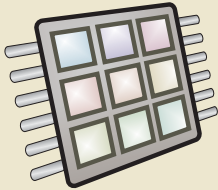




Universität Karlsruhe (TH)
Research University · founded 1825

Institute for Program Structures
and Data Organization (IPD)



Multicore Software Engineering Young Investigator Group

HowTo-Guide

Getting started with OpenMP

2007-10-24

Victor Pankratius (Editor)
Wolfgang Schnerring

<http://www.multicore-systems.org/research/>

© University of Karlsruhe, Germany

All rights reserved. The authors and editors have taken great care in the preparation of this report, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information contained therein.

1 Getting started with OpenMP

OpenMP (Open Multi-Processing) is an API for shared memory multiprocessing programming in C/C++ and Fortran. This section describes how to set up your environment to be able to compile programs like the OpenMP-“Hello world” example shown in figure 1.

```
#include <omp.h>
#include <stdio.h>

int main(int argc, char* argv[]) {
    int id;
    #pragma omp parallel private(id)
    {
        id = omp_get_thread_num();
        printf("%d: Hello World!\n", id);
    }
    return 0;
}
```

Figure 1: A hello-world program using OpenMP.

1.1 Linux

GCC supports OpenMP since version 4.2, simply install it using your distribution’s package manager. You should now be able to compile programs using OpenMP e. g. with

```
$ gcc -fopenmp -o hello hello-openmp.c
```

1.2 Windows

First install the POSIX thread library as described in “Getting started with POSIX threads”¹

Download GCC 4.2 from the MinGW project <http://www.mingw.org/>, you need `gcc-core-4.2.xyz.tar.gz` and optionally `gcc-g++-4.2.xyz.tar.gz` if you want to use OpenMP with C++. Unpack the archives into your MinGW root directory.

Edit `$MINGW/lib/gcc/mingw32/4.2.1-sjlj/libgomp.spec` to contain

```
*link_gomp: -lgomp -lpthreadVC2
```

instead of the original contents.

You should now be able to compile programs using OpenMP, e. g. with

```
C:\>gcc-sjlj.exe -fopenmp -o hello hello-openmp.c
```

1.3 Eclipse

Create a new “Managed Makefile” project.

Open the Properties dialog from the Project menu. In the section C/C++ Build add `-fopenmp` to the Miscellaneous flags for GCC C++ Compiler, GCC C Compiler, and GCC C++ Linker as shown in figure 2.

¹available at <http://www.multicore-systems.org/research/howto.html>

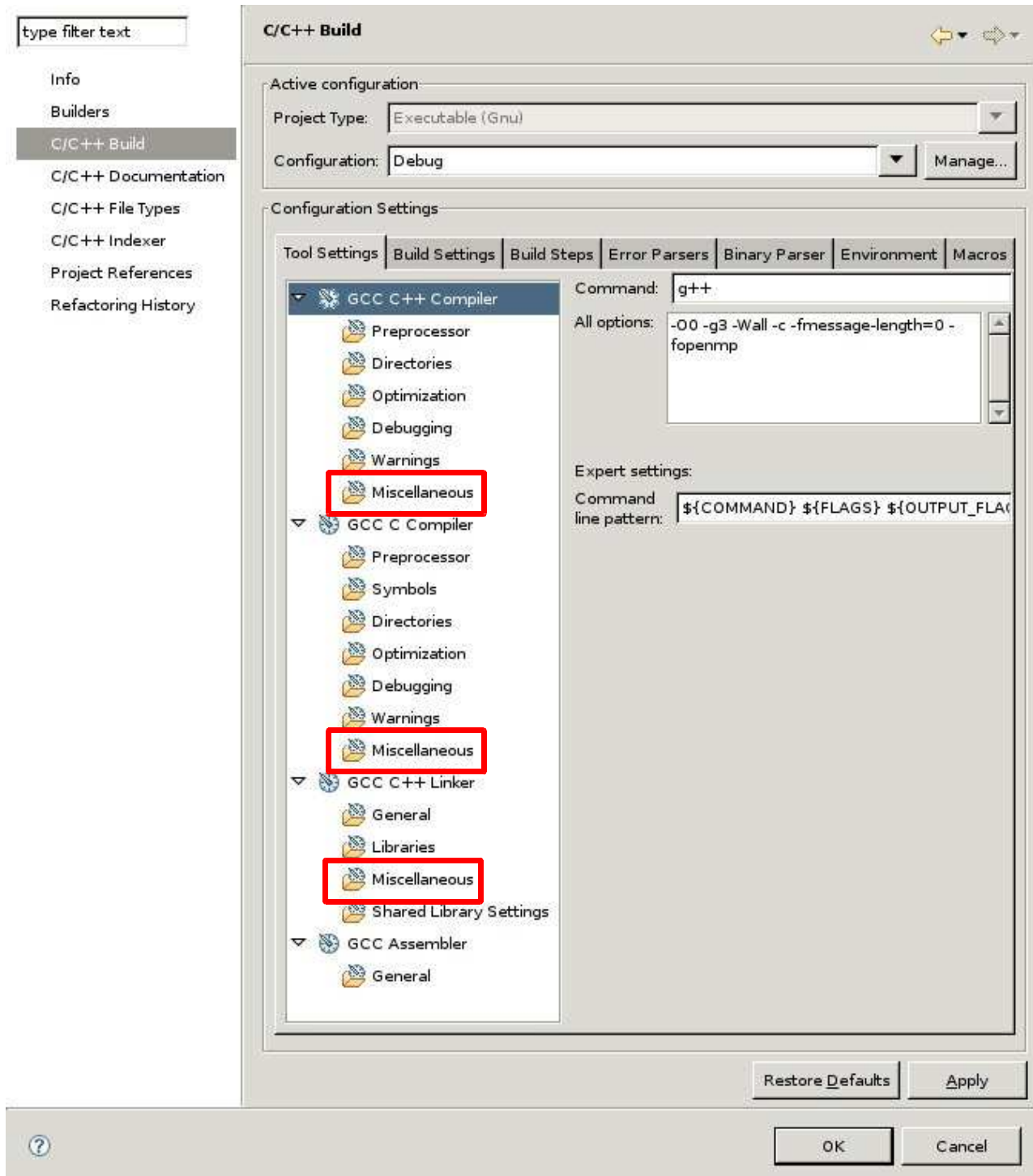


Figure 2: Setting compiler flags for OpenMP in Eclipse.