idl2mat
IDL-Matlab Communication Package

Finn Årup Nielsen

1998 February 5, *Revision* : 1.1

idl2mat is a collection of small programs written in IDL[1] and C to enable IDL to communicate with matlab. A Matlab *engine* can be opened from IDL and matrices can be constructed in IDL and copied to the engine. Matlab command strings can be written in the IDL session and executed by the engine and matrices can be fetch from matlab back to IDL.

1 Installation

The package requires:

- **IDL.** Version 5.0
- **Matlab.** Version 5.0, perhaps will version 4.2 also work.
- C compiler. The package has been compiled under Linux and SGI IRIX.

Unpack with something like:

```
mkdir idl2mat
cd idl2mat
... (download idl2mat.tar.gz)
gunzip idl2mat.tar.gz
tar xvf idl2mat.tar
```

You should edit in the `Makefile` to make it reflect your local environment: If matlab is not installed in its default directory (`/usr/local/matlab`) edit the `MATLAB` path. You should also edit `ARCH` to reflect the arcitecture matlab is running under. Look in the `/usr/local/matlab/extern/lib/` directory: `ARCH` should be set to the subdirectory name in that directory.

After this you should be able to build the program, i.e. the dynamic linked library file `idl2mat.so`, with the `make` command.

You have to set the `LD_LIBRARY_PATH` path so the operating system can find the idl2mat.so library and the matlab engine library. If you are on an SGI with the `tcsh` shell and start IDL in the directory containing the `idl2mat.so` file you should write:

```
setenv LD_LIBRARY_PATH .:/usr/local/matlab/extern/lib/sgi
```

2 Running

After the installation you should be able to start idl and write `idl2mat_test`. Then a small test program will be executed testing all the functions of the package. Perhaps the easiest way to use the package is to copy the IDL commands from the `idl2mat_test.pro` file.

The first thing you should do is to open a matlab engine with `idl2mat_open`. This function returns a engine pointer that should be used in all subsequent operations — so you have to assign that to a variable:

```
ep = idl2mat_open()
```

[1] By IDL is meant Interactive Data Langauge a computer environment from "Research Systems, Inc" for interactive analysis and visualization. It should not be confused with the other language with the same acronym IDL — the - Interface Definition Language which is something completely different

IDL function	Description
`idl2mat_open`	Open a engine
`idl2mat_close`	Close a engine
`idl2mat_put`	Copy a matrix from IDL to matlab
`idl2mat_get`	Copy a matrix from matlab to IDL
`idl2mat_eval`	Execute a matlab string
`idl2mat_evals`	Execute a matlab string and show the output on stdin

Table 1: Overview of the IDL functions.

The variable is a long integer (and it will not work if you typecast it to something else). With the matlab engine open you now be able to pipe commands to it:

```
result = idl2mat_evals(ep, "help")
```

Or:

```
result = idl2mat_evals(ep, "whos")
```

... which will give you the varibles defined in matlab[2].

You can also define variables in IDL and copy them to matlab. The package work only with double-precision floats, and only with 2-dimensional structures, i.e., matrices. That is matlab's default type. Define, e.g.:

```
a = dindgen(2,3)
```

... and put this matrix to the engine:

```
result = idl2mat_put(ep, a, "M")
```

The matlab engine has now defined a matrix called "M". Remember, matlab is — unlike IDL — case sensitive. You can now make matlab commands with this matrix, e.g.:

```
result = idl2mat_eval(ep, "K = sin(M)")
```

... and get the result back from the "K" matrix:

```
result = idl2mat_get(ep, "K", B)
```

At the end of your session you should close the engine:

```
result = idl2mat_close(ep)
```

In the case you have a hanging matlab engine, e.g., after you forgot to close the engine and exited IDL, you should terminate the session with `kill` or `killall` from the Unix prompt.

3 Background

The package is/was written by Finn Årup Nielsen and Peter Toft on the initiative of Stephen C. Strother, — all working within the Human Brain Project: Finn and Peter in the Department of Mathematical Modelling at the Technical University of Denmark and Stephen at the PET Imaging Service, VA Medical Center in Minneapolis.

The package is available from:

```
http://hendrix.imm.dtu.dk/software/atlant
```

There is no exclusive mail address for package: Comments and bugs should be sent to `fn@imm.dtu.dk` — Finn's email address.

[2] If the output text is truncated it is because the `BUFFER_LENGTH` variable defined in `idl2mat.c` is too small. You can edit this and recompile the library.