


# Développer en C ou C++ avec Eclipse

Par Laurent Barbareau 

Date de publication : 1 octobre 2012

Dernière mise à jour : 3 avril 2013

DÉBUTANT

Cet article a pour objectif de vous aider à mettre en place un environnement de développement C/C++ avec Eclipse. Il s'adresse aux débutants en programmation C/C++ et à ceux qui commencent leur prise en main de cet  **EDI** qu'est Eclipse. À la fin de cet article, vous aurez l'environnement requis pour parvenir à compiler et à déboguer vos programmes C ou C++, tout en disposant de l'éditeur de code source C/C++ et de l'ensemble des fonctionnalités offertes par la plate-forme Eclipse.

Bien que cet article soit davantage utile aux utilisateurs développant sous Windows (du fait de l'absence d'outils C/C++ installés en standard sur celui-ci), les utilisateurs Linux et Mac OS X pourront également en tirer profit.

Pour réagir à ce support de cours, un espace de dialogue vous est proposé sur le forum

I - Introduction.....	3
II - Installer et Configurer Eclipse.....	3
II-A - Installer Eclipse à l'aide du package dédié au développement C/C++.....	3
II-B - Ajouter le plug-in CDT à une installation d'Eclipse existante.....	4
III - Installation de l'outillage C/C++ sous Windows.....	4
III-A - Installation de MinGW.....	5
III-B - Installation de Cygwin.....	7
IV - Vérification du fonctionnement dans Eclipse.....	11
IV-A - Création d'un Projet C/C++.....	12
IV-B - Exécution en mode Debug.....	13
IV-C - Ajuster le paramétrage d'un Projet C/C++.....	15
V - Conclusion.....	17
VI - Liens utiles.....	17

## I - Introduction

Le développement en C/C++ avec Eclipse requiert le  **plug-in officiel CDT**, ainsi qu'un compilateur C/C++, l'outil make et un débogueur.

Sous **Linux**, il est probable que ceux-ci soient déjà installés. Dans le cas contraire, il vous faudra les installer par l'intermédiaire du package **GNU Toolchain**, lequel doit être disponible au sein de votre distribution Linux.

Sous **Mac OS X**, si nécessaire, vous trouverez le **GNU Toolchain Apple**, dans l'**EDI Xcode**.

Sous **Windows**, par défaut il n'y a pas ce genre d'outils, il faut donc se charger de les installer. C'est ce que nous verrons après la configuration d'Eclipse.

## II - Installer et Configurer Eclipse


En premier lieu, il faut rassembler les éléments nécessaires en ce qui concerne Eclipse.

Deux choix possibles :

- 1 Partir de zéro et installer le package Eclipse, pré-assemblé avec le plug-in CDT.
- 2 Ajouter le plug-in CDT à une installation d'Eclipse existante.

Quelle que soit l'option choisie, vous disposerez à l'issue des mêmes composants vous permettant de développer en C/C++ sous Eclipse.

### II-A - Installer Eclipse à l'aide du package dédié au développement C/C++

Eclipse est un  **EDI** qui permet l'ajout de toutes sortes de plug-ins, notamment CDT, spécifiquement conçu pour le développement C/C++. Ainsi, si vous disposez déjà d'une installation d'Eclipse ne comportant pas encore ce plug-in, l'ajouter sera en principe aisé et devrait suffire à vous permettre de développer en C/C++.

Pour autant, faire le choix de partir de zéro en installant le package Eclipse pré-assemblé pour le C/C++, permet de commencer sur des bases saines et dans des conditions optimales. De surcroît, « cloisonner » ses développements en ayant une version d'Eclipse pour le développement C/C++ et une autre pour un autre langage ou une autre technologie, reste un choix plutôt judicieux. Il n'est pas rare en effet que le cumul de plug-ins apporte son lot de petits désagréments. Donc, à partir du moment où vous disposez de suffisamment de mémoire RAM (1Go par instance d'Eclipse est en général confortable) et que vos projets dans différentes technologies ne sont pas liés, il est bon d'avoir plusieurs versions/packages d'Eclipse avec des *workspaces* séparés.

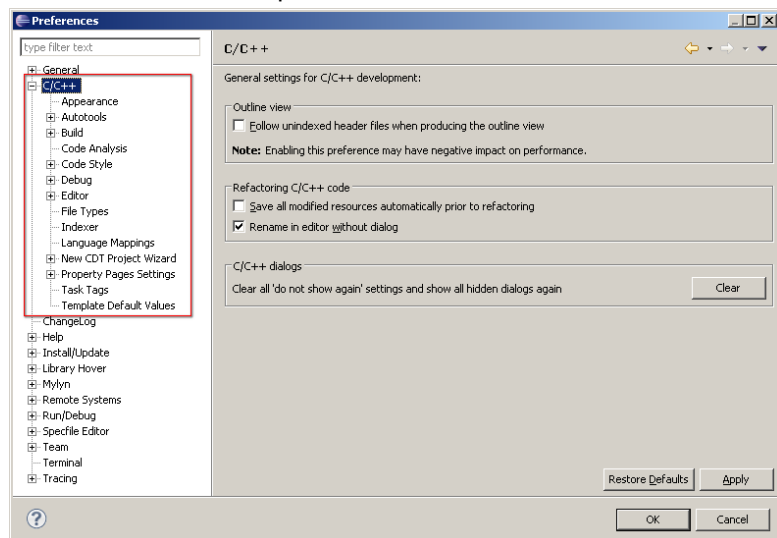
Donc, pour partir d'une version d'Eclipse pré-assemblée pour le développement C/C++, il faut télécharger le package **Eclipse IDE for C/C++ Developers**, disponible sur cette page <http://www.eclipse.org/downloads/>.

L'installation de celui-ci se résume ensuite à décompresser le fichier récupéré à un endroit approprié (de grande préférence proche de la racine d'un disque), avec comme seul pré-requis le fait de disposer d'une *machine virtuelle Java*, sans laquelle Eclipse ne pourra pas démarrer.

Pour de plus amples explications, vous pouvez consulter le lien suivant, afin de vous faciliter son installation et son exécution : [FAQ Comment installer Eclipse sur ma machine ?](#).








## II-B - Ajouter le plug-in CDT à une installation d'Eclipse existante

Si vous disposez d'une installation d'Eclipse pour laquelle vous ignorez si elle est déjà équipée du plug-in CDT, un moyen rapide de le vérifier consiste à contrôler la présence de la rubrique C/C++ au niveau du menu **Window** > **Preferences** dans Eclipse.



Si vous disposez d'une installation d'Eclipse qui ne contient pas encore le plug-in CDT, alors suivez les étapes suivantes :

- 1 Allez dans le menu **Help** > **Install New Software...**
- 2 Dans la liste **Work with**, sélectionnez l'*update-site* correspondant à votre version, c'est-à-dire :  
[EclipseVersion - <http://download.eclipse.org/releases/eclipseversion>]  
 Par exemple, pour **Eclipse Indigo (3.7)**, vous aurez l'*update-site* suivant :  
[Indigo - <http://download.eclipse.org/releases/indigo>]
- 3 Une fois le bon *update-site* sélectionné, vous devez cocher les plug-ins concernant le C/C++. Au minimum, prenez soin de sélectionner ceux-ci :

<input checked="" type="checkbox"/>		Programming Languages	
<input checked="" type="checkbox"/>		Autotools support for CDT (Incubation)	3.0.1.201202152032
<input checked="" type="checkbox"/>		C/C++ Development Tools	8.0.2.201202111925
<input type="checkbox"/>		C/C++ Development Tools SDK	8.0.2.201202111925
<input checked="" type="checkbox"/>		C/C++ Library API Documentation Hover Help (Incubation)	0.3.0.201202152032
<input type="checkbox"/>		C/C++ Library API Documentation Hover Help (Incubation)	0.3.0.201108301805
<input type="checkbox"/>		CDT Visual C++ Support	1.0.0.201202111925

- 4 Cliquez sur **Finish** et suivez les instructions jusqu'au redémarrage d'Eclipse.

Une fois Eclipse redémarré, vous pouvez d'ores et déjà créer des projets C/C++ et ouvrir les fichiers source dans l'éditeur de code C/C++. Cependant, si votre système ne comporte pas encore de **compilateur**, outil **make** et **débogueur**, vous ne pouvez pas compiler et exécuter vos programmes. Le cas échéant, vous devez installer ces outils, parallèlement à Eclipse, comme indiqué aux chapitres suivants.

## III - Installation de l'outillage C/C++ sous Windows

Eclipse et son plug-in CDT n'étant pas suffisants pour développer des programmes en C/C++, il convient de compléter ceux-ci avec les outils permettant de produire des programmes exécutables.

Pour cela, il y a notamment **MinGW** et **Cygwin**, lesquels rassemblent les éléments permettant d'obtenir une chaîne de compilation cohérente et complète. Ceux-ci embarquent tous deux le package **binutils** lequel contient entre autres

l'éditeur de liens **ld**, le générateur d'archives **ar**, le générateur d'index pour les bibliothèques **ranlib...**, mais aussi la suite **GCC** avec différents compilateurs tels que C, C++, Fortran, ADA... et leurs bibliothèques.


De plus, MinGW et Cygwin sont également pourvus de tout un système d'émulation de Linux sous Windows. Ceci permet par exemple de profiter des **Autotools**, autorisant ainsi le lancement de scripts de configuration dans le but de compiler des outils supplémentaires.

À noter que MinGW et Cygwin ne sont pas les seules collections d'outils disponibles mais dans ce tutoriel nous nous focaliserons principalement sur celles-ci. Toutefois, vous pourrez trouver en fin d'article des références vers d'autres collections.

Le choix de l'utilisation de MinGW ou de Cygwin doit se faire en fonction du type d'exécutable que vous souhaitez produire. Nous allons donc voir quelles sont leurs différences à ce niveau-là et comment les installer.

**Remarque :** Vous pouvez les installer parallèlement, sans que cela ne pose de quelconque problème. Eclipse saura déterminer l'ensemble des outils dont il peut disposer, à partir du moment où ceux-ci auront été référencés dans la variable d'environnement **PATH** du système ou dans la configuration d'un projet.

### III-A - Installation de MinGW

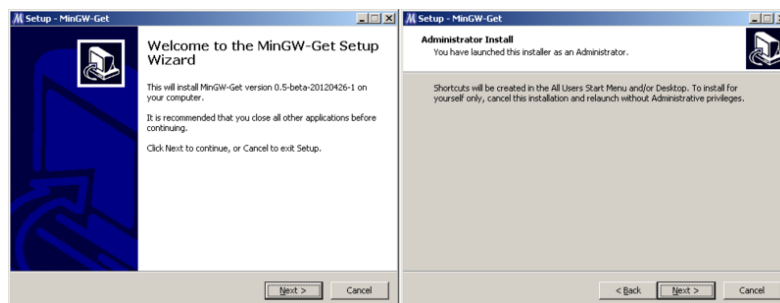
**MinGW** étant la contraction de « *Minimalist GNU for Windows* », cela signifie que c'est un environnement sous **licence GNU**, dédié au développement d'applications natives pour Windows. Il ne requiert pas d'autre bibliothèque que celles fournies par le système Windows mais ne fournit pas d'environnement d'exécution  **POSIX**.

MinGW apporte les éléments suivants :

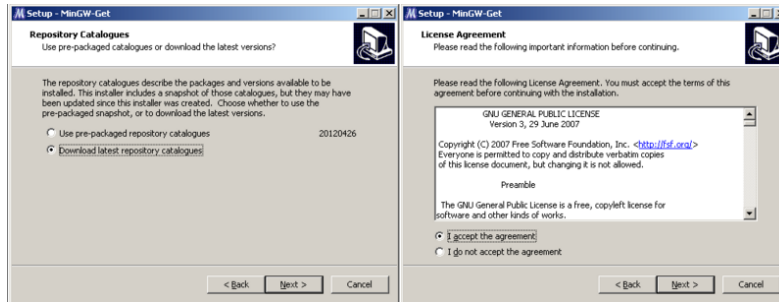
- un portage de la collection d'outils de compilation **GNU (GCC)**, incluant les compilateurs **C, C++, ADA** et **Fortran** ;
- **GNU Binutils** pour Windows (assembleur, linker, gestionnaire d'archives) ;
- un installateur en ligne de commande (**mingw-get**) pour le déploiement de **MinGW** et **MSYS** sur Windows ;
- une IHM graphique (**mingw-get-inst**) permettant d'installer **MinGW** et les composants souhaités.

Pour installer MinGW, voici comment faire :

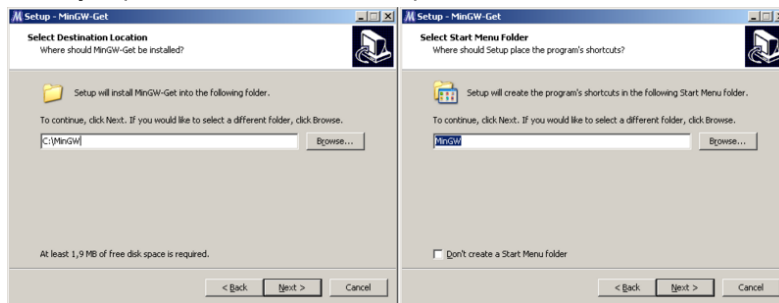
- 1 En premier lieu, il faut récupérer la dernière version de MinGW que vous trouverez ici **Installateur mingw-get-inst**.
- 2 Ensuite, exécutez le fichier récupéré, puis validez en cliquant sur **Next >**, jusqu'à la fenêtre **Repository Catalogues**.



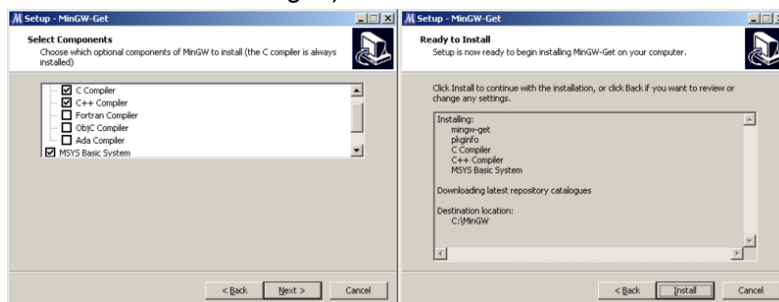
- 3 Sélectionnez l'option **Download latest repository catalogues** afin d'être sûr de récupérer les dernières versions des outils, puis cliquez sur **Next >**, acceptez la licence et cliquez de nouveau sur **Next >**.



- 4 En ce qui concerne le répertoire d'installation, il est préférable de conserver le chemin par défaut, notamment pour éviter les chemins trop longs, ou ceux contenant des espaces car cela pourrait s'avérer problématique dans certains cas. Vous pouvez néanmoins choisir un lecteur différent du lecteur **C:**. Cliquez ensuite sur **Next >**, jusqu'à la sélection des composants à installer.



- 5 Parmi les composants présentés, sélectionnez au minimum le compilateur **C++** ou **C**, ainsi que **MSYS Basic System**. Ensuite cliquez sur **Next >**, puis **Install**, ce qui provoquera le téléchargement et l'installation des composants (cette opération est potentiellement assez longue).



- 6 Pour finaliser l'installation de MinGW, il reste à ajouter à la variable d'environnement **PATH**, le chemin vers le répertoire **bin** de MinGW.



Si vous ignorez comment effectuer cette opération, voici ce qu'il faut faire :

- 1 Allez dans le menu **Démarrer** de Windows, puis sélectionnez **Paramètre > Panneau de configuration > Système**.
- 2 Dans la fenêtre qui s'est ouverte, placez-vous sur l'onglet **Avancé**, puis cliquez sur le bouton **Variables d'environnement** situé au bas de la fenêtre.
- 3 Dans cette fenêtre, au niveau du bloc inférieur **Variables système**, repérez l'emplacement de la variable **PATH** (ou **Path**), puis cliquez sur **Modifier**.
- 4 Vous devez à présent ajouter, dans la zone **Valeur de la variable**, le chemin vers le répertoire **bin** de MinGW. Par exemple, ajoutez le chemin `;C:\MinGW\bin` à la fin de la chaîne. Vérifiez que le caractère `;`, servant de séparateur, est bien présent entre la fin du chemin existant et celui que vous aurez ajouté. Il ne faut bien évidemment pas toucher au reste des valeurs pré-existantes, donc si vous n'avez pas l'habitude de cette manipulation, soyez prudent.
- 5 Validez en cliquant sur **Ok**, jusqu'à sortir des **Propriétés Système**.
- 6 Pour vérifier la bonne prise en compte de votre paramétrage, ouvrez une nouvelle **Invite de commande DOS** (via **Démarrer > Exécuter... > cmd.exe**) et tapez la commande `set`. Ceci fait alors apparaître la liste des variables d'environnement, au sein desquelles sera présente la variable **PATH**, complétée du chemin que vous venez d'ajouter.  
Si jamais ce n'est pas le cas, revoyez la configuration de la variable d'environnement.

Remarque : Si votre Eclipse est démarré, pensez à le relancer afin qu'il puisse prendre en compte les modifications de la variable PATH.


## III-B - Installation de Cygwin

Comme expliqué précédemment, Cygwin est composé d'une collection d'outils qui servent à émuler le « *Look and Feel* » Linux, sous Windows. Autrement dit, Cygwin permet de retrouver l'apparence et les usages de l'environnement Linux à l'aide des mêmes programmes et utilitaires présents sur ce système.

D'autre part, a contrario de MinGW, Cygwin permet de créer des exécutables au standard  **POSIX**, donc il vous revient de savoir si cela vous convient, car dans ce cas, vos applications devront pouvoir disposer de l'environnement  **POSIX** pour s'exécuter.

*Il est important de noter que Cygwin semble avoir du mal à suivre les évolutions des outils de compilation et donc à en fournir des versions récentes. Heureusement, si vous souhaitez pouvoir disposer de celles-ci, il est possible de pallier ce défaut. Donc, une fois l'installation de Cygwin terminée, vous pourrez si vous le souhaitez le compléter avec les versions récentes des outils de compilation. Il vous faudra alors récupérer, compiler et installer les composants **GCC**, **GMP**, **MPC** et **MPFR**.*

*Pour vous aider dans cette tâche, vous avez notamment le tutoriel suivant  **How to install a newer version of GCC**.*

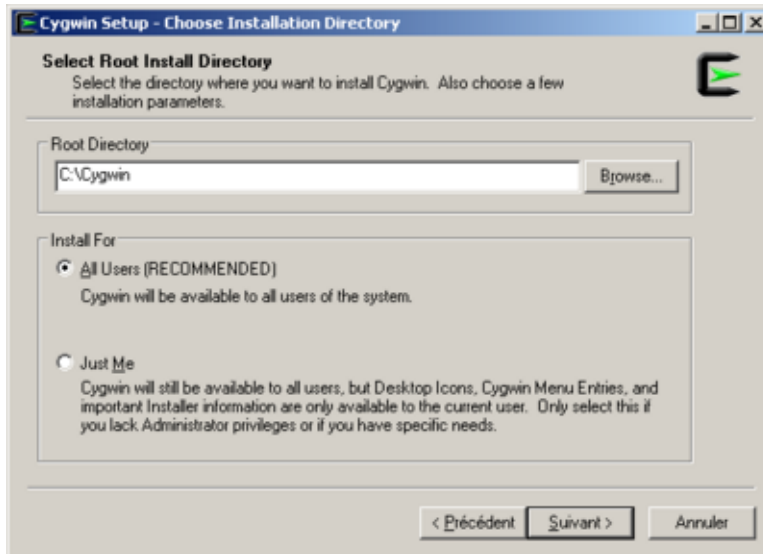
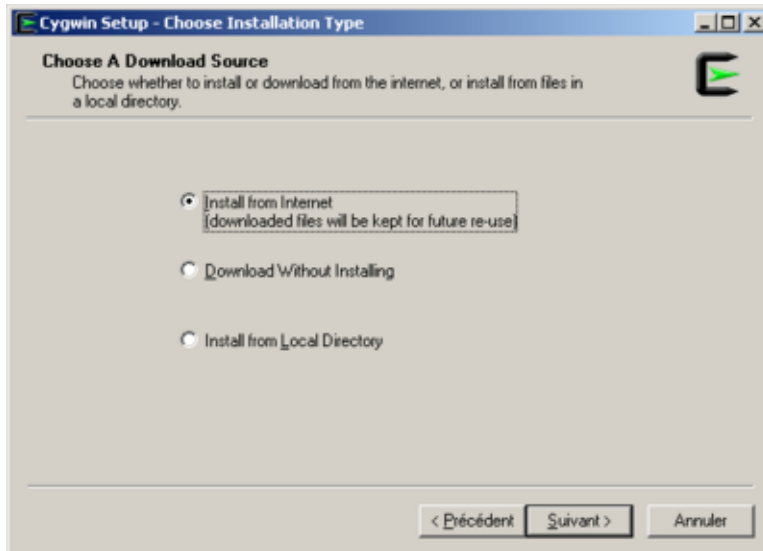
 *Notez cependant que c'est une opération relativement délicate et qui prend un certain temps, notamment en ce qui concerne la phase de compilation. De plus, ce tutoriel n'est pas forcément en phase avec les dernières versions des packages à installer et parfois il faudra peut-être ignorer certaines étapes, ou surmonter des problèmes spécifiques... Difficile de donner davantage de précision car le document étant de type WIKI, toute indication ou contre-indication peut rapidement devenir inutile ou obsolète.*

*Quoi qu'il en soit, personnellement je suis parvenu à intégrer la version **4.7.2** du compilateur sur la base de ce tutoriel. Celui-ci s'appuyait sur les packages `gcc-4.7.2.tar.bz2`, `gmp-5.0.5.tar.bz2`, `mpc-1.0.1.tar.gz` et `mpfr-3.1.1.tar.bz2`. Ce qui n'est pas expliqué dans ce tutoriel, dans le cadre d'Eclipse, c'est qu'il faudra compléter l'opération par une modification des paramètres de projet. Le chapitre **V. Ajuster le paramétrage d'un Projet C** indique comment effectuer cette dernière étape.*

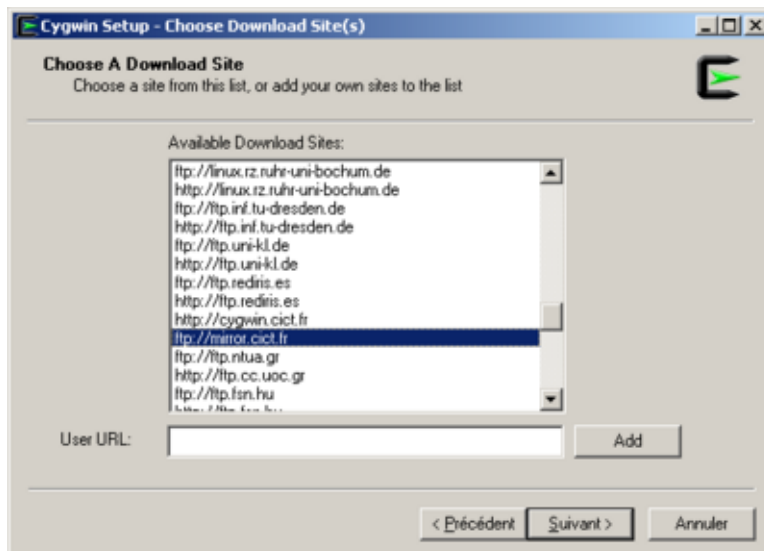
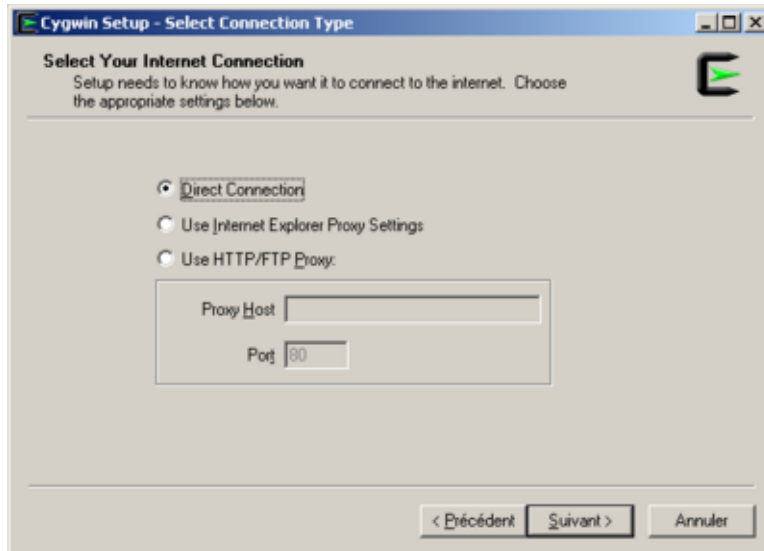
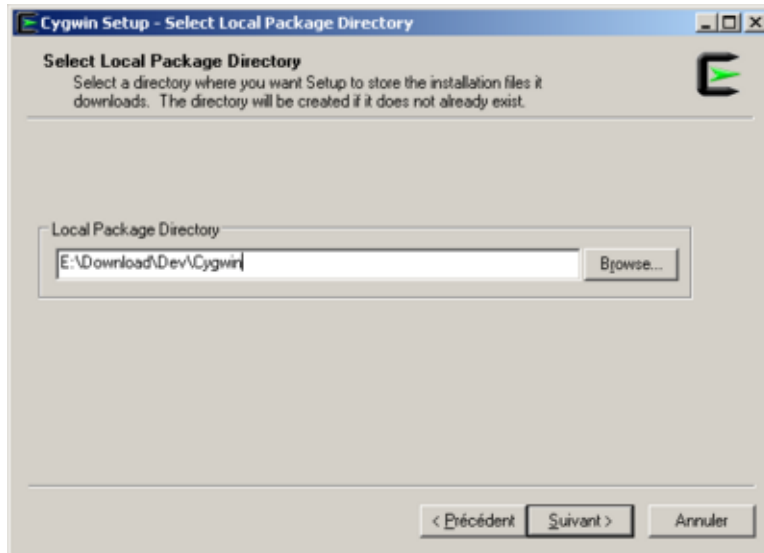
Pour installer Cygwin, procédez comme suit :

- 1 Téléchargez la dernière version de Cygwin à cette adresse <http://cygwin.com/setup.exe>, puis exécutez-là. Lorsque la première fenêtre s'ouvre, laissez vous guider en sélectionnant les options qui vous conviennent et cela, jusqu'au formulaire de sélection des packages.

**Remarque** : au moment de sélectionner un site de téléchargement pour le rapatriement des outils, il est préférable, pour des questions de débit, de choisir un site miroir proche de votre localisation. Néanmoins, si vous constatez que ça met du temps à venir, n'hésitez pas à tester un autre site miroir.



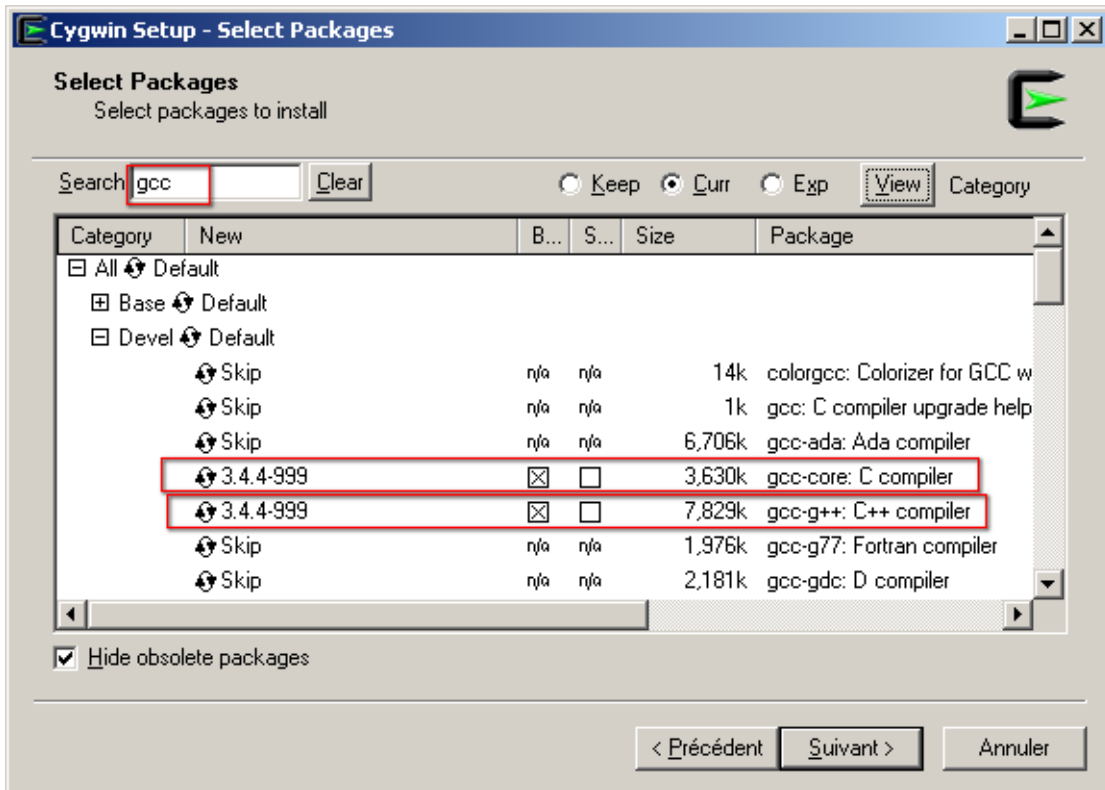




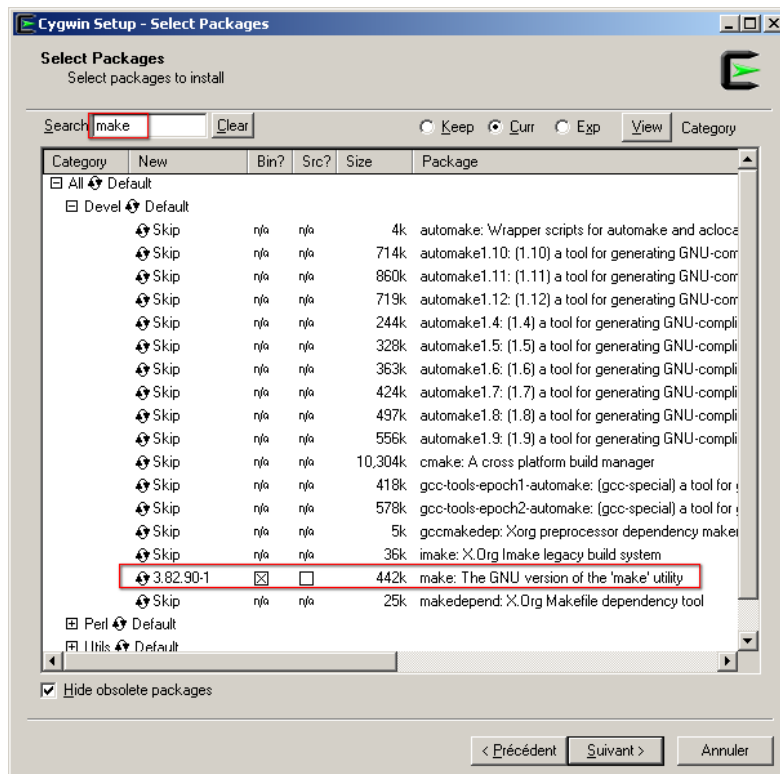
- 2 Par défaut, un certain nombre de packages sont pré-sélectionnés mais en ce qui concerne les outils C/C++, ce n'est pas le cas. De fait, il faut rechercher et cocher un à un les packages, pour le compilateur, l'outil make et le débogueur. Tous les packages qui nous intéressent se trouvent dans la rubrique **Develop**. Pour retrouver les éléments à sélectionner, aidez-vous de la zone **Search**.

Commencez par taper **gcc** dans celle-ci, vous obtenez ainsi une liste réduite, dans laquelle il vous faut sélectionner **gcc-core** et/ou **gcc-g++**.

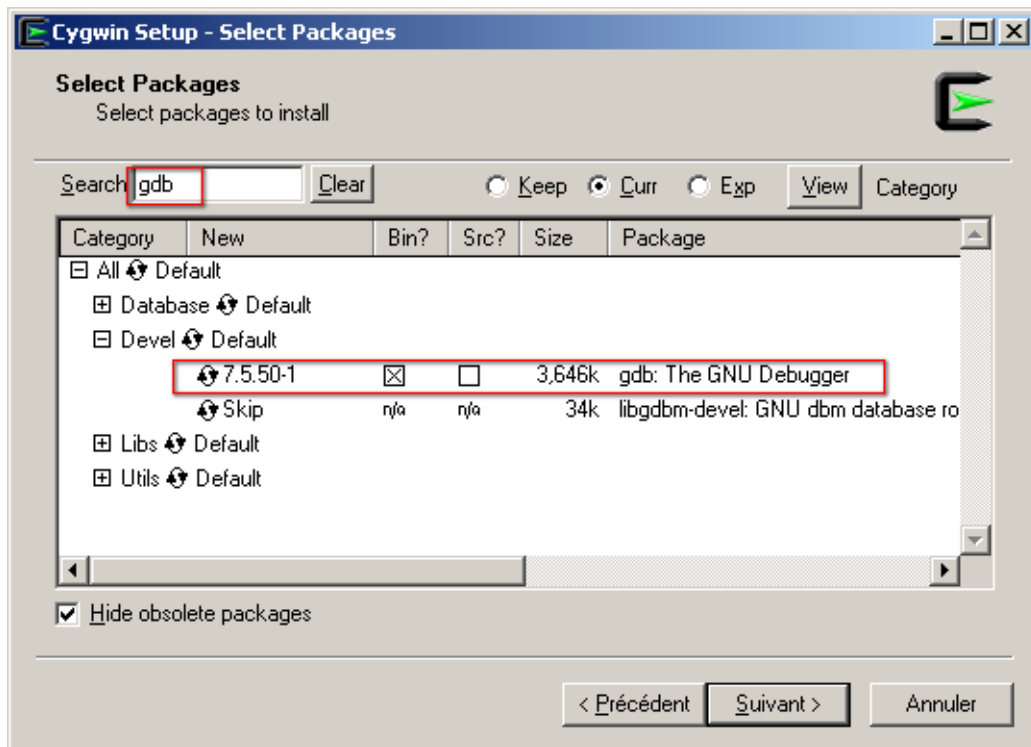
Dans la colonne **New**, vous verrez apparaître la version du programme sélectionné, en lieu et place de l'état **Skip**.



3 Faites de même pour l'outil **make** :



4 Idem pour le débogueur :



- 5 Finissez l'installation en cliquant sur **Suivant**.
- 6 Lorsque c'est fait, assurez-vous que les outils en **gcc.exe**, **make.exe** et **gdb.exe** sont bien présents dans le répertoire **bin** de votre installation de Cygwin.
- 7 Pour finaliser l'installation de Cygwin, il reste à ajouter le chemin vers son répertoire **bin**, à la variable d'environnement PATH.  
Si vous ignorez comment effectuer cette opération, voici ce qu'il faut faire :
  - 1 Allez dans le menu **Démarrer** de Windows, puis sélectionnez **Paramètre > Panneau de configuration > Système**.
  - 2 Dans la fenêtre qui s'est ouverte, placez-vous sur l'onglet **Avancé**, puis cliquez sur le bouton **Variables d'environnement** situé au bas de la fenêtre.
  - 3 Dans cette fenêtre, au niveau du bloc inférieur **Variables système**, repérez l'emplacement de la variable PATH (ou **Path**), puis cliquez sur **Modifier**.
  - 4 Vous devez à présent ajouter, dans la zone **Valeur de la variable**, le chemin vers le répertoire **bin** de Cygwin. Par exemple, ajoutez le chemin ;C:\Cygwin\bin à la fin de la chaîne. Vérifiez que le caractère ';', servant de séparateur, est bien présent entre la fin du chemin existant et celui que vous aurez ajouté. Il ne faut bien évidemment pas toucher au reste des valeurs pré-existantes, donc si vous n'avez pas l'habitude de cette manipulation, soyez prudent.
  - 5 Validez en cliquant sur **Ok**, jusqu'à sortir des **Propriétés Système**.
  - 6 Pour vérifier la bonne prise en compte de votre paramétrage, ouvrez une nouvelle **Invite de commande DOS** (via **Démarrer > Exécuter... > cmd.exe**) et tapez la commande set. Ceci fait alors apparaître la liste des variables d'environnement, au sein desquelles sera présente la variable **Path**, complétée du chemin que vous venez d'ajouter.  
Si jamais ce n'est pas le cas, revoyez la configuration de la variable d'environnement.

Remarque : Si votre Eclipse est démarré, pensez à le relancer afin qu'il puisse prendre en compte les modifications de la variable **PATH**.

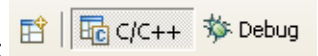
## IV - Vérification du fonctionnement dans Eclipse

Cygwin et MinGW sont dédiés aux environnements Windows, donc une fois ceux-ci installés convenablement, Eclipse dispose en principe du nécessaire pour compiler ou déboguer un programme C/C++. Sur Linux ou Mac OS X, Eclipse devrait pouvoir en faire de même, à partir du moment où les conditions évoquées en introduction sont réunies.

Pour vérifier qu'Eclipse est 100 % opérationnel pour développer en C/C++, nous allons créer un projet de type *Hello World*, le compiler, l'exécuter, puis nous passerons au mode **Debug**.

Si ce n'est déjà fait, lancez Eclipse et placez-vous dans la perspective C/C++, afin notamment de pouvoir accéder facilement aux outils dédiés à ce type de développement. Si besoin, cette perspective peut-être activée en la

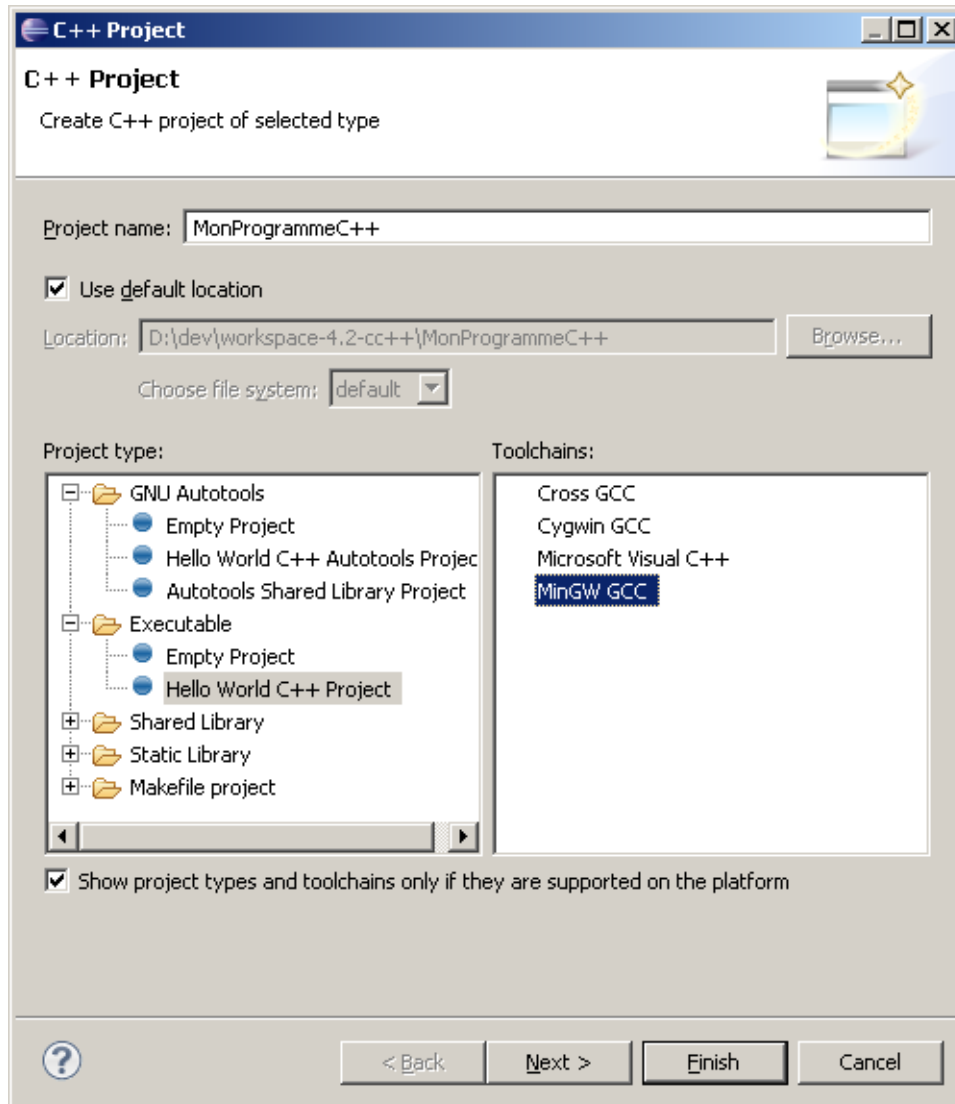
sélectionnant par l'intermédiaire de son icône, lequel est par défaut placé en haut à droite :



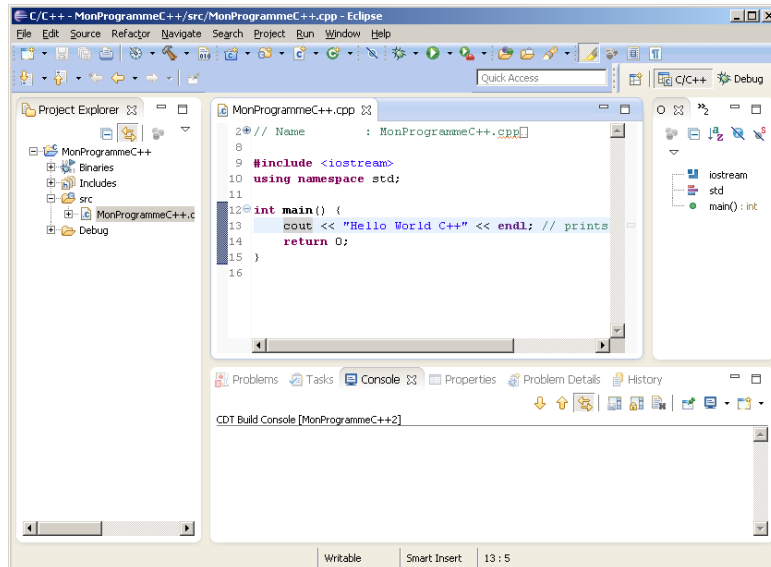
## IV-A - Création d'un Projet C/C++

Voici les étapes pour créer un projet :

- 1 Allez dans le menu **File > New > C++ Project**.  
 Dans l'assistant qui s'affiche, commencez par donner un nom à votre projet, par exemple MonProgrammeC++.  
 Au niveau du cadre **Project Type**, sélectionnez **Executable > Hello World C++ Project**.  
 Puis dans le cadre **Toolchains**, sélectionnez **Cygwin GCC** ou **MinGW GCC** en fonction de vos choix précédents, puis cliquez sur **Next >** jusqu'au dernier formulaire ou cliquez directement sur **Finish**.



- 2 Votre projet est en place et vous pouvez voir le code source du fichier principal.



- 3 Sélectionnez votre fichier source ou votre projet, puis allez dans le menu **Project > Build Project** afin de lancer la compilation.

Vous devez voir apparaître des traces telles que celles-ci, dans la console :

```
20:52:50 **** Rebuild of configuration Debug for project MonProgrammeC++ ****
Info: Internal Builder is used for build
g++ -O0 -g3 -Wall -c -fmessage-length=0 -o "src\MonProgrammeC++.o" "..\src\MonProgrammeC++.cpp"
g++ -o MonProgrammeC++.exe "src\MonProgrammeC++.o"

20:52:52 Build Finished (took 1s.532ms)
```

- 4 Pour finir, faites un clic droit sur votre projet ou sur le fichier source, puis **Run As > Local C/C++ Application**, ce qui provoque l'affichage de la chaîne attendue, dans la console :

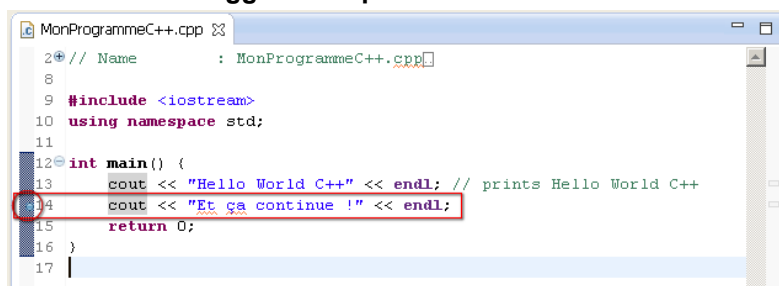
```
Hello World C++
```

## IV-B - Exécution en mode Debug

Pour exécuter l'application en mode Debug, voici comment procéder :

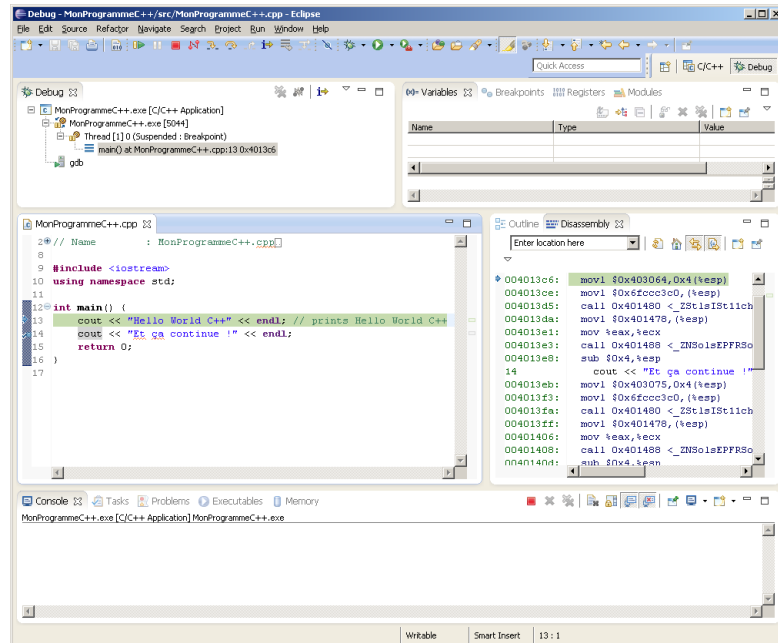
- 1 Tout d'abord, afin de rendre le test un peu plus intéressant, on va ajouter une ligne simple de code supplémentaire. Par exemple, ajoutez la ligne qui suit, juste avant le return.
 

```
cout << "Et ça continue !" << endl;
```
- 2 Dans la marge à gauche de la ligne que vous venez d'ajouter, double-cliquez afin de faire apparaître un point bleu représentant un point d'arrêt. Vous pouvez faire la même opération via le clic droit, toujours dans la marge gauche, puis en sélectionnant **Toggle Breakpoint**.

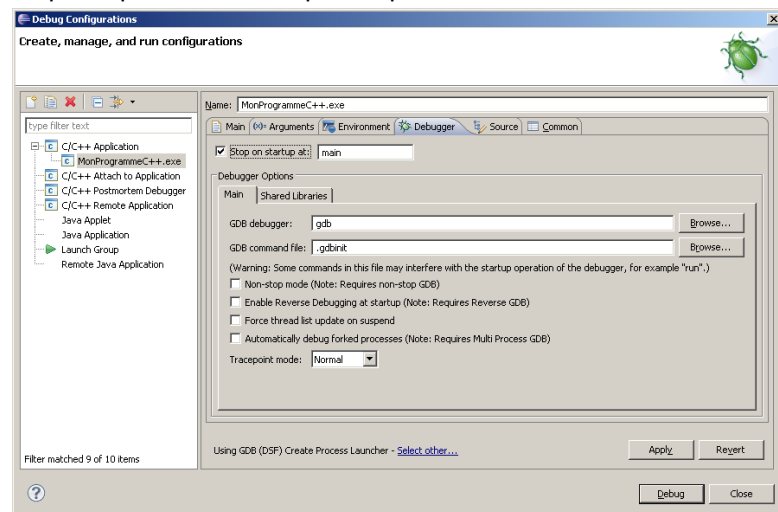


- 3 Exécutez à présent l'application en mode Debug, en allant dans le menu **Run > Debug** ou en appuyant sur **F11**.

Vous devriez constater que l'application est automatiquement compilée avant de s'exécuter.



- 4 Le mode Debug s'arrête sur la première ligne du corps de la méthode main et non pas sur votre point d'arrêt. En fait, cela est paramétrable au niveau du menu **Run > Debug Configurations**, dans l'onglet **Debugger** de la configuration d'exécution de votre application. Ainsi, sur cette page, vous pouvez configurer le mode Debug afin qu'il s'arrête ou non sur la méthode indiquée. En d'autre termes, par défaut la méthode main sert de premier point d'arrêt, quels que soient ceux placés par vos soins.



- 5 Si vous appuyez sur **F8**, le mode Debug continue son exécution et s'arrête sur le premier point d'arrêt suivant rencontré, donc dans le cas présent, sur celui que vous avez ajouté. Pour relancer le déroulement, il suffit de nouveau d'appuyer sur la touche **F8**.

En résumé les touches utiles pour le mode Debug sont les suivantes :

- **F5** : rentre dans la méthode invoquée sur la ligne courante ;
- **F6** : exécute l'instruction courante sans entrer dans la méthode invoquée ;
- **F7** : sort de la méthode courante ;
- **F8** : continue l'exécution jusqu'au prochain point d'arrêt ou la fin du programme.

Le mode Debug est extrêmement utile et permet d'agir de façon précise sur l'exécution, donc n'hésitez pas à l'étudier davantage afin de vous familiariser avec celui-ci.

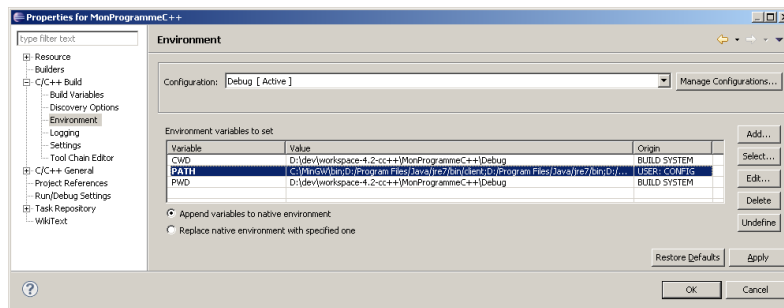
## IV-C - Ajuster le paramétrage d'un Projet C/C++

Eclipse offre de nombreuses possibilités de configuration au sein d'un projet C/C++. Nous n'allons cependant pas les passer en revue ici car ce n'est pas l'objet de ce tutoriel. En revanche, nous allons voir comment il est possible, a posteriori, de modifier l'outillage C/C++ d'un projet, c'est-à-dire de passer par exemple de Cygwin à MinGW ou inversement. Cette opération vous permettra d'avoir un premier aperçu de ce qui caractérise un projet C/C++ au sein d'Eclipse.

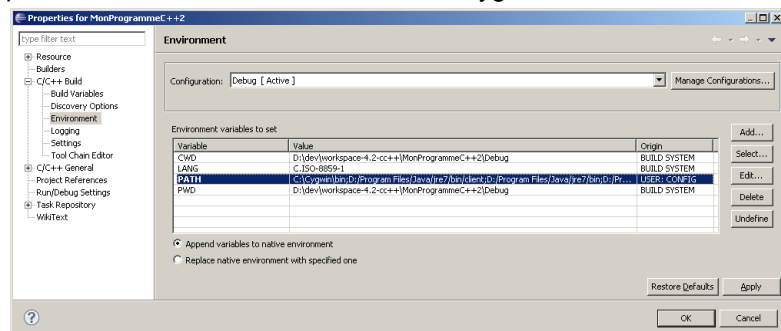
*Dans le cas où vous auriez basé votre projet sur l'outillage de Cygwin et que de plus, vous ayez décidé de le mettre à jour avec la dernière version du compilateur (comme indiqué au chapitre III-B. [Installation de Cygwin](#)), il vous faudra dès lors faire en sorte que celle-ci soit accessible, en ajustant la variable PATH du projet. En d'autres termes, vous devrez modifier cette variable pour que le chemin vers la nouvelle version du compilateur soit pris en compte, comme expliqué ci-après.*

Pour faire basculer l'outillage C/C++ de MinGW vers Cygwin ou inversement, voici comment procéder :

- 1 Tout d'abord, faites un clic droit sur votre projet dans la vue **Project Explorer**, puis allez dans le menu **Properties > C/C++ Build > Environment**.  
Si vous avez choisi l'outillage de **MinGW**, vous devez obtenir la fenêtre suivante avec la variable PATH qui commence par le chemin vers les exécutables de MinGW :



Si en revanche vous avez choisi l'outillage de **Cygwin**, vous devez obtenir la fenêtre suivante avec la variable PATH qui commence par le chemin vers les exécutables de Cygwin :

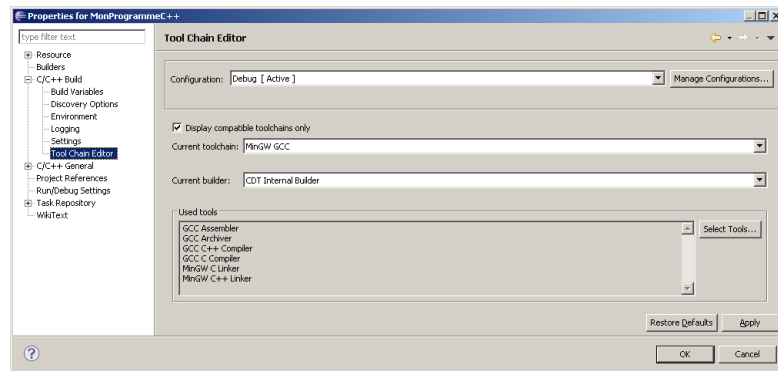


Notez que les valeurs indiquées peuvent varier en fonction de votre environnement et de vos choix.

- 2 À présent, modifiez la variable PATH à l'aide du bouton **Edit...**, en indiquant le chemin vers l'outillage qui vous convient, l'important étant que celui-ci soit placé en premier dans cette variable. Vous pouvez à votre guise choisir de supprimer ou laisser figurer le chemin de l'ancien outillage dans le PATH, du moment qu'il n'est pas en premier.

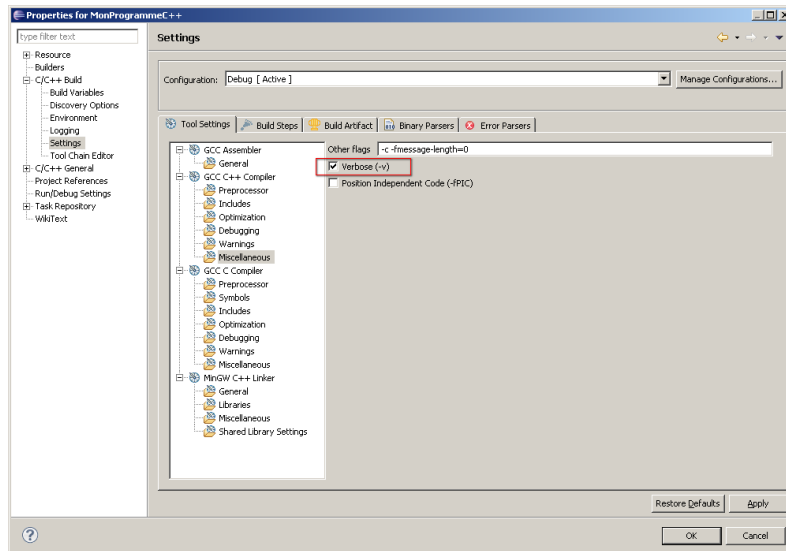
*Si vous souhaitez simplement faire en sorte d'utiliser la **dernière version du compilateur** que vous venez de déployer dans **Cygwin**, insérez le chemin **C:\cygwin\usr\local\bin;** au début de cette variable. Ce chemin étant par défaut celui qui aura reçu les exécutables résultant de la compilation des outils **GCC**, **GMP**, **MPC** et **MPFR** (comme indiqué au chapitre III-B. [Installation de Cygwin](#)).*

- 3 Pour que le changement d'outillage soit complet, il reste à l'indiquer explicitement, au niveau du menu **Tool Chain Editor**. Ainsi, il vous faut modifier le **Current toolchain** en conformité avec la modification de la variable PATH.



- 4 À ce niveau-là, la modification de l'outillage est terminée, il ne vous reste plus qu'à cliquer sur **Ok** pour valider la modification. Toutefois, pour qu'il soit facile de vérifier que tout est bien pris en compte, vous pouvez ajouter une option qui permettra de visualiser les versions et emplacements des outils utilisés au moment de la compilation.

Pour cela, allez dans le menu **Settings**, puis dans l'onglet **Tool Settings**, sélectionnez par exemple **GCC C++ Compiler > Miscellaneous**, et enfin cochez l'option **Verbose (-v)** comme indiqué ci-dessous :

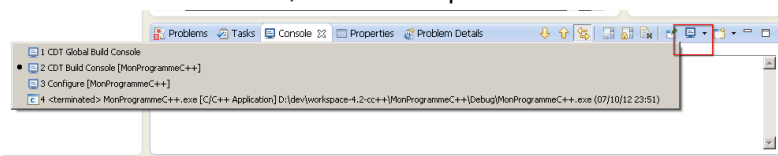


Pour terminer cliquez sur **Ok**, sélectionnez votre projet, puis effectuez un **Project > Clean...**, suivi d'un **Project > Build**.

Ainsi, dans la console d'Eclipse, vous devriez voir apparaître un certain nombre d'informations supplémentaires, notamment le chemin du compilateur, sa version...

À noter que vous aurez bien évidemment tout intérêt à désactiver cette option par la suite, afin d'éviter d'être systématiquement submergé par ce type d'information.




**Remarque :** Si vous constatez que la console n'affiche plus rien, ou pas ce qui vous intéresse, vérifiez que vous êtes sur la bonne instance de console, en la sélectionnant parmi celles présentes dans la liste qui s'affiche en cliquant sur l'icône dédiée à cela, comme indiqué ci-dessous :





## V - Conclusion

Ainsi s'achève ce tutoriel vous permettant de démarrer le développement d'une application C/C++ avec Eclipse.

Nous avons vu qu'Eclipse a besoin d'outils externes de façon à pouvoir compiler et exécuter des programmes C/C++ mais il est important de savoir qu'il existe d'autres outils en dehors de MinGW et Cygwin, tels que  **SDK Windows**,  **MinGW-w64**,  **TDM-GCC**...

Les versions successives d'Eclipse apportent leur lot de nouveautés et on peut dire que le projet CDT est en constante évolution. Il en est à sa version **8.1** (au moment de la rédaction de ce tutoriel) et d'autres évolutions le concernant sont au programme. Notamment, il est prévu qu'Eclipse intègre les particularités de la **version 11** du langage C++.

## VI - Liens utiles

Voici quelques liens utiles.

- Aide officielle pour développer avec Eclipse en C/C++ :  [http://help.eclipse.org/juno/topic/org.eclipse.cdt.doc.user/concepts/cdt\\_o\\_home.htm?cp=11](http://help.eclipse.org/juno/topic/org.eclipse.cdt.doc.user/concepts/cdt_o_home.htm?cp=11).
- La page officielle CDT :  <http://www.eclipse.org/cdt/>.
- Le Wiki CDT :  <http://wiki.eclipse.org/CDT>.
- Les nouveautés de la version **CDT 8.2** :  <http://wiki.eclipse.org/CDT/User/NewIn82>.  
Les nouveautés de la version **CDT 8.1** :  <http://wiki.eclipse.org/CDT/User/NewIn81>.  
Les nouveautés de la version **CDT 8.0** :  <http://wiki.eclipse.org/CDT/User/NewIn80>.
- Site officiel de MinGW :  <http://mingw.org/>.
- Site officiel de Cygwin :  <http://cygwin.com/>.
- D'autres outils de compilation C/C++ :
  -  **SDK Windows** ;
  -  **MinGW-w64** ;
  -  **TDM-GCC**.
- Si vous rencontrez des bugs au niveau du fonctionnement du plug-in CDT, vous pouvez vérifier si ceux-ci sont déjà référencés et dans le cas contraire, les rapporter vous-même. La rubrique concernant la partie CDT est celle-ci <https://bugs.eclipse.org/bugs/describecomponents.cgi?product=CDT>.