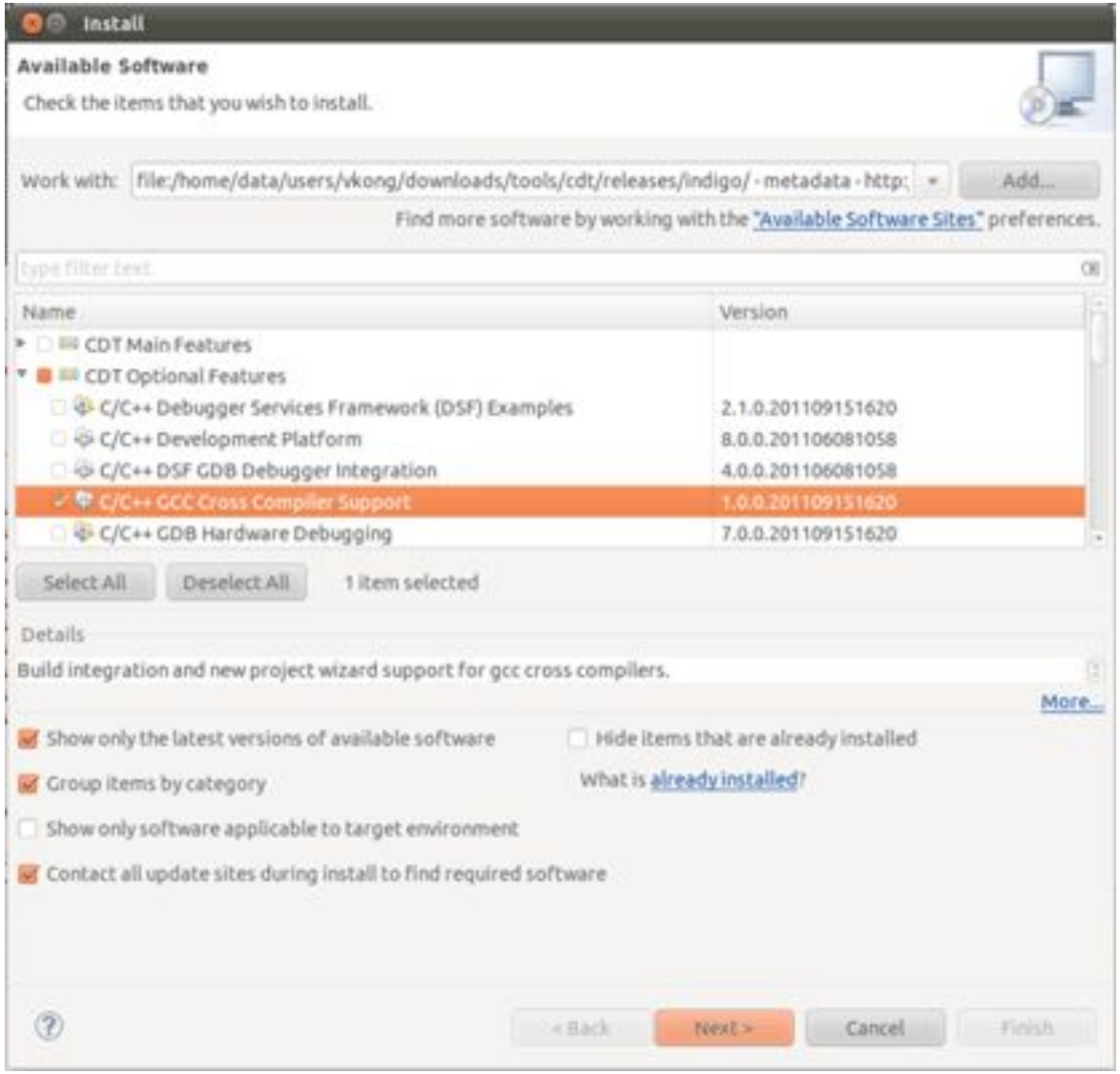
# Лекція №4.9
# Крос-компілятор з Linux у Windows та створення мейкфайлів

**Крос-компілятор з Linux у Windows**

Ще одиним варіантом побудови у Windows виконуваних файлів з Eclipse на Linux є варіант використанні крос-компілятора. Для цього ,CDT має мати відповідне розширення ,а крос-компілятор має бути встановленим. В Ubuntu 10.10, установка пакету mingw32 надає все необхідне для побудови зразка проекту з використанням крос-компіляторі MinGW. Отже, давайте встановимо CDT підтримку крос компіляції.

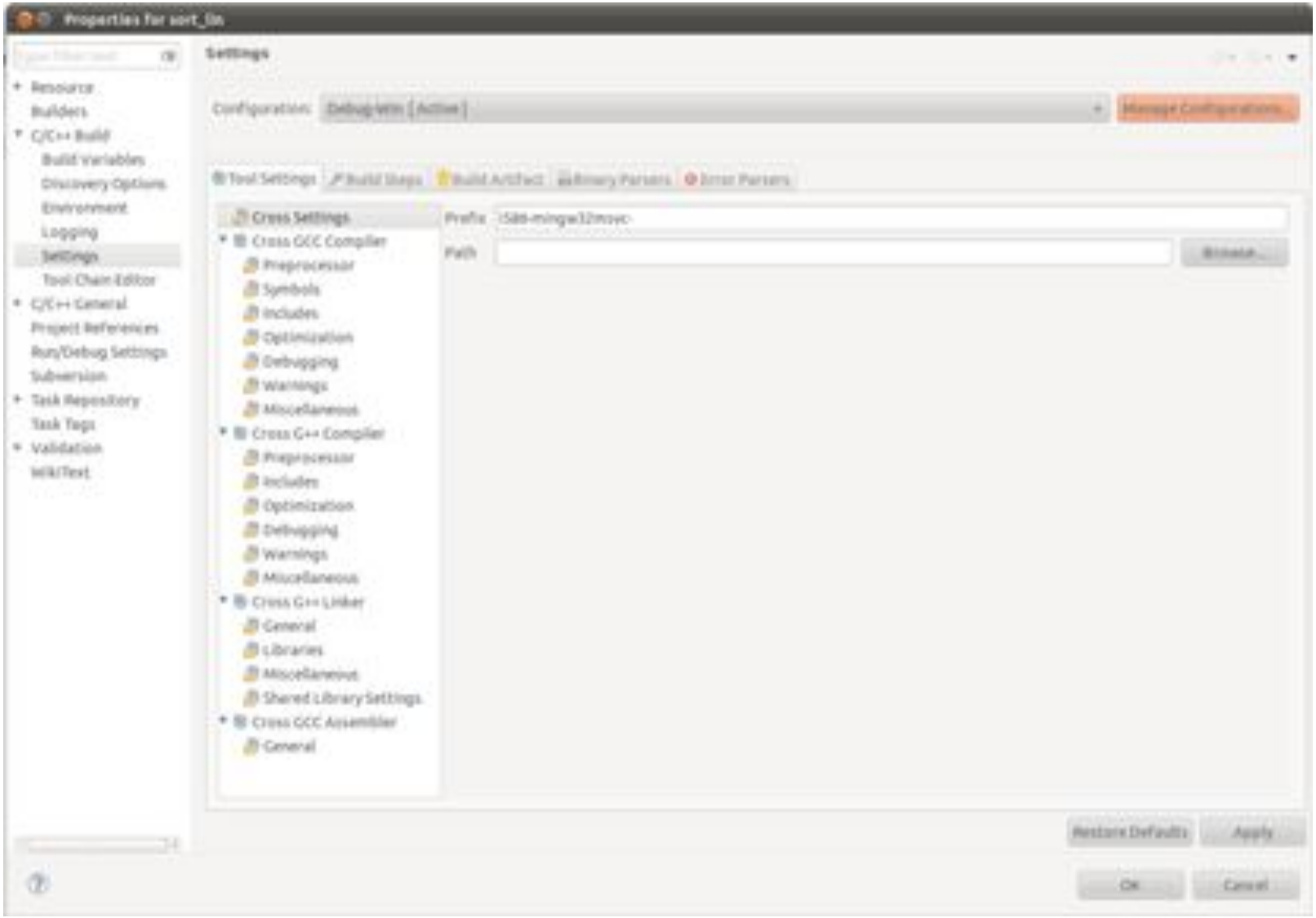
Щоб встановити CDT підтримку крос компіляції, треба перейти до розділу Help → Install New Software ... У текстовому полі, введіть cdt, для відбору налаштуваннь CDT та виберіть опцію CDT. Розгорніть CDT Optional Features і встановіть опцію C/C++ Cross Compiler Support . Далі дотримуйтесь інструкцій, щоб завершити установку (мал. 8).




**Малюнок 8: Налаштування Windows, крос-компіляцію на Linux.**

Встановлення CDT підтримки крос-компіляції надає нові можливості для побудови параметрів конфігурації. Для того щоб побачити ці параметри, дотримуйтесь наведених інструкцій. У вікні Import Predefined встановіть: Executable→Cross GCC→Debug, та Executable→Cross GCC→Release. Створіть нову конфігурацію під назвою Debug-Win і виберіть опцію Cross GCC → Debug.

При виборі цієї конфігурації, в налаштуваннях проекту ви помітите нову опцію під назвою Cross Settings. Відкрийте налаштування проекту, виберіть Cross Settings і введіть i586-mingw32msvc- в текстове поле. У випадку якщо воно залишається порожнім, IDE буде використовувати стандартні системні інструменти на посилання і компіляції (малюнок 9).


Properties for sort_lis
Settings
Resource
Builders
C/C++ Build
Build Variables
Discovery Options
Environment
Logging
Settings
Tool Chain Editor
C/C++ General
Project References
Run/Debug Settings
Subversion
Task Repository
Task Tags
Validation
WikiText
Configuration:
Tool Settings
Build Steps
Build Artifact
Binary Parsers
Error Parsers
Cross Settings
Cross GCC Compiler
Preprocessor
Symbols
Includes
Optimization
Debugging
Warnings
Miscellaneous
Cross G++ Compiler
Preprocessor
Includes
Optimization
Debugging
Warnings
Miscellaneous
Cross G++ Linker
General
Libraries
Miscellaneous
Shared Library Settings
Cross GCC Assembler
General
Prefix:
Path:
Browse...
Restore Defaults
Apply
OK
Cancel

## Кілька гарних можливостей

Eclipse пропонує кілька варіантів для створення крос-платформних проектів. Це займає трохи часу, але розвиток між Linux і Windows, прекрасно працює, як тільки проект налаштований. Є й інші варіанти, які ми не розглядаємо такі як використання GCC на Linux і Windows, або створення makefiles вручну. Кожен варіант має свій власний набір примх, які включають каталоги, ремонтопридатність, сторонні бібліотеки і налагодження.

У цій статті ми розглянули пункти, що включають каталоги і варіанти, що є можливо, найпростішим для підтримки (з Eclipse, автоматично обробляє makefile). Якщо ви хочете налагодити на обох платформах або використовувати сторонні бібліотеки, це вимагатиме більше роботи, щоб зробити в залежності від обраного варіанту для крос-платформної розробки.

## Makefile

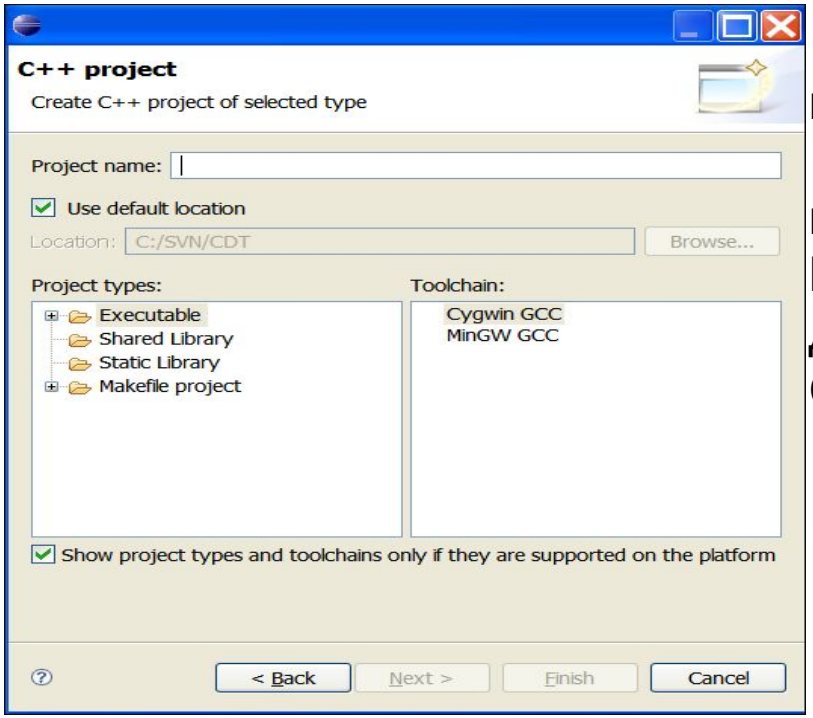
Makefile являє собою текстовий файл, який створюється за допомогою команди, яка описує будівлю цілей, і містить таку інформацію, як залежностей на рівні вихідного коду і залежностей від порядку побудови.
CDT може створювати Makefile для вас, такі проекти називаються Керовані проекти Марка. Деякі проекти, відомі як Standard Make project, дозволяють визначити свій власний Makefile.

## Побудова Makefile project

Це керівництво описує процес створення нового C+ + проект , який включає в себе створення Makefile і налагодження програми. Вам потрібно створити проект, щоб утримувати свій вихідний код і пов'язані з ним файли. Проект має пов'язаний builder, який може поступово скомпонувати компіляції вихідних файлів.

# Створити проект:

Оберіть **File > New > Project**.




При створенні нового проекту, ви повинні вказати тип проекту. Цей тип проекту визначатиме набір інструментів, даних і вкладки, які CDT використовує / відображає. Виберіть тип проекту для створення. Для цієї команди, оберіть C/ C + + папку і виберіть C + + проект .

1. CDT фільтрує Toolchain і Типи проектів, які в даний час відображаються в цих списках, які засновані на мовної підтримки майстра C + + Project, який ви обрали для цього уроку.
2. У полі Ім'я проекту введіть HelloWorld.
3. Залиште прапорець Use Default Location.
Далі, ви повиннівибрати тип проекту для створення. У реалізації проекту будівництва нового CDT.

Ви можете вибрати один з наступних типів проектів:
**Executable** - Забезпечує виконуваного програми. Ця папка тип проекту містить три шаблони.

i. **Hello World C++ Example** забезпечує простий C++ Hello World додаток з головним().
ii. **Hello World ANSI C Example** забезпечує простий C Hello World додаток з головним().
iii. **Empty Project** надає єдину вихідну папку проекту, який не містить файлів.

Після вибору цього шаблону, результат це проект тільки з файлами мета-даних, необхідних для типу проекту. Ви повинні надати вихідні файли для мети проекту.

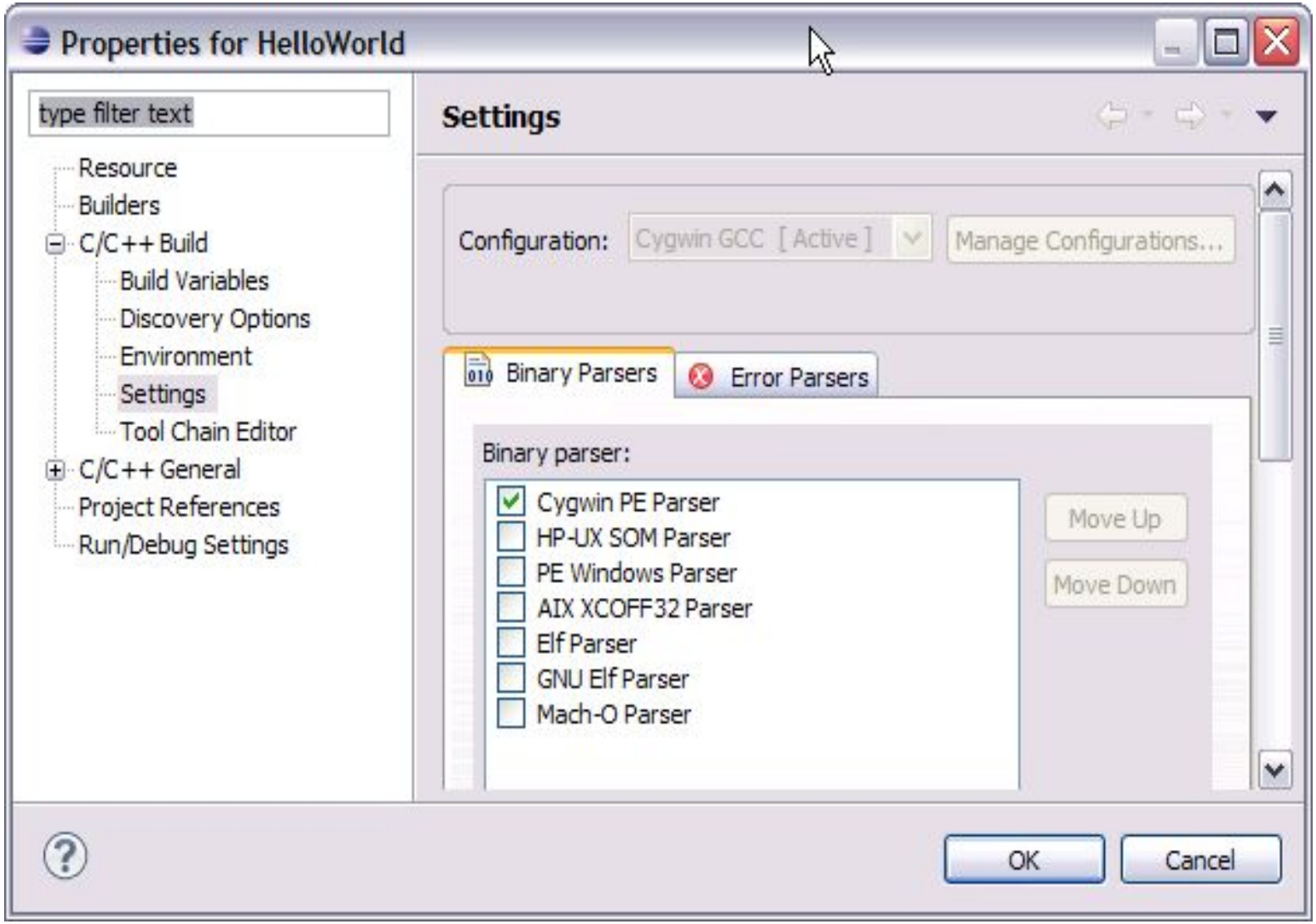
b. The makefile для проекту типу **Executable** автоматично будується CDT.

c. **Shared Library** - виконуваний модуль, який буде зібрано і пов'язано окремо. При створенні проекту, що використовує поділювану бібліотеку (libxx.so), ви визначаєте проект вашої бібліотеки в якості посилання на проект для вашої програми. Для цього типу проекту, CDT поєднує об'єктні файли разом і з'єднує їх, так що вони переміщуються і можуть бути розділені між багатьма процесами. Спільні бібліотеки іменуються формат libxx.so.version, де версія є кількість та значення за замовчуванням 1. Файл libxx.so зазвичай є символічною посиланням на останню версію. Makefile для цього типу проекту автоматично створюється на CDT.

d. **Static Library** - колекція об'єктних файлів, які можна зв'язати в інший додаток (libxx.a). CDT поєднує об'єктні файли (наприклад, *. O) в архів (*.), що безпосередньо пов'язано з виконуваним файл. Makefile для цього типу проекту автоматично створюється на CDT.

e. **Makefile Project** - Створює порожній проект без файлів мета-даних. Цей вибір корисний для імпорту та модифікації існуючих проектів Makefile основі; новий Makefile не створений для цього типу проекту.
CDT фільтрує набір інструментів і типів проектів, які в даний час відображенні в цих списках на основі мовної підтримки майстра C + + Project, який ви обрали для цього уроку.

4. Зі списку типів проектів розгорніть проект Makefile і виберіть Порожній проект. Цей проект дозволяє ввести ім'я вихідного файлу і Makefile самостійно.
5. Виберіть необхідний набір інструментів зі списку Toolchain. Інструментарій являє собою набір інструментів (таких як компілятор, компонувальник і асемблер) призначений для створення проекту. Додаткові інструменти, такі як відладчик, можуть бути пов'язані з інструментарієм.? Там може бути кілька Toolchains доступні, в залежності від укладачів встановлених в системі.
6. Натисніть **Next**.
7. (Необов'язково) На сторінці Вибір налаштування натисніть кнопку Додаткові параметри. Розгорнути C / C + +> Установки. Перейдіть на вкладку Бінарні аналізаторами.


Properties for HelloWorld
type filter text
Resource
Builders
C/C++ Build
Build Variables
Discovery Options
Environment
Settings
Tool Chain Editor
C/C++ General
Project References
Run/Debug Settings
Settings
Configuration:
Cygwin GCC [ Active ]
Manage Configurations...
Binary Parsers
Error Parsers
Binary parser:
Cygwin PE Parser
HP-UX SOM Parser
PE Windows Parser
AIX XCOFF32 Parser
Elf Parser
GNU Elf Parser
Mach-O Parser
Move Up
Move Down
OK
Cancel

У списку Binary Parser виберіть синтаксичний аналізатор помилку, який підходить для вашої платформи. Інші параметри проекту можуть бути змінені також і тут.
Для забезпечення точності точнічть C / C + + Projects і здатність успішно виконувати і налагоджувати програми, вибрати правильний синтаксичний аналізатор важливо. Після того як ви вибрали правильний синтаксичний аналізатор для вашого середовища розробки і збірки проекту, ви можете переглядати компоненти .o файлу в поданні C/ C + + Projects. Ви можете також переглянути вміст .o файлу в C/ C ++ редактор.

Натисніть OK, щоб закрити вікно Властивості проекту.
8. Натисніть кнопку Finish.
9. Якщо вікно повідомлення запропонує вам змінити перспективи, натисніть кнопку Так.
Ваш новий проект відображає у вигляді Project Explorer. Ваш проект порожній, бо ви ще не створили файли для вашого проекту. Ви можете побачити помилку, так як нічого не побудовано для вашого проекту. Тепер Ви можете почати писати код для вашої програми HelloWorld.
Порада: Ви можете переглядати і змінювати властивості вашого HelloWorld проекту в будь-який час, клацнувши правою кнопкою миші на проекті в поданні Project Explorer і виберіть пункт Properties.

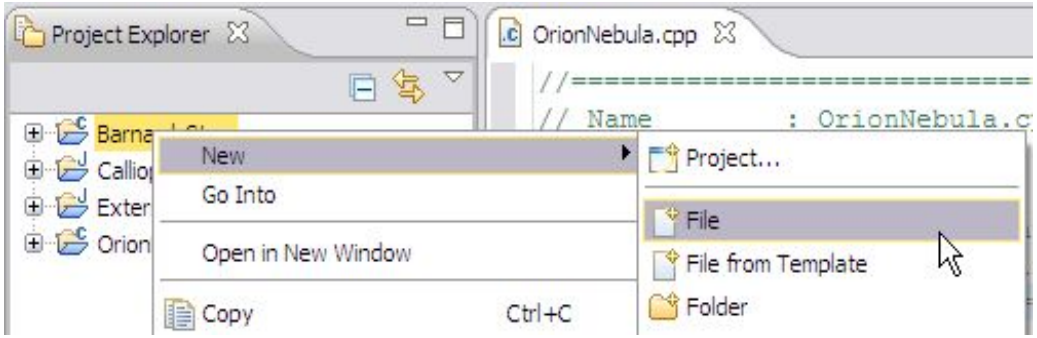
**Побудова makefile**

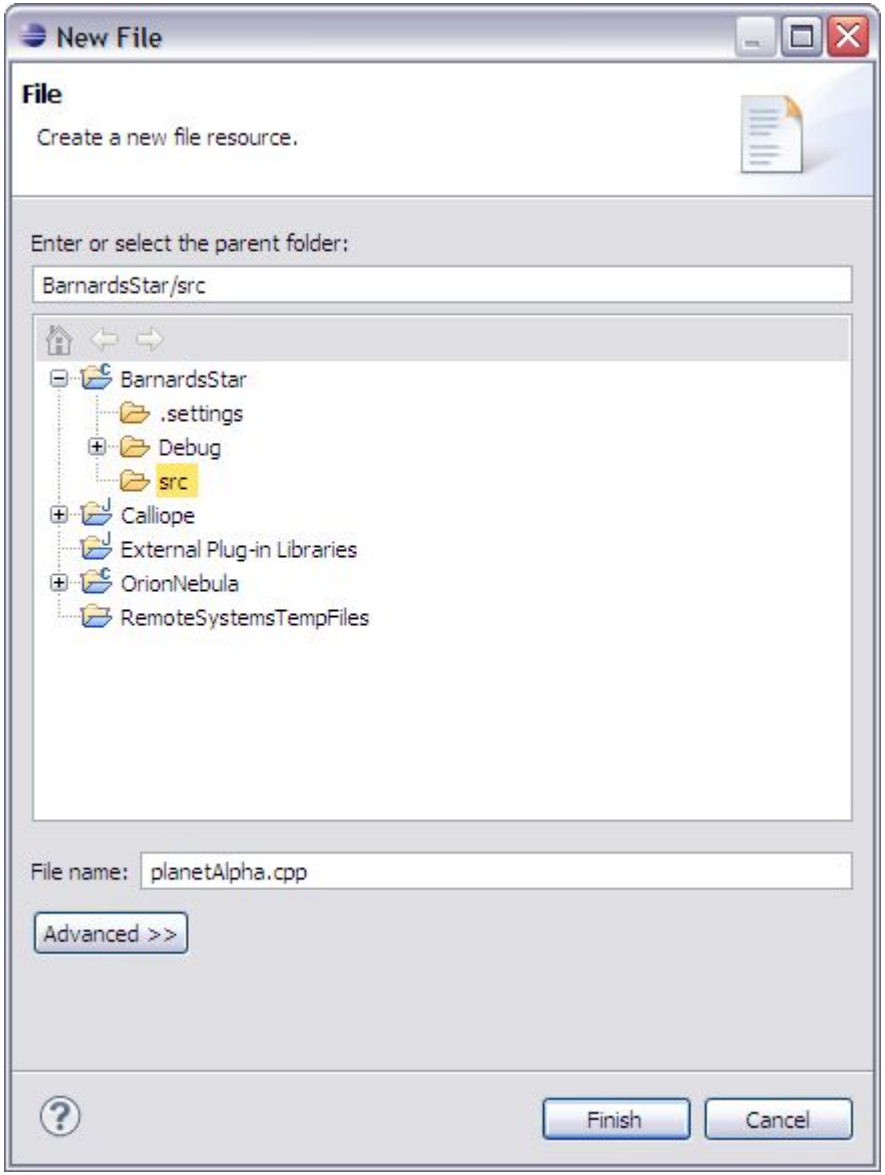
Якщо ви створили Standard Make C/C++ Project, необхідно забезпечити **makefile**.

При створенні проекту, висновок відображається у вікні консолі. Makefile дії відображаються у вигляді цілей.

Для створення цього файлу:

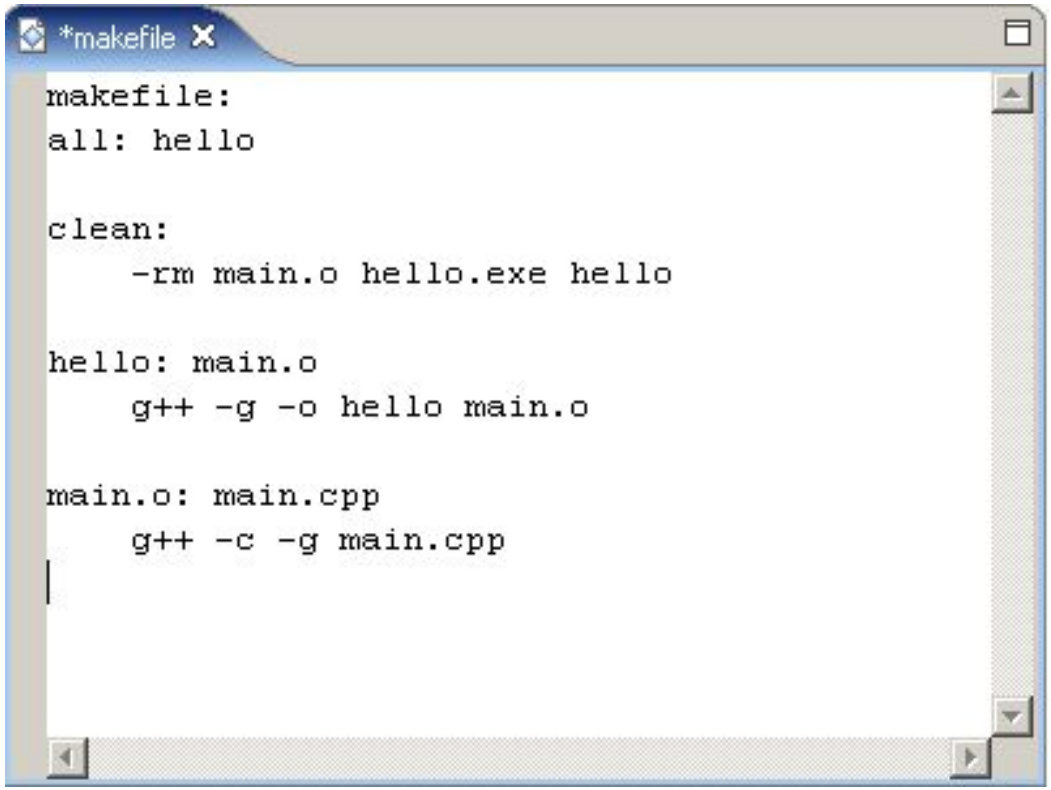
У C++ Projects клацніть правою кнопкою миші проект і виберіть New> File.






2. У полі File name введіть Makefile.
3. У списку проектів, переконайтеся, що обрано правильний проект.
4. Натисніть **Finish**

The C/C++ editor відкриваєтсья. Наберіть інструкції до  makefile в the C/C++ editor.




Натисніть **File > Save**.