

TalkingBox Demo

Dr. Dilip K. Prasad

School of Computer Science & Engineering

Nanyang Technological University

Singapore

**How to Configure the SWI-Prolog with MATLAB?**

The steps below are to configure SWI-Prolog with MATLAB. These steps will set up the environment path for the operating system. Then the operating system will allow MATLAB to interact with SWI-Prolog.

1. Install SWI-Prolog (32 or 64 bits) – Version SWI-Prolog 7.2.3 or later, if SWI-Prolog has not been installed.

* http://www.swi-prolog.org/download/stable
* Matlab 32 bits with SWI-Prolog 32 bits
* Matlab 64 bits with SWI-Prolog 64 bits

1. Update the file **librarypath.txt** that is located in **<Matlab\_Root>\toolbox\local** . Insert the following lines (<SWI-Prolog\_Root > in the SWI-Prolog root path.):

* <SWI-Prolog\_Root>
* <SWI-Prolog\_Root>\bin
* <SWI-Prolog\_Root>\library

Example:

* C:/Program Files/swipl
* C:/Program Files/swipl/bin
* C:/Program Files/swipl/library

1. Created an environment variable called SWI\_HOME\_DIR pointing to the SWI-Prolog root path.

* SWI\_HOME\_DIR=<SWI-Prolog\_Root>

1. Add in the PATH environment variable the following directories

* <SWI-Prolog\_Root>\bin
* <SWI-Prolog\_Root>\library

1. Restart the computer to apply the changes in the environment variables configuration.

**Installation of TalkingBox Demo**

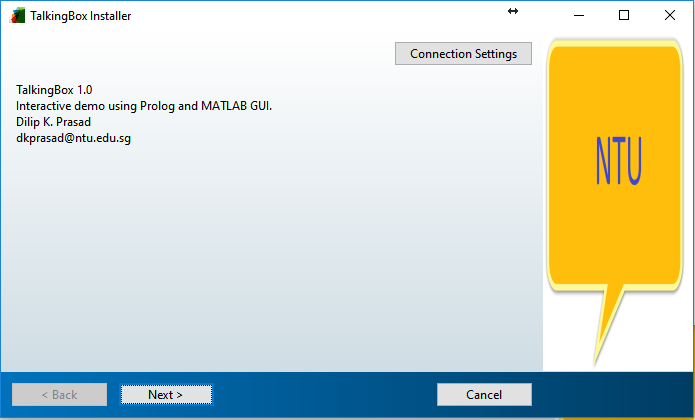
MATLAB Compiler

1. Prerequisites for Deployment

. Ensure you have installed version 9.2 (MATLAB 2017a).

Option 1 – Run the TalkingBoxInstaller\_web.exe

Follow the installation steps. Web installer will automatically find the suitable MATLAB runtime version from the web (internet connection is required)



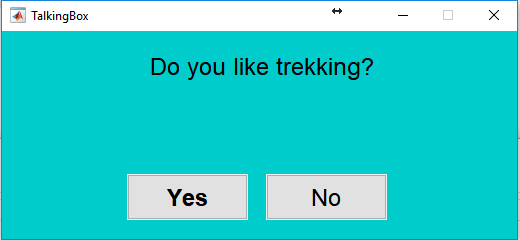
Click Next and follow the installation steps. If MATLAB runtime version 9.2 (R2017a) is not installed then you will be prompted to download ~460 MB MATLAB Runtime installation file. After the runtime installation, TalkingBox demo will be installed in the system. If the runtime version 9.2 (R2017a) is installed then it will not prompt the download and directly install the TalkingBox Demo in the system.

Option2: Set the path of MATLAB to TalkingBox\_Sourcecode

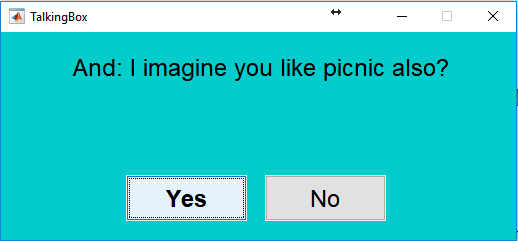
-Run the “Script\_demo\_interaction.m” file

**Run the TalkingBox from the desktop shortcut or startup menu**

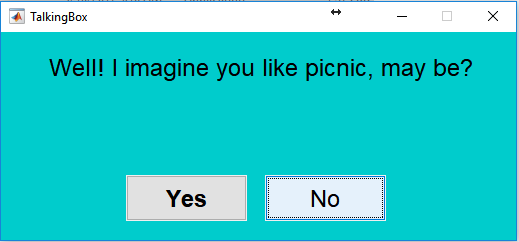
You will be prompted the first question “Do you like trekking?”



If Click Yes then you may see the below screen



If click No you may see the below screen



Note: If your system has speaker then you will hear the voice as well.

The TalkBox demo app is an example of multiplatform interactive Prolog project. The source code of this project will be shared with you. You will see the following files

1. Demo\_Interaction.fig – MATLAB GUI figure file
2. Demo\_Interaction.m- Matlab function file related to GUI
3. Exclaims.m – Matlab function to support this demo
4. QueryProlog1.m – Matlab function to query SWI-Prolog and interact with the Prolog compiler
5. tts.m- Matlab function for converting text to audio.
6. Script\_demo\_interaction.m – Matlab startup function for TalkingBox demo
7. basic\_interactive1.pl – The initial Prolog script.
8. interactive.pl – The prolog script which changes dynamically during the run time of the demo as per user response.

Note: Only interactive.pl script will be generated by the demo program.

We hope that this example demo will be useful in understanding the practical utility of Prolog Programming. This demo is a sample project to give you an idea of assignment project you will be working on.

Option 3 : If MATLAB software is not available then prolog script “basic\_interactive1.pl” can be executed alone using SWIProlog software.

Given below is the prolog script for TalkingBox

-------------------------------------------------

ask(trekking,0).

ask(X,Y):-

like(Y), related(X,Y).

ask(X,Y):-

random(X).

/\* member of a list\*/

member(X,[X|\_]).

member(X,[\_|R]) :- member(X,R).

/\* takeout a member from a list\*/

takeout(X,[X|R],R).

takeout(X,[F|R],[F|S]) :- takeout(X,R,S).

/\* append a member to a list\*/

append([A | B], C, [A | D]) :- append(B, C, D).

append([], A, A).

related(X,Y):- romantic(L),member(X,L),member(Y,L).

/\*,succ(Count\_romantic,Count\_romantic).\*/

related(X,Y):- outdoorsy(L),member(X,L),member(Y,L).

/\*,succ(Count\_outdoorsy,Count\_outdoorsy).\*/

related(X,Y):- social(L),member(X,L),member(Y,L).

/\*,succ(Count\_social,Count\_social).\*/

related(X,Y):- health\_freak(L),member(X,L),member(Y,L).

/\*,succ(Count\_health\_freak,Count\_health\_freak).\*/

related(X,Y):- loner(L),member(X,L),member(Y,L).

random(X):- romantic(L),member(X,L).

/\*,succ(Count\_romantic,Count\_romantic).\*/

random(X):- outdoorsy(L),member(X,L).

/\*,succ(Count\_outdoorsy,Count\_outdoorsy).\*/

random(X):- social(L),member(X,L).

/\*,succ(Count\_social,Count\_social).\*/

random(X):- health\_freak(L),member(X,L).

/\*,succ(Count\_health\_freak,Count\_health\_freak).\*/

random(X):- loner(L),member(X,L).

/\*,succ(Count\_loner,Count\_loner).\*/

romantic([gifts,wine,dinner,candlelight,rains,tea,concert,night,poetry,music,movie,dating,magic,novels,stories,roses,bouquets,courtship,chickflicks,cruise,breeze,diaries,painting]).

outdoorsy([picnic,trekking,soccer,sports,jogging,kayaking,parks,event,woods,mountains,beaches,cricket,action\_movies,dogs,lakes,fairs,swimming,breeze,fitness,water]).

social([coffee,picnic,friends,party,beer,music,concert,event,movie,soccer,dinner,gifts,gardens, roses, flowers,bouquet,cricket,board\_games,cafe,netflix,dogs,networking,exhibitions,fairs,debates]).

health\_freak([sports,tea,fruitjuice,smoothie,trekking,training,jogging,soccer,sweating,sleeping,swimming,lakes,exercise,burpees,fitness,cats,water,music]).

loner([books,coffee,woods,candlelight,sleeping,training,jogging,night,tea,beaches,poetry,tea,rains,fiction, novels,board\_games,netflix,cats,music,painting,sketching,writing,diaries]).

a.

like(nothing).

dislike(nothing).

a.

This script sample run will give log as below



Option 4: Independent and interactive Prolog script – “TalkingBoxPrologOnly.pl” can be executed alone using SWIProlog software.

Description: It also achieve the same goal of option 1, 2, &3. It’s different from “basic\_interactive1.pl” in terms two aspects – (a) conflict management is achieved via Prolog, (b) It provides better user interactivity via command prompt.

Ask(0). Is starting command and to provide the feedback y/n/q are the options

y. – yes, n. – no, q. – Quit the script

The users choices will be stored in like(X) and dislike(X)

like(X). – will store yes choices, dislike(X) – will store no choices (nothing is dummy initialization only)

“TalkingBoxPrologOnly.pl” script sample run will give log as below:

