

- You do not have permissions to view this page - please try [logging in](#).

# Windows build prerequisites

[Redirected from en/Windows Build Prerequisites](#)

- [Edit](#)
- [Watch](#)

## Table of contents

1. [Introduction](#)
2. [Hardware requirements](#)
3. [Software requirements](#)
  - 3.1. [Microsoft Visual C++ \(MSVC\)](#)
  - 3.2. [Microsoft Windows SDK](#)
  - 3.3. [Microsoft DirectX SDK](#)
  - 3.4. [MozillaBuild / Pymake](#)
4. [Opening a Build Command Prompt](#)
  - 4.1.1. [Command Prompt Tips and Caveats](#)
5. [Common problems, hints and restrictions](#)
  - [Tags](#)
  - [Files](#)

[Page Notifications Off](#)

## Table of contents

1. [Introduction](#)
2. [Hardware requirements](#)
3. [Software requirements](#)
  - 3.1. [Microsoft Visual C++ \(MSVC\)](#)
  - 3.2. [Microsoft Windows SDK](#)
  - 3.3. [Microsoft DirectX SDK](#)
  - 3.4. [MozillaBuild / Pymake](#)
4. [Opening a Build Command Prompt](#)
  - 4.1.1. [Command Prompt Tips and Caveats](#)
5. [Common problems, hints and restrictions](#)

Build Instructions:

- [Requirements](#)
- [Source](#)
- [Configuration](#)
- [Build](#)
- [Frequently Asked Questions](#)
- [More...](#)

## Introduction

The Mozilla build process requires many tools that are not installed on most Windows systems. In addition to Visual Studio, you must install MozillaBuild, which is a bundle of software including just the right versions of bash, GNU make, autoconf, Mercurial, and much more.

This document is a guide to the MozillaBuild environment, which works with the following versions of the Mozilla codebase available as Mercurial code repositories:

- [mozilla-central](#)/mozilla-aurora/mozilla-beta/mozilla-release (Firefox 4 and later)
- mozilla-1.9.2 (1.9.2/Firefox 3.6)
- tracemonkey
- [comm-central](#)/comm-aurora/comm-beta/comm-release (Thunderbird 5/SeaMonkey 2.1 and later)
- comm-1.9.2 (Thunderbird 3.1)

It works with the following compilers ("Official" means this is what we use for the builds we release as the Firefox product):

	VC8 (Visual Studio 2005)	VC9 (Visual Studio 2008)	VC10 (Visual Studio 2010)
<b>mozilla-1.9.2/comm-1.9.2</b>			
<b>Gecko 1.9.2</b>	Yes (Official)	Yes	No
<b>Firefox 3.6</b>			
<b>mozilla-central/tracemonkey/comm-central</b>	Yes <sup>1</sup> (Official)	Yes	Yes
<b>Gecko 2 &amp; later</b>			
<b>Firefox 4 &amp; later</b>			
<b>Firefox 13 &amp; later</b>	Yes	Yes	Yes (Official)

For building older versions of the codebase, see the [Windows build prerequisites \(1.9.0/1.9.1 branches\)](#) and the [Windows build prerequisites using cygwin](#). If you are

building very old versions of the Mozilla source code, from the 1.0 branch and earlier, follow the [nmake build instructions](#).

<sup>1</sup>VC8 SP1 (service pack 1) is now required.

## Hardware requirements

Mozilla takes time to build. You need a development-class system:

- A fast processor, ideally with several cores.
- At least 2 GB of RAM, 4 GB or higher recommended.
- 1.5 GB NTFS or 3 GB FAT disk space, or more
- Windows Vista or later recommended (the build should work on Windows XP though).

## Software requirements

The following software needs to be installed for a standard Windows build:

- Microsoft Visual C++ tools
- Microsoft Windows SDK(s)
- MozillaBuild
- Microsoft DirectX SDK (optional)

Mozilla will **not** build if some tools are installed at a path that contains **spaces** or other breaking characters such as pluses, quotation marks, or meta characters. The Visual C++ tools may be installed in a directory which contains spaces (the default install location is preferred).

### Microsoft Visual C++ (MSVC)

Depending on what you're building, you will need to use one of several available versions of Microsoft Visual C++ (MSVC) and/or Microsoft Visual Studio. See the chart at the top of this page to ensure that the version you've selected is compatible with the branch you want to build.

If you want to enable jemalloc, you must be using [Visual Studio 2005 SP1](#), Visual Studio 2008 SP1, Visual Studio 2010 or Visual Studio 2010 SP1. Other versions of Visual Studio will not work.

- Visual C++ 10 is available from Microsoft as part of the free [Visual C++ 2010 Express Edition](#).
- Visual C++ 9 is available from Microsoft as part of the free [Visual C++ 2008 Express Edition with SP1](#) (make sure you get this and not the one without SP1!). You do not need to install the Silverlight runtime or the SQL Server

2008 Express Edition when offered. Alternatively you can install the command-line compiler as part of the Windows 7 SDK, but you are then expected to use the included WinDbg debugger.

- The Visual C++ 8 command-line compiler is included with the Vista SDK, but you are then expected to use the included WinDbg debugger.

## Microsoft Windows SDK

See [Windows SDK versions](#).

## Microsoft DirectX SDK

The DirectX SDK is required for building the [ANGLE](#) GLES-on-D3D9 renderer, which is used on Windows for WebGL rendering. This part of ANGLE is an optional component, and is built if the DirectX SDK is available. Official Firefox builds include ANGLE. As part of the DirectX SDK install, you must install the End-User Redistributable Packages; don't uncheck it in the installer.

Install at least the [June 2010 DirectX SDK](#). The latest DirectX SDK can be found from the [DirectX Developer Center](#).

MSDN claims that MSVC 2005 is not compatible with the June 2010 SDK, but as far as we're concerned it works for release builds. However you cannot use the compiler provided with the Vista SDK to compile debug builds at all, and you need a [hotfix](#) to be able to compile debug builds with the full MSVC 2005.

## MozillaBuild / Pymake

The MozillaBuild package contains the other software prerequisites necessary for building Mozilla. This includes the MSYS build environment, [Mercurial](#), autoconf-2.13, CVS, Python, YASM, NSIS, and UPX, as well as optional but useful tools such as wget and emacs.

[Download the current MozillaBuild package](#).

Note: MozillaBuild includes mercurial.

By default, the package installs to `c:\mozilla-build`. It is recommended to use the default path. Don't use a path that contains spaces. The installer does not modify the Windows registry. Note that some binaries may require [Visual C++ Redistributable package](#) to run.

**Expectation setting:** Note that the "UNIX-like" environment provided by MozillaBuild is only really useful for building and committing to the Mozilla source. Most command line tools you would expect in a modern Linux distribution are not present, and those tools that are provided can be as much as a decade or so old

(especially those provided by MSYS). It's the old tools in particular that can cause problems, since they often don't behave as expected, are buggy, or don't support command line arguments that have been taken for granted for years. For example, copying a source tree using `cp -rf src1 src2` does not work correctly because of an old version of `cp` (it gives "cp: will not create hard link" errors for some files). In short, MozillaBuild supports essential developer interactions with the Mozilla code, but beyond that don't be surprised if it trips you up in all sorts of exciting and unexpected ways.

The MozillaBuild package includes the "make" program that originated on the UNIX and Linux platforms. This version of make does not make use of Windows' multi-core architecture. [pymake](#) is a Python implementation of Make that uses Windows' multi-core architecture and thus significantly speeds up build time. See [Mike Conley's blog post](#) for more information about installing and using pymake.

## Opening a Build Command Prompt

After the prerequisites are installed, launch one of the following batch files from the `c:\mozilla-build` directory:

- `start-msvc8.bat` (VS 2005)
- `start-msvc9.bat` (VS 2008)
- `start-msvc10.bat` (VS 2010)
- `start-msvc11.bat` (VS 2011)

**Even if you are on 64-bit Windows, do not use the `start-msvcX-x64.bat` files** (unless you know what you're doing). Those files are **experimental** and **unsupported**. See the [Build:MozillaBuild For x64 wiki page](#). This will launch a MSYS command prompt properly configured to build one of the aforementioned code bases.

All further commands should be executed in this command prompt window. Read on for some tips and caveats, or continue now with the build steps outlined at [Build Documentation](#) (probably with getting the source code).

### Command Prompt Tips and Caveats

- To paste into this window, you must right-click on the window's title bar, move your cursor to the **Edit** menu and click **Paste**.
- If you have cygwin installed, make sure that the MozillaBuild directories come before any cygwin directories in the search path (use `echo $PATH` to see your search path).
- In the MSYS shell, the c: drive looks like a directory called `c` under the root directory (`/`). So if you want to change to the directory `c:\mydir`, in the

MSYS shell you would use `cd /c/mydir` Note the UNIX-style forward slashes (/) in the prompt instead of the Windows-style backward slashes (\)

- The MSYS root directory is located in `c:\mozilla-build\msys` if you used the default installation directory. It's a good idea not to build anything under this directory. Instead use something like `/c/mydir`.

## Common problems, hints and restrictions

- [Debugging Mozilla on Windows FAQ](#): Tips on how to debug Mozilla on Windows.
- The build may fail if your PATH environment variable contains quotes ("). Quotes are not properly translated when passed down to MozillaBuild subshells. Quotes are usually not needed so they can be removed.
- Prior to Gecko 5.0 (Firefox 5.0 / Thunderbird 5.0 / SeaMonkey 2.2) , your OBJDIR (build directory) cannot be within the MSYS root directory. If you try this, you will see linker failures (for example, if you are building into `/mydir/some/path`, it will complain about being unable to find `mydir:\some\path`).
- The build may fail if you have cygwin installed. Make sure that the MozillaBuild directories (`c:\mozilla-build` and subdirectories) come before any cygwin directories in your PATH environment variable. If this does not help, remove the cygwin directories from PATH, or try building on a clean PC with no cygwin.
- Building with versions of NSIS other than the version that comes with the latest supported version of MozillaBuild is not supported and will likely fail.
- If you're building on VS2005 and the build fails with the error message "C2118: negative subscript" on `win_util.cc`, you need to install the VS2005 Service Pack 1. Furthermore, if you get "LNK1103: debugging information corrupt", try installing this hotfix: <http://support.microsoft.com/kb/949009>
- If you intend to distribute your build to others, you will need to set `WIN32_REDIST_DIR=$VCINSTALLDIR\redist\x86\Microsoft.VC80.CRT` in your `mozconfig` to get the Microsoft CRT DLLs packaged along with the application.
  - Prior to Gecko 9 (FF 9, SM 2.6, TB 9) , this was needed only if you had not built with `--enable-jemalloc`.
- The [Microsoft Antimalware service](#) can interfere with compilation, often manifesting as an error related to `conftest.exe` during build. To remedy this, add at least your object directory to the [exclusion settings](#).
- If you encounter an error that `atlbase.h` cannot be found and you have installed Visual Studio 10 Express together with the platform SDK, you may have to delete a registry entry so that `guess-msvc.bat` doesn't detect VC10 installed. The key is `HKLM\SOFTWARE\Microsoft\VisualStudio\10.0\Setup\VC`.
- If you encounter an error like "second C linkage of overloaded function

'\_interlockedbittestandset' not allowed", it happens when `intrin.h` and `windows.h` are included together. Use a `#define` to redefine one instance of the function's name. See more on [using `intrin.h`](#).



**Warning:** If you follow the below advice, your computer will not be protected against attacks that exploit this vulnerability. Make sure you take adequate precautions.

- If you still get random crashes when running MAKE, you may be encountering interference from Windows Security Updates that prevent proper operation of the Bash shell on some Windows systems. In this case, if you're on Windows XP, you will need to uninstall Windows XP Security Updates KB933729 and KB970238 using Add or Remove Programs from the Control Panel. The first of these Security Updates is also incorporated into Windows XP Service Pack 3 (KB936929), so if you have already installed SP3 you'll need to uninstall it and then make sure that Service Pack 2, including all Windows updates (EXCEPT KB933729, KB936929 and KB970238) get installed. To prevent these updates from being installed automatically, select "Notify me but don't automatically download or install them" in the Control Panel's "Automatic Updates" dialog. Then de-select them when any of these items appear in the list of recommended updates, and when asked, indicate that you don't want to be asked about these de-selected updates in the future.
- Parallel builds (-jN) do not work with GNU make on Windows. You need to use [pymake](#) instead.

[Return to the Build Documentation](#)

## Tags (6)

### [Edit tags](#)

- [Build documentation](#)
- [Developing Mozilla](#)
- [NeedsEditorialReview](#)
- [NeedsMarkupWork](#)
- [NeedsTechnicalReview](#)
- [NeedsUpdate](#)

## Attachments (0)

## Images 0

No images to display in the gallery.

Attach file

Page last modified 14:58, 20 Apr 2012 by dbolter

---



© 2012 Mozilla Developer Network

Content is available under [these licenses](#) • [About MDN](#) • [Privacy Policy](#) • [Help](#)

- [Sign in](#)

### What's this?

MDN has switched to [BrowserID](#), a safe and simple way to sign in with just your e-mail address. [Learn more about why Mozilla is using BrowserID](#).

**Returning members:** sign in with BrowserID and you'll be connected to your MDN profile (all your information is still here).

**New members:** sign in with BrowserID first, then you'll be able to set up your new MDN profile.

[Sign in](#)