

Matoffice and Demoplayer Installation

10-Apr-2010, Version 1.4

1. General note

This software is distributed by the Institute of Information Technology of the University of Ulm ("ITUU"). It is intended as add-on for the lectures

a) Nachrichtentechnik I / II

b) Communications Engineering I / II

and it is only allowed for private use. By using this software you accept these conditions.

TRADEMARKS - Used trademarks, company names, product names and logos of other manufacturers are accepted and belong to the corresponding owner of that trademark. This is also valid if at concerning positions this is not mentioned again.

PROPERTY RIGHTS - The content of this web-offer is copyrighted. Copying or distribution of the software or parts of it in any form is only allowed with previous acceptance of ITUU.

LIMITATION OF LIABILITY - ITUU does not take any warranty for the correctness, the completeness or the applicability of the software for a certain purpose. The use is on your own responsibility. For disadvantages in conjunction with the use of this software ITUU does not take any responsibility. ITUU is not obligated to keep the software up to date. ITUU keeps the right to change the software at any time without prior notice.

2. Installation

The software is based on Matlab m-files, but the zip package contains p-files which were created from these m-files. p-files can be started under Matlab like m-files.

The demoplayer software is part of a more general software package called *Matoffice*, which was created over many years. It consists of large number of individual programs and supporting routines. The supporting routines are usually grouped in an appropriate named m-file and they can be called individually by using a "key" string as the first argument of the call. For more details see **4. Additional notes**.

There is a given directory structure with the folder "Matoffice" as root folder. The Matoffice folder and its subfolders must be known by Matlab, i.e. they have to be added to the list of Matlab search path's before using Matoffice. The following steps are needed:

Step 1: Unzip

Unzip the zip file, e. g. `Matoffice7p2v1p4.zip`. 7p2 stand for Matlab 7.2 and v1p4 means Matoffice version 1.4. As a target folder for unzip you can use any folder for Matoffice. Before step 2 you can also move the whole Matoffice directory to any other location, but keep the Matoffice subfolder structure. The additional example album in the zip package can be moved to a different location, but the subfolder structure must also be preserved.

Step 2: Run copyman and set the Matlab search path list

Start Matlab and write the following command in the command window:

```
run(copymanwithfullpath)
```

An example for copymanwithfullpath is

```
Copymanwithfullpath = 'C:\Matoffice7p2v1p4\Matoffice\Matofficesvc\copyman'
```

copyman is an abbreviation for Copymanager which is the central gui program of Matoffice. C:\ Matoffice7p2v1p4\ in this example is the unzip path, i.e. the path leading to the folder which contains Matoffice as a subfolder. A more common target path is C:\Programme\, e.g., but usually an unzip program proposes C:\ Matoffice7p2v1p4\, which is the name of the zip file.

The run command starts Copymanager with all necessary Matoffice paths added temporarily to the current Matlab path list. There are appropriate addpath statements inside the copyman routine.

To save Matoffice paths permanently use the Matlab menu “File - Set Path”. In the Set Path window you only need to save the current path setting with the save button. For further info see **Note 5**.

Step 3: Run Matoffice

There are three main gui programs in Matoffice: *Copymanager*, *Demoplayer* and *Alben*. The Copymanager plays a central role, because it allows to open the others. So run it first, see Step 2. If the path settings are correct, you can type “copyman” in the Matlab command window. Demoplayer can be started by typing “demoplayer” or with the Copymanager menu “Programme”. For alternatives and more info see **Note 5**.

Step 4: Check Java version

Some Java class-files are used in Matoffice. They again use the Java Virtual Machine (JVM) installed on the operating system, independently of Matlab.

If problems with Java routines occur, check the current JVM version with the Matoffice command “matofficesvc('JAVA-VERSION')”. Some Java routines are version dependent, so change the version if necessary.

Three versions are provided in the zip package: Versions 1.4, 1.5, 1.6. The current class files on the Java search path are version 1.6 files. To change this, copy the files from the appropriate subfolder

- ... Matoffice/Generell/Java/JavaClass14/, ..., ...16/
- to the Java class path ... Matoffice/Generell/Java/ and replace the old ones.

2. Notes

Note 1: Alben

Alben is a program for filing of all file types. A first sample album is provided in the zip package to start properly. You can copy it with its subfolders to any location. Preserve the directory structure. After starting the album program by clicking on the Copymanager menu “Alben”, you will see the main window. At the left there is a listbox (“Albenbox”), at the right an empty album page. There are three alternatives to get the first album running:

- Alternative 1
Double click to “Weitere Alben” in the Albenbox. Search for the example album, find the file “ExampleAlbum.mat” in the folder “00Beschreibung” which is a subfolder of “Alben”.
- Alternative 2
Use “Album importieren” in the menu “Datei” and follow further instructions to find the “ExampleAlbum.mat” as in Alternative 1.
- Alternative 3
Create a new Album by selecting “Neues Album (leer)” the menu “Datei”. Follow the instructions.

With the first album opened, an album page can be seen at the right. More albums can be added to the album directory. For further help see menu “Hilfe”.

Note 2: Right mouse button, tooltips

In all Matoffice Guis it is common to use the LEFT and RIGHT mouse buttons for different tasks when clicking on a Gui object (a pushbutton e.g.). Point with the mouse on an Gui object to see the tooltip string for short information.

Note 3: Additional readme files

There are additional readme files with some specific information in all Matoffice subfolders.

Note 4: Windows xp and further Gui settings

Concerning the Gui colors in Matoffice, the standard setting for Windows xp has less degrees of freedom than the “Windows classical” setting. For Windows xp change the Matoffice setting with Copymanager menu “Einstellungen – Win xp”. Depending on the screen, it may also be appropriate – independent of the Windows style – to change the dpi setting for the font size to 120dpi (i.e. 125%).

Note 5: More info

Saving the currently added paths in the Matlab path list is not always possible, e.g. if a multiuser installation of Matlab is used. Then the only remaining starting procedure for Matoffice (and Copymanager, Demoplayer, ...) is using the `run(...)` command explained in step 2. Of course, you can write this run command (with a long string in the argument) into

your own m-file with a shorter name and call this m-file, but again, then the path to this m-file must be in the Matlab path list.

With a running Copymanager gui further possible calls are used internally or can be used in the Matlab command window:

- a) `matofficesvc('RUN-COPYMAN')` or `copyman`
- b) `matofficesvc('RUN-ALBUM')` or `album`
- c) `matofficesvc('RUN-DEMOPLAYER')`

With all these calls a total initialization is made, where **all** global variables are set to initial values. Also font defaults are set. Alternatives without total initialization are

- d) `dm700copymanager`
- e) `ph000alben('OPEN')`

Additionally, for each call a) to c) there is a check: if at least one of those three programs is running, no total initialization is made. There is **no such check** for other Matoffice Gui programs, e. g. for `too300svchelp` (see 4. Additional notes). If these programs are running and an a), b) or c) call happens, they will fail to work after that. But normally they will be started from Copymanager, Alben or Demoplayer.

To create a start icon for copymanager under Windows drop first a link for Matlab onto the desktop e.g.. With a right button mouse click on the icon and then on “properties” to see the run command

```
MALABPATH\bin\win32\MATLAB.exe
```

for starting Matlab. It can be modified to

```
MALABPATH\bin\win32\MATLAB.exe /r matofficesvc('RUN-COPYMAN')
```

Observe the blanks after `exe` and `/r`. A double click on this icon will start Matlab and then Copymanager. A precondition is that the Matoffice path's were saved in the Matlab path list before. If not, there is a further possibility by using `eval`. For the example in step 2 this would be:

```
MALABPATH\bin\win32\MATLAB.exe /r eval(  
['run(''C:\Matoffice7p2vlp4\Matoffice\Matofficesvc\copyman'')'])
```

One string is meant here, without line break.

4. Additional notes

matofficesvc.m (in folder ... \Matoffice\Matofficesvc\) is the main start and basic service routine for Matoffice.

The whole Matoffice package consists of many individual routines, usually packed in appropriate named m-files. For more information run “too300svchelp” in Matlab command window or “SVC-Help – generell” in Copymanager menu “Programme”.

General info concerning Matoffice files (“name” can be any m-filename string):

a) Gui files

name.m	gui routine
namecb.m	callback (cb) ...
nameget.m	get ...
nameset.m	set ...
namesvc.m	service ...
name.mat	mat-file
nameglobal.m	describes global struct variables

b) general files

namesvc.m	general service routine
nameget.m	... get ...
nameset.m	... set ...
name.mat	mat-file

c) non-Matlab files

.dll	Windows dll, source = c/mex-file
.vbs	Windows vbs routines
.class	Java routines, source = java-files

In general one program or gui (name.m) needs cb/get/set/svc routines and mat-files for storing internal parameters. Example:

```
demoplayer.m
demoplayercb.m
demoplayerget.m
demoplayerset.m
demoplayer.mat
demoplayerglobal.m
```

In this example, the demoplayer routines do not need a svc-routine.

This general naming scheme is used in Matoffice throughout. On this basis tool routines can create and update other routines (like a gui) in a more or less automated way.

Matoffice cb-routines have an internal structure which separates individual sections. Each section contains m-statements which are called by a mouse click on a gui object which belongs to this section. get/set/svc-routines have a similar structure inside. Here each section performs a special task and it is addressed by a call of the routine with a “key”, e.g. demoplayerset with the key 'DISPLAY-SNR':

```
demoplayerset('DISPLAY-SNR',snrindb)
```

Parameters which are needed inside are in the argument list following the key, in this example snrindb. The tool too300svchelp.m (in folder ... \Matoffice\Tools\) can be used to see a short description for the keys of a certain routine. The right mouse button toggles to a window

with more details. For use in new m-files the call is copied to the clipboard and can be pasted where needed.