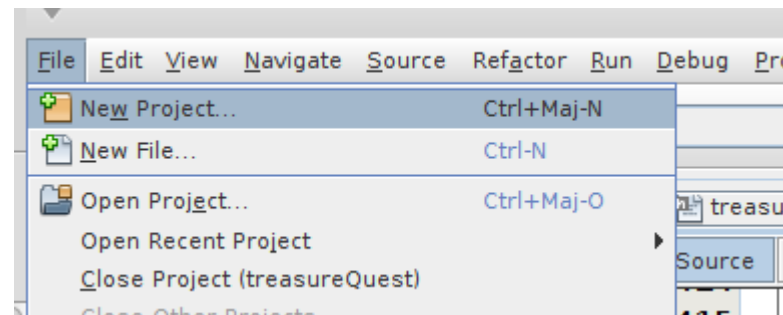
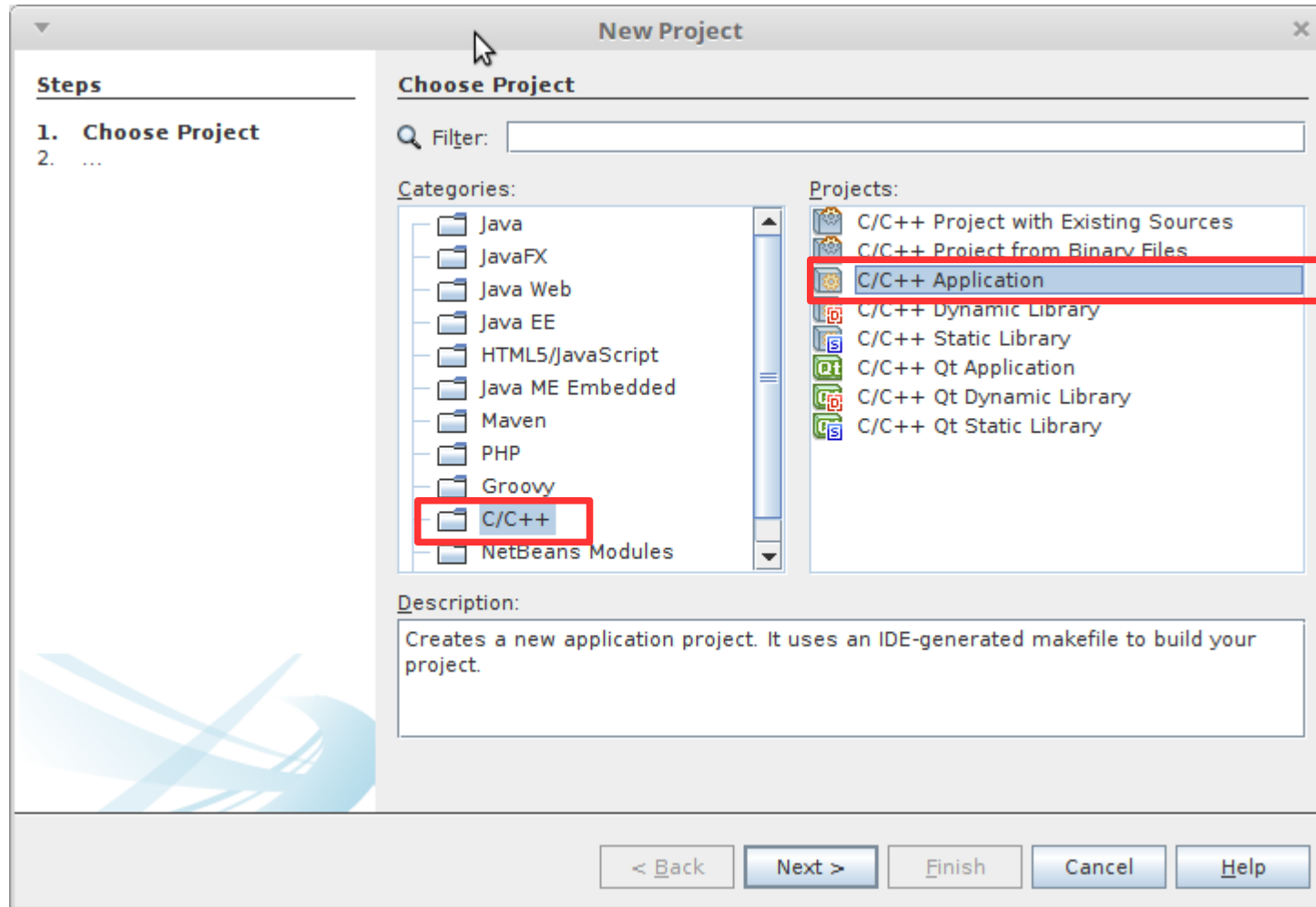


Première utilisation de netbeans pour le développement d'application en langage C

Cliquez sur File>New Project



Choisissez C/C++ dans Categories puis C/C++ Applications dans Projects



Nom du projet

Emplacement
du projet

The image shows a 'New C/C++ Application' dialog box. On the left, a 'Steps' panel lists '1. Choose Project' and '2. Project Name and Location'. The main area is titled 'Project Name and Location' and contains several input fields: 'Project Name' (with 'CppApplication_1'), 'Project Location' (with '/tmp' and a 'Browse...' button), 'Project Folder' (with '/tmp/CppApplication_1'), 'Project Makefile Name' (with 'Makefile'), a checked 'Create Main File' checkbox with 'main' in the adjacent field, 'Build Host' (with 'localhost'), and 'Tool Collection' (with 'Default (GNU (GNU))'). A language dropdown menu is open, showing 'C' selected. At the bottom are buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. Three yellow callout boxes with arrows point to the 'Project Name' field (labeled 'Nom du projet'), the 'Project Location' field (labeled 'Emplacement du projet'), and the 'C' option in the language dropdown (labeled 'Choisir C').

Steps

1. Choose Project
2. **Project Name and Location**

Project Name and Location

Project Name: CppApplication_1

Project Location: /tmp Browse...

Project Folder: /tmp/CppApplication_1

Project Makefile Name: Makefile

☒ Create Main File main C

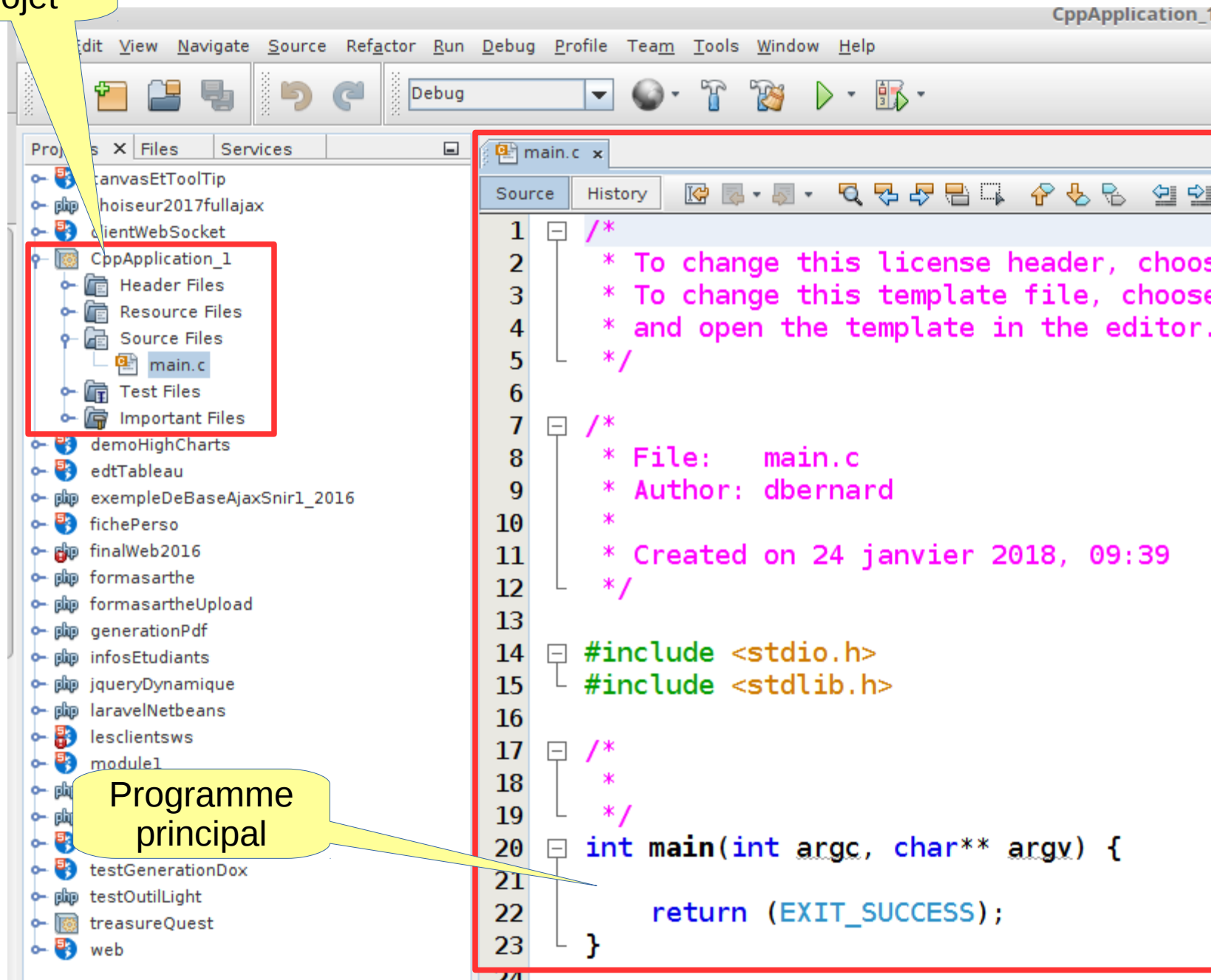
Build Host: localhost

Tool Collection: Default (GNU (GNU))

< Back Next > Finish Cancel Help

Choisir C

Arborescence
du projet



Compiler
le programme

The screenshot shows the NetBeans IDE interface. The top menu bar includes Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The toolbar contains icons for file operations, running, and debugging. The 'Debug' button, represented by a hammer icon, is highlighted with a red rectangle. The left sidebar shows a project tree for 'CppApplication_1' with folders for Header Files, Resource Files, Source Files (containing 'main.c'), Test Files, and Important Files. The main editor window displays the source code of 'main.c' with line numbers 1 through 23. The code includes a license header, preprocessor directives for `<stdio.h>` and `<stdlib.h>`, and a `main` function that returns `EXIT_SUCCESS`. The bottom output window, titled 'Output - CppApplication_1 (Build)', shows the execution of the `make` command, including directory creation, compilation of `main.o`, and linking of the final executable. The output concludes with 'BUILD SUCCESSFUL (total time: 1s)'.

Messages du compilateur

```
1  /*
2  * To change this license header, choose
3  * To change this template file, choose
4  * and open the template in the editor.
5  */
6
7  /*
8  * File:   main.c
9  * Author: dbernard
10 *
11 * Created on 24 janvier 2018, 09:39
12 */
13
14 #include <stdio.h>
15 #include <stdlib.h>
16
17 /*
18 *
19 */
20 int main(int argc, char** argv) {
21
22     return (EXIT_SUCCESS);
23 }
```

```
cd '/tmp/CppApplication_1'
/usr/bin/make -f Makefile CONF=Debug
"/usr/bin/make" -f /tmp/CppApplication_1/project/Makefile-Debug.mk QMAKE= SUBPROJECTS= .build-conf
make[1]: on entre dans le répertoire « /tmp/CppApplication_1 »
"/usr/bin/make" -f /tmp/CppApplication_1/project/Makefile-Debug.mk dist/Debug/GNU-Linux/cppapplication_1
make[2]: on entre dans le répertoire « /tmp/CppApplication_1 »
mkdir -p build/Debug/GNU-Linux
rm -f "build/Debug/GNU-Linux/main.o.d"
gcc -c -g -MMD -MP -MF "build/Debug/GNU-Linux/main.o.d" -o build/Debug/GNU-Linux/main.o main.c
mkdir -p dist/Debug/GNU-Linux
gcc -o dist/Debug/GNU-Linux/cppapplication_1 build/Debug/GNU-Linux/main.o
make[2]: on quitte le répertoire « /tmp/CppApplication_1 »
make[1]: on quitte le répertoire « /tmp/CppApplication_1 »

BUILD SUCCESSFUL (total time: 1s)
```

Exécuter
le programme

