

K-Meleon**BuildKMeleonDistribution**

Content
- Building Mozilla
- What You Need to Build Mozilla for K-Meleon 1.5.x to 1.6.x
- Getting a command prompt for building GRE 1.9.1.x
- Getting a command prompt for building GRE 1.8.1.x
- What You Need to Build for K-Meleon 0.9 to 1.1.x
- Unzip the Source
- How to Configure
- Build the Mozilla Source Code
- Building K-Meleon
- Creating the Chrome Package
- Creating the installer

KMeleonWiki > Documentation > BuildKMeleonDistribution

To create a K-Meleon distribution, you need the Gecko engine, K-Meleon's executable file and its plugins, and the chrome files. In addition, K-Meleon's standard distribution uses an installer created using NSIS. This document covers these four sections:

- Building Mozilla
- Building K-Meleon
- Creating the chrome package
- Creating the installer

Building Mozilla

The section is mostly derived from Mozilla's own build documentation, so you should consult that document first if you encounter any problem.

What You Need to Build Mozilla for K-Meleon 1.5.x to 1.6.x**Based on Mozilla Build Documentation for the 1.8.1 and 1.9.1 Branches**

- a PC running a NT based Windows (Win2000, XP or newer) with the Microsoft Service Packs required by the MS Visual C++ Version You want to use
- a hard drive with about 7 GB free space or more
- the Microsoft Visual C++ compilers, tools and their respective updates
- the appropriate Microsoft SDKs that are needed by Your Visual C++ version
- the Mozilla Build package that contains the other software prerequisites needed for building Mozilla. This includes the MSYS build environment (Cygwin) and useful tools. Best use the latest package

Notes:

Install all items in their default location.

Commercial Versions of the VC++ Compilers and Tools by Microsoft

Either VC7.1 (Visual Studio 2003), VC8 (Visual Studio 2005) or VC9 (Visual Studio 2008) can be used to build GRE 1.9.1.x.

Either Microsoft Visual C++ 6 (Service Pack 5), VC7.1 (Visual Studio 2003) or VC8 (Visual Studio 2005) can be used to build GRE 1.8.1.x. The GREs created with MSVC6 and MSVC 7.1 work well with project standard VC 7.1 generated K-Meleon.exe and kplugins. To work with a VC 8 generated GRE You also need to create a K-Meleon.exe with that compiler.

Free Versions of the VC++ Compilers and Tools distributed/sponsored by Microsoft

Free tools that you can use to try how it works.

Visual Studio 2005 Express-Edition additionally SP1

System requirements for VC2005 EE:

- W2K SP4
- XP SP2
- W2k3 SP1
- XP/W2k3 x64
- Vista

Visual Studio 2008 Express-Edition with SP1

System requirements for VC2008 EE:

- XP SP2
- W2k3 SP1
- XP/W2k3 x64
- Vista x86/x64

Microsoft Visual C Toolkit 2003. The article [Visual C++ Toolkit 2003 still in use by Mateusz Loskot](#) gives additional info on what is needed. Most extras come from Microsoft .NET Framework 1.1 which must be installed before Toolkit 2003. The Missing ml.exe, link.exe and other tools ship in the bin directory of Windows Server 2003 Service Pack 1 (SP1) Driver Development Kit (DDK). Default location for the installed files .\Winddk\3790.1830\bin\x86.

Another item You will need to build with free Toolkit 2003 is a batch file to get a command prompt so start compilation with Toolkit 2003 from the mozilla-build environment. Unpack the archive into the folder of any mozilla-build tools version higher than 1.3 and start the MSYS command prompt with start-msvc71 toolz.bat.

The 3 free compilers require that You to install the Microsoft Windows Server 2003 R2 Platform SDK in order to include some libraries and includes.

90-days test version of Visual Studio 2008 Team Suite

Getting a command prompt for building GRE 1.9.1.x

When the prerequisites are set up, launch one of the batch files from the c:\mozilla-build directory:

- start-msvc71.bat (VS 2003)
- start-msvc8.bat (VS 2005)
- start-msvc9.bat (VS 2008)

The batch file will automatically display a MSYS command prompt window with all the needed environment variables set, provided the software requirements were met. To paste commands into this window, you must right-click on the window's title bar to open its context menu.

The following assumes VC8 which is the default compiler/tool for K-Meleon 1.6.

Lets move to partition c:/ and create a directory called c:/projects to work there.

cd c: # go to c:/ which is a directory for bash/MSYS command prompt

mkdir projects # create the directory projects.

cd projects # move into projects

Create a text file with an text editor with the extension bat called clone.bat and write inside:

```
hg clone http://hg.mozilla.org/releases/mozilla-1.9.1/ 191src > .clone.log 2>&1
```

```
cd 191src
```

The two commands will download the needed source into the subfolder C:/projects/191src. The last part of the first line "> .clone.log 2>&1" creates a log so that you can check whether all was done or if you had errors.

Start the batch file via the command prompt:

```
clone.bat
```

Your firewall will prompt You that hg.exe wants to access the net. Allow it. The process will get you the Mozilla source.

Alternatively you can download the latest release source archive and unpack it to ./projects. You can use either the SeaMonkey 2.0.14. or the Firefox 3.5.19 version of the source.

Next You need to download and unpack K-Meleon specific GRE 1.9.1x source files: GRE1.9.1build.7z

Unpack the archive. The files and directories go to:

```
./projects/191src/ # If You pulled the source directly - like described above, I will use this as default
```

```
./projects/Your SeaMonkey_Source_Directory_Name/Mozilla/ # If You use SeaMonkey
```

sources

The archive contains

- ./kmeleon # a subfolder with K-Meleon specific GRE sources needed to build K-Meleon's 1.9.1.x GRE
- patch.bat # a file to start a patch process from the command prompt. You can alternatively write to the command prompt:

```
patch -p1 -b < moz1.9.1.patch > .patch_build.log 2>&1 # start to patch and write
.patch_build.log
```

- moz1.9.1.patch # the actual file that does the patching
- .mozconfig # the file that configures the build parameters
- build.bat # the file that starts the actual build process. You can alternatively write to the command prompt:

```
make -f client.mk build > .GRE_build.log 2>&1 # build and write .GRE_build.log
```

Start ./mozilla-build/start-msvc8.bat

cd c:

cd projects

cd 191src

patch.bat

build.bat

When the command prompt with build.bat stops blinking after an hour or several the 1.9.1 GRE for K-Meleon 1.6 is finished.

You will find the new 1.9.1 GRE in: ./projects/191src/.testVC9/kmeleon/dist/bin Delete all exe files in ./dist/bin. Do a file compare of ./components/ against the ./components folder of K-Meleon 1.6 beta2. All newly created files that have no equivalent in K-Meleon 1.6 beta2 can be deleted. Files that are in K-Meleon 1.6.beta2 ./components and are not newly build are copied to the new GRE.

You will find the components of the SDK to help to compile K-Meleon.exe and kplugins in:

```
./projects/191src/.testVC9/kmeleon/dist/SDK/bin
```

```
./projects/191src/.testVC9/kmeleon/idl
```

```
./projects/191src/.testVC9/kmeleon/include
```

```
./projects/191src/.testVC9/kmeleon/lib
```

Create a: ./projects/mozilla/mozilla/dist/ folder and copy the folders idl, include, lib and sdk/bin into it

After that you create getkm.bat inside ./projects with an editor. Write into the file these 3 lines that get you into K-Meleon CVS, get the newest source code and logout again.

```
cvs -d:pserver:anonymous@kmeleon.cvs.sourceforge.net:/cvsroot/kmeleon login
```

```
cvs -z3 -d:pserver:anonymous@kmeleon.cvs.sourceforge.net:/cvsroot/kmeleon
checkout .
```

```
cvs -d:pserver:anonymous@kmeleon.cvs.sourceforge.net:/cvsroot/kmeleon logout
```

Start the bash command prompt.

cd c:

cd projects

getkm.bat

Press Enter when You are prompted for the password. Allow cvs.exe to access the Internet when your firewall asks You. It will copy the K-Meleon sources to ./projects/K-Meleon.

Trouble shooting:

Info about a fake custom comsupp.lib needed for Toolkit 2003. Or download the fake Comsupp.lib already pre-compiled with VC7.1. Another Firefox thread for build problems with MS Toolkit 2003. Do not forget the Cygwin tools of old have been replaced by the mozilla-build package. The rest of the info is usable when You want to build GRE 1.9.1.x or GRE 1.8.1.x.

Vc 7.1, VC8 Express and VC9 Express cannot build: ac_add_options --enable-jemalloc
Comment the feature out or disable it.

Free tools: Vc 7.1, VC8 Express and VC9 Express. Cannot find altthunk.lib when using Microsoft Platform SDK for Windows Server 2003 R2. Either disable ~ActiveX or patch atlwinn.h as described by Kairo in How to set up a SeaMonkey tinderbox with Microsoft Visual C++ 2005 Express Edition.

Getting a command prompt for building GRE 1.8.1.x

When the prerequisites are set up, launch one of the batch files from the c:\mozilla-build directory:

- start-msvc6.bat (Visual C6)
- start-msvc71.bat (VS 2003)
- start-msvc8.bat (VS 2005)

The batch file will automatically display a MSYS command prompt window with all the needed environment variables set, provided the software requirements were met. To paste commands into this window, you must right-click on the window's title bar to open its context menu.

cd c: # go to c:/ which is a directory for bash

mkdir projects # create the directory projects.

cd projects # move into projects

Either get a last source archive of SeaMonkey 1.1.19 and extract it try to pull the newest source directly.

cvs up -r MOZILLA_1_8_BRANCH client.mk # get the client that makes the magic for this branch

make -f client.mk checkout MOZ_CO_PROJECT=suite > .GRE_checkout.log 2>&1 #
make checkout & write the result to .GRE_checkout.log

make -f client.mk build > c:/mozilla/.GRE_build.log 2>&1 # make the GRE & write a build log into .GRE_build.log in the current folder

Additionally You need a .mozconfig for GRE 1.8.1 like the one that Dorian posted in the development forum.

Place it into Your SeaMonkey source code directory (default c:/mozilla) and start the bash.

cd c:

cd mozilla

make -f client.mk build > .GRE_build.log 2>&1 # make the build and write .GRE_build.log

Continue with Building K-Meleon

What You Need to Build for K-Meleon 0.9 to 1.1.x

Based on Mozilla Build Documentation for the 1.7 and 1.8 Branches

- a PC running Windows 95 or a more recent Windows operating system
- a hard drive with at least 1.5 GB NTFS / 3 GB FAT of free space
- Microsoft Visual C++ 6 (with Service Pack 5) or higher
- Cygwin with at least the following packages installed:
 - ash
 - cvs
 - coreutils
 - diffutils
 - findutils
 - gawk
 - grep
 - make
 - patchutils
 - perl
 - sed
 - unzip

- zip
- Netscape wintools
- Mozilla source code, currently SeaMonkey 1.0.4 is used
- K-Meleon source code, currently at version 1.0

Notes:

- SeaMonkey 1.0.x is based on the Mozilla 1.8.0 branch, which does not support VC8 (VS 2005).
- For those requirements, your total download size will be around 50 MB.
- If you are using Windows 9x, you probably need to give the DOS environment more memory space by running:

command /E:4096

Unzip the Source

Create a directory where everything will go. Say, C:\projects. Create two more in there, one named mozilla and the other kmeleon. Now, unzip the respective source code packages with your favorite archive program in their respective directories.

How to Configure

You'll need a batch file in your projects directory to set up the environment prior to compiling:

```
rem --- Set VCVARS to wherever the MSVC vcvars.bat file is found
set VCVARS=C:\Program Files\Microsoft Visual Studio 6\VC98\Bin\VCVARS32.BAT

rem --- Set MOZ_TOOLS to wherever you have the moztools packaged installed
set MOZ_TOOLS=C:\moztools

rem --- Set CYGWINBASE to wherever cygwin is installed
set CYGWINBASE=C:\cygwin

rem --- Prepend Cygwin path and append moztools path
rem --- This is necessary so that cygwin find is ahead of windows find.exe in the PATH, but cygwin link is after MSVC link
set PATH=%CYGWINBASE%\bin;%PATH%;%MOZ_TOOLS%

rem --- Set MSVC environment vars
call "%VCVARS%"
```

Next, you'll need a .mozconfig file to place in your mozilla source code directory:

```
mk_add_options MOZ_OBJDIR=@TOPSRCDIR@/seamonkey
ac_add_options --enable-application=suite
ac_add_options --with-user-appdir="K-Meleon"

ac_add_options --enable-optimize
ac_add_options --enable-strip
ac_add_options --enable-crypto
ac_add_options --enable-extensions=cookie,wallet,xml-rpc,xmlextras,p3p,pref,transformiix,inspector,universalchardet,t

ac_add_options --disable-debug
ac_add_options --disable-logging
ac_add_options --disable-dtd-debug
ac_add_options --disable-tests
ac_add_options --disable-ldap
ac_add_options --disable-postscript
ac_add_options --disable-jsd
ac_add_options --disable-composer
ac_add_options --disable-activex
ac_add_options --disable-activex-scripting
ac_add_options --disable-installer
ac_add_options --disable-xprint
ac_add_options --disable-mailnews
ac_add_options --disable-xpinstall
ac_add_options --disable-chatzilla
```

For more information on mozconfig options, read this page:
http://developer.mozilla.org/en/docs/Configuring_Build_Options

Build the Mozilla Source Code

Open a command prompt, and go to the projects directory with it. Run your batch file. Next, enter the mozilla directory, and use the following command to start building:

```
make -f client.mk build
```

Depending on the speed of your PC, this might take anywhere from half an hour to several hours.

Building K-Meleon

There is a detailed build instruction in [BuildKMeleon](#). Because you have built Mozilla, you don't need to download the SDK anymore. If you follow the previous section, the Mozilla include files would be in C:\projects\mozilla\seamonkey\dist\include\.

Creating the Chrome Package

Note: This section is outdated.

%kmeleon% is the directory of an existing version of K-Meleon (the binary release).

Copy %kmeleon%\components\txmgr.dll to %mozilla%\dist\bin\components

Copy %kmeleon%\nssckbi.dll to %mozilla%\dist\bin

Rename %kmeleon%\chrome\embed.jar to embed.zip, and unpack it.

Rename %mozilla%\dist\bin\chrome\embed.jar to embed.zip, and unpack it.

Copy the unpacked files from %kmeleon%\chrome\embed (or wherever you put them) into %mozilla%\dist\bin\chrome\embed, overwriting the existing stuff.

Zip %mozilla%\dist\bin\chrome\embed to %mozilla%\dist\bin\chrome\embed.zip, and rename it to embed.jar.

Creating the installer

Not yet written.

[EditThisPage](#) [BackLinks](#) [PageInfo](#) last changed on Fri Dec 9 19:00:50 2011

(c) 2000-2010 kmeleonbrowser.org. All rights reserved.
design by splif.