MathCode C++
Installation instructions and administration

MathCode C++ Version 1.4.4, March 1, 2011

1 Important to remember

When you start MathCode:

Remember to **close all Mathematica sessions** before using MathCode.

Remember to use always the **MathCode C++ button** available in your Start Menu. This button starts a batch file which sets up correct environment for your C++ compiler, and then starts Mathematica.

2 Installation step by step

Please follow these steps for successful MathCode C++ installation.

1.1 Check your Mathematica and Windows version

- Mathematica 6.0, 7.0, 8.0 are supported.
- Windows 2000, XP Pro and XP Home, Vista, Windows7 are supported.

During the installation you will need to specify the release number and location of *Mathematica*. You should select the *Mathematica* directory manually. This should be the directory containing `MathKernel.exe`.

1.2 Determine your $MachineID

The $Machine ID is needed for registration. It is the identity of the machine you want a license for. To find out your $Machine ID, evaluate the following in *Mathematica*:

```
$MachineID
```

1.3 Obtain license key for purchased license

You should register to get a key file that will enable you to use the software. If you purchased the software you can register it online at the following URL:

`http://www.mathcore.com/register.html`

Please do not use this page for demo(trial) licenses, please use

`http://www.mathcore.com/products/mathcode/`

instead.

When you start installaion of MathCode you can click the button **Register** to register your software.

Within two business days you should receive an e-mail with the key file attached. Save the attachment to a file. Remember where you saved it; you will need to select this location during MathCode C++ installation.

1.4 Obtaining license key for demo(trial) license

You apply for demo(trial) license using online demo request form at
http://www.mathcore.com/products/mathcode/
and click on **Download Trial version**

When you start installation of MathCode you should not click the **Register** button to register your software.

Within two business days you should receive an e-mail with the key file attached. Save the attachment to a file. Remember where you saved it; you will need to select this location during MathCode C++ installation.

1.5 Check for the latest release

Since MathCode relies on many other software products that often change their versions and properties please **always download the latest version** from the address you get from us together with your key file; currently it is
http://www.mathcore.com/products/mathcode/download/downloadframe.shtml

1.6 Chose a compiler

If you use Mathematica 8.0 then you must install **MicroSoft Visual C++ 2010 Express Edition and platform SDK, see Section 1.7.**

If you use Mathematica 6.0 or 7.0 then MathCode already includes a C++ compiler for you..

1.7 Install MicroSoft Visual C++ 2010 Express Edition

MathCode already includes a C++ compiler that works with Mathematica 6.0 and 7.0.

However if you have to use another free C++ compiler, MicroSoft Visual C++ 2010 Express, you should install it before Mathcode installation.

Currently the toolkit is available here:

http://www.microsoft.com/express/download/#webInstall

1.8 Install MicroSoft Visual C++ 2008 Express Edition (optional)

MathCode already includes a C++ compiler that works with Mathematica 6.0 and 7.0.

However if you have to use another free C++ compiler, MicroSoft Visual C++ 2008 Express, you should install it before Mathcode installation.

1.9 Start setup

Run the program you downloaded or found on a CD and follow the on-screen instructions. The Section 1.10 explains how the setup system determines which C++ compiler you will use. Use the **MathCode C++** button in **Start menu** in order to start MathCode C++.

1.10 Choose which C++ compiler you will use.

The setup system asks you to select a compiler:Currently these compilers are suggested:

(A) MingW32 1.1 (installed together with MathCode)
(B) MinfW32 (lookup manually)
(C) Microsoft Visual Studio 2005
(D) Visual C++ 2005 Express
(E) Visual Studio 2008
(F) Visual Studio 2010
(G) CygWin 1.5.2 (or earlier) gcc
(H) CygWin 1.7 (or later) gcc

If Microsoft Visual Studio 2005 is installed on your computer, the "MicroSoft Visual C++ 2005 Express" is disabled because parallel installation of these tools is not supported.

1.10.1 Mingw32

Mingw32 is free and included in MathCode installation setup automatically. It should be used if you have no MicroSoft Visual C++ installed. It is not compatible with Mathematica 8.

1.10.2 MicroSoft Visual C++

If you have MicroSoft Visual C++ installed in your machine it will be detected.

1.10.3 MicroSoft Visual C++ .NET 2003 toolkit

This tool is detected or not detected depending on your specific setup of this tool. It cannot be detected automatically if it has been installed by a different user.

1.10.4 Advanced case: compiler is installed but it is not detected during setup.

During installation you have to click on the Browse button and browse to any other existing directory on your disk. A batch file should exist in this directory. This match file should set all necessary environment variables for your compiler.

1.10.5 Advanced case: if you want to change the C++ compiler after MathCode C++ installation

Please re-install MathCode C++. The setup will detect a new compiler automatically.

1.10.6 Advanced case: if you want to change the C++ compiler without re-installation.

Execute **Configurator.nb** in Mathematica. When you choose a compiler name there, it will append compiler definition string to the end of **MathCodeConfig.m**. These strings are explained in Section 2.2.

Edit **StartMathCode.bat** and modify the path of the batch file which set up correct environment variables for the new compiler.

If you did not peform these operations correctly you will typically get linking errors during compilation of your C++ code into executable.

1.11 Always use the MathCode button in Start Menu

Remember to always use the **MathCode C++ button** available in your Start Menu. This button starts a batch file which sets up correct environment for your C++ compiler, and then starts Mathematica.

If you start Mathematica without using this button, you will typically get **errors** during compilation phase since the C++ compiler will not be found.

2 Details on manual configuration of C++ compiler

2.1 C/C++ compilers on Windows

The GNU compilers

The port of EGCS to Windows used by *MathCode* is called "MinGW32" which is an acronym for "Minimalist GNU on Win32". Compiled code depends only on libraries that are always provided by Windows, so generated executables can run on any Windows computer. This is *not* the latest release of MinGW32.

Detailed information about MinGW32 can be found on the following URL:

```
http://www.xraylith.wisc.edu/~khan/software/gnu-win32/
```

or

```
http://www.mingw.org
```

Specifying Visual C++ as the default compiler

After the installation, you should open Configurator.nb in Mathematica and execute whole notebook. All possible C++ complers will be suggested. When you choose one, MathCode is re-configured for use of your chosen compiler. In this process several MathCode library files are copied and the file `MathCodeConfig.m` in the main *MathCode* directory is updated. A new line such as

```
DefaultCompiler["C++"->"mingw32"]
```

or

```
DefaultCompiler["C++"->"vc60"]
```

is added to it.

For more information about `DefaultCompiler[]` see Section 2.2.

Using a different unsupported compiler

The precompiled object files distributed are compiled using MingW32 version and Visual C++ versions mentioned above. If you wish to use another compiler, you need to recompile the array library and re-generate code for the standard packages.

Recompiling the array library

1. Open a command prompt
2. go to the directory `MathCode\lib\lightmat\src`
3. Edit `compwin.bat`. Select there a compiler that matches best your choice,duplicate the entry for it and adjust the Makefile for it if necessary.
4. type `compwin.bat`

This will recompile the array library for your current version. You should also change the default compiler as described above.

Regenerating code for standard packages

Open the notebook `lib/stdpackages/src/System.nb` in *Mathematica*. Select "Evaluate Notebook" from the "Kernel" menu. This will regenerate compiled code for this package.

2.2 More on compiler definitions

The file `MathCodeConfig.m` in the main *MathCode* directory controls the *MathCode* runtime configuration. This file is really a *Mathematica* package that contains some configuration directives; currently `DefineCompiler[]` and `DefaultCompiler[]`.

`DefineCompiler[]` is used to associate a symbolic compiler name (a string) with a make file, a command template, and a build command. You don't normally need to bother with these details.

`DefaultCompiler[]` is used to select the default compiler definition for a language. Currently the only language supported for code generation is C++. In `MathCodeConfig.m` you might find a line

```
DefaultCompiler["C++"->"mingw32"];
```

This tells *MathCode* to use the included `"mingw32"` compiler definition when generating C++ code. If you wish to use Visual C++ instead (assuming you are on the Windows platform), you should change this to read:

```
DefaultCompiler["C++"->"vc60"];
```

If there are several `DefaultCompiler` definitions, the last one is taken into account.

3 License management

3.1 What are licenses?

For each machine you wish to run *MathCode* on, you should obtain one key file containing the license. *MathCode* uses the same MathID as *Mathematica* does to distinguish between machines. A key file is a text file containing a mix of letters and digits. Key files should be put into the `Licensing` subdirectory of the *MathCode* installation. The names of the key files do not matter.

3.1.1 Adding a license

When you register for a new *MathCode* license, you will receive a file that should be put in the `Licensing` subdirectory of your *MathCode* installation.

3.1.2 The license index file

MathCode will use an index file `index.m` in the `Licensing` directory to speed up license lookups. If a new license is added, `index.m` is rebuilt automatically as needed.

If you experience problems with the licensing, you can remove the `index.m` file, forcing *MathCode* to rebuild it on the next license check.

For a site installation, users might not have write permissions to the `Licensing` subdirectory. In this case, the system administrator should rebuild the index file by evaluating the following in *Mathematica*:

```
Needs["MathCode`"];
RebuildIndex[ToFileName[{$MCRoot,"Licensing"}]];
```

If `index.m` didn’t exist, you will se an error message about opening it. This error message can safely be ignored.