

Mozilla Configuration with mozptch

Table of Contents

Mozilla Configuration with mozptch.....	1
The program mozptch.exe.....	1
 License.....	1
 Quick getting started.....	2
 The differences between handling mozilla, firefox and thunderbird.....	2
 Starting mozptch.....	2
 Logging.....	3
 The program flow	
 or: what does the program in which sequence.....	3
 The configuration file.....	4
 Structure.....	4
 Block 1: program flow control.....	4
 Block 2: Variables.....	5
 Block 3: prefs.js entries.....	7
 Block 4: mime types helper applications.....	8
 Example configurations.....	9

If mozilla should be used in Organizations as standard browser and mail client, at first you need the possibility, of a non interactive, automatic installation. All you need in this case, you will find on the demo cd from opsi.org or on unattended.sourceforge.net.

But this is only the first step. In the second one you have to configure and personalize the installations. For this purpose we developed the program mozptch.

The main features of mozptch are:

- You can configure in which path mozilla has to place its profile directory. So it is possible to place it on your central file server.
- You can define the security rules for mozilla
- You have the possibility for a non interactive and fully automatic personalization. You can also choose interactive and semi automatic way.
- You can widely configure where the program has to get together for the needed informations for configuration and personalization
- You may choose special applications for the handling of selected document mime types.
- mozptch handles installations of Mozilla, Firefox and Thunderbird

The program mozptch.exe

License

The program mozpatch.exe is open source software and is published under the terms of the General Public License (GPL).

Quick getting started

1. Download and install your mozptch copy. Don't forget the needed libraries.
2. The sample files are interactive, so that you don't have to configure much the first time. So you can try to start mozptch. But there is a good chance that mozptch will fail, because it don't know where your mozilla is installed. So if it fail, open the configuration file mozptch.ini (or the one you user by the -f argument) and look for %mozprogdire% =c:\programme\mozilla in the section [localDefinedVariable] and correct it. Save the configuration file and start mozptch again. If it still don't start, please send the log file to mozptch@uib.de.
3. If you want to configure thunderbird so please use: mozptch.exe -f tunderbird.ini. If you want to configure firefox so please use: mozptch.exe -f firefox.ini. In both cases it is possible that you fail because (like mentioned above) the Path to the executable file isn't correct.
4. Answer the interactive questions (In case of firefox there will be no questions). mozptch will check them and ask for confirmation before it (re)creates a mozilla profile and patches your prefs.js. (Even if mozptch have messages in English, German and French, the questions will be in English because they are in the configuration file). Check the log file and your prefs.js for the results.
5. The next big step is to make your own configuration file:
 6. configure your mozilla like you need it
 7. make a copy of a sample configuration file to e.g. my_mozptch.ini
 8. cut and paste the content of the prefs.js file to the section [general_prefsjs_entries]
 9. look for entries that have to personified and replace them with a variable
 - 10.user_pref("mail.smtpserver.smtp1.hostname", "%OutgoingServer%");
 - 11.try your configuration file by calling mozptch with -f <path>/my_mozptch.ini
- 12.The final step is to define where to get the needed informations for the variables. A simple test is to get them from the environment.

The differences between handling mozilla, firefox and thunderbird

Most things are very similar at the configuration at these three programs. So mozptch can handle them all. But there are some important differences in the mozptch configuration file. You can see these differences by comparing the sample files. Here are the main differences:

The Path and Name of the executable file (mozilla.exe / firefox.exe)

Name , Path and construction of the registry file. Mozilla uses a binary registry.dat. Firefox >= 0.9 and Thunderbird >=0.8 uses text file named profiles.ini.

Starting mozptch

The program mozptch is a generic mozilla configuration program. It is widely driven by a configuration file.

If the program is started with out any option, it search for a configuration file 'mozptch.ini' in that directory, from where the mozptch.exe was started.

If you start it with the command line option '-f' you can define which configuration file should be used. The start command is in this case:

mozptch.exe -f <path_and_name_of_configuration_file>

With the command line option '-s' (silent) you disable any user interaction. The program tries to configure mozilla according to the used configuration file.

Example 1:

```
c:\programme\uib\mozpatch\mozpatch.exe
```

Mozpatch.exe will use the configuration file mozpatch.ini from the directory
c:\programme\uib\mozpatch\

Example 2:

```
c:\programme\uib\mozpatch\mozpatch.exe -f p:\mail\my_mozpatch.ini
```

Mozpatch.exe will use the configuration file my_mozpatch.ini from the directory P:\mail.

Example 3:

```
c:\programme\uib\mozpatch\mozpatch.exe -s -f p:\mail\my_mozpatch.ini
```

Mozpatch.exe will use the configuration file my_mozpatch.ini from the directory P:\mail and configure mozilla without any user interaction.

Linux examples:

```
/usr/local/mozptch/run_mozptch.sh -f /usr/local/mozptch/firefox_linux.ini
```

```
/usr/local/mozptch/run_mozptch.sh -f /usr/local/mozptch/thunderbird_linux.ini
```

```
/usr/local/mozptch/run_mozptch.sh -f /usr/local/mozptch/mozilla_linux.ini
```

Logging

The Program writes an very detailed log file with will help you to find problems in the configuration process.

Windows:

The log file name is maileinr.log and will be placed in the directory c:\tmp if it exists. If c:\tmp doesn't exist, it will be placed where ever the environment variable TEMP points.

Linux:

The log file name is mozptch.log and will be placed in the directory \$HOME/.mozptch.

The program flow or: what does the program in which sequence

The Program mozptch works in the following steps:

1. opens the log file
2. checks the system language and switch to translated messages if possible.
(default is English)
3. parses the command line for interactive mode (-s Option)
4. opens the configuration file
5. read the Blocks 1 and 2 of the configuration file (see chapter below)
6. checks the collected information for plausibility:

7. - does the profile target directory exist or can it be created
8. - are the mail hosts resolvable
9. - is the OutgoingServer host reachable on port 25 (smtp)
- 10.- verify user name at sendmail (option, see block 1)
- 11.- is the IncomingServer reachable at one of these Ports:
110 (pop3), 143 (imap2), 220 (imap3), 993 (simap4), 995 (spop3)
- 12.if interactive then show user information and asks for editing prefs.js or not
- 13.start programs that have to executed before (if there any, see block 1)
- 14.if there is a old profile (prefs.js) in the target directory and in interactive mode then user information about what will happen to existing profile
- 15.any running instance of mozilla will be killed
- 16.Creates the mozilla profile target path (if it doesn't exists)
- 17.deletes existing xul.mfl file
- 18.deletes existing registry.dat (or profiles.ini) and pluginreg.dat
- 19.In case of mozilla: starts mozilla with the -createProfile Option to create the new profile in the target directory and write this down in a new registry.dat.
In case of Firefox / Thunderbird: writes a new profiles.ini and copies the default profile files from the program directory to the new profile directory.
- 20.if copy_to_profile_from (see block 1) it copy file from this directory to the new profile directory
- 21.after backing up the mimetype.rdf file it will be patched according to configuration block 4
- 22.if the prefs.js file should be patched it will be patched according to configuration block 3
- 23.start programs that have to executed after (if there any, see block 1)
- 24.if interactive then show user information about finished process

The configuration file

Structure

The configuration file is an text file structured like a ini-file. It consists out of four blocks.

The first block (section [general_configuration]) is used for the program control.

The second block (section [variables], [localDefinedVariable], [environment], [inifile-<variable>], [file-<variable>], [ask-<variable>]) is used to configure where the program will find the needed information for organization or user specific data. These data will be used in variables which can be used afterwards.

The third block (section [general_prefsjs_entries], [imap_prefsjs_entries]) is used to hold the pref.js entries which should be used during configuration of mozilla. In these entries you can use the variables form block two.

The fourth block (section [mimetype_helper], [install-<mimetype>]) is used to define helper applications which mozilla should use to open selected mime types.

Block 1: program flow control

In this block (section [general_configuration]) are informations for the program control

defined. There are following configuration possibilities:

- Should the user name verified by the defined sendmail server (verify_user_at_sendmail) ?
- Should be after the user confirmed the start of the mozilla configuration a other program be stated be the configuration take place (execute_before)?
- Should after the configuration is finished a other program be started (execute_after)?
- Should after the creation of the profile and before the configuration of pref.js and mimetype.rdf some file be copied to the to the user profile directory (copy_to_profile_from) ?
- There is a switch checkuser_imap_folder which is custom specific and not commented here. It should be set to false normally.

Example for block 1

```
[general_configuration]
// should we try to verify the user name at the sendmail daemon
// this sendmail feature is often disabled because of security
verify_user_at_sendmail=true
//should we test the existence of imap folders in the users home directory
// normally: false
// for hupsi: true
checkuser_imap_folder=false
// if you want to start a program after the user confirmed to go on
// and before we start so give here the full path and options
// eg. c:\utils\winst32.exe /batch mywinstscript.ins
// otherwise say no
execute_before=no
//execute_before=c:\utils\winst32.exe /batch P:\mail\startbefore.ins
// if you want to start a program after everything is done
// so give here the full path and options//eg. c:\utils\winst32.exe /batch mywinstscript.ins
// otherwise say no
//execute_after=no
execute_after=c:\utils\winst32.exe /batch P:\mail\startafter.ins
// If you want to copy some files to the new profile directory
// give here the the path to copy from (source path)
// otherwise say no
copy_to_profile_from=p:\install\mozilla\mozdir\mozmail\files
```

Block 2: Variables

For configuring mozilla the program mozptch needs some system wide and some user specific informations. Examples here fore are the system wide value path to mozilla.exe and the values used for personalization of the mail client like user name host names for smtp,

pop or imap. In the section [variables] are defined a lot of variables and it is described witch method should be used at runtime to get the value of each variable. Possible values for the method are environment variables, files, ini-files, constants and interactive input dialogs. Based on this flexibility in gathering the information to gather, in most cases you will be able to find all (or nearly all) informations needed at run time. The variables will be resolved in exactly that sequence that used for defining the variables in the Examples below. This is important because it is possible to use a variable that is defined on top for defining a variable that follows later. Important: changing the sequence of the variables in the section does not (!) change the the sequence the program mozptch will resolve them.

The definition of a variable needs two steps. In the first step (in the section [variables] it is defined which method should be used. possible values are:

- environment= from environment variable like defined in section [environment]
- localDefinedVariable= in section [localDefinedVariable] defined
- inifile= from inifile like in section [inifile-<variablename>] defined
- file=content of file that is in section [file-<variablename>] defined
- ask=interactive dialog with message like defined in section [ask-<variablename>]

There is special value "detect=calculated internally". This can and must be used only for the variables: %completemozdir% and %localMozAppdataDir%

In the second step we find for every method mentioned in step one detailed information. An Example shows this:

[variables]

```
%sUname%=environment
%IncomingServer%=environment
%OutgoingServer%=environment
%EmailAddress%=environment
%sHome%=inifile
%replyToAdress%=file
%fullName%=ask
%mozprogrdir%=localDefinedVariable
```

[localDefinedVariable]

```
%mozprogrdir%=c:\programme\mozilla
```

[environment]

```
%sUname%=UNAME
%IncomingServer%=POPHOST
%OutgoingServer%=SMTPHOST
%EmailAddress%=EMAIL
```

[inifile-%sHome%]

```
filename=%HelpVar5%\%sUname%.ini
```

```
section=userdefs
```

```
key=HOME
```

```
[file-%replyToAdress%]
```

```
filename=%completemozdir%\interinf.txt
```

```
[ask-%fullname%]
```

```
text=Please type Your full name
```

```
default=%sUname%
```

Block 3: prefs.js entries

This Section holds the prefs.js entries. They are entered in the exact same format as you find them in the prefs.js. So it is possible to 'cat and paste' them from a existing Installation. In addition you have replace personalized values with the corresponding variables. It is possible to use comments in this section (starting with //). Comments will not be written to the prefs.js.

```
[general_prefs_entries]
```

```
// Place for Mozilla Cache
```

```
user_pref("browser.cache.disk.parent_directory", "%localMozAppdataDir%");
```

```
// Starting mail reply above the old mail...
```

```
user_pref("mailnews.reply_on_top", 1);
```

```
// forward mails as inline (2)
```

```
user_pref("mail.forward_message_mode", 2);
```

```
user_pref("general.startup.mail", true);
```

```
//Preferences/Navigator/Tabbed Browsing/Tab instead Window at middle Mouse click
```

```
user_pref("browser.tabs.opentabfor.middleclick", true);
```

```
//Preferences/Navigator/Start page address:
```

```
user_pref("browser.startup.homepage", "http://mozptch.mozdev.org");
```

```
// *** don't use mozilla.org as first start page
```

```
user_pref("browser.startup.homepage_override.mstone", "rv:1.5");
```

```
user_pref("browser.bookmarks.added_static_root", true);
```

```
// Preferences/Navigator/Downloads/Download close progress dialog when finished
```

```
user_pref("browser.download.progressDnldDialog.keepAlive", false);
```

```
// Preferences/Navigator/Downloads/open progress dialog
```

```
user_pref("browser.downloadmanager.behavior", 1);
```

```
// Now follows the mail account settings
```

```
user_pref("mail.account.account1.identities", "id1");
```

```
user_pref("mail.account.account1.server", "server1");
```

```
user_pref("mail.account.account2.server", "server2");
```

```
user_pref("mail.accountmanager.accounts", "account1,account2");
```

```
user_pref("mail.accountmanager.defaultaccount", "account1");
```

```
user_pref("mail.accountmanager.localfoldersserver", "server2");
```

```
user_pref("mail.smtp.defaultserver", "smtp1");
```

```
//
```

```
user_pref("mail.smtpserver.smtp1.auth_method", 1);
```

```
// Mail & Newsgroups AccountPreferences/outgoing Server/Server name
```

```
user_pref("mail.smtpserver.smtp1.hostname", "%OutgoingServer%");
```

```
// ???
```

```

user_pref("mail.smtpserver.smtp1.try_ssl", 0);
// Mail & Newsgroups accountpreferences/outgoing Server/user name
user_pref("mail.smtpserver.smtp1.username", "%sUname%");
user_pref("mail.smtpservers", "smtp1");
user_pref("mail.root.imap", "%completemozdir%\\imapMail");
// Mail & Newsgroups accountpreferences/Copies & Folders/Drafts folder on
//(Andere) Entwuerfe auf Mail für user@servername
user_pref("mail.identity.id1.stationery_folder", "imap://%sUname%@%IncomingServer%/
Vorlagen");
// Mail & Newsgroups accountpreferences/Copies & Folders/Templates folder on
user_pref("mail.identity.id1.draft_folder", "imap://%sUname%@%IncomingServer%/
Entwuerfe");
// Mail & Newsgroups accountpreferences/Copies & Folders/Place copy in
user_pref("mail.identity.id1.fcc_folder", "imap://%sUname%@%IncomingServer%/
gesendet");
// Mail & Newsgroups accountpreferences/account preferences/E-Mail address
user_pref("mail.identity.id1.useremail", "%EmailAddress%");
// ***
user_pref("mail.identity.id1.valid", true);
// ***
user_pref("mail.identity.id1.smtpServer", "smtp1");
// Mail & Newsgroups accountpreferences/Locale folder/locale folder
user_pref("mail.root.none", "%localMozAppdataDir%\\mozilla\\localmail");
user_pref("mail.server.server2.directory", "%localMozAppdataDir%\\localmail");
user_pref("mail.server.server2.hostname", "Locale folder");
user_pref("mail.server.server2.name", "Locale folder");
user_pref("mail.server.server2.type", "none");
user_pref("mail.server.server2.userName", "nobody");

// Mail & Newsgroups accountpreferences/server preferences/Server name
user_pref("mail.server.server1.hostname", "%IncomingServer%");
// ???
user_pref("mail.server.server1.name", "Mail fÃ¼r %sUname%@%IncomingServer%");
// Mail & Newsgroups accountpreferences/account preferences/reply address
user_pref("mail.identity.id1.reply_to", "%replyToAdress%");
// Mail & Newsgroups accountpreferences/account preferences/your Name
user_pref("mail.identity.id1.fullname", "%fullname%");
//(....)

[imap_prefjs_entries]
// for future use...;-)

```

Block 4: mime types helper applications

```

[mimetype_helper]
// describes entries for inserting help applications in mimetyeps.rdf
// In this section for every application one entry in the style
//name=install
//
// beschreibt Einträge für Hilfsapplikationen zum Eintrag in die mimetyeps.rdf
// In dieser Sektion für jede Applikation ein Eintrag im Stil
// bezeichner=install
//
// For every name ther must be one section [install-name]

```



```
// with the following entries
//
// Für jeden bezeichner eine eigne Sektion [install-bezeichner]
// mit folgenden Einträgen:
//
// mimetype: mimetype
// prospath: path ohne abschließenden slash (path without trailing slash)
// progname: reiner exename (pure name of exe)
// extension: extension, mehrere durch komma separiert (multiple extensions separated by
comma)
mstnef=install
msdocviewer=install
```

```
[install-mstnef]
mimetype=application/ms-tnef
extension=tnf
prospath=C:\Programme\Mozilla\Plugins
progname=fentun.exe
```

```
[install-msdocviewer]
mimetype=application/msword
prospath=C:\Programme\msview\wordview
progname=wordview.exe
extension=doc,rtf
```

Example configurations

[mozilla configuration for windows](#)

[thunderbird configuration for windows](#)

[firefox configuration for linux](#)