zaragoza, Gilles Beloeil, Gillis Rombouts, Gillis d'Hondecoeter, Gina Pellón, Gino Severini, Giocondo Albertolli, Giorgio Cavallon, Giorgio De Vincenzi, Giorgio Giulio Clovio, Giorgio Morandi, Giorgio de Chirico, Giorgione, Giotto, Giovanni Antonio Galli, Giovanni Battista Cipriani, Giovanni Battista Gaulli, Giovanni Battista Innocenzo Colombo, Giovanni Battista Piazzetta, Giovanni Battista Piranesi, Giovanni Battista Tiepolo, Giovanni Bellini, Giovanni Bernardino Asoleni, Giovanni Bernardino Azzolini, Giovanni Bernardino Mazzolini, Giovanni Boldini, Giovanni Fattori, Giovanni Francesco Barbieri, Giovanni Giacometti, Giovanni Lanfranco, Giovanni Paolo Cavagna, Giovanni Paolo Pannini, Giovanni Pelliccioli, Girolamo Muziano, Giuseppe Abbati, Giuseppe Antonio Petrini, Giuseppe Arcimboldo, Giuseppe Avanzi, Giuseppe Bernardino Bison, Giuseppe Camuncoli, Giuseppe De Nittis, Giuseppe Grisoni, Giuseppe Tominz, Gladys Dawson, Gladys Kathleen Bell, Glen Angus, Glen Keane, Glenn Fabry, Glennray Tutor, Glenys Cour, Gloria Stoll Karn, Glòria Muñoz, Godfrey Blow, Godfried Schalcken, Gong Kai, Gong Xian, Gonzalo Endara Crow, Gordon Browne, Gordon Parks, Goro Fujita, Gottfried Helnwein, Govert Dircksz Camphuysen, Govert Flinck, Goyō Hashiguchi, Grace Clements, Grace Cossington Smith, Grace English, Grace Pailthorpe, Grace Polit, Graham Forsythe, Graham Sutherland, Grandma Moses, Grant Wood, Grayson Perry, Greg Hildebrandt, Greg Rutkowski, Greg Spalenka, Greg Staples, Gregorio Lazzarini, Gregorio Prestopino, Gregorius Sickinger, Gregory Crewdson, Gregory Gillespie, Gregory Manchess, Grete Stern, Grethe Jürgens, Grigoriy Myasoyedov, Grillo Demo, Griselda Allan, Grytë Pintukaitë, Grzegorz Domaradzki, Grzegorz Rutkowski, Gu An, Gu Hongzhong, Gu Kaizhi, Gu Zhengyi, Guan Daosheng, Guido Borelli da Caluso, Guido Reni, Guillermo del Toro, Guity Novin, Guo Chun, Guo Xi, Gustaf Munch-Petersen, Gustaf Tenggren, Gustav Doré, Gustav Klimt, Gustave Baumann, Gustave Boulanger, Gustave Caillebotte, Gustave Courbet, Gustave Dore, Gustave Doré, Gustave Moreau, Gustave Van de Woestijne, Guskuma Seihō, Gusztáv Kelety, Guy Denning, Guy Rose, Gwen Barnard, Gwen John, Gwenny Griffiths, Gwilym Prichard, Gyoshū Hayami, Gyula Aggházy, Gyula Basch, Gyula Batthyány, Gyula Benczúr, Gyula Derkovits, György Rózsahegyi, György Vastagh, Gérard Ernest Schneider, Géza Dósa, Géza Mészöly, Géza Udvary, gaston bussiere, gustav dore

H:

H. R. (Hans Ruedi) Giger, H. R. Giger, H.P. Lovecraft, H.R. Giger, H.R. Millar, HAP Grieshaber, Haddon Sundblom, Hal Foster, Hallsteinn Sigurðsson, Hamilton Sloan, Hamish MacDonald, Han Gan, Hanabusa Itchō, Hanabusa Itchō II, Hanna-Barbera, Hannah Frank, Hannah Höch, Hannah Tompkins, Hans Katz, Hans Asper, Hans Baldung, Hans Baluschek, Hans Beat Wieland, Hans Bellmer, Hans Bol, Hans Burgkmair, Hans Eduard von Berlepsch-Valendas, Hans Erni, Hans Falk, Hans Fischer, Hans Fries, Hans Gude, Hans Hartung, Hans Hinterreiter, Hans Hofmann, Hans Holbein the Elder, Hans Holbein the Younger, Hans Leu the Elder, Hans Makart, Hans Memling, Hans Mertens, Hans Sandreuter, Hans Schwarz, Hans Werner Schmidt, Hans von Aachen, Hans von Bartels, Harald Giersing, Hariton Pushwagner, Harold Elliott, Harold Gilman, Harold Harvey, Harold Sandys Williamson, Harold Shapinsky, Harold von Schmidt, Harriet Backer, Harriet Powers, Harriet Zeitlin, Harrington Mann, Harrison Fisher, Harry Beckhoff, Harry Clarke, Harry Haenigsen, Harry Morley, Harry Shoulberg, Harumi Hironaka, Harvey Dunn, Harvey Kurtzman, Harvey Pratt, Harvey Quaytman, Hasegawa Settan, Hasegawa Tōhaku, Hashimoto Gahō, Hasui Kawase, Haukur Halldórsson, Hayao Miyazaki, Hazel Armour, Heather Hudson, Hedda Sterne, Hedi Xandt, Heinrich Bichler, Heinrich Brocksieper, Heinrich Danioth, Heinrich Herzig, Heinrich Hofmann, Heinrich Kley, Heinrich Lefler, Heinrich Maria Davringhausen, Heinz Anger, Heinz Edelman, Helen Berman, Helen Biggar, Helen Binyon, Helen Dahm, Helen Edwards, Helen Frankenthaler, Helen Stevenson, Helen Thomas Dranga, Helene Schjerfbeck, Helmut Federle, Helmut Newton, Hendrick Avercamp, Hendrick Bloemaert, Hendrick Cornelisz Vroom, Hendrick Cornelisz. van Vliet, Hendrick Goudt, Hendrick Terbrugghen, Hendrick van Balen, Hendrick van Streeck, Hendrik Gerritsz Pot, Hendrik Goltzius, Hendrik Martenszoon Sorgh, Hendrik Willem Mesdag, Hendrik van Steenwijk I, Hendrik van Steenwijk II, Henning Jakob Henrik Lund, Henri Alphonse Barnoin, Henri Bellechose, Henri Biva, Henri Cartier-Bresson, Henri Fantin-Latour, Henri Harpignies, Henri Le Sidaner, Henri Matisse, Henri Michaux, Henri Rousseau, Henri de Toulouse-Lautrec, Henri-Edmond Cross, Henri-Julien Dumont, Henric Trenk, Henricus Hondius II, Henriett Seth F., Henriette Grindat, Henriette Wyeth, Henrik Weber, Henry B. Christian, Henry Bright, Henry Carr, Henry Fuseli, Henry Heerup, Henry Ives Cobb, Jr., Henry Justice Ford, Henry Lamb, Henry Macbeth-Raeburn, Henry Moore, Henry O. Tanner, Henry Ossawa Tanner, Henry Otto Wix, Henry Raeburn, Henry Raleigh, Henry Scott Tuke, Henry Snell Gamley, Henry Tonks, Henry Wallis, Henry Woods, Henry van de Velde, Henryk Rodakowski, Henryk Siemiradzki, Henryk Stazewski, Henryka Beyer, Herb Aach, Herb Ritts, Herbert Abrams, Herbert Bayer, Herbert James Gunn, Herbert MacNair, Hercules Seghers, Herman Saftleven, Herman van Swanevelt, Hermann Feierabend, Hermann Rüdissühli, Hermenegildo Anglada Camarasa, Hermione Hammond, Hew Lorimer, Hidari Jingorō, Hideyuki Kikuchi, Hieronim Bosch, Hieronymus Bosch, Hieronymus Bosch, Hikari Shimoda, Hilda Annetta Walker, Hilda May Gordon, Hilde Goldschmidt, Hilma af Klint, Hinchel Or, Hirohiko Araki, Hiromitsu Takahashi, Hiromu Arakawa, Hirosada II, Hiroshi Honda, Hiroshi Nagai, Hiroshi Yoshida, Hiroshige, Hiroyuki Tajima, Hishida Shunsō, Hishikawa Moronobu,

Hisui Sugiura, Hokusai, Holger Roed, Hong Ren, Honoré Daumier, Horace Vernet, Horatio McCulloch, Horatio Nelson Poole, Horst Antes, Hovsep Pushman, Howard Arkley, Howard Butterworth, Howard Chandler Christy, Howard Chaykin, Howard Finster, Howard Hodgkin, Howard Kanovitz, Howard Knotts, Howard Lyon, Howard Mehring, Howard Pyle, Howardena Pindell, Hristofor Zhefarovich, Hristofor Žefarović, Hsiao-Ron Cheng, Hu Jieqing, Hu Zao, Hu Zaobin, Hua Yan, Huang Binhong, Huang Ding, Huang Gongwang, Huang Guangjian, Huang Ji, Huang Shen, Huang Tingjian, Hubert Robert, Hubert van Eyck, Hubert van Ravesteyn, Hubert von Herkomer, Hugh Adam Crawford, Hugh Ferriss, Hugh Hughes, Hugh William Williams, Hugo Anton Fisher, Hugo Heyrman, Hugo Kārlis Grotuss, Hugo Scheiber, Hugo Simberg, Hugo Sánchez Bonilla, Hugo van der Goes, Humberto Castro, Hundertwasser, Hyacinthe Rigaud, Hyman Bloom, hajime sorayama, helen huang, hr giger

I:

I Ketut Soki, IAN SPRIGGS, INO, Ian Fairweather, Ian Hamilton Finlay, Ian McQue, Ian Miller, Ib Eisner, Ibrahim Kodra, Ibram Lassaw, Ida Rentoul Outhwaite, Ignacio Bazan-Lazcano, Ignacio Zuloaga, Ignacy Witkiewicz, Ignat Bednarik, Igor Grabar, Igor Kieryluk, Igor Kufayev, Igor Morski, Igor Zenin, Ithor Podolchak, Ike no Taiga, Ikuo Hirayama, Ilka Gedő, Illarion Pryanishnikov, Ilya Glazunov, Ilya Kuvshinov, Ilya Ostroukhov, Ilya Repin, Ilya Yefimovich Repin, Ina Wong, Ingrida Kadaka, Inoue Naohisa, Inshō Dōmoto, Ion Andreescu, Irene Bache, Irene Lieblich, Irene and Laurette Patten, Irma Stern, Irvin Bomb, Isaac Grünewald, Isaac Levitan, Isaac Soyer, Isaac van Ostade, Isabel Codrington, Isabel Naftel, Isamu Noguchi, Isidor Kaufman, Isidore Bonheur, Ismael Nery, Ismail Acar, Ismail Gulgee, Ismail Inceoglu, Isobel Heath, Isobelle Ann Dods-Withers, Israel Tsvaygenbaum, Istvan Banyai, Istvan Horkay, István Csók, István Nagy, István Orosz, István Regős, István Réti, István Szőnyi, István Árkossy, Itagaki Yoshio, Italo Mus, Itshak Holtz, Itō Jakuchū, Itō Ogura Yonesuke, Itō Sei, Itō Shinsui, Ivan Aivazovsky, Ivan Albright, Ivan Bilibin, Ivan Generalić, Ivan Grohar, Ivan Kramskoi, Ivan Lacković Croatia, Ivan Meštrović, Ivan Mrkvička, Ivan Ranger, Ivan Shishkin, Ivan Trush, Ivan Yakovlevich Vishnyakov, Ivana Kobilca, Ivor Davies, Ivor Williams, Iwao Takamoto, Iwasa Matabei, Izidor Kršnjavi, Izzy Medrano, irakli nadar

J:

J. Alden Weir, J. C. Leyendecker, J. Frederick Smith, J. J. Grandville, J. L. Lund, J. M. W. Turner, J. Massey Rhind, J. Ottis Adams, J. W. Tristram, J.C. Leyendecker, J.M.W. Turner, Jaakko Mattila, Jacek Andrzej Rossakiewicz, Jacek Malczewski, Jacek Sempoliński, Jacek Yerka, Jack Boul, Jack Butler Yeats, Jack C. Mancino, Jack Davis, Jack Gaughan, Jack Kirby, Jack Levine, Jack M. Ducker, Jack Roth, Jack Smith, Jackson Pollock, Jacob Adriaensz Backer, Jacob Burck, Jacob Collins, Jacob Duck, Jacob Esselens, Jacob Gerritsz. Cuyp, Jacob Jordaens, Jacob Kainen, Jacob Koninck, Jacob Lawrence, Jacob Maris, Jacob Marrel, Jacob More, Jacob Ochtervelt, Jacob Philipp Hackert, Jacob Pynas, Jacob Savery, Jacob Steinhart, Jacob Toorenvliet, Jacob Willemszoon de Wet, Jacob de Gheyn II, Jacob de Heusch, Jacob van Campen, Jacob van Ruisdael, Jacob van Utrecht, Jacob van der Ulft, Jacopo Amigoni, Jacopo Baccarini, Jacopo Bassano, Jacopo Bellini, Jacopo Pontormo, Jacopo de' Barbari, Jacques Blanchard, Jacques Callot, Jacques Daret, Jacques Hérold, Jacques Maroger, Jacques Sablet, Jacques Villon, Jacques-François Ochart, Jacques-Laurent Agasse, Jacques-Louis David, Jacqui Morgan, Jaime Colson, Jaime Jones, Jakob Emanuel Handmann, Jakob Gauer mann, Jakob Häne, Jakub Husnik, Jakub Rozalski, Jakub Różalski, Jakub Schikaneder, James Abbott McNeill Whistler, James Ardern Grant, James Ballantine, James Bard, James Barry, James Bateman, James Baynes, James Bolivar Manson, James Brooks, James C. Christensen, James Cadenhead, James Campbell Noble, James Cowie, James Cromar Watt, James Dickson Innes, James E. Brewton, James Ensor, James Giles, James Gilleard, James Gillick, James Gillray, James Gurney, James Guthrie, James Humbert Craig, James Jarvaise, James Jean, James Jonathan Jeffries, James McBey, James McIntosh Patrick, James McNeill Whistler, James Montgomery Flagg, James Morris, James Morrison, James Paick, James Paterson, James Peale, James Pittendrigh MacGillivray, James Rosenquist, James Ryman, James Sessions American painter, James Thomas Watts, James Tissot, James Warhola, James Wood, Jamie Hewlett, Jamie Wyeth, Jan Antonisz van Ravesteyn, Jan Asselijn, Jan Baptist Weenix, Jan Brett, Jan Brueghel the Elder, Jan Brueghel the Younger, Jan Cornelisz Vermeyen, Jan Cox, Jan Davidsz de Hem, Jan Davidsz. de Heem, Jan Dirksz Both, Jan Gossaert, Jan Gregoor, Jan Griffier, Jan Hackaert, Jan Henryk Rosen, Jan Jahn, Jan Karpíšek, Jan Kip, Jan Konůpek, Jan Kupecký, Jan Lievens, Jan Matejko, Jan Miel, Jan Miense Molenaer, Jan Müller, Jan Nieuwenhuys, Jan Pynas, Jan Rustem, Jan Sawka, Jan Stanisławski, Jan Steen, Jan Tegnagel, Jan Theuninck, Jan Toorop, Jan Verkolje, Jan Victors, Jan Wijnants, Jan Wyck, Jan Zrzavý, Jan de Baen, Jan de Bray, Jan van Bijlert, Jan van Eyck, Jan van Goyen, Jan van Huysum, Jan van Kessel the Elder, Jan van Kessel the Younger, Jan van Mieris, Jan van de Cappelle, Jan van der Heyden, Jane Carpanini, Jane Frank, Jane Freeman, Jane Freilicher, Jane Graverol, Jane Hawkins, Jane Kelly, Jane Nasmyth, Jane Small, Janet Archer, Janet Dawson, Janet Fish, Jang Seung-eop, Jankel Adler, January Suchodolski, Jaroslav Doubrava, Jaroslava Korol, Jarosław Jaśnikowski, Jasmine Becket-Griffith, Jason A. Engle, Jason Benjamin, Jason Chan, Jason Edmiston, Jason Felix, Jason Teraoka, Jasper Francis Cropsey, Jasper Johns, Jay Hambidge, Jay Milder, Jean Antoine Watteau, Jean Arp, Jean Auguste Dominique Ingres, Jean Baptiste Debret, Jean Béraud, Jean Clark, Jean Colombe, Jean Crotti, Jean Cunningham, Jean Delville, Jean Dubuffet, Jean Dufy, Jean Fautrier, Jean Fouquet, Jean Giraud, Jean Hey, Jean Huber Voltaire, Jean Hé lion, Jean Jouvenet, Jean Louis Petitot, Jean Malouel, Jean Messagier, Jean Metzinger, Jean Micheal Basquiat, Jean Petitot, Jean Tabaud, Jean Xceron,

Jean-Antoine Watteau, Jean-Auguste-Dominique Ingres, Jean-Baptiste Monge, Jean-Baptiste Pater, Jean-Baptiste van Loo, Jean-Baptiste-Camille Corot, Jean-Baptiste-Siméon Chardin, Jean-François Millet, Jean-François de Troy, Jean-Honoré Fragonard, Jean-Jacques Henner, Jean-Louis-Ernest Meissonier, Jean-Léon Gérôme, Jean-Marc Nattier, Jean-Michel Atlan, Jean-Michel Basquiat, Jean-Paul Laurens, Jean-Paul Riopelle, Jean-Pierre Norblin de La Gourdaie, Jean-Pierre Saint-Ours, Jean-Yves Couliou, Jean-Étienne Liotard, Jeanna bauck, Jeanne Hébuterne, Jeanne du Maurier, Jeff A. Menges, Jeff Easley, Jeff Koons, Jeff Miracola, Jeffrey Catherine Jones, Jeffrey Smith, Jeka Kemp, Jemima Blackburn, Jenaro de Urrutia Olanar, Jennifer Bartlett, Jennifer Janesko, Jenny Eakin Delony, Jenny Saville, Jens Ferdinand Willumsen, Jens Juel, Jens Jørgen Thorsen, Jens Søndergaard, Jenő Barcsay, Jenő Gyárfás, Jeong Seon, Jeremiah Ketner, Jeremy Chong, Jeremy Geddes, Jeremy Henderson, Jerry Eisenberg, Jerry Pinkney, Jerry Schatzberg, Jerry Weiss, Jerry Wilkerson, Jerzy Kossak, Jesper Ejsing, Jesper Knudsen, Jesper Myrfors, Jesse Richards, Jessie Alexandra Dick, Jessie Algie, Jessie M. King, Jessie Newbery, Jessie Willcox Smith, Jesús Mari Lazkano, Ji Sheng, Jiang Tingxi, Jiao Bingzhen, Jim Burns, Jim Davis, Jim Dine, Jim Lee, Jim Manley, Jim Murray, Jim Nelson, Jim Steranko, Jimmy Boyle, Jimmy Ernst, Jin Homura, Jin Nong, Jiro Yoshihara, Jitish Kallat, Jo Baer, JoWonder, Joachim Patinir, Joan Ayling, Joan Brown, Joan Miro, Joan Miró, Joan Snyder, Joan Tuset, Joan Warburton, Joanna Carrington, Joaquín Clausell, Joaquín Sorolla, Joaquín Torres García, Jodorowsky, Joe Bowler, Joe Fenton, Joe Jusko, Joe Machine, Joe Mangrum, Joe Shuster, Joe Sorren, Joe Stefanelli, Joe de Mers, Joel Shapiro, Johan Christian Dahl, Johan Edvard Mandelberg, Johan Grenier, Johan Jongkind, Johan Lundbye, Johann Balthasar Bullinger, Johann Berthelsen, Johann Bodin, Johann Caspar Füssli, Johann Christian Brand, Johann Friedrich Overbeck, Johann George Schmidt, Johann Gottfried Steffan, Johann Heinrich Bleuler, Johann Heinrich Meyer, Johann Jakob Biedermann, Johann Kaspar Füssli, Johann Kretzschmer, Johann Liss, Johann Ludwig Aberli, Johann Ludwig Bleuler, Johann Pucher, Johann Rudolf Byss, Johann Zoffany, Johanna Marie Fosie, Johannes Bosschaert, Johannes Cornelisz Verspronck, Johannes Fabritius, Johannes Heisig, Johannes Helgeson, Johannes Itten, Johannes Lingelbach, Johannes Martini, Johannes Mytens, Johannes Vermeer, Johannes Voss, Johannes van Haensbergen, Johfra Bosschart, John Alexander, John Altoon, John Anster Fitzgerald, John Armleder, John Armstrong, John Atherton, John Atkinson Grimshaw, John Avon, John Backderf, John Bauer, John Bellany, John Berkey, John Blair, John Blanche, John Bourne, John Brack, John Broadley, John Brown, John Brown Abercromby, John Button, John Byrne, John Cale, John Carpenter, John Chamberlain, John Clayton, John Clayton Adams, John Collier, John Constable, John Covert, John Crawford Brown, John Doman Turner, John Duncan Fergusson, John E. Berninger, John Ellsworth Weis, John Elwood Bundy, John Everett Millais, John Eyre, John F. Francis, John F. Peto, John Fabian Carlson, John Frederick Herring, Jr., John Frederick Herring, Sr., John Frederick Kensett, John French Sloan, John Fulton Folinsbee, John George Sowerby, John Gibson, John Haberle, John Harris, John Henderson, John Henry Kingsley, John Henry Lorimer, John Henry Tachtman, John Howe, John Hoyland, John Hutchison, John Hutton, John J Park, John James Audubon, John Kay, John Keane, John Kelso Hunter, John Kingsley Cook, John La Gatta, John Lavery, John Laviers Wheatley, John Linnell, John Lowrie Morrison, John Luke, John Macdonald Aiken, John Maggs, John Marin, John Martin, John Matson, John Maxwell, John McLaughlin, John Michael Wright, John Moonan, John Mossman, John Murdoch, John Nelson Battenberg, John Nicolson, John Noble Barlow, John Opie, John Parker, John Perceval, John Pettie, John Philip Falter, John Platt, John Plumb, John Quidor, John Quinton Pringle, John Rhind, John Robertson Reid, John Romita Jr, John Runciman, John Russell Fulton, John Salminen, John Singer Sargent, John Singleton Copley, John Skinner Prout, John Sloan, John Souch, John Steell, John Steuart Curry, John Stevenson Rhind, John Stuart Ingle, John Thomson of Duddingston, John Trumbull, John Uzzell Edwards, John Watson Gordon, John William Godward, John William Waterhouse, John Wilson, John Wollaston, John Wonnacott, John Zephaniah Bell, Jon Coffelt, Jon Foster, Jon Whitcomb, Jonas De Ro, Jonathan Solter, Joong Keun Lee, Joop Sanders, Joos de Momper, Jordan Grimmer, Jorge Jacinto, Jorge Velarde, Joris van der Haagen, Josan Gonzalez, Josef Abel, Josef Albers, Josef Block, Josef Dande, Josef Jackerson, Josef Mánes, Josef Navrátil, Josef Čapek, Josef Šima, Josefina Tanganelli Plana, Josep Rovira Soler, Joseph Badger, Joseph Beuys, Joseph Bowler, Joseph Clement Coll, Joseph Cornell, Joseph DeCamp, Joseph Delaney, Joseph Ducreux, Joseph Dwight Strong, Joseph Heintz the Elder, Joseph Henderson, Joseph Ignaz Appiani, Joseph Keppler, Joseph Kleitsch, Joseph Morgan Henninger, Joseph Noel Paton, Joseph Pickett, Joseph Pisani, Joseph Raphael, Joseph Severn, Joseph Stella, Joseph Werner, Joseph Wright of Derby, Joseph Yoakum, Joseph von Führich, Joseph-Marie Vien, Josephine Wall, Josetsu, Josh Bayer, Joshua Reynolds, Josse Lieferinxe, José Clemente Orozco, José Comas Quesada, José Malhoa, Joy Garnett, Joyce Ballantyne Brand, Joze Ciuha, Jozef Czapski, Jozef Israëls, Jozef Summler, João Artur da Silva, Ju Chao, Ju Lian, Juan Carlos Stekelman, Juan Fernando Cobo, Juan Giménez, Juan Gris, Juan Luna, Juan O'Gorman, Juan Villafuerte, Juan de Flandes, Juan de Valdés Leal, Judith Brown, Judith Gutierrez, Judith Leyster, Judy Cassab, Judy Takács, Juergen von Huendeberg, Jules Bastien-Lepage, Jules Breton, Jules Chéret, Jules Joseph Lefebvre, Jules Olitski, Jules Pascin, Jules Robert Auguste, Jules Tavernier, Julia Margaret Cameron, Julia Pishtar, Julian Allen, Julian Fałat, Julian Hatton, Julian Onderdonk, Julian Schnabel, Julie Bell, Juliette Leong, Juliette Wytzman, Julio González, Julio Larraz, Julius Edvard Marak, Julius Exner, Julius Hatofsky, Julius Klinger, Julius LeBlanc Stewart, Juliusz Kossak, Jung Park, Junji Ito, Junpei Satoh, Junsaku Koizumi, Juraj Julije Klović, Juriaen van Streeck, Justin Currie, Justin Gerard, Justin Sweet, Justus van Gent, János Donát, János Kass, János Nagy Balogh, János Saxon-Szász, János Tornyai, János Valentiny, János Vaszary, Jóhannes Geir Jónsson, Jóhannes Sveinsson Kjarval, Józef Brandt, Józef Mehoffer, Józef Pankiewicz, Józef Szermentowski, József Borsos, József Breznay, József Egry, József Koszta, József Rippl-

Rónai, Jörg Immendorff, Jørgen Nash, Jørgen Roed, Július Jakoby, Júlíana Sveinsdóttir, Jürg Kreienbühl, james christensen, jean moebius giraud, jeonseok lee, jessica rossier, joseph christian leyendecker

K:

Kaburagi Kiyokata, Kadir Nelson, Kaff Gerrard, Kagaku Murakami, Kahlo, Kaigetsudō Anchi, Kaigetsudō Ando, Kai Higashiyama, Kailash Chandra Meher, Kaja Foglio, Kalervo Palsa, Kamagurka, Kameda Bōsai, Kamisaka Sekka, Kamāl ud-Dīn Behzād, Kanbun Master, Kandinsky, Kanzan Shimomura, Kanō Eitoku, Kanō Hōgai, Kanō Masanobu, Kanō Mitsunobu, Kanō Motonobu, Kanō Naizen, Kanō Sanraku, Kanō Sansetsu, Kanō Shōsenin, Kanō Takanobu, Kanō Tan'yū, Kanō Tanshin, Karel Appel, Karel Dujardin, Karel Havlíček, Karel Klíč, Karel van Mander, Karel Štěch, Karl Ballmer, Karl Bodmer, Karl Bryullov, Karl Buesgen, Karl Gerstner, Karl Hagedopoor, Karl Hagedorn, Karl Hofer, Karl Jauslin, Karl Kopinski, Karl Matzek, Karl Otto Götz, Karl Pümpin, Karl Schmidt-Rottluff, Karl Stauffer-Bern, Karl Thiart, Karl Völker, Karl Walser, Karl Zerbe, Karol Bak, Kate Beaton, Kate Greenaway, Katerina Wilczynski, Kathleen Allen, Kathleen Browne, Kathleen Guthrie, Kathleen Scott, Kathleen Walne, Kati Horna, Katia Chausheva, Katsuchika Hokusai, Katsukawa Shun'ei, Katsukawa Shunchō, Katsukawa Shunkō I, Katsukawa Shunsen, Katsukawa Shunshō, Katsushika Hokusai, Katsushika Ōi, Kawabata Ryūshi, Kawai Gyokudō, Kawanabe Kyōsai, Kawase Hasui, Kay Nielsen, Kay Sage, Kazimierz Alchimowicz, Kazimierz Wojniakowski, Kazimir Malevich, Kazys Varnelis, Ke Jiusi, Kees Bol, Kees Maks, Kees Scherer, Kees van Dongen, Keisai Eisen, Keith Haring, Keith Henderson, Keith Mallett, Keith Parkinson, Kelly Freas, Kelly McKernan, Kelly Sueda, Ken Danby, Ken Elias, Ken Howard, Ken Messer, Ken Sugimori, Kenneth Noland, Kentaro Miura, Keos Masons, Ker-Xavier Roussel, Keraca Visulčeva, Kerembeyit, Keren Katz, Kev Walker, Kevin A. Short, Kevin Connor, Khalil Gibran, Kieran Yanner, Kikuchi Yōsai, Kilian Eng, Kim Deuk-sin, Kim Du-ryang, Kim Eung-hwan, Kim Hong-do, Kim Hwan-gi, Kim Jeong-hui, Kim Keever, Kim Myeong-guk, Kim Tschang Yeul, Kinichiro Ishikawa, Kinuko Craft, Kinuko Y. Craft, Kishi Ganku, Kitagawa Utamaro, Kitao Shigemasa, Kiyohara Tama, Kiyoshi Yamashita, Klimt, Knox Martin, Knud Agger, Kobayashi Kiyochika, Kogan Gegei, Koloman Sokol, Konrad Grob, Konrad Klapheck, Konrad Krzyzanowski, Konrad Witz, Konstantin Korovin, Konstantin Makovsky, Konstantin Savitsky, Konstantin Somov, Konstantin Vasilyev, Konstantin Westchilov, Konstantin Yuon, Konstantinas Ciurlionis, Koryusai Isoda, Kose Kanaoka, Koson Ohara, Krenz Cushart, Kristian Kreković, Kristian Zahrtmann, Kristin Nelson, Krzysztof Boguszewski, Ksenia Milicevic, Kubisi art, Kume Keiichiro, Kun Can, Kuno Veeber, Kuroda Seiki, Kurt Roesch, Kurt Schwitters, Kurt Seligmann, Kurt Trampedach, Kurt Wenner, Kusama, Kusumi Morikage, Kyffin Williams, Kyle Lambert, Károly Brocky, Károly Ferenczy, Károly Kernstok, Károly Kisfaludy, Károly Lotz, Károly Markó the Elder, Károly Patkó, Käthe Kollwitz, Kōno Bairei, Kōno Michisei, Kōshirō Onchi, karlkka, karolis strauntniekas, katsuya terada, ken kelly

L:

L. A. Ring, L. Alcopley, L. S. Lowry, L. Vance Phillips, Ladrönn, Lajos Berán, Lajos Bruck, Lajos Gulácsy, Lajos Kassák, Lajos Tihanyi, Lajos Vajda, Lale Westvind, Lam Qua, Lambert Doomer, Lambert Jacobsz, Lan Ying, Lari Pittman, Larry D. Alexander, Larry Elmore, Larry Fink, Larry Rivers, Larry Zox, Lars Grant-West, Lars Jonson Haukaness, Lasar Segall, Lattanzio Gambaro, Laura Ford, Laura Knight, Laura Muntz Lyall, Laura Theresa Alma-Tadema, Laura Wheeler Waring, Laurel Burch, Laurie Lipton, Laurits Tuxen, Lawren Harris, Lawrence Alma-Tadema, Lawrence Calcagno, Lawrence Harris, Lawton S. Parker, LeConte Stewart, LeRoy Neiman, Leandro Erlich, Lee Gatch, Lee Jeffries, Lee Loughridge, Lee Madgwick, Lefevre James Cranstone, Leiko Ikemura, Leila Faithfull, Leland Bell, Lena Alexander, Leng Mei, Lennie Lee, Leo Goetz, Leo Leuppi, Leo Lionni, Leo Michelson, Leo Valledor, Leo and Diane Dillon, Leon Chwistek, Leon Golub, Leon Kapliński, Leon Kossoff, Leon Kroll, Leon Polk Smith, Leon Wyczółkowski, Leona Wood, Leonaert Bramer, Leonard Appelbee, Leonard Bahr, Leonard Daniels, Leonard Long, Leonard Ochtman, Leonardo da Vinci, Leonid Afremov, Leonid Pasternak, Leonor Fini, Leonora Carrington, Leopold Gottlieb, Les Edwards, Lesser Ury, Leticia Gillett, Lev Lvovich Kamenev, Lewis Henry Meakin, Li Cheng, Li Chevalier, Li Di, Li Fangying, Li Gonglin, Li Kan, Li Keran, Li Mei-shu, Li Rongjin, Li Shan, Li Shida, Li Shixing, Li Song, Li Tang, Li Tiefert, Li Zai, Liam Wong, Liang Kai, Liao Chi-chun, Lichtenstein, Lili Árkayné Sztéhló, Lilia Alvarado, Lilla Cabot Perry, Lillian Bassman, Lily Delissa Joseph, Limbourg brothers, Lin Liang, Lin Tinggui, Linda Sutton, Lindsay Bartholomew, Lionel Lindsay, Lionel Walden, Lisa Frank, Lisa Milroy, Lisa Nankivil, Lisa Yuskavage, Lise Deharme, Liu Haisu, Liu Jue, Liu Jun, Liubov Popova, Liza Donnelly, Lizzy Ansingh, Lodewijk Bruckman, Loic Zimmerman, Lois Dodd, Lois Mailou Jones, Lois van Baarle, Lojze Logar, Loren Munk, Lorentz Frölich, Lorenzo Lotto, Lorraine Fox, Lotte Reiniger, Louis Anquetin, Louis Buvelot, Louis Comfort Tiffany, Louis Eilshemius, Louis Faure, Louis Glackens, Louis Grell, Louis Hersent, Louis Janmot, Louis Le Nain, Louis Marcoussis, Louis Mathieu Verdilhan, Louis Pohl, Louis Schanker, Louis Stettner, Louis Valtat, Louis Wain, Louis de Caullery, Louis le Brocqy, Louis-François Aubry, Louis-Léopold Boilly, Louis-Michel van Loo, Louisa Chase, Louisa Matthíasdóttir, Louisa Puller, Louise Abbéma, Louise Bourgeois, Louise Catherine Breslau, Louise Nevelson, Lovis Corinth, Lu Guang, Lu Zhi, Lubin Baugin, Lubna Agha, Luc Tuymans, Luc-Olivier Merson, Luca Zontini, Luca della Robbia, Lucas Cranach the Elder, Lucas Cranach the Younger, Lucas Graciano, Lucas Vorsterman, Lucas van Leyden, Lucebert, Lucette Barker, Lucia Peka, Lucian Freud, Lucien Pissarro, Lucio Fontana, Lucy Angeline Bacon, Lucy Madox Brown, Ludolf Bakhuizen, Ludolf Leendertz de Jongh,

Ludovico Carracci, Ludovit Fulla, Ludwig Bemelmans, Ludwig Knaus, Ludwig Mestler, Ludwik Konarzewski, Ludwik Konarzewski Jr, Luděk Marold, Luigi Kasimir, Luis Enrique Camej, Luis Marsans, Luis Miranda, Luis Molinari, Luis Paret y Alcazar, Luis Royo, Lujo Bezeredi, Luma Rouge, Luo Mu, Luo Ping, Luo Zhichuan, Lydia Field Emmet, Lyle Tuttle, Lynda Benglis, Lynn Pauley, Lyonel Feininger, Lyubov Popova, László Balogh, László Beszédes, László Dombrovsky, László Mednyánszky, László Moholy-Nagy, László Paál, Léon Bakst, Líviusz Gyulai, Lü Ji, loish, lovecraft

M:

M. C. Escher, M. F. Husain, M.C. Escher, MC Escher, Ma Lin, Ma Quan, Ma Shi, Ma Wan, Ma Yuan, Ma Yuanyu, Mab Graves, Mabel Rollins Harris, Mac Conner, Maciej Kuciara, Madge Oliver, Maeda Masao, Maeda Seison, Magali Villeneuve, Magdalene Bärens, Maggie Hamilton, Maginel Wright Enright Barney, Magnus Colcord Heurlin, Mahmoud Farshchian, Maki Haku, Makoto Aida, Makoto Shinkai, Maksimilijan Vanka, Maksymilian Gierymski, Makuzu Kōzan, Malcolm Drummond, Malcolm Morley, Malcom Howie, Malczewski, Maler Müller, Malevich, Malvin Gray Johnson, Man Ray, Manabu Mabe, Mandy Jurgens, Manfredi Beninati, Manjit Bawa, Manolo Millares, Manuel Franquelo, Manuel Ortiz de Zarate, Manuel Rendón Seminario, Marc Bell, Marc Chagall, Marc Simonetti, Marc Tetro, Marc-Aurèle de Foy Suzor-Coté, Marcel Chagall, Marcel Duchamp, Marcell Chagall, Marcelle Bergerol, Marcello Bacciarelli, Marcin Zaleski, Marco Mazzoni, Marcos Restrepo, Mardi Barrie, Marek Okon, Margaret Backhouse, Margaret Bingham, Margaret Boden, Margaret Bruce Wells, Margaret Brundage, Margaret Garland, Margaret Geddes, Margaret Graeme Niven, Margaret Keane, Margaret Leiteritz, Margaret Macdonald Mackintosh, Margareta Alströmer, Margareta Sterian, Margit Anna, Margo Hoff, Marguerite Zorach, Maria Helena Vieira da Silva, Maria Johanna Görtz, Maria Oriana Galli Bibiena, Maria Sibylla Merian, Maria van Oosterwijk, Marianne North, Marianne von Werefkin, Marie Angel, Marie Bashkirtseff, Marie Bracquemond, Marie Courtois, Marie Krøyer, Marie Laurencin, Marie Vassilieff, Marie-Gabrielle Capet, Marie-Suzanne Giroust, Marilyn Bendell, Marina Abramović, Mario Bardi, Mario Comensoli, Mario Cooper, Mario Dubsky, Mario Sironi, Marion Ancrum, Marion Wachtel, Mariotto Albertinelli, Marius Borgeaud, Marià Fortuny, Marjorie Frances Bruford, Marjorie May Bacon, Mark A. Brennan, Mark Arian, Mark Boyle, Mark Brooks, Mark English, Mark Gertler, Mark Poole, Mark Rothko, Mark Ryden, Mark Tedin, Mark Tobey, Mark Zug, Marsden Hartley, Marshall Arisman, Marten Post, Martin Benka, Martin Deschambault, Martin Desjardins, Martin Johnson Heade, Martin Kober, Martin Schoeller, Martin Schongauer, Martina Krupičková, Martine Johanna, Martinus Rørbye, Martiros Saryan, Maruja Mallo, Maruyama Ōkyo, Mary Abbott, Mary Adthead, Mary Agnes Yerkes, Mary Anne Ansley, Mary Audsley, Mary Beale, Mary Black, Mary Blair, Mary Byfield, Mary C. Davidson, Mary Callery, Mary Cameron, Mary Cassatt, Mary Corkling, Mary Curtis Richardson, Mary Davis, Mary Dignam, Mary Elizabeth Groom, Mary Elizabeth Price, Mary Grant, Mary Hallock Foote, Mary Jane Begin, Mary McCrossan, Mary McEvoy, Mary Moser, Mary Syme Boyd, Maryam Hashemi, Masamitsu Ōta, Masamune Shirow, Masolino, Master MS, Master of Affligem, Master of Saint Giles, Master of the Bambino Vispo, Master of the Embroidered Foliage, Master of the Legend of Saint Lucy, Matej Sternen, Mathias Kollros, Mathieu Le Nain, Mati Klarwein, Maties Palau Ferré, Matija Jama, Matsui Fuyoko, Matsumura Goshun, Matsuno Chikanobu, Matt Bors, Matt Cavotta, Matt Groening, Matt Stewart, Matteo Pérez, Matthew D. Wilson, Matthew Smith, Matthias Grünewald, Matthias Jung, Matthias Stom, Matthias Weischer, Matthijs Maris, Matthijs Naiveu, Mattias Adolfsson, Mattise, Maud Naftel, Maude Kaufman Eggemeyer, Maurice Boitel, Maurice Braun, Maurice Denis, Maurice Esteve, Maurice Prendergast, Maurice Reymond, Maurice Sendak, Maurice Utrillo, Maurice de Vlaminck, Maurycy Gottlieb, Max Beckmann, Max Buri, Max Dauthendey, Max Dupain, Max Ernst, Max Gubler, Max Klinger, Max Liebermann, Max Magnus Norman, Max Pechstein, Max Slevogt, Max Walter Svanberg, Max Weber, Max Švabinský, Maxfield Parrish, Maximilian Cercha, Maximilien Luce, Maxwell Bates, Maxwell Gordon Lightfoot, May Louise Greville Cooksey, May de Montravel Edwardes, Małgorzata Serwatka, McKendree Long, Mead Schaeffer, Mei Qing, Meindert Hobbema, Melchior Broederlam, Melchior Lorck, Melchior d'Hondecoeter, Melissa A. Benson, Melissa Benson, Melozzo da Forlì, Menez, Meredith Dillman, Meredith Garniss, Mi Fu, Mia Brownell, Miao Fu, Micha Klein, Michael Aloysius Sarisky, Michael Ancher, Michael Andrews, Michael Cheval, Michael Dahl, Michael Deforge, Michael Flohr, Michael Ford, Michael Garmash, Michael Goldberg, Michael Gustavus Payne, Michael James Smith, Michael Komarck, Michael Leunig, Michael Malm, Michael Ray Charles, Michael Sittow, Michael Sutfin, Michael Whelan, Michaelangelo, Michal Karcz, Michalis Oikonomou, Michał Karcz, Michel De Caso, Michel Delacroix, Michel Kikoine, Michel Tapié, Michelangelo, Michelangelo Buonarroti, Michelangelo Buonarroti, Michelangelo Merisi da Caravaggio, Michelangelo Unterberger, Michiel Jansz. van Mierevelt, Michiel van Musscher, Miho Hirano, Mihály Munkácsy, Mihály Zichy, Mikalojus Konstantinas Ciurlionis, Mikalojus Konstantinas Čiurlionis, Mike "Beeples" Winkelmann, Mike Bierck, Mike Deodato, Mike Mignola, Mike Winkelmann, Mikhail Evstafiev, Mikhail Larionov, Mikhail Lebedev, Mikhail Nesterov, Mikhail Vrubel, Mikhail Yuryevich Lermontov, Miklós Barabás, Miklós Borsos, Mikoláš Aleš, Mildred Anne Butler, Mildred Eldridge, Millard Sheets, Milton Avery, Milton Caniff, Milton Glaser, Milton Menasco, Miltos Manetas, Mimmo Rotella, Min Zhen, Minerva J. Chapman, Mirabello Cavalori, Mirko Rački, Miroslava Sviridova, Mitchell Johnson, Miwa Komatsu, Miyagawa Chōshun, Miyagawa Isshō, Miyagawa Shunsui, Miyamoto, Miyazaki, Modest Urgell, Moebius, Mollie Forestier-Walker, Mona Moore, Mondrian, Monet, Mordecai Ardon, Morgan Russell, Mori Sosen, Morris Kestelman, Morris Louis, Morris Louis Bernstein, Mort Künstler, Moses Soyer,

Moses van Uyttenbroeck, Moïse Kisling, Mstislav Dobuzhinsky, Muggur, Muirhead Bone, Munakata Shikō, Munch, Muqi, Murakami, Muriel Brandt, Murray Tinkelman, Mustafa Rakim, Mykola Burachek, Myles Birket Foster, Mym Tuma, Myra Landau, Márta Lacza, Méret Oppenheim, Mór Adler, Mór Than, Mārtiņš Krūmiņš, Möbius, mads berg, mark keathley, maxim verehin, mckadesinsanity, miles johnston, mucha

N:

N. C. Wyeth, N.C. Wyeth, NEVERCREW, Nabil Kanso, Nadim Karam, Nadir Afonso, Nadya Rusheva, Nagasawa Rosetsu, Naka Bokunen, Nakahara Nantenbō, Nam Gye-u, Nan Goldin, Nancy Carline, Nancy Graves, Nancy Spero, Nancy Stahl, Naoko Takeuchi, Naomi Okubo, Naondo Nakamura, Nara Yoshitomo, Narashige Koide, Narayan Shridhar Bendre, Nassos Daphnis, Natalia Goncharova, Natasha Tan, Nathalie Rattner, Nathan Oliveira, Nathan Wyburn, Nathaniel Hone, Nathaniel Pousette-Dart, Naza, Nazmi Ziya Güran, Ndoc Martini, Neal Adams, Ned M. Seidler, Neil Blevins, Neil Boyle, Neil Welliver, Neil Williams, Nele Zirmite, Nell Dorr, Nelson Alexander Ross, Nene Thomas, Nevin Çokay, Neysa McMein, Ni Duan, Ni Tian, Ni Yuanlu, Ni Zan, Niccolò dell' Abbate, Nicholas Hilliard, Nicholas Marsicano, Nicholas Roerich, Nick Fudge, Nick Gentry, Nicola Samori, Nicolaes Eliaszoon Pickenoy, Nicolaes Maes, Nicolaes Pieterszoon Berchem, Nicolas Carone, Nicolas Froment, Nicolas Lancret, Nicolas Poussin, Nicolas Toussaint Charlet, Nicolas de Staël, Nicolette Macnamara, Nicomachus of Thebes, Niels Lergaard, Nikita Veprikov, Niklaus Manuel, Niko Henrichon, Nikola Avramov, Nikolai Alekseyevich Kasatkin, Nikolai Astrup, Nikolai Ge, Nikolai Yaroshenko, Nikolaj Abraham Abildgaard, Nikolay Makovsky, Nikolay Nikanorovich Dubovskoy, Nil Gleyen, Nils Hamm, Nils von Dardel, Nina Hamnett, Nina Petrovna Valetova, Nishida Shun'ei, Nishikawa Sukenobu, Niyazi Selimoglu, Noah Bradley, Noe Canjura, Noel Counihan, Nora Cundell, Norah Neilson Gray, Noriyoshi Ohrai, Norma Bull, Norman Garstin, Norman Hepple, Norman Lewis, Norman Rockwell, Norman Saunders, Normand Baker, Noémi Ferenczy, Nuno Gonçalves, Nyuju Stumpy Brown, Nándor Katona, Nína Tryggvadóttir, Nōami, national geographic, nicoletta ceccoli

O:

O'Keeffe, Odd Nerdrum, Odhise Paskali, Odilon Redon, Ogata Gekkō, Ogata Kenzan, Ogata Kōrin, Ohara Koson, Oka Yasutomo, Okada Beisanjin, Okada Hanko, Okamoto Tarō, Okuda Gensō, Okumura Masanobu, Okumura Togyu, Olaf Gulbransson, Olaf Rude, Oleg Lipchenko, Oleg Oprisco, Olga Boznańska, Olga Rozanova, Olha Darchuk, Olive Mudie-Cooke, Oliver Sin, Olivia Peguero, Olivia de Berardinis, Oluf Høst, Orazio Gentileschi, Orovida Camille Pissarro, Orshi Drozdik, Osamu Tezuka, Oscar Rodríguez Naranjo, Oskar Kokoschka, Oskar Lūthy, Oskar Schlemmer, Osman Hamdi Bey, Ossip Zadkine, Osvaldo Romberg, Oswald Achenbach, Oswald Birley, Oswaldo Guayasamín, Oswaldo Viteri, Oszkar Tordai Schilling, Ota Bubeníček, Otakar Kubín, Otakar Sedloň, Otake Chikuha, Otis Kaye, Otomo Katsuhiro, Oton Gliha, Oton Iveković, Ottilie Maclaren Wallace, Otto Abt, Otto Dix, Otto Eckmann, Otto Frölicher, Otto Lange, Otto Marseus van Schrieck, Otto Meyer-Amden, Otto Morach, Otto Piene, Otto Pilny, Otto Placht, Otto Stark, Ottó Baditz

P:

P.C. Skovgaard, Pablo Carpio, Pablo Munoz Gomez, Pablo Picasso, Pablo Rey, Pacita Abad, Pamela Ascherson, Pamela Coleman Smith, Pamela Drew, Pamphilus, Pan Tianshou, Pan Yuliang, Paolo Parente, Paolo Uccello, Paolo Veronese, Parmigianino, Pascal Blanché, Pascal Dagnan-Bouveret, Pascale Champion, Pat Adams, Pat Oliphant, Patrick Adam, Patrick Brown, Patrick Caulfield, Patrick Ching, Patrick Dougherty, Patrick Hall, Patrick Henry Bruce, Patrick Heron, Patrick Nagel, Patrick Nasmyth, Patrick Pietropoli, Patrick Pye, Patrick Woodroffe, Paul Bird, Paul Bodmer, Paul Bril, Paul Cadmus, Paul Cezanne, Paul Cornoyer, Paul Cézanne, Paul Davis, Paul Delvaux, Paul Dirmeikis, Paul Emmert, Paul Feeley, Paul Gauguin, Paul Georges, Paul Guigou, Paul Gustav Fischer, Paul Gustave Fischer, Paul Harvey, Paul Henry, Paul Howard Manship, Paul Jacob Naftel, Paul Kane, Paul Kelpé, Paul Klee, Paul Lehr, Paul Lohse, Paul Lucien Dessau, Paul Mavrides, Paul Monnier, Paul Nash, Paul Pelletier, Paul Ranson, Paul Resika, Paul Signac, Paul Wunderlich, Paul Émile Chabas, Paul-Albert Besnard, Paul-Émile Borduas, Paula Modersohn-Becker, Paula Rego, Paulus Decker, Paulus Moreelse, Paulus Potter, Pavel Fedotov, Pavel Filonov, Paweł Kluzka, Pearl Frush, Peder Severin Krøyer, Pedro Figari, Pedro Pedraja, Pedro Álvarez Castelló, Peggy Angus, Peggy Bacon, Penelope Beaton, Penleigh Boyd, Penny Patricia Poppycock, Penny Rimbaud, Penny Williams, Per Kirkeby, Per Krohg, Perin del Vaga, Perle Fine, Persis Goodale Thurston Taylor, Pete Morhbacher, Pete Venters, Peter Alexander Hay, Peter Bagge, Peter Basch, Peter Benjamin Graham, Peter Birmann, Peter Blume, Peter Brandes, Peter Brook, Peter Churcher, Peter Doig, Peter Elson, Peter Fiore, Peter Gric, Peter Helck, Peter Lanyon, Peter Lely, Peter Lindbergh, Peter Madsen, Peter Max, Peter Maxwell Ewart, Peter McArdle, Peter Michael, Peter Mohrbacher, Peter Paul Rubens, Peter Prendergast, Peter Rockwell, Peter S. Pezzati, Peter Scott, Peter Snow, Peter Wells, Peter Wtewael, Peter Zumthor, Peter de Seve, Peter de Sève, Petr Brandl, Petros Afshar, Petrus Christus, Petrus Van der Velden, Phil Foglio, Phil Koch, Philip Absolon, Philip Evergood, Philip Guston, Philip Wilson Steer, Philip de Koninck, Philip de László, Philipp Veit, Philippe Druillet, Philips Wouwerman, Phillip Otto Runge, Phillip Peter Price, Phyllis Bone, Phyllis Bray, Phyllis Ginger, Pia Fries, Picasso, Piero della Francesca, Piero di Cosimo, Pierre Adolphe Valette, Pierre Alechinsky, Pierre Auguste Cot, Pierre Bonnard, Pierre Brissaud, Pierre Laffillé, Pierre Mion, Pierre Pellegrini, Pierre Puvis de Chavannes, Pierre Roland Renoir, Pierre Roy, Pierre

Soulages, Pierre Toutain-Dorbec, Pierre-Auguste Renoir, Pierre-Joseph Redouté, Piet Mondrian, Pieter Aertsen, Pieter Anthonisz. van Groenewegen, Pieter Bruegel, Pieter Bruegel the Elder, Pieter Brueghel the Younger, Pieter Claesz, Pieter Codde, Pieter Cornelisz van Slingelandt, Pieter Franciscus Dierckx, Pieter Huys, Pieter Janssens Elinga, Pieter Jansz Quast, Pieter Jansz Saenredam, Pieter Jansz van Asch, Pieter Lastman, Pieter Mulier II, Pieter Mulier the Elder, Pieter de Bloot, Pieter de Grebber, Pieter de Hooch, Pieter de Ring, Pieter van Anraedt, Pieter van Laer, Pieter van der Werff, Pietro Faccini, Pietro Longhi, Pietro Lorenzetti, Pietro Perugino, Pietro Testa, Pietro da Cortona, Pinchus Kremegne, Pinturicchio, Piotr Michałowski, Piranesi, Pisanello, Pixar, Pogus Caesar, Pollock, Pompeo Batoni, Porfirio DiDonna, Primrose Pitman, Prince Hoare, Prudence Heward, Pruet Carter, Pu Hua, Puru, Pál Balkay, Pál Böhm, Pál Szinyei Merse

Q:

Qi Baishi, Qian Du, Qian Gu, Qian Xuan, Qiu Ying, Qu Leilei, Quentin Blake, Quentin Matsys, Quint Buchholz, Quinton Hoover, Quirijn van Brekelenkam, Quirizio di Giovanni da Murano

R:

R. B. Kitaj, R. R. McLan, R.O. Blechman, RHADS, Rachel Reckitt, Rachel Ruysch, Rachel Whiteread, Rackstraw Downes, Radi Nedelchev, Rafael Ritz, Rafail Levitsky, Rafal Olbinski, Raffaeello Ossola, Rainer Maria Latzke, Raja Ravi Varma, Rajesh Soni, Rajmund Kanelba, Ralph Albert Blakelock, Ralph Burke Tyree, Ralph Earl, Ralph Horsley, Ralph McQuarrie, Ram Chandra Shukla, Ramon Casas i Carbó, Ramon Pichot, Ramón Piaguaje, Ramón Silva, Randall Schmit, Randolph Caldecott, Randolph Schwabe, Randy Gallegos, Randy Post, Randy Vargas, Raoul De Keyser, Raoul Dufy, Raoul Ubac, Raphael, Raphael Kirchner, Raphael Lacoste, Raphael Soyer, Raphaelle Peale, Raphaël Collin, Ras Akyem, Ravi Zupa, Ray Caesar, Ray Croke, Ray Howard-Jones, Ray Parker, Raymond Briggs, Raymond Coxon, Raymond Duchamp-Villon, Raymond Han, Raymond Leech, Raymond Norman, Raymond Saunders, Raymond Swanland, Raymond Teague Covern, Raúl Martínez, Rebecca Guay, Rebecca Horn, Reginald Richard Redford, Rei Kamoi, Reinier Nooms, Relja Penezic, Rembrandt, Rembrandt Peale, Rembrandt van Rijn, Remedios Varo, Ren Bonian, Ren Hang, Ren Renfa, Ren Xiong, Ren Xun, Rene Magritte, Renoir, René Auberjonois, René Burri, René Magritte, Reuben Nakian, Reuben Tam, Reynolds Beal, Rezső Bálint, Rhea Carmi, Riad Beyrouti, Ric Estrada, Ric Nagualero, Ricardo Bofill, Richard Anuszkiewicz, Richard Artschwager, Richard Avedon, Richard Benning, Richard Carline, Richard Corben, Richard Dadd, Richard Demarco, Richard Diebenkorn, Richard Doyle, Richard Estes, Richard Gerstl, Richard Gordon Kendall, Richard Gruelle, Richard Hamilton, Richard Hess, Richard Kane Ferguson, Richard Mayhew, Richard Mortensen, Richard Parkes Bonington, Richard Pionk, Richard Pousette-Dart, Richard Randolph Rubenstein, Richard Rockwell, Richard Schmid, Richard Sedlon, Richard T. Scott, Richard Wilson, Richard Wright, Richmond Barthé, Richter, Rick Amor, Rick Griffin, Ridolfo Ghirlandaio, Rihard Jakopič, Rinaldo Cuneo, Rita Angus, Riusuke Fukahori, Riza Abbasi, Roar Kjernstad, Rob Alexander, Rob Gonsalves, Rob Liefeld, Robbie Trevino, Robert Adamson, Robert Alwyn Hughes, Robert Antoine Pinchon, Robert Bain, Robert Ballagh, Robert Bateman, Robert Beatty, Robert Beauchamp, Robert Bechtle, Robert Bery, Robert Bevan, Robert Brackman, Robert Brough, Robert Bryden, Robert C. Barnfield, Robert Campin, Robert Childress, Robert Colescott, Robert Colquhoun, Robert Combas, Robert Crozier, Robert Crumb, Robert Delaunay, Robert Dickerson, Robert Edmonstone, Robert Falk, Robert Fawcett, Robert Feke, Robert Freebairn, Robert Gavin, Robert Goodnough, Robert Griffier, Robert Henderson Blyth, Robert Henri, Robert J. Brawley, Robert Jacobsen, Robert Koehler, Robert Lee Eskridge, Robert Lenkiewicz, Robert M. Cunningham, Robert MacBryde, Robert Macaulay Stevenson, Robert Maguire, Robert Mapplethorpe, Robert McCall, Robert McGinnis, Robert Medley, Robert Motherwell, Robert Noble, Robert Peak, Robert Rauschenberg, Robert Reid, Robert Richenburg, Robert S. Duncanson, Robert Scott Lauder, Robert Sivell, Robert Storm Petersen, Robert Thomas, Robert Walker Macbeth, Robert Weaver, Robert Weir Allan, Robert William Vonnoh, Robert Zünd, Roberto Ferri, Roberto Matta, Roberto Parada, Roberto da Matta, Robin Guthrie, Robt L. Williams, Robt. L. Williams, Rockwell Kent, Rodel Gonzalez, Rodney Joseph Burn, Rodney Matthews, Rodolfo Amoedo, Rodolfo Escalera, Rodolfo Morales, Rodolphe Töpffer, Rodolphe Wytzman, Roelant Savery, Roelof van Vries, Roger Ballen, Roger Cecil, Roger Dean, Roger Swainston, Roger Wilson Dennis, Rogier van der Weyden, Rolf Armstrong, Rolf Iseli, Romaine Brooks, Romaine Brooks, Roman Bezpalkiv, Romano Vio, Romare Bearden, Romero Britto, Ron English, Ron Spears, Ron Spencer, Ron Walotsky, Ronald Davis, Ronald Rae, Ronnie Landfield, Rory McEwen, Rosa Bonheur, Rosalie Emslie, Rose Henriques, Rose Maynard Barton, Rose O'Neill, Rosemary Allan, Ross Bleckner, Ross Tran, Rowena Meeks Abdy, Roy Dalgarno, Roy De Maistre, Roy DeCarava, Roy Lichtenstein, Roy Newell, Roy Petley, Roz Chast, Ruan Jia, Rube Goldberg, Rubens Peale, Rudolf Ernst, Rudolf Freund, Rudolf Hausner, Rudolf Koller, Rudolf Schlichter, Rudolf von Alt, Rudolph Belarski, Rudolph F. Ingerle, Rudy Siswanto, Rufino Tamayo, Rumiko Takahashi, Rupert Bunny, Rupert Shephard, Russell Chatham, Russell Dongjun Lu, Russell Drysdale, Russell Patterson, Ruth Abrahams, Ruth Abrams, Ruth Brandt, Ruth Collet, Ruth Deckard, Ruth Hollingsworth, Ruth Jên, Ruth Orkin, Ruth Sanderson, Ruth Simpson, Ruud Janssen, Ryan Barger, Ryan Pancoast, Ryan Stegman, Ryan Yee, Ryohei Hase, Ryoji Ikeda, Ryuzaburo Umehara, Ryōhei Koiso, Ryūkōsai Jokei, Ryūsei Kishida, Róbert Berény, reyna rochin, ridley scott, roger deakins, rossdraws

S:

S J "Lamorna" Birch, Sadamichi Hirasawa, Sadao Watanabe, Sailor Moon, Saitō Kiyoshi, Sakai Hōitsu, Sally Haley, Salomon Koninck, Salomon de Bray, Salomon van Abbé, Salomon van Ruysdael, Salvador Dali, Salvador Dalí, Sam Black, Sam Bosma, Sam Charles, Sam Dillemans, Sam Francis, Sam Havadtoy, Sam Spratt, Samu Börtsök, Samuel Birmann, Samuel Colman, Samuel Dirks van Hoogstraten, Samuel F. B. Morse, Samuel Hieronymus Grimm, Samuel Peploe, Samuel Prout, Samuel Scott, Samuel Shelley, Samuel Silva, Samuel Washington Weis, Sandra Chevrier, Sandro Botticelli, Saneatsu Mushanokōji, Sanford Robinson Gifford, Santiago Caruso, Santiago Martínez Delgado, Santiago Rusiñol, Sara Saftleven, Sarah Gough Adamson, Sarah Louisa Kilpack, Sarah Lucas, Sarah Morris, Sardar Sobha Singh, Sargent Johnson, Sarper Baran, Sasha Putrya, Satake Yoshiatsu, Satoshi Kon, Saul Steinberg, Saul Tepper, Saul Yaffie, Saurabh Jethani, Sava Šumanović, Sawa Sekkyō, Scarlett Hooft Graafland, Schelte a Bolswert, Scott Gustafson, Scott Listfield, Scott M. Fischer, Scott Naismith, Scott Samuel Summers, Sean Scully, Seb McKinnon, Sebastian Spreng, Sebastian Vranckx, Sebastiano Ricci, Sengai, Serafino De Tivoli, Serge Poliakoff, Serge Sudeikin, Sergei Sviatchenko, Sergio Burzi, Sergio Larrain, Serhii Vasylykivsky, Sesshū Tōyō, Seuss Dr, Shaddy Safadi, Shang Xi, Shani Rhys James, Shao Mi, Sheikh Hamdullah, Sheila McClean, Sheila Mullen, Sheilah Beckett, Shekhar Gurera, Shen Che-Tsai, Shen Quan, Shen Shichong, Shen Zhou, Sheng Mao, Sheng Maoye, Shi Rui, Shi Zhonggui, Shiba Kōkan, Shibata Zeshin, Shigeru Aoki, Shin Saimdang, Shin Yun-bok, Shingei, Shinji Aramaki, Shinoda Toko, Shirley Teed, Shitao, Shog Janit, Shukei Sesson, Shunbaisai Hokuei, Shunkōsai Hokushū, Shōzaburō Watanabe, Shōzō Shimamoto, Shūbun Tenshō, Sidney Nolan, Sidney Richard Percy, Sidney Simon, Siegfried Haas, Sigmar Polke, Sigmund Freudenberg, Sigrid Hjertén, Sigurd Swane, Silvestro Lega, Silvia Dimitrova, Silvia Pelissero, Sim Sa-jeong, Simon Bisley, Simon Gaon, Simon Marmion, Simon Stalenhag, Simon Stålenhag, Simon Ushakov, Simon Vouet, Simon de Vlieger, Simone Martini, Sin Wi, Siona Shimshi, Sir Alfred Munnings, Sir Jacob Epstein, Sir John Tenniel, Sir William Orpen, Sir William Russell Flint, Slava Raškaj, Sławomir Maniak, Slobodan Pejić, Sofonisba Anguissola, Sohrab Sepehri, Solomon Gessner, Soma Orlai Petrich, Song Maojin, Song Xu, Sonia Delaunay, Sonia Delaunay-Terk, Sophia Beale, Sophie Anderson, Sophie Gengembre Anderson, Sophie Pemberton, Sophie Taeuber-Arp, Spencer Gore, Stan Galli, Stan Stokes, Stan and Jan Berenstain, Stanhope Forbes, Stanislas Lépine, Stanislaw Zhukovsky, Stanislaw Samoźrzelnik, Stanisław Ignacy Witkiewicz, Stanisław Masłowski, Stanisław Tondos, Stanisław Witkiewicz, Stanisław Wyspiański, Stanley Bahe, Stanley Matthew Mitruk, Stanley Spencer, Stanley Twardowicz, Stanton Macdonald-Wright, Stefan Gierowski, Stefan Lochner, Stella Schmolle, Stephan Martiniere, Stephan Martinière, Stephen Bone, Stephen Gammell, Stephen Gilbert, Stephen Greene, Stephen Little, Stephen Pace, Stevan Dohanos, Steve Argyle, Steve Brodner, Steve Dillon, Steve Ditko, Steve Hanks, Steve McCurry, Steve Prescott, Steven Belledin, Steven Campbell, Steven James Petruccio, Stokely Webster, Storm Thorgerson, Stuart Davis, Studio Ghibli, Sudip Roy, Sugimura Jihci, Sun Junze, Sun Kehong, Sun Long, Sunil Das, Susan Crile, Susan Heidi, Susan Weil, Susy Pilgrim Waters, Suzanne Duchamp-Crotti, Suzanne Valadon, Suzuki Harunobu, Suzy Rice, Sven Erixson, Sven Nordqvist, Svend Rasmussen Svendsen, Svetlin Velinov, Svetoslav Roerich, Syd Barrett, Syd Mead, Sydney Carline, Sydney Prior Hall, Sylvester Shchedrin, Sylvia Molloy, Sylvia Sleigh, Sylvia Snowden, Sylvia Wishart, Szymon Czechowicz, Sándor Bihari, Sándor Bortnyik, Sándor Brodsky, Sándor Liezen-Mayer, Sō Shiseki, Sōami, Sōtarō Yasui, senior artist, senior character artist, senior environment artist, sparth, stanley artgerm, stephen bliss, sung choi, sylvain sarrailh

T:

T. C. Steele, T. K. Padmini, T. S. Sullivant, Tadanori Yokoo, Tadao Ando, Tadashi Nakayama, Tadashige Ono, Taddeo Gaddi, Tadeusz Ajdukiewicz, Tadeusz Brzozowski, Tadeusz Dominik, Tadeusz Kantor, Tadeusz Makowski, Tadeusz Pruszkowski, Tahir Salahov, Taiyō Matsumoto, Takahashi Yuichi, Takashi Murakami, Takato Yamamoto, Takehisa Yumeji, Takeshi Obata, Takeuchi Seihō, Tamara Lempicka, Tamara de Lempicka, Tamas Galambos, Tan Ting-pho, Tanaka Isson, Tang Di, Tang Sin Yun Sandara, Tang Yifen, Tang Yin, Tani Bunchō, Tanomura Chikuden, Taravat Jalali Farahani, Taro Okamoto, Taro Yamamoto, Tarsila do Amaral, Tatiana Hordienko, Tatsuyuki Tanaka, Tawaraya Sōtatsu, Ted DeGrazia, Ted Nasmith, Telemaco Signorini, Teobaldo Nina Mamani, Teresa Copnall, Teresa Fasolino, Terese Nielsen, Terrell James, Terry Marks, Terry Morris, Terry Oakes, Terry Redlin, Tetsugoro Yorozu, Tex Avery, The Family Circus, The Mazeking, Theo Constanté, Theo van Doesburg, Theodor Philipsen, Theodore Earl Butler, Theodore Major, Theodore Robinson, Theophanes the Greek, Thierry Bisch, Thomas Aquinas Daly, Thomas Baines, Thomas Barker, Thomas Blackshear, Thomas Bock, Thomas Campbell, Thomas Cantrell Dugdale, Thomas Carr, Thomas Cole, Thomas Cornell, Thomas Corsan Morton, Thomas Couture, Thomas Crane, Thomas Dalziel, Thomas Dewing, Thomas Doughty, Thomas Eakins, Thomas Fogarty, Thomas Frederick Worrall, Thomas Furlong, Thomas Gainsborough, Thomas Gambier Parry, Thomas Hart Benton, Thomas Hill, Thomas Häfner, Thomas Kinkade, Thomas Kluge, Thomas Lawrence, Thomas M. Baxa, Thomas Mann Baynes, Thomas Millie Dow, Thomas Moran, Thomas Nast, Thomas Rowlandson, Thomas Scholes, Thomas Stothard, Thomas Struth, Thomas Stuart Burnett, Thomas Symington Halliday, Thomas Tudor, Thomas Wijck, Thomas de Keyser, Thornton Oakley, Thornton Willis, Thota Vaikuntham, Thyrza Anne Leyshon, Théodore Chassériau, Théodore Géricault, Théodore Rousseau, Théodule Ribot, Théophile Steinlen, Tibor Czorba, Tibor Rényi, Tilo Baumgärtel, Tim Biskup, Tim Doyle, Tim Hildebrandt, Tim Okamura, Tim White, Tim and Greg Hildebrandt, Tina Blondell, Tina Modotti, Tintoretto, Titian, Titian

Peale, Titus Lunter, Tivadar Alconiere, Tivadar Csontváry Kosztka, Tobias Stimmer, Todd Lockwood, Tom Bagshaw, Tom Bonson, Tom Carapic, Tom Chambers, Tom Gourdie, Tom La Padula, Tom Lovell, Tom Palin, Tom Phillips, Tom Roberts, Tom Scott RSA, Tom Thomson, Tom Wesselmann, Tom Whalen, Tom Wänerstrand, Tomasz Jedruszek, Tomek Setowski, Tomer Hanuka, Tomi Ungerer, Tomioka Tessai, Tommaso Dolabella, Tommaso Masaccio, Tommaso Redi, Tomokazu Matsuyama, Tomás Barceló, Tony DiTerlizzi, Tony Sart, Tony Szczudlo, Tony Tuckson, Tooth Wu, Torii Kiyomasu, Torii Kiyomasu II, Torii Kiyomitsu, Torii Kiyomoto, Torii Kiyonaga, Torii Kiyonobu I, Toriyama Sekien, Toros Roslin, Tosa Mitsunobu, Tosa Mitsuoki, Toshiko Okanoue, Toss Woollaston, Totte Mannes, Tove Jansson, Toyen, Toyohara Chikanobu, Toyohara Kunichika, Tracey Emin, Tracey Moberly, Tracy Harris, Tran Nguyen, Trevor Brown, Troels Wörsel, Tsubasa Nakai, Tsuchida Bakusen, Tsuchiya Koitsu, Tsuguharu Foujita, Tsuji Kakō, Tsukioka Yoshitoshi, Tsuruko Yamazaki, Tuomas Korpi, Tuvia Beeri, Tyler Edlin, Tyler Jacobson, Tytus Czyżewski, Tōichi Katō, Tōshi Yoshida, Tōshūsai Sharaku, the Brothers Hildebrandt, theCHAMBA, tomasz alen kopera

U:

Ueda Fumito, Uemura Shōen, Ugo Nespolo, Ulrich Leman, Ulrika Pasch, Umberto Boccioni, Un'ichi Hiratsuka, Unkoku Togan, Urakami Gyokudō, Urakusai Nagahide, Ursula Edgcumbe, Ursula Wood, Utagawa Hirokage, Utagawa Hiroshige II, Utagawa Kunimasa, Utagawa Kunimasu, Utagawa Kunisada, Utagawa Kunisada II, Utagawa Kunisada III, Utagawa Kuniyoshi, Utagawa Toyoharu, Utagawa Toyokuni, Utagawa Yoshiiku, Utagawa Yoshitaki, Utagawa Yoshitora, Utagawa Yoshitsuya, Uwe Wittwer

V:

Vadym Meller, Valentin Aleksandrovich Serov, Valentine Hugo, Valerie Petts, Valéria Dénes, Van Gogh, Vanessa Beecroft, Vanessa Bell, Vasile Hutopila, Vasily Andreevich Tropinin, Vasily Perov, Vasily Polenov, Vasily Surikov, Vasily Vereshchagin, Vassily Maximov, Veikko Törmänen, Veno Pilon, Vermeer, Verónica Ruiz de Velasco, Vicente Juan Masip, Victo Ngai, Victor Adame Minguez, Victor Brauner, Victor Enrich, Victor Meirelles, Victor Moscoso, Victor Mosquera, Victor Nizovtsev, Victor Noble Rainbird, Victor Surbek, Victor Vasarely, Victor Wang, Victoria Francés, Victorine Foot, Vija Celmins, Viktor Madarász, Viktor Oliva, Viktor Vasnetsov, Viktor de Jeney, Vilhelm Bissen, Vilhelm Kyhn, Vilhelm Lundstrøm, Villard de Honnecourt, Vilmos Aba-Novák, Vincent Di Fate, Vincent Evans, Vincent Lefevre, Vincent Pepi, Vincent Proce, Vincent Van Gogh, Vincenzo Cabianca, Vincenzo Irolli, Vint Lawrence, Viola Paterson, Violet Fuller, Violet Oakley, Virgil Finlay, Virginia Lee Burton, Vito D'Ancona, Vittore Carpaccio, Vivian Maier, Vivien Blackett, Vladimir Baranov-Rossine, Vladimir Borovikovsky, Vladimir Kush, Vladimir Makovsky, Vladimir Novak, Vladimir Tatlin, Vladimir Tretchikoff, Vladimir Vašiček, Vlady Kibalchich Russakov, Vlaho Bukovac, Volkan Baga, Vytautas Kasulis, Václav Brožík, Victor Manuel García Valdés

W:

W. Lindsay Cable, WLOP, Wadim Kashin, Walasse Ting, Waldo Peirce, Walenty Wańkowitz, Wally Wood, Walt Disney, Walt Reed, Walter Bayes, Walter Beach Humphrey, Walter Crane, Walter Emerson Baum, Walter Haskell Hinton, Walter Humphrey, Walter Leighton Clark, Walter Osborne, Walter Percy Day, Walter Sickert, Walter Stuempfig, Walther Jervolino, Wang Duo, Wang E, Wang Fu, Wang Guxiang, Wang Hui, Wang Jian, Wang Lü, Wang Meng, Wang Mian, Wang Shimin, Wang Shishen, Wang Wei, Wang Wu, Wang Ximeng, Wang Yi, Wang Yuan, Wang Yuanqi, Wang Zhenpeng, Wang Zhongyu, Warhol, Warren Eugene Brandon, Warren Mahy, Warwick Goble, Washington Allston, Wassily Kandinsky, Watanabe Kazan, Watanabe Shōtei, Wayne Barlowe, Wayne England, Wayne Reynolds, Wayne Thiebaud, Weiwei, Wen Boren, Wen Jia, Wen Tong, Wen Zhengming, Wen Zhenheng, Wenceslas Hollar, Wendell Minor, Wendy Froud, Wenzel Lorenz Reiner, Werner Andermatt, Werner Gutzeit, Werner Tübke, Wes Anderson, Wes Wilson, Wesley Burt, Whitney Darrow, Whitney Sherman, Wifredo Lam, Wilfredo Lam, Wilhelm Bendz, Wilhelm Freddie, Wilhelm Hammershøi, Wilhelm Heise, Wilhelm Leibl, Wilhelm Marstrand, Wilhelm Sasnal, Wilhelm Schnarrenberger, Wilhelm Trübner, Wilhelmina Weber Furlong, Will Barnet, Will Eisner, Will Ellis, Willard Metcalf, Willard Mullin, Willem Claeszoon Heda, Willem Cornelisz Duyster, Willem Drost, Willem Hondius, Willem Jacobsz Delff, Willem Kalf, Willem Labeij, Willem Maris, Willem Pieterszoon Buytewech, Willem de Kooning, Willem de Poorter, Willem van Aelst, Willem van Haecht, Willem van Mieris, Willem van de Velde the Elder, Willem van de Velde the Younger, Willem van der Vliet, William Berra, William Birnie Rhind, William Blake, William Blake Richmond, William Bliss Baker, William Bonnar, William Brodie, William Coldstream, William Congdon, William Conor, William Crosbie, William Crozier, William Dargie, William Didier-Pouget, William Dobell, William Dobson, William Dring, William Edouard Scott, William Edward West, William Etty, William Fettes Douglas, William Forsyth, William Gear, William Geissler, William George Gillies, William Glackens, William Gow Ferguson, William Grant Stevenson, William Gropper, William H. Mosby, William Harnett, William Hoare, William Hogarth, William Holman Hunt, William Holmes Sullivan, William Home Lizars, William Jacob Baer, William Jennys, William John Thomson, William Kentridge, William Langson Lathrop, William MacTaggart, William McGregor Paxton, William McTaggart, William Merritt Chase, William Michael Harnett, William Miller, William Morris, William Mossman, William Mustart Lockhart, William Nicholson, William Powhida, William Quiller Orchardson, William Simpson, William Steig, William Stott,

William Stout, William Thon, William Trost Richards, William Turner, William Turner of Oxford, William Twigg-Smith, William Woodward, William Yellowlees, William York Macgregor, William Zorach, William-Adolphe Bouguereau, Willian Murai, Willie Ito, Willy Bo Richardson, Willy Finch, Wilson Irvine, Winona Nelson, Winslow Homer, Winsor McCay, Winston Churchill, Witold Pruszkowski, Witold Wojtkiewicz, Wladyslaw Strzeminski, Wlastimil Hofman, Wlodzimierz Tetmajer, Wojciech Gerson, Wojciech Korneli Stattler, Wojciech Kossak, Wojciech Siudmak, Wojciech Weiss, Wolf Huber, Wolf Kahn, Wolf Vostell, Wolfgang Letti, Wolfgang Zelmer, Wouter Pietersz Crabeth, Wu Bin, Wu Changshuo, Wu Daozi, Wu Guanzhong, Wu Hong, Wu Li, Wu Shixian, Wu Wei, Wu Zhen, Wu Zuoren, Wuzhun Shifan, Wyke Bayliss, Wylie Beckert, Wyndham Lewis, Władysław Czachórski, Władysław Malecki, Władysław Podkowiński, Władysław Ślewiński, wolfgang letti

X:

Xanthus Russell Smith, Xavier Blum Pinto, Xi Gang, Xia Chang, Xia Gui, Xia Shuwen, Xia Yong, Xiang Shengmo, Xiao Yuncong, Xie He, Xie Huan, Xie Shichen, Xie Sun, Xu Beihong, Xu Wei, Xu Xi, Xuande Emperor, Xul Solar

Y:

Yaacov Agam, Yahoo Kusama, Yamagata Hiro, Yamaguchi Kayo, Yamamoto Shōun, Yan Hui, Yan Liben, Yanagawa Nobusada, Yanagawa Shigenobu, Yang Borun, Yang Buzhi, Yang J, Yang Jin, Yang Weizhen, Yanjun Cheng, Yao Tingmei, Yasar Vurdem, Yasuo Kuniyoshi, Yasushi Sugiyama, Yasutomo Oka, Yayoi Kusama, Yayou Kusama, Ye Xin, Yehia Dessouki, Yeong-Hao Han, Yerkaland, Yi Inmun, Yi Insang, Yi Jaegwan, Yi Yuanji, Yoann Lossel, Yokoyama Taikan, Yosa Buson, Yoshida Hanbei, Yoshihiko Wada, Yoshio Markino, Yoshitaka Amano, Yoshitoshi ABe, Yoshitoshi Mori, Yoshiyuki Tomino, Youri Messen-Jaschin, Youssef Howayek, Yousuf Karsh, Yu Zhiding, Yuan Jiang, Yuan Yao, Yue Minjun, Yuki Ogura, Yukihiko Yasuda, Yukimasa Ida, Yuko Shimizu, Yuko Tatsushima, Yumihiko Amano, Yun Du-seo, Yun Shouping, Yun-Fei Ji, Yuri Ivanovich Pimenov, Yuumei, Yves Klein, Yves Tanguay, Yves Tanguy, Yvonne Jacquette, yoji shinkawa

Z:

Zacharias Martin Aagaard, Zack Snyder, Zack Stella, Zaha Hadid, Zahari Zograf, Zao Wou-Ki, Zbigniew Nowosadzki, Zdeněk Burian, Zdzislaw Beksinski, Zdzislaw Beksinski, Zeen Chin, Zelma Blakely, Zeng Jing, Zha Shibiao, Zhan Ziqian, Zhang Han, Zhang Kechun, Zhang Lu, Zhang Sengyao, Zhang Shengwen, Zhang Shunzi, Zhang Shuqi, Zhang Wo, Zhang Xiaogang, Zhang Xuan, Zhang Yan, Zhang Yin, Zhang Zeduan, Zhang Zongcang, Zhao Mengfu, Zhao Yong, Zhao Yuan, Zhao Zhiqian, Zhao Zuo, Zheng Xie, Zhichao Cai, Zhou Chen, Zhou Fang, Zhou Jichang, Zhou Shuxi, Zhou Wenjing, Zhu Da, Zhu Derun, Zinaida Serebriakova, Zlatyu Boyadzhiev, Zofia Stryjenska, Zoltan Boros, Zoltán Joó, Zoltán Nuridsány, Zoran Mušič, Zou Yigui, Zou Zhe, Zoë Mozart, Zsolt Bodoni, Zsuzsa Máthé, Zvest Apollonio, Zygmunt Vogel, Zygmunt Waliszewski, zdzislaw beksinski

Á:

Ángel Botello, Ásgrímur Jónsson

É:

Édouard Debat-Ponsan, Édouard Detaille, Édouard Manet, Édouard Vuillard, Élisabeth Vigée Le Brun, Émile Bernard, Étienne Aubry, Évariste Vital Luminais

Ó:

Oscar Domínguez

Ö:

Ödön Márffy

Þ:

Þórarinn B. Þorláksson

Ș:

Ștefan Luchian

3.42.14.6.9 Get Medium

[SDL.Get Medium / SDL.Gme](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL.Get Medium / SDL.Gme

Retrieves art mediums based on a given index number, which ranges from 0 to 95.

Intention

The `SDL.Get Medium` and its shorthand `SDL.Gme` are designed to retrieve art mediums.

These commands return a specific art medium based on an index number provided by the user.

The result of this command can be appended or prepended to the prompt for Stable Diffusion (SD) processing to generate images that align with the retrieved art medium.

The index must be between 0 and 95; otherwise, the command will return "a photo" (index 29).

Alternative to the numeric index you can also write something for example "pointilism" and the system will try to get the best match for you.

The commands are part of the Stable Diffusion Local (SDL) suite, which works locally on the user's system.

Syntax

SDL.Get Medium [| P1] [| P2]

Parameter Explanation

- **P1:** (Optional) An index number between 0 and 95. If omitted, the default value of 39 is used, and the result will be "a photo".
- **P2:** (Optional) A variable that will be used to store the art medium name retrieved. If omitted, the result is placed on TOS (Top Of Stack).

Example

```

'*****
' SDL.-Sample for retrieving art mediums
'*****
$$IDX = 7

```

```

SDL.Get Medium|$$IDX|$$MED
' Result will be "a child's drawing"
DBP. The art medium for index $$IDX is: $$MED
ENR.

```

Part 1: Digital and Modern Art Forms

Number	Style Name	Description
0	A 3D Render	A three-dimensional computer graphic.
1	A Photorealistic Painting	A painting that closely resembles a high-resolution photograph.
2	An Anime Drawing	A style of drawing associated with Japanese animation.
3	A Digital Rendering	A digital representation of an object or scene.
4	A Comic Book Panel	A single frame or scene as part of a larger comic book.
5	Digital Art	Artwork created using digital technology.
6	A Detailed Painting	A painting with a high level of detail.
7	A Child's Drawing	A drawing that resembles the work of a young child.
8	An Ambient Occlusion Render	A shading and rendering technique.
9	Lineart	A drawing that consists of distinct straight or curved lines.
10	A Low Poly Render	A 3D model with a small number of polygons.
11	A Statue	A three-dimensional representation of a person or object.

12	Cyberpunk Art	Art inspired by cyberpunk themes, often futuristic and dystopian.
13	An Illustration of Origami	A graphical representation of the Japanese art of paper folding.
14	A Black and White Photo	A photograph without colors, only shades of grey.
15	Pixel Art	A digital art form where images are created on the pixel level.
16	An Abstract Drawing	A drawing that does not represent physical objects.
17	A Diagram	A simplified drawing showing the appearance or workings of something.
18	A Digital Painting	A type of digital art but created traditionally by hand using software.
19	A Cartoon	A drawing representing a humorous or satirical situation.

Part 2: Traditional Art Forms and Techniques

Number	Style Name	Description
20	A Drawing	A picture or diagram made with a pencil, pen, or crayon rather than paint.
21	A Fine Art Painting	A visual art considered to have been created primarily for aesthetic purposes.
22	A Flemish Baroque	Style of Baroque art developed in the

		Southern Netherlands.
23	A Gouache	A type of watercolor painting that is more opaque.

Part 3: Varied Art Forms and Techniques (Continued)

Number	Style Name	Description
24	A Hologram	A three-dimensional image formed by the interference of light beams.
25	A Hyperrealistic Painting	A painting that is extremely realistic and detailed.
26	A Jigsaw Puzzle	An image broken into pieces to be reassembled.
27	A Cave Painting	Prehistoric wall art commonly found in caves.
28	A Macro Photograph	A close-up photograph, typically of very small subjects.
29	A Manga Drawing	A style of drawing associated with Japanese comic books.
30	A Marble Sculpture	A sculpture made from marble.
31	A Matte Painting	A painted representation that is used as a backdrop in movies.
32	A Microscopic Photo	A photograph taken through a microscope.
33	A Mid-nineteenth Century Engraving	A detailed print made in the mid-19th century.
34	A Minimalist Painting	A painting that uses minimal elements for maximum effect.

35	A Mosaic	Art made by assembling small pieces of colored glass, stone, or other materials.
36	A Painting	A representation or image made using pigments on a surface.
37	A Pastel	An art medium in the form of a stick, made of pigment and a binder.
38	A Pencil Sketch	A preliminary drawing made using a pencil.
39	A Photo	A photograph.
40	A Photocopy	A reproduction of a document or print.
41	A Detailed Drawing	A drawing with a high level of detail.
42	A Picture	A representation of something in a visual form.
43	A Pointillism Painting	A painting technique using small, distinct dots of color.
44	A Polaroid Photo	A type of instant photograph.
45	A Pop Art Painting	Art inspired by popular culture or consumerism.
46	A Portrait	A painting, drawing, or photograph of a person.
47	A Poster	A large printed picture used for decoration.

Part 4: Varied Art Forms and Techniques (Continued)

Number	Style Name	Description
48	A Raytraced Image	An image created with computational ray tracing techniques for realistic lighting.

49	A Renaissance Painting	A painting from the Renaissance period, known for its attention to detail and realism.
50	A Screenprint	A print made by forcing ink through a prepared screen of fine material.
51	A Screenshot	An image taken to record the visible items displayed on a computer monitor.
52	A Silk Screen	A printing technique where a mesh is used to transfer ink onto a substrate.
53	A Sketch	A quick, informal drawing.
54	A Computer Rendering	An image created from a computer model.
55	A Still Life	A work of art depicting inanimate subject matter.
56	A Stipple	An art style that uses numerous small dots or specks to create shading and texture.
57	A Stock Photo	A photo made available for public use, often used in advertising or media.
58	A Storybook Illustration	An illustration typically found in a children's book.
59	A Surrealist Painting	A painting in the surrealist style, often featuring dream-like scenes.
60	A Surrealist Sculpture	A sculpture in the surrealist style, often featuring dream-like forms.

61	A Tattoo	A form of body art where ink is inserted into the dermis layer of the skin.
62	A Tilt Shift Photo	A photo manipulated to give a miniature scale appearance.
63	A Watercolor Painting	A painting technique using water-soluble pigments.
64	A Wireframe Diagram	A visual representation of a three-dimensional object.
65	A Woodcut	A type of printmaking from a wooden block.
66	A Charcoal Drawing	A drawing made with charcoal sticks.
67	An Abstract Painting	A painting that does not depict a person, place, or thing in the natural world.
68	An Abstract Sculpture	A sculpture that does not depict a person, place, or thing in the natural world.
69	An Acrylic Painting	A fast-drying paint containing pigment suspension in acrylic polymer emulsion.
70	An Airbrush Painting	A painting created with an airbrush.

Part 5: Varied Art Forms and Techniques (Continued)

Number	Style Name	Description
71	An Album Cover	The front of the packaging of a commercially released audio recording.

72	A Color Pencil Sketch	A sketch made using colored pencils.
73	A Bronze Sculpture	A sculpture made from bronze.
74	An Art Deco Painting	A painting in the Art Deco style, characterized by rich colors and bold geometry.
75	An Art Deco Sculpture	A sculpture in the Art Deco style, characterized by rich colors and bold geometry.
76	An Engraving	The practice of incising a design onto a hard, flat surface.
77	An Etching	A print produced by the process of etching.
78	A Cubist Painting	A painting in the cubist style, emphasizing the two-dimensionality of the canvas.
79	An Impressionist Painting	A painting in the impressionist style, characterized by small, thin, visible brushstrokes.
80	An Ink Drawing	A drawing made using ink.
81	An Oil on Canvas Painting	A painting made using oil paints on canvas.
82	An Oil Painting	A painting made using oil-based pigments.
83	An Ultrafine Detailed Painting	A painting with an extremely high level of detail.
84	Chalk Art	Artwork made using chalk.
85	Computer Graphics	Artwork created using computer software.

86	Concept Art	Art designed to convey an idea or concept.
87	A Cross Stitch	A form of sewing and a popular form of counted-thread embroidery.
88	A Character Portrait	A portrait focused on depicting a specific character.
89	Egyptian Art	Artwork in the style of ancient Egyptian art.
90	Graffiti Art	Artwork made on walls or other public or private surfaces.
91	A Colorized Photo	A black and white photo that has been converted to include color.
92	A Detailed Matte Painting	A highly detailed painting used to create the illusion of a landscape or set.
93	Poster Art	Art specifically designed to be made into posters.
94	Vector Art	Artwork created in vector graphics programs.
95	An Illustration of	A detailed drawing or painting that provides a visual representation of something.

3.42.14.6.10 Get Movement

[SDL.Get Movement / SDL.Gmv](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL.Get Movement / SDL.Gmv

Retrieves art movements based on a given index number, which ranges from 0 to 200.

Intention

The `SDL.Get Movement` and its shorthand `SDL.Gmv` are designed to retrieve **art movements**.

Is also a sort of Painting styles.

These commands return a specific art movement based on an index number provided by the user.

You can append or prefix these to the positive prompt, to have the result going into that direction.

The commands are part of the Stable Diffusion Local (SDL) suite, which works locally on the user's system.

Syntax

```
SDL.Get Movement [ | P1 ] [ | P2 ]  
SDL.gmv [ | P1 ] [ | P2 ]
```

Parameter Explanation

- **P1: (Optional)** An index number between 0 and 200. If omitted, the default value of 0 is used, and the result will be empty ("").

You can also specify a Text like "modern art" etc. then the command will try to find the best match Movement for you.

Below you will find tables with the implemented Art-Movements.

- **P2: (Optional)** A variable that will be used to store the art movement name retrieved. If omitted, the result is placed on TOS (Top Of Stack).

Example


```

!*****
! SDL.-Sample for retrieving art movements
!*****
$$IDX = 25
SDL.Get Movement|$$IDX|$$ART
DBP. The art movement for index $$IDX is: $$ART
ENR.
    
```

Part 1: Abstract and Modern Movements

Number	Art Movement	Description
0		No specific art movement.
1	Abstract Art	Art that does not attempt to represent an accurate depiction of a visual reality.
2	Abstract Expressionism	Post-World War II art movement focused on expressive, abstract forms.
3	Abstract Illusionism	A painting style that creates the illusion of three-dimensional space.
4	Academic Art	Art influenced by the standards of art academies.
5	Action Painting	A style of painting in which paint is spontaneously dribbled or splashed onto the canvas.
6	Aestheticism	Late 19th-century movement that believed in "art for art's sake."
7	Afrofuturism	Cultural, aesthetic genre that involves a mix of science fiction and African culture.
8	Altermodern	A term coined by Nicolas Bourriaud, relating to how artists

		are responding to our globalized world.
9	American Barbizon School	Inspired by the French Barbizon, focused on natural landscapes.
10	American Impressionism	American version of the 19th-century art movement focusing on light and color.
11	American Realism	Art style focused on depicting American life in a naturalistic manner.
12	American Romanticism	19th-century American art and literature movement focusing on emotion and nature.
13	American Scene Painting	Depicts scenes of rural or small-town America primarily in the first half of the 20th century.
14	Analytical Art	Art that focuses on different forms and spatial relationship.
15	Antipodeans	A group of Australian artists who protested against abstract art.
16	Arabesque	Islamic art decoration consisting of surface decorations based on rhythmic linear patterns.
17	Arbeitsrat für Kunst	A council for art established in Berlin during the Weimar Republic.
18	Art & Language	A conceptual artists' collaboration that questioned the critical assumptions of mainstream modern art practice.
19	Art Brut	Raw or rough art, often outsider art or naïve art.

Part 2: Art Deco to Brutalism

Number	Art Movement	Description
20	Art Deco	A style of visual arts, architecture, and design that first appeared in France before World War I.
21	Art Informel	A style similar to Abstract Expressionism that emerged in France during the 1940s.
22	Art Nouveau	An international style of art and architecture that peaked in popularity at the turn of the 20th century.
23	Art Photography	Photography considered as an art form.
24	Arte Povera	An Italian contemporary art movement that emerged in the late 1960s.
25	Arts and Crafts Movement	An international movement in the decorative and fine arts that originated in Britain and flourished in Europe and North America.
26	ASCII Art	Art made up of computer characters.
27	Ashcan School	An art movement in the United States that depicted scenes of daily life in New York's poorer neighborhoods.
28	Assemblage	An artistic process in which a three-dimensional artwork is

		made from putting together found objects.
29	Australian Tonalism	An Australian art movement that emerged in the late 20th century, emphasizing tone over color.
30	Auto-Destructive Art	A term coined by Gustav Metzger, referring to artworks which are destroyed in the process of their creation.
31	Barbizon School	A 19th-century French art movement that had a strong influence on the introduction of Realism in art.
32	Baroque	An art style that used exaggerated motion and detail to produce drama, tension, and grandeur.
33	Bauhaus	A German art school operational from 1919 to 1933 that combined crafts and the fine arts.
34	Bengal School of Art	An art movement and a style of Indian painting that originated in Bengal, primarily Kolkata and Shantiniketan.
35	Berlin Secession	An art association founded by Berlin artists in 1898 as an alternative to the conservative state-run Association of Berlin Artists.
36	Black Arts Movement	A cultural or artistic movement that took place in the USA and other nations.

37	Brutalism	An architectural style which emerged in the mid-20th century and gained popularity in the late 1950s and 1960s.
----	-----------	---

Part 3: Classical Realism to Cubo-Futurism

Number	Art Movement	Description
38	Classical Realism	A 20th-century art movement that places a high value upon skill and beauty.
39	Cloisonnism	A style of post-Impressionist painting with bold and flat forms separated by dark contours.
40	Cobra	A European avant-garde movement active from 1948 to 1951.
41	Color Field	A style of abstract painting that emerged in New York City during the 1940s and 1950s.
42	Computer Art	Art made with the assistance of computers.
43	Conceptual Art	Art for which the idea (or concept) behind the work is more important than the finished product.
44	Concrete Art	An art movement with a strong emphasis on abstraction.
45	Constructivism	An artistic and architectural philosophy that originated in Russia beginning in 1913.

46	Context Art	Art that is created based on the surrounding environment and culture.
47	Crayon Art	Art created using crayons.
48	Crystal Cubism	A distilled form of Cubism consistent with a shift, between 1915 and 1916.
49	Cubism	An early 20th-century art movement that brought European painting and sculpture into modernity.
50	Cubo-Futurism	A Russian art movement which combined aspects of Cubism and Futurism.

Part 4: Cynical Realism to Digital Art

Number	Art Movement	Description
51	Cynical Realism	A contemporary movement in Chinese art, particularly in painting, that began in the 1990s.
52	Dada	An avant-garde movement that began in Europe during World War I.
53	Danube School	A circle of painters of the first third of the 16th century in Bavaria and Austria.
54	Dau-al-Set	A Catalan artistic group founded in Barcelona in 1948.

55	De Stijl	A Dutch art movement focused on geometric abstraction.
56	Deconstructivism	An architectural movement that started in the late 20th century.
57	Digital Art	An artistic work or practice that uses digital technology as part of the creative process.

Part 5: Ecological Art to Feminist Art

Number	Art Movement	Description
58	Ecological Art	Art that is centered on ecological awareness and is created to improve the health of ecosystems.
59	Environmental Art	Art that helps improve our relationship with the natural world.
60	Excessivism	A modern art movement that emphasizes the excessive use of certain elements.
61	Expressionism	An art movement in which the representation of reality is not the objective but to express the artist's feelings.
62	Fantastic Realism	A form of magic realism in visual arts.
63	Fantasy Art	A genre of art that depicts magical or other supernatural themes, ideas, creatures, or settings.

64	Fauvism	An early 20th-century art movement that emphasized painterly qualities and strong color.
65	Feminist Art	Art that seeks to challenge the dominance of male artists and question the role of women in society.

Part 6: Figuration Libre to Geometric Abstract Art

Number	Art Movement	Description
66	Figuration Libre	A French art movement that broke away from abstraction to embrace a free, expressive style.
67	Figurative Art	Art that is clearly derived from real object sources, representing them through forms and figures.
68	Figurativism	Art that retains strong references to the real world and particularly to the human figure.
69	Fine Art	A visual art considered to have been created primarily for aesthetic and intellectual purposes.
70	Fluxus	An art movement that emphasized the artistic process over the finished product.
71	Folk Art	Art produced from an indigenous culture or by

		peasants or other laboring tradespeople.
72	Funk Art	An American art movement that was a reaction against the non-objectivity of abstract expressionism.
73	Furry Art	Art related to the Furry fandom, depicting fictional anthropomorphic animal characters with human personalities and traits.
74	Futurism	An early 20th-century art movement that captured the dynamism and energy of the modern world.
75	Generative Art	Art that has been generated, composed, or constructed in an algorithmic manner.
76	Geometric Abstract Art	A form of abstract art based on geometric forms and structures.

Part 7: German Romanticism to International

Typographic Style

Number	Art Movement	Description
77	German Romanticism	German variant of Romanticism, emphasizing national history, folklore, and cultural identity.

78	Gothic Art	Art that originated from the Medieval period, characterized by intricate details and religious themes.
79	Graffiti	Art created on walls or other public surfaces, often in an unauthorized or unsanctioned manner.
80	Gutai Group	A post-war Japanese art movement that focused on the relationship between body and matter.
81	Happening	A performance, event, or situation meant to be considered art, usually as performance art.
82	Harlem Renaissance	A cultural, social, and artistic explosion centered in Harlem, New York, spanning the 1920s.
83	Heidelberg School	An Australian art movement that achieved international recognition in the late 19th and early 20th centuries.
84	Holography	A technique to produce 3D photographic imagery.
85	Hudson River School	A 19th-century American art movement embodied by landscape painters influenced by Romanticism.
86	Hurufiyya	An art movement that sought to integrate Islamic calligraphy into modern visual art.
87	Hypermodernism	A cultural, artistic, and architectural movement

		that extends and intensifies modernism.
88	Hyperrealism	An art style that mimics high-resolution photography, creating illusionistic paintings.
89	Impressionism	A 19th-century art movement characterized by small, thin, visible brush strokes and focus on light and color.
90	Incoherents	A French art movement in the late 19th century, noted for absurd or humorous works.
91	Institutional Critique	An art practice that reflects critically on its own housing in galleries and museums.
92	Interactive Art	Art that involves the spectator in some way.
93	International Gothic	A period of Gothic art that began in Burgundy, France, and Northern Italy in the late 14th and early 15th century.
94	International Typographic Style	A graphic design style developed in Switzerland in the 1950s, emphasizing cleanliness, readability, and objectivity.

Part 8: Kinetic Art to Vorticism

Number	Art Movement	Description
95	Kinetic Art	Art that contains movement perceivable by the viewer or

		depends on motion for its effect.
96	Kinetic Pointillism	A technique combining pointillism and kinetic art, often involving optical illusions.
97	Kitsch Movement	Art considered to be in poor taste because of excessive garishness or sentimentality.
98	Land Art	Art that is made directly in the natural environment, often involving earthworks.
99	Les Automatistes	A Canadian art movement emphasizing spontaneous, subconscious creation.
100	Les Nabis	A group of Post-Impressionist avant-garde artists in France.
101	Letterism	A French avant-garde movement, established in Paris in the mid-1940s.
102	Light and Space	An art movement founded in the 1960s focusing on the viewer's perception of light and space.
103	Lowbrow	A populist art movement with origins in the underground comix world, punk music, and hot-rod street culture.
104	Lyco Art	A lesser-known art movement focusing on the use of light and color.
105	Lyrical Abstraction	A type of freewheeling abstract painting

		inspired by the emotional resonance of color and shape.
106	Magic Realism	A genre where magical or unreal elements play a natural part in an otherwise realistic environment.
107	Magical Realism	A literary and artistic genre in which realistic narrative is combined with surreal elements.
108	Mail Art	Art that uses the postal system as a medium.
109	Mannerism	A style in European art that emerged in the later years of the Italian High Renaissance.
110	Massurrealism	A portmanteau of Mass Media and Surrealism, often involving the use of technology in the creation of art.
111	Maximalism	An aesthetic of excess, contrasting with minimalism.
112	Metaphysical Painting	A style of painting that flourished in the 1910s, characterized by eerie, archaic scenes.
113	Mingei	A Japanese folk art movement founded in the late 1920s and 1930s.
114	Minimalism	An art movement characterized by extreme simplicity of form and a literal approach.
115	Modern European Ink Painting	Modern interpretations of traditional East Asian ink painting techniques.

116	Modernism	A broad movement that developed in the late 19th and early 20th centuries, characterized by a break with traditional styles.
117	Modular Constructivism	An architectural movement that utilizes crystalline or agglomerated forms in a non-repetitive sequence.
118	Naive Art	Art created by untrained artists, often characterized by simplicity and a lack of sophistication.
119	Naturalism	A style and theory of representation based on the accurate depiction of detail.
120	Neo-Dada	A movement that combines aspects of Dadaism and abstract expressionism, and includes found art.
121	Neo-Expressionism	A style of modern painting and sculpture that emerged in the late 1970s and dominated the art market until the mid-1980s.
122	Neo-Fauvism	An art movement that revives the bright colors and wild brushwork of Fauvism.
123	Neo-Figurative	Art that includes recognizable objects, particularly the human figure, in a more modern context.
124	Neo-Primitivism	A Russian art movement which took its name and

		attributes from the non-academic art of the early 20th century.
125	Neo-Romanticism	A subjectively emotional approach originating in the arts during the early 20th century.
126	Neoclassicism	A Western movement in the fine arts that drew inspiration from the classical art and culture of ancient Greece or ancient Rome.
127	NeoGeo	An art movement of the 1980

Part 9: Neogeo to Vorticism

Number	Art Movement	Description
127	NeoGeo	An art movement of the 1980s and 1990s that is an abbreviation for Neo-Geometric conceptualism.
128	Neoism	A contemporary art movement with roots in the 1980s, known for its subversion of social norms.
129	Neoplasticism	A style founded by Piet Mondrian, characterized by the use of horizontal and vertical lines and primary colors.
130	Net Art	Art made on the internet or that uses the internet as its medium.
131	New Objectivity	A movement in Germany in the 1920s that aimed to depict objective

		reality as opposed to romantic idealization.
132	New Sculpture	A movement in late 19th-century British sculpture marked by a departure from traditional formats and styles.
133	Northwest School	A regional art movement centered in the Seattle area, involving both visual and performing arts.
134	Nuclear Art	Art inspired by or commenting on nuclear energy and its implications.
135	Objective Abstraction	Abstract art that aims to portray the objective world, often through geometric forms.
136	Op Art	A style of visual art that makes use of optical illusions.
137	Optical Illusion	Art designed to visually deceive the viewer into perceiving something other than what is actually represented.
138	Orphism	A French art movement that sought to combine the musical qualities of color with Cubist abstraction.
139	Panfuturism	A futurist art movement that seeks to combine various styles and mediums.
140	Paris School	A group of artists who lived and worked in Paris, particularly during the first half of the 20th century.

141	Photorealism	A genre of art in which paintings and drawings are created to resemble high-resolution photographs.
142	Pixel Art	A form of digital art where images are edited at the pixel level.
143	Plasticien	A Canadian art movement in the 1950s focused on the aesthetic qualities of geometric shapes.
144	Plein Air	Painting done outdoors, capturing landscapes and natural light.
145	Pointillism	A painting technique using tiny dots of color to form an image.
146	Pop Art	Art based on modern popular culture and the mass media.
147	Pop Surrealism	A movement that combines the whimsy and fantasy of Surrealism with the techniques and subjects of Pop Art.
148	Post-Impressionism	A French art movement that followed Impressionism, embracing various styles and techniques.
149	Postminimalism	A term used to describe changes in Minimalist art in the late 1960s and 1970s, moving towards complexity.
150	Pre-Raphaelitism	A British art movement that sought to return to the style and techniques of art before the High Renaissance.

151	Precisionism	A style characterized by the depiction of industrial and architectural subjects with precision and clarity.
152	Primitivism	A Western art movement that borrows visual forms from non-Western or prehistoric peoples.
153	Private Press	A movement to produce books by hand, often with a focus on high-quality materials and design.
154	Process Art	Art that emphasizes the process of its making, rather than the finished product.
155	Psychedelic Art	Art inspired by the experience or aesthetics of hallucination and psychedelic experiences.
156	Purism	An art movement that aimed to simplify geometric forms and remove extraneous detail.
157	Qajar Art	A style of Persian art from the Qajar dynasty, characterized by elaborate ornamentation and portraiture.
158	Quito School	A Latin American school of art that is a blend of Indigenous and European styles.
159	Rasquache	A Chicano art movement that makes the most from the least, often using found materials.

160	Rayonism	A Russian abstract art movement that sought to depict the rays of light as dynamic forms.
161	Realism	An art movement that seeks to depict objects or scenes as they appear objectively.
162	Regionalism	An American art movement that emerged in the Midwest in the early 20th century, focusing on rural scenes.
163	Remodernism	A movement that aims to reintroduce traditional techniques and spirituality into modern art.

Part 10: Renaissance to Vorticism

Number	Art Movement	Description
164	Renaissance	A cultural, artistic, and intellectual movement that originated in Italy, emphasizing classical forms.
165	Retrofuturism	Artistic and creative disciplines that explore the themes of what the future may hold, based on the past.
166	Rococo	An 18th-century artistic movement and style, affecting many aspects of the arts including painting.

167	Romanesque	A style of art and architecture that was common in Europe from the 10th to the 12th century.
168	Romanticism	A movement emphasizing emotion and individualism as well as glorification of the past and nature.
169	Samikshavad	An Indian art movement that focuses on social issues.
170	Serial Art	Art that adheres to a strict order or sequence, often realized in multiple parts.
171	Shin Hanga	A 20th-century art movement in Japan that revitalized traditional ukiyo-e art.
172	Shock Art	Art that incorporates shocking or provocative elements.
173	Socialist Realism	A style of idealized realistic art, developed in the Soviet Union and used to glorify Communist values.
174	Sots Art	A Russian art movement that arose in the 1970s, a fusion between Socialist Realism and Pop Art.
175	Space Art	Artistic representations of astronomical phenomena or space exploration.
176	Street Art	Visual art created in public spaces, often unsanctioned.
177	Stuckism	An international art movement that

		promotes figurative painting in opposition to conceptual art.
178	Sumatraism	A lesser-known movement focusing on the art and culture of the Sumatra region.
179	Superflat	A postmodern art movement founded by Takashi Murakami, influenced by manga and anime.
180	Suprematism	A Russian art movement focused on geometric shapes like squares, circles, and lines.
181	Surrealism	A movement that seeks to express the workings of the subconscious through art.
182	Symbolism	A movement that seeks to depict the more emotional and subjective aspects of human experience.
183	Synchromism	An art movement founded by Stanton Macdonald-Wright and Morgan Russell in Paris in 1913.
184	Synthetism	A term used by post-Impressionist artists to describe their method of creating and defining form.
185	Sōsaku Hanga	A 20th-century Japanese art movement focusing on creative prints.
186	Tachisme	A French style of abstract painting in the 1940s and 1950s, similar to Abstract Expressionism.

187	Temporary Art	Art that is made to exist only for a short period of time.
188	Tonalism	An art style that emerged in the 1880s in the US, characterized by soft, diffused light and muted colors.
189	Toyism	A contemporary art movement that originated in the 1990s in the Netherlands.
190	Transgressive Art	Art that aims to transgress or violate basic mores and sensibilities of the status quo.
191	Ukiyo-e	A genre of Japanese art which flourished from the 17th through 19th centuries, often depicting landscapes.
192	Underground Comix	Small press or self-published comic books that are often socially relevant or satirical in nature.
193	Unilalianism	A less-known contemporary art movement focused on universal language through visual art.
194	Vancouver School	A term applied to a group of conceptual photographers based in Vancouver, Canada.
195	Vanitas	A genre of still-life painting that has been common in the Netherlands in the early 17th century.
196	Verdadism	A form of art and philosophy emphasizing

		social change, founded in 1992 by Soraida Martinez.
197	Video Art	Art which relies on moving pictures and comprises video and/or audio data.
198	Viennese Actionism	A short and violent movement in 20th-century art, associated with Vienna.
199	Visual Art	Art forms that create works that are primarily visual in nature.
200	Vorticism	An early 20th-century movement in British art and poetry, influenced by Cubism and Futurism.

3.42.14.6.11 Get_Number

SDL.Get Number

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL.Get Number

Returns specified Numbers that are results of function calls from SDL. and SDO.



Intention

This command will return a specified number in **P2** from the internal Registers **P1**.

```
SDL.Get Number|1|$$RET
MBX.Number of Engines for SDO.: $$RET
```

Number	Variable	Source
1	SDO_Num_Engines	SDO_Call_StabilityAI_ListEngines_INT
2	SD_subseed_strength	SDL_Analyze_and_Save_Stable Diffusion
3	SD_subseed	SDL_Analyze_and_Save_Stable Diffusion
4	SD_seed	SDL_Analyze_and_Save_Stable Diffusion

Syntax

SDL.Get Number [| P1] [| P2]

Parameter Explanation

P1 - Number of the Internal Register that you want to read.

P2 - (*optional*) Variable for the result, if omitted the result is placed on TOS.

Example

```
! *****  
!  
! *****
```

Remarks

-

Limitations:

-

See also:

-

3.42.14.6.12 Get_Several

SDL.Get Several

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL.Get Several

Return internal AI-Registers to the User

Intention

The `SDL.Get Several` Command will return a lot of different internal Register values from the SD.-AI System.

Which value shall be return depends on the given Parameter.

```
SDL.Get Several|5|$$RET
MBX, The Style used is: $$RET
```

Case Number	Operation returns
1	SD_Clip_Guidance
2	SD_Engine
3	SD_Last_Body
4	SD_Sampler
5	SD_Style
6	SD_Return
7	StabilityAI_API_Generation_Endpoint_TXT2I MG
8	StabilityAI_API_Generation_Endpoint_IMG2I MG
9	SDO_Model
10	SD_PrintOut
11	SDL_Model

Syntax

SDL.Get Several [|P1] [|P2]

Parameter Explanation

P1 - (*optional*) Is the "Value Number" of the Parameter that should be returned in **P2** or on **TOS**.

P2 - (*optional*) Variable for the Return Value, if omitted the Value is placed on TOS.

Example

```
| *****  
|  
| *****
```

Remarks

-

Limitations:

-

See also:

-

3.42.14.6.13 Set_Scale

SDL.Set Scale

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL.Set Scale

Set the CfgScale for SDO. and SDL.

Configure the adherence level of the Stable Diffusion process to the provided prompt text.

Intention

The `SDL.Set cfgScale` command allows users to determine how strictly the Stable Diffusion (**SDL.** and **SDO.**) process should adhere to the prompt text provided. By adjusting the `cfgScale` value, users can control the degree to which the resultant output mirrors or diverges from the initial prompt, offering flexibility in achieving desired results.

Technical Explanation - `cfgScale`

The `cfgScale` parameter plays a pivotal role in guiding the SD.- process:

Adherence to Prompt: A higher `cfgScale` value ensures that the diffusion process remains more faithful to the provided prompt text. This means that the generated output will closely resemble or align with the initial prompt's intent.

Flexibility & Creativity: Conversely, a lower `cfgScale` value allows the SD.- process more latitude to deviate from the prompt, potentially leading to more unique or unexpected results. This setting can be useful when seeking more varied or creative outputs.

Balanced Approach: The default `cfgScale` value of 7 offers a balanced approach, ensuring a reasonable blend of adherence to the prompt while allowing for some creative freedom.

Syntax

SDL.Set Scale [| P1]

Parameter Explanation

P1 - (optional) <CfgScale Value>: Optional. Specifies the desired `cfgScale` value that determines the adherence level of the SD.- process to the prompt text. The valid range is between **0 and 35**. If omitted, the **default** value is set to **7**.

Example

```
|*****  
|  
|*****
```

Remarks

-

Limitations:

-

See also:

-

3.42.14.6.14 Set_Seed

[SDL.Set Seed / SDL.See](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL.Set Seed / SDL.See

Sets a new seed value for the random number generator in the Stable Diffusion image generation process for both SDO and SDL.

Intention

The SDL.Set Seed command is specifically designed to set a new seed value for the random number generator used in the Stable Diffusion image generation process. This command is applicable for both SDO and SDL commands.

By setting a new seed, you can control the randomness in the image generation process.

Same Seed means - under some circumstances (Sampler plays a role) -> same result. Additionally, the command places the current seed value on the TOS for further use or inspection.

Syntax

SDL.Set Seed [|P1]

Parameter Explanation

P1: (Optional) The new seed value between 0 and 4294967295.

This value will be used for the random number generator in the Stable Diffusion image generation process. If omitted, a default value of 50 is used.

Example

```

'*****
' SDL.-Sample for setting the seed value in Stable Diffusion
'*****
' Also available as SDL.Ss
SDL.Set Seed|100
' The new seed value for Stable Diffusion is now set to 100
' The current seed value will be placed on TOS
ENR.

```

Remarks

-

Limitations:

-

See also:

-

3.42.14.6.15 Set_Size

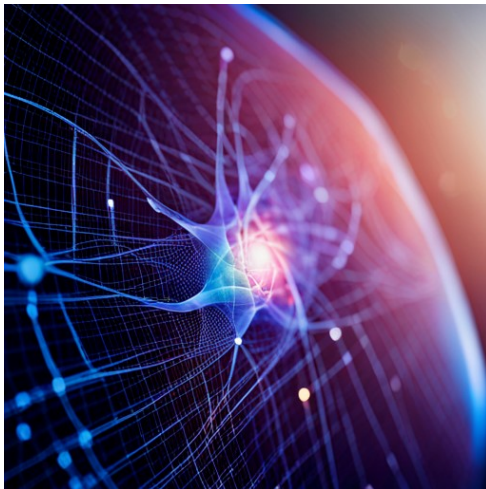
SDL.Set_Size

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDO.Set_Size

Set correct image size for SDO. and SDL.



Intention

This command is used to set the size of the image that will be generated by the Stable Diffusion Online (SDO) service.

The size is specified in pixels and must be a multiple of 64 and greater than or equal to 128.

The first parameter, **P1**, specifies the width of the image.

If the second parameter, **P2**, is provided, it specifies the height of the image. If **P2** is omitted, the height will be the same as the width.

The third parameter, **P3**, is optional and specifies the engine to be used for image generation.

The engine must be one of the following:

- esrgan-v1-x2plus
- stable-diffusion-xl-1024-v0-9
- stable-diffusion-xl-1024-v1-0
- stable-diffusion-v1
- stable-diffusion-v1-5
- stable-diffusion-512-v2-0
- stable-diffusion-768-v2-0
- stable-diffusion-depth-v2-0
- stable-diffusion-512-v2-1
- stable-diffusion-768-v2-1
- stable-diffusion-xl-beta-v2-2-2
- stable-diffusion-x4-latent-upscaler
- stable-inpainting-v1-0
- stable-inpainting-512-v2-0

There are rules for the possibly image sizes depending on the used Engine. This command will internally try to fix malformed size parameters.

If you do not want that use "SDO.Set_Size_Free".

General Rules for the Image Size:

Both the height and width of the image must be multiples of 64 and greater than or equal to 128.

The default value for both dimensions is 512.

Engine-Specific Rules:

512 Engines: For engines that have a resolution of 512 (like stable-diffusion-512-v2-0 or stable-inpainting-512-v2-0), the product of the height and width of the image must be between 262,144 and 1,048,576. This means that the dimensions of your image could range from 512x512 (which equals 262,144) up to approximately 1024x1024 (which equals 1,048,576).

768 Engines: For engines that have a resolution of 768 (like stable-diffusion-768-v2-0), the product of the height and width of the image must be between 589,824 and 1,048,576. This means that the dimensions of your image could range from approximately 768x768 (which equals 589,824) up to approximately 1024x1024 (which equals 1,048,576).

SDXL Beta: For the stable-diffusion-xl-beta-v2-2-2 engine, the dimensions can be as low as 128 and as high as 896, as long as neither dimension exceeds 512. If either dimension is greater than 512, then the other dimension can be at most 512.

SDXL v0.9 and v1.0: For the stable-diffusion-xl-1024-v0-9 and stable-diffusion-xl-1024-v1-0 engines, the valid dimensions are 1024x1024, 1152x896, 1216x832, 1344x768, 1536x640, 640x1536, 768x1344, 832x1216, or 896x1152. These specific combinations ensure that the product of the height and width falls within the acceptable range for these engines.

Examples:

```
' This command sets the image size to 512x768 pixels and uses the stable-diffusion-768-v2-0 engine.  
SDO.SetSize|512|768|stable-diffusion-768-v2-0
```

```
' This command sets the image size to 1024x1024 pixels and uses the stable-diffusion-xl-1024-v1-0 engine.  
SDO.SetSize|1024|stable-diffusion-xl-1024-v1-0
```

```
' This command sets the image size to 640x640 pixels and uses the default engine.  
SDO.SetSize|640
```

Syntax

SDO.SetSize | P1 [| P2] [| P3]

SDL.SetSize | P1 [| P2]

Parameter Explanation

P1 - The desired height of the image in pixels. This value must be a multiple of 64 and at least 128. The actual maximum value depends on the chosen engine and the width of the image.

P2 - (Optional) The desired width of the image in pixels. If omitted, the width will be the same as the height. This value must also be a multiple of 64 and at least 128. The actual maximum value depends on the chosen engine and the height of the image.

P3 - (Optional) This Parameter makes mostly sense for SDO.

The name of the engine to be used for generating the image. If omitted, the default engine will be used.

The available engines are currently:

- esrgan-v1-x2plus
- stable-diffusion-xl-1024-v0-9
- stable-diffusion-xl-1024-v1-0
- stable-diffusion-v1
- stable-diffusion-v1-5
- stable-diffusion-512-v2-0
- stable-diffusion-768-v2-0
- stable-diffusion-depth-v2-0
- stable-diffusion-512-v2-1
- stable-diffusion-768-v2-1
- stable-diffusion-xl-beta-v2-2-2
- stable-diffusion-x4-latent-upscaler
- stable-inpainting-v1-0
- stable-inpainting-512-v2-0

Each engine has specific requirements and limitations regarding the image dimensions. For example the esrgan engines can not be used for text-to-image. Also the "Inpainting" engines have special uses.

Example

```
| *****
|
| *****
```

Remarks

-

Limitations:

-

See also:

- [1.6.1. Program Flow Control](#)^[134]
- [! Smart Package Robot 's Parallel Robot Operations](#)^[3745]
- [1.5. Features and Hints](#)^[93]

3.42.14.6.16 Set_Size_free

[SDL.Set_Size Free](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL.Set_Size_Free

Set image size for SDO. and SDL.



Intention

This command is used to set the size of the image that will be generated by the Stable Diffusion Online (SDO) service.

The size is specified in pixels and must be a multiple of 64 and greater than or equal to 128.

The first parameter, P1, specifies the width of the image.

If the second parameter, P2, is provided, it specifies the height of the image. If P2 is omitted, the height will be the same as the width.

There are rules for the possibly image sizes depending on the used Engine. This command will NOT try to fix malformed size parameters.

If you do want that use "SDL.Set_Size".

General Rules for the Image Size:

Both the height and width of the image must be multiples of 64 and greater than or equal to 128.

The default value for both dimensions is 512.

Engine-Specific Rules:

512 Engines: For engines that have a resolution of 512 (like stable-diffusion-512-v2-0 or stable-inpainting-512-v2-0), the product of the height and width of the image must be between 262,144 and 1,048,576. This means that the dimensions of your image could range from 512x512 (which equals 262,144) up to approximately 1024x1024 (which equals 1,048,576).

768 Engines: For engines that have a resolution of 768 (like stable-diffusion-768-v2-0), the product of the height and width of the image must be between 589,824 and 1,048,576. This means that the dimensions of your image could range from approximately 768x768 (which equals 589,824) up to approximately 1024x1024 (which equals 1,048,576).

SDXL Beta: For the stable-diffusion-xl-beta-v2-2-2 engine, the dimensions can be as low as 128 and as high as 896, as long as neither dimension exceeds 512. If either dimension is greater than 512, then the other dimension can be at most 512.

SDXL v0.9 and v1.0: For the stable-diffusion-xl-1024-v0-9 and stable-diffusion-xl-1024-v1-0 engines, the valid dimensions are 1024x1024, 1152x896, 1216x832, 1344x768, 1536x640, 640x1536, 768x1344, 832x1216, or 896x1152. These specific combinations ensure that the product of the height and width falls within the acceptable range for these engines.

Examples:

```
' This command sets the image size to 512x768 pixels  
SDL.SetSize Free|512|768
```

```
' This command sets the image size to 1024x1024 pixels  
SDL.SetSize Free|1024
```

```
' This command sets the image size to 640x640 pixels.  
SDL.SetSize Free|640
```

This command lets you set any size, however if the size is not supported you will get an error-message from the Server instead of a picture.

Syntax

```
SDO.Set Size Free|P1 [|P2]  
SDL.Set Size Free|P1 [|P2]
```

Parameter Explanation

P1 - The desired height of the image in pixels. This value must be a multiple of 64 and at least 128. The actual maximum value depends on the chosen engine and the width of the image.

P2 - (Optional) The desired width of the image in pixels. If omitted, the width will be the same as the height. This value must also be a multiple of 64 and at least 128. The actual maximum value depends on the chosen engine and the height of the image.

Example

```
| *****  
|  
| *****
```

Remarks

-

Limitations:

The internal Registers for **SDO**. and **SDL**. for Height and With are the same. Any change here will also change the Size Variables for **SDL**. This way you can more easy switch between Local and Online Rendering.

See also:

-

3.42.14.6.17 Set_Steps

SDL.Set_Steps

[Previous](#) 1492 [Top](#) 49 [Next](#) 1499

MiniRobotLanguage (MRL)

SDL.Set_Steps

Set the number of steps for the Stable Diffusion (SD) process for SDL. and for SDO.





"Beauty of Nations". Left side "native Indian", right side "African". Created with the SDL-Command and Stable Diffusion Local.

Intention

The `SDL.Set Steps` command configures the number of iterations or cycles the Stable Diffusion (SD) process will undergo. By adjusting the number of steps, users can influence the depth and precision of the diffusion process, directly affecting the quality and detail of the resultant output.

Technical Explanation - Steps in Diffusion Process

In the context of the Stable Diffusion process, "steps" refer to the number of iterations the algorithm undergoes to transform, blend, or diffuse the input data into the final output. Each step can be seen as a refinement phase, where the algorithm further processes the data, adding details, enhancing features, or smoothing out

inconsistencies.

The number of steps plays a pivotal role in determining the quality of the output:

- **Depth of Process:** More steps mean the algorithm has more opportunities to refine and process the data, leading to richer and more detailed outputs.
- **Quality & Detail:** As the number of steps increases, the output tends to have more clarity, depth, and detail. This can lead to sharper images, more intricate patterns, or smoother gradients.
- **Sampler Dependency:** The impact of steps on quality is also influenced by the chosen sampler. Some samplers might produce optimal results with fewer steps, while others might require more iterations to achieve their best output.
- **Computational Time:** While more steps can lead to higher quality, they also introduce more computational demands. This means the process will require more processing time to complete.

Example Usage:

```
'This example configures the SD process to undergo 100 steps.
SDL.Set Steps|100
```

```
'In this instance, the number of steps for the SD process defaults to 70.
SDL.Set Steps
```

While increasing the number of steps can lead to enhanced quality and detail in the Stable Diffusion process, it's crucial to balance this with the associated computational costs and the characteristics of the chosen sampler.

Syntax

SDL.Set_Steps [| P1]

Parameter Explanation

P1 - (optional) <Number of Steps>: Optional. The desired number of steps for the SD process. This should be a number between 10 and 150. If omitted, the default value is set to 70.

Example

```
! *****
!
! *****
$$PRO=wooden Geodesic Dome,honeycomb, natural, a rainbow
$$NEG=ugly,comic,unrealistic,fat,unhealty,malformed faces
$$FIL=?exeloc\Sample_?.png
```

```
SDL.SetSteps|150
FOR.$$LM8|1|25
  RND.1|21|$$SEM
  SDL.Set_Sampler|$$SEM
  POP.$$SAM
  SDL.Set Model Free|dreamlike-photoreal-2.0.safetensors
  SDL.Set Extra Parameter|"restore_faces": true
  VAN.$$TIM=#dtime#
  SDL.gtf|$$PRO|$$FIL|$$NEG|4
  VAN.$$TIM=#dsince#|i
  DBP. Set $$SAM Sampler. In $$TIM Seconds.
NEX.
DMP.6
MBX.!
```

ENR.

Remarks

-

Limitations:

-

See also:

- [1.6.1. Program Flow Control](#)¹³⁴
- [! Smart Package Robot 's Parallel Robot Operations](#)³⁷⁴⁵
- [1.5. Features and Hints](#)⁹³

3.42.14.6.18 Show Error

[SDO.Show Error](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDO.Show_Error

Retrieve and display the latest errors encountered during Stable Diffusion processes using the SDO. and SDL. commands.

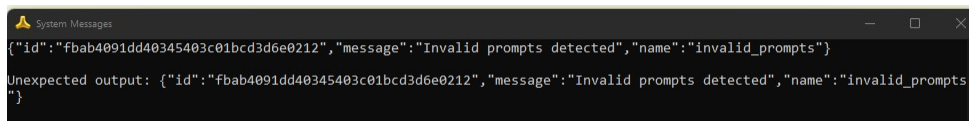


Intention

The "SDO.Show Error" command is designed to aid in debugging and error handling by providing detailed information on the last errors that occurred during the execution of Stable Diffusion Online (SDO) and Stable Diffusion Local (SDL) commands.

This command opens a console window that displays these errors and presents a message box that the user must confirm to close the console window. This command is especially useful for understanding what might have gone wrong during a Stable Diffusion process and for informing corrective action.

Result of AIC.Show Error



```
System Messages
{"id":"fbab4091dd40345403c01bcd3d6e0212","message":"Invalid prompts detected","name":"invalid_prompts"}
Unexpected output: {"id":"fbab4091dd40345403c01bcd3d6e0212","message":"Invalid prompts detected","name":"invalid_prompts"}
```

Syntax

SDO. [| P1]

Parameter Explanation

P1 - (*optional*): Variable to return the last Error Messages during the Stable Diffusion Call.

If omitted the result is placed on TOS.

Example

```
! *****
!
! *****
$$PRO=native women, natural smile, beautiful, long hair rainbow colors, honeycomb,
$$PRO=slim Siren, long golden color hair, on rock over moonlit Rhine
$$NEG=ugly
SDO.Set_Samples|1
SDO.Set_Sampler|7|$$SMP
DBP.Sampler: $$SMP
SDO.Set Steps 120
SDO.Set Scale|1.5
$$FIL=?path\Lorea_?.jpg
FOR.$$STY|0|16
    $$FIL=?path\Lorea_$$STY.jpg
    SDO.gtf|$$PRO|$$FIL|$$NEG|$$STY
    POP.$$PAT
    POP.$$SUC
    IVV.$$SUC=1
        ANA.load|0|$$PAT
        ANA.Resizeto|0|1|640|640
        SDL.Get Several|5|$$STA
        %PicPrint SPR/SDO. generated: Style $$STA
        ANA.show|1
        ANA.Save|1|$$PAT
    ELS.
        SDL.Get Several|6|$$RET
        DBP.$$RET
        SDL.Show Error
        MBT.No File generated.
EIF.
```

NEX.

Remarks

-

Limitations:

-

See also:

- [1.6.1. Program Flow Control](#)^[134]
- [! Smart Package Robot 's Parallel Robot Operations](#)^[3745]
- [1.5. Features and Hints](#)^[93]

3.42.14.7 SDL. - Stable Diffusion Local System

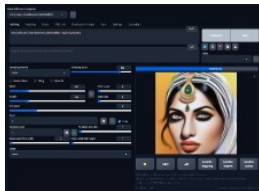
Using Stable Diffusion to generate Pictures [Previous](#) [Top](#) [Next](#)

Stable Diffusion on your own Computer (Local)

To use Stable Diffusion Online, you will need to

[Install and prepare Local Stable Diffusion](#)

You can then test your installation using the WEB-GUI from Automatic1111. Once tested, you can use SD with the SPR. The local version of Stable diffusion is uncensored. Therefore watch what you enter.



Once tested you can use Stable Diffusion with the SPR.

The following Sample-Code will take your Theme-Prompt, ask Open AI "ChatGPT" to improve it.

And then generate the Picture with **SDL**.

You need to have your [OpenAI API-Key](#) saved locally in the Scriptfolder using the [Save_Key](#) Command.

```

VAR.$$THE="neural network beautiful golden girl Galaxy robot digital photorealistic"
VAR.$$PAO=Please make me a prompt below 231 characters, for stable diffusion use:
' Use Image Register 0
VIN.$$IMR=0

' Set OpenAI API-Key from the saved File
AIC.SetKey|File
' Set Model-Temperature
AIC.Set_Temperature|0

' Set Max-Tokens (Possible lenght of answer, depending on the Model up to 2000 Tokens)
' The more Tokens you use the more you need to pay. But the longer Input and Output
AIC.SetMax_Token|300

FOR.$$STP|1|25
  AIC.Ask_Chat|$$PAO|$$RET
  DBP. Got: <<$$RET >>
  $$FIR=?path\Sample Pics\Testresult_?.png
  FIL.gen|$$FIR|1|0|$$FIL
  SDL.gtf|$$RET|$$FIL
  ANA.Load|$$IMR|$$FIL
  ' This will print a Text inside the Picture
  $$TXT=SPR/SDO. generated: $$STP Steps
  $$COA=&HFFFFFFF
  $$COB==&H0

```

```
$$BGC=-2
$$XPA=30
$$YPA=450
ANA.PrintAt|$$IMR|$$TXT|$$COA|$$BGC|$$XPA|$$YPA|24
ANA.PrintAt|$$IMR|$$TXT|$$COB|$$BGC|($$XPA+1)|($$YPA+2)|24
' Mow we load the picture
DBP.Loaded in IR:$$IMR -> $$FIL
ANA.Show|$$IMR!
NEX.
ENR.
```


3.42.14.7.1 !SDL - IMG2IMG

Possible Parameters

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL. - Image to Image

Use an provided Image and change it using AI.

Intention

Here are all supported Parameters for Img2Img, using SDL. and the Automatic1111-API.

```

'*****
' SDL.-Sample
'*****
' Sample Script that shows how to use these Parameters.
$$PRO=hero,propaganda (0.6),epic,russia(0.4),sexy
$$EXT=
$$EXT+"prompt": "$$PRO",
$$EXT+"denoising_strength": 0.75,
$$EXT+"image_cfg_scale": 4,
$$EXT+"batch_size": 2,
$$EXT+"n_iter": 3,
$$EXT+"steps": 70,
$$EXT+"cfg_scale": 7,
$$EXT+"width": 512,
$$EXT+"height": 512,
$$EXT+"restore_faces": true,
$$EXT+"tiling": false,
SDL.Set Extra Parameter|$$EXT

$$IMG=?exeloc\TP01.jpg
$$TAR=?exeloc\TP02.png
SDL.Img2Img|$$IMG|$$TAR

SDL.Show Error
'MBT.No File generated.
ENR.

```



Details and Parameter description

Parameter	Default Value	Value Range	Explanation
init_images	["string"]	Array of strings	Initial image(s) for the model Base64 encoded
resize_mode	0	Integer	Mode for resizing images, <ul style="list-style-type: none"> • 0: Just Resize • 1: Crop and Resize • 2: Resize and Fill
denoising_strength	0.75	0.0 - 1.0	Strength of denoising filter, higher means more AI Influence in the result
image_cfg_scale	0	Integer	Scale configuration for image
mask	"string"	String	Mask applied to the image Base64 encoded
mask_blur	0	Integer	Amount of blur applied to mask
mask_blur_x	4	Integer	X-axis blur for mask
mask_blur_y	4	Integer	Y-axis blur for mask
inpainting_fill	0	Integer	Fill value for inpainting
inpaint_full_res	true	Boolean	Whether inpainting is done at full

			resolution
inpaint_full_res_padding	0	Integer	Padding for inpainting at full resolution
inpainting_mask_invert	0	Integer	Inversion of inpainting mask
initial_noise_multiplier	0	Integer	Multiplier for initial noise
prompt	""	String	Text prompt for the model
styles	["string"]	Array of strings	Styles applied to the model
seed	-1	Integer	Random seed
subseed	-1	Integer	Subseed for randomization
subseed_strength	0	Integer	Strength of subseed effect
seed_resize_from_h	-1	Integer	Seed resize height
seed_resize_from_w	-1	Integer	Seed resize width
sampler_name	"string"	String	Name of the sampling algorithm
batch_size	1	Integer	Number of images processed in a batch
n_iter	1	Integer	Number of iterations
steps	50	Integer	Number of steps for the algorithm
cfg_scale	7	Integer	Configuration scale
width	512	Integer	Width of the output image
height	512	Integer	Height of the output image
restore_faces	false	Boolean	Whether to restore faces in the image
tiling	false	Boolean	Whether tiling is enabled
do_not_save_samples	false	Boolean	Flag to prevent saving of samples
do_not_save_grid	false	Boolean	Flag to prevent saving of grid
negative_prompt	"string"	String	Negative text prompt for the model

eta	0	Integer	Learning rate parameter
s_min_uncond	0	Integer	Minimum unconditional step size
s_churn	0	Integer	Churn rate
s_tmax	0	Integer	Maximum temperature
s_tmin	0	Integer	Minimum temperature
s_noise	1	Integer	Noise level
override_settings	{}	JSON Object	Settings to override
override_settings_restore_afterwards	true	Boolean	Whether to restore settings afterwards
script_args	[]	Array	Additional script arguments
sampler_index	"Euler"	String	Index of the sampler
include_init_images	false	Boolean	Whether to include initial images

Parameter Explanation

[The denoising_strength parameter is used in the img2img/inpaint module.](#)

It is a float that represents the strength of denoising applied to an image. [According to a discussion on GitHub, the following values are recommended for the denoising_strength parameter:](#)

- If you are confident about the model you are using and want fewer modifications on the image, lower your denoising_strength to ~0.4. Switch your hires. fix upscaler to non-Latent upscalers (e.g. LANCZOS, ESRGAN). You will get some nice images.
- If you want more modifications and more potential, apply a higher denoising_strength (~0.55 to ~0.7). Pick a favorite Latent upscaler. You will get just what you want.
- For a general solution, don't use hires. fix. Use extra->upscale to upscale your image (make your image sharper if possible), send the upscaled image to img2img. If you want more potential, using Inpaint and Inpaint Sketch is better. Process and finetune. You will get some awesome images (just wasting too much time compared with other alternative solutions).

[The image_cfg_scale parameter is used in the img2img/inpaint module.](#)

It is a float that represents the scale of the image configuration.

[According to a discussion on GitHub, the image_cfg_scale parameter is added as an option to the XYZ Plot script on the img2img tab when an InstructPix2Pix model is loaded.](#)

[However, there is another discussion on GitHub where a user reported that the](#)

[image_cfg_scale](#) slider for InstructPix2Pix models provided in the `img2img` tab seems to not do anything.

[The mask parameter is used in the `img2img/inpaint` module.](#)

It is a binary image that represents the region of the image to be inpainted.

[According to a discussion on GitHub, the mask parameter can be used in conjunction with the `mask_blur` option to apply a Gaussian blur on the section of the image that has the mask.](#)

[This can help to produce a smoother blend with the original image and force the regions inside to match up more smoothly with the parts at the edges which aren't changing as much.](#)

Example

```
! *****  
! IRS.-Sample  
! *****
```

Remarks

-

Limitations:

-

See also:

- [1.6.1. Program Flow Control](#)^[134]
- [! Smart Package Robot 's Parallel Robot Operations](#)^[3745]
- [1.5. Features and Hints](#)^[93]

3.42.14.7.1.1 !SDL. - IMG2IMG - JSON

SDL. - Image to Image

[Previous](#)  [Top](#)  [Next](#) 

MiniRobotLanguage (MRL)

SDL. - Image to Image

Use an provided Image and change it using AI.

Intention

Here are all supported Parameters for Img2Img, using SDL. and the Automatic1111-API.

JSON API:

These terms can be used with the SDL.IMG2IMG Command and the SDL.Set Extra Parameter Command:

```
{
  "init_images": [
    "string"
  ],
  "resize_mode": 0,
  "denoising_strength": 0.75,
  "image_cfg_scale": 0,
  "mask": "string",
  "mask_blur": 0,
  "mask_blur_x": 4,
  "mask_blur_y": 4,
  "inpainting_fill": 0,
  "inpaint_full_res": true,
  "inpaint_full_res_padding": 0,
  "inpainting_mask_invert": 0,
  "initial_noise_multiplier": 0,
  "prompt": "",
  "styles": [
    "string"
  ],
  "seed": -1,
  "subseed": -1,
  "subseed_strength": 0,
  "seed_resize_from_h": -1,
  "seed_resize_from_w": -1,
  "sampler_name": "string",
  "batch_size": 1,
  "n_iter": 1,
  "steps": 50,
  "cfg_scale": 7,
  "width": 512,
  "height": 512,
  "restore_faces": false,
  "tiling": false,
  "do_not_save_samples": false,
  "do_not_save_grid": false,
  "negative_prompt": "string",
  "eta": 0,
  "s_min_uncond": 0,
  "s_churn": 0,
  "s_tmax": 0,
  "s_tmin": 0,
  "s_noise": 1,
  "override_settings": {},
  "override_settings_restore_afterwards": true,
  "script_args": [],
  "sampler_index": "Euler",
  "include_init_images": false,
```

```
"script_name": "string",  
"send_images": true,  
"save_images": false,  
"alwayson_scripts": {}  
}
```


3.42.14.7.1.2 !SDL.- More Parameter

[Adding Parameters](#)
[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

More Parameters

You can add other Parameters using the `SDL.Set Extra Parameter Command`

Intention

There are a lot of Parameters that are not represented by internal Registers. These Parameters must be added using

```
SDL.Set Extra Parameter|$$EXT
```

You can just copy those lines that you need for your project.

Please be aware of the new Option to append to a variable using:

```
' First command will SET Variable TO (=)
$$EXT="resize_mode": 0,
' Following Commands APPEND to that variable (+)
$$EXT+"denoising_strength": 0.35
```

Here you can see all optional Parameters that are added th

The following Parameters must use the provided internal Re
`"negative_prompt", "height", "width", "cfg_scale", "steps", "s`

Therefore you must use `"SDL.Set Prompt|$$PRO"` to set the Prompt same for the other Parameters that require the use of internal Registers.

```
' Parameters with internal Registers
$$EXT="prompt": "Positive Prompt Text",
$$EXT="negative_prompt": "Negative Prompt Text",
$$EXT+"styles": ["Styles that are available"],
$$EXT+"sampler_name": "SD_Sampler",
$$EXT="steps": SD_Steps,
$$EXT="cfg_scale": SD_cfg_scale,
$$EXT="width": SD_width,
$$EXT="height": SD_height,

' Parameters with NO Internal Registers
$$EXT+"resize_mode": 0,
$$EXT+"denoising_strength": 0.35,
```

```
$$EXT+"image_cfg_scale": 9,

$$EXT+"mask_blur": 0,
$$EXT+"mask_blur_x": 4,
$$EXT+"mask_blur_y": 4,
$$EXT+"inpainting_fill": 0,
$$EXT+"inpaint_full_res": true,
$$EXT+"inpaint_full_res_padding": 0,
$$EXT+"inpainting_mask_invert": 0,
$$EXT+"initial_noise_multiplier": 0,

$$EXT+"seed": -1,
$$EXT+"subseed": -1,
$$EXT+"subseed_strength": 0,
$$EXT+"seed_resize_from_h": -1,
$$EXT+"seed_resize_from_w": -1,
$$EXT+"batch_size": 1,
$$EXT+"n_iter": 1,

$$EXT+"restore_faces": false,
$$EXT+"tiling": false,
$$EXT+"do_not_save_samples": false,
$$EXT+"do_not_save_grid": false,

$$EXT+"eta": 0,
$$EXT+"s_min_uncond": 0,
$$EXT+"s_churn": 0,
$$EXT+"s_tmax": 0,
$$EXT+"s_tmin": 0,
$$EXT+"s_noise": 1,
$$EXT+"override_settings": {},
$$EXT+"override_settings_restore_afterwards": true,
$$EXT+"script_args": [],
$$EXT+"sampler_index": "Euler",
$$EXT+"include_init_images": false,
$$EXT+"script_name": "string",
$$EXT+"send_images": true,
$$EXT+"save_images": false,
$$EXT+"always_on_scripts": {}

SDL.Set Extra Parameter|$$EXT
```

Example

```
! *****  
! IRS.-Sample  
! *****
```

Remarks

-

Limitations:

-

See also:

- [1.6.1. Program Flow Control](#)¹³⁴
- [! Smart Package Robot 's Parallel Robot Operations](#)³⁷⁴⁵
- [1.5. Features and Hints](#)⁹³

3.42.14.7.1.3 Img2img

[SDL.Img2Img](#)

[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL . Img2Img

Initiates the Image-to-Image process using the Stable Diffusion API from Automatic1111, generating one or multiple images based on the parameters set.

Intention

The `SDL . Img2Img` command is designed to start the Image-to-Image process using the Stable Diffusion API.

It takes an original image (**P1**) and a target path (**P2**) to generate new images.

The command can produce multiple images depending on the parameters set using the `SDL.Set Extra Parameters` command.

An optional parameter (**P3**) can be used to return the paths of all generated images.

Important:

This command does not rely on any other commands like "SDL.Set Prompt" or "SDL.Set Style" etc.

Using this command you have to specify ALL Parameters using the `SDL.Set Extra Parameters` command.

The Image-Parameters need a special processing and are therefore specified directly with the command,

Syntax

SDL . Img2Img | P1 | P2 [| P3]

Parameter Explanation

- **P1 - Original Pic:** Specifies the file path of the original image to be processed.
- **P2 - Target Path:** Specifies the target file path where the new image(s) will be saved. Can contain a "?" to indicate multiple images.
- **P3 - (Optional): Variable for Result Path:** A variable to store the file paths of all generated images. If omitted, these are placed on the Top Of Stack (TOS).

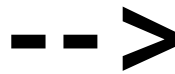
Returns a list of Filenames of all generated files, separated using the "*".

Example

! *****

```
' SDL.-Sample
!*****
$$PRO=girl,hero,propaganda,sexy,epic,russia
$$NEG=ugly,unnatural, fat
$$EXT=
$$EXT+"prompt": "$$PRO",
$$EXT+"negative_prompt": "$$NEG",
$$EXT+"denoising_strength": 0.75,
$$EXT+"image_cfg_scale": 4,
$$EXT+"batch_size": 3,
$$EXT+"n_iter": 1,
$$EXT+"steps": 70,
$$EXT+"cfg_scale": 7,
$$EXT+"width": 512,
$$EXT+"height": 512,
$$EXT+"restore_faces": true,
$$EXT+"tiling": false,
SDL.Set Extra Parameter|$$EXT

$$IMG=?exeloc\TP01.jpg
$$TAR=?exeloc\SPA.png
SDL.Img2Img|$$IMG|$$TAR|$$RES
MBX.Generated Files: $$RES
ENR.
```



Remarks

-

Limitations:

-

See also:

-

3.42.14.7.1.4 Img2Img Intern

`SDL.Img2Img Intern`[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL.Img2Img Intern

Initiates the Image-to-Image process using the Stable Diffusion API from Automatic1111, generating one or multiple images based on the parameters set.

Intention

The `SDL.Img2Img Intern` command is designed to start the Image-to-Image process using the Stable Diffusion API.

It takes an original image (**P1**) and a target path (**P2**) to generate new images.

The command can produce multiple images **depending** on the **Internal Parameter registers** that are set using these commands:

```
SDL.Set Style Free
SDL.Set Prompt
SDL.Set negative Prompt
SDL.Set Sampler
SDL.Set Steps
SDL.Set Scale
SDL.Set Size
```

All other parameters must be set using the `SDL.Set Extra Parameters` command.

An optional parameter (**P3**) can be used to return the paths of all generated images.

The Image-Parameters need a special processing and are therefore specified directly with the command,

Syntax

SDL.Img2Img Intern | P1 | P2 [| P3]

Parameter Explanation

- **P1 - Original Pic:** Specifies the file path of the original image to be processed.
- **P2 - Target Path:** Specifies the target file path where the new image(s) will be saved. Can contain a "?" to indicate multiple images.
- **P3 - (Optional): Variable for Result Path:** A variable to store the file paths of all generated images. If omitted, these are placed on the Top Of Stack (TOS).

Returns a list of File names of all generated files, separated using the "*".

Example

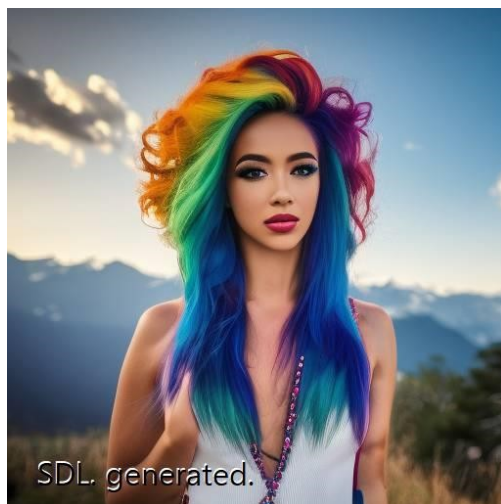
```

'*****
' SDL.-Sample
'*****
$$PRO=girl,hero,propaganda,sexy,epic,russia
$$NEG=ugly,unnatural, fat
$$EXT=
$$EXT+"denoising_strength": 0.75,
$$EXT+"batch_size": 3,
$$EXT+"n_iter": 1,
$$EXT+"restore_faces": true,
$$EXT+"tiling": false,
SDL.Set Extra Parameter|$$EXT

SDL.Set Prompt|$$PRO
SDL.Set negative Prompt|$$NEG
SDL.Set Scale|7
SDL.Set Size|512|512
SDL.Set Steps|90

$$IMG=?exeloc\TP01.jpg
$$STAR=?exeloc\IPA_?.png
SDL.Img2Img Intern|$$IMG|$$STAR|$$RES
MBX.Generated Files: $$RES
ENR.

```



Remarks

-

Limitations:

-

See also:

-

3.42.14.7.1.5 Img2Msk

[SDL.Img2Msk](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL . Img2Msk

Initiates the Image-to-Image process using the Stable Diffusion API from Automatic1111, generating one or multiple images based on the parameters set.

Intention

The `SDL . Img2Msk` command is designed to start the Image-to-Image process using the Stable Diffusion API.

It takes an original image (**P1**) and a Mask-Picture can be used to generate picture(s) in the target path (**P3**).

The command can produce multiple images depending on the parameters set using the `SDL.Set Extra Parameters` command.

An optional parameter (**P4**) can be used to return the paths of all generated images.

Important:

This command does not rely on any other commands like "SDL.Set Prompt" or "SDL.Set Style" etc.

Using this command you have to specify ALL Parameters using the `SDL.Set Extra Parameters` command.

The Image-Parameters need a special processing and are therefore specified directly with the command,

Syntax

SDL . Img2Msk | P1 | P2 | P3 [| P4]

Parameter Explanation

- **P1 - Original Pic:** Specifies the file path of the original image to be processed.
- **P2 - Mask Path:** Specifies the File name and Path for the Mask-File
- **P3 - Target Path:** Specifies the target file path where the new image(s) will be saved. Can contain a "?" to indicate multiple images.
- **P4 - (Optional): Variable for Result Path:** A variable to store the file paths of all generated images. If omitted, these are placed on the Top Of Stack (TOS).

Returns a list of File names of all generated files, separated using the "*".

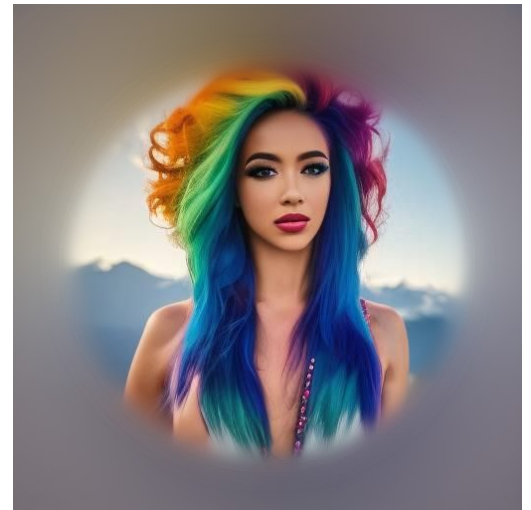
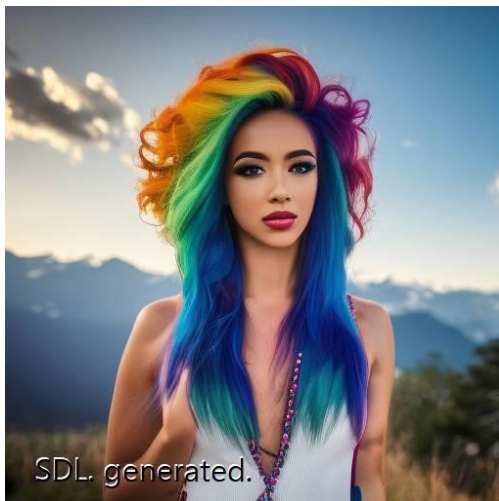
Example

```

!*****
! SDL.-Sample
!*****
$$PRO=girl,hero,propaganda,sexy,epic,russia
$$NEG=ugly,unnatural, fat
$$EXT=
$$EXT+"prompt": "$$PRO",
$$EXT+"negative_prompt": "$$NEG",
$$EXT+"denoising_strength": 0.75,
$$EXT+"image_cfg_scale": 4,
$$EXT+"batch_size": 3,
$$EXT+"n_iter": 1,
$$EXT+"steps": 70,
$$EXT+"cfg_scale": 7,
$$EXT+"width": 512,
$$EXT+"height": 512,
$$EXT+"restore_faces": true,
$$EXT+"tiling": false,
SDL.Set Extra Parameter|$$EXT

$$MSK=?exeloc\Masks\Weiser Kreis B 512.png
$$IMG=?exeloc\TP01.jpg
$$TAR=?exeloc\ZPA_?.png
SDL.Img2Msk|$$IMG|$$MSK|$$TAR|$$RES
MBX.Generated Files: $$RES
ENR.

```



Remarks

-

Limitations:

-

See also:

-

3.42.14.7.1.6 Img2Msk Intern

[SDL.Img2Msk Intern](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL.Img2Msk Intern

Initiates the Image-to-Image process using the Stable Diffusion API from Automatic1111, generating one or multiple images based on the parameters set - including a Mask-Picture.

Intention

The `SDL.Img2Msk Intern` command is designed to start the Image-to-Image process using the Stable Diffusion API. It takes an original image (**P1**) and a Mask-Picture (**P2**) and the target path (**P3**) to generate new images.

The command can produce multiple images **depending** on the **Internal Parameter registers** that are set using these commands:

```
SDL.Set Style Free
SDL.Set Prompt
SDL.Set negative Prompt
SDL.Set Sampler
SDL.Set Steps
SDL.Set Scale
SDL.Set Size
```

All other parameters must be set using the `SDL.Set Extra Parameters` command.

An optional parameter (**P4**) can be used to return the paths of all generated images.

The Image-Parameters need a special processing and are therefore specified directly with the command,

Syntax

SDL.Img2Msk Intern | P1 | P2 [| P3]

Parameter Explanation

- **P1 - Original Pic:** Specifies the file path of the original image to be processed.
- **P2 - Mask Path:** Specifies the File name and Path for the Mask-File
- **P3 - Target Path:** Specifies the target file path where the new image(s) will be saved. Can contain a "?" to indicate multiple images.
- **P4 - (Optional): Variable for Result Path:** A variable to store the file paths of all generated images. If omitted, these are placed on the Top Of Stack (TOS).

Returns a list of File names of all generated files, separated using the "*".

Example

```

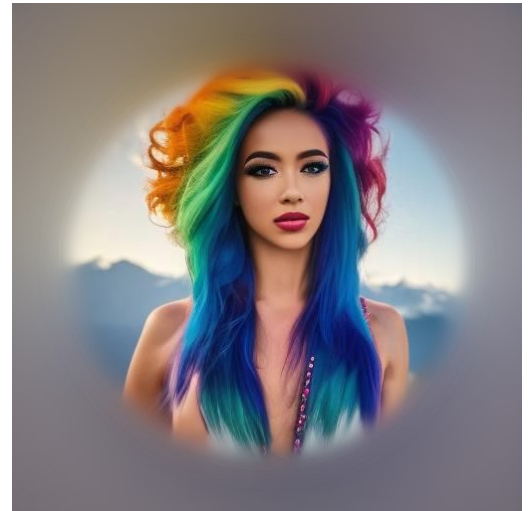
!*****
! SDL.-Sample
!*****
$$PRO=girl,hero,propaganda,sexy,epic,russia
$$NEG=ugly,unnatural, fat
$$EXT=
$$EXT+"denoising_strength": 0.25,
$$EXT+"batch_size": 3,
$$EXT+"n_iter": 1,
$$EXT+"restore_faces": true,
$$EXT+"tiling": false,
$$EXT+"mask_blur": 10,
$$EXT+"mask_blur_x": 14,
$$EXT+"mask_blur_y": 14,
$$EXT+"inpainting_fill": 0,
$$EXT+"inpaint_full_res": true,
$$EXT+"inpaint_full_res_padding": 0,
$$EXT+"inpainting_mask_invert": 1,
$$EXT+"initial_noise_multiplier": 0.5,

SDL.Set Extra Parameter|$$EXT

SDL.Set Prompt|$$PRO
'SDL.Set negative Prompt|$$NEG
SDL.Set Scale|7
SDL.Set Size|512|512
SDL.Set Steps|90

$$MSK=?exeloc\Masks\Weiser Kreis B 512.png
$$IMG=?exeloc\TP01.jpg
$$STAR=?exeloc\ZPA_?.png
SDL.Img2Msk Intern|$$IMG|$$MSK|$$STAR|$$RES
MBX.Generated Files: $$RES
ENR.

```



Remarks

-

Limitations:

-

See also:

-

3.42.14.7.2 !SDL - TXT2IMG

[SDL. - Sample Scripts and Prompts](#)[Previous](#) [Top](#) [Next](#)

MiniRobotLanguage (MRL)

SDL. - Sample Scripts and Prompts

These Scripts are "ready to run"

[Script 1, see Prompt](#)

As officially there are no human races and telling the opposite is seen as "racist", Stable diffusion can show you the visible differences between people living at different continents.

We will show this using the Script below.

Before you can run the Script, you will however need to:

- have Stable Diffusion local or in your Network installed,
- have Stable diffusion running with --api Parameter, see [AI - Install and prepare Local Stable Diffusion](#)

Here are some of the resulting pictures. All these pictures are AI generated on the local Computer using the SDL.-Command, and the Script below.

As Stable Diffusion can only render Pictures of certain resolution, we are using the SPR and its ANA.-Command to automatically re-size the picture and save them as ".jpg".

Stable Diffusion will generally produce only ".png"-Pictures. Using the SPR you have the complete control over the Output of Stable Diffusion.

Unlike "Cloud-based Stable Diffusion" - which is available using the SDO.-Command, "Local Stable Diffusion" - which is behind the SDL.-Command, is completely uncensored.