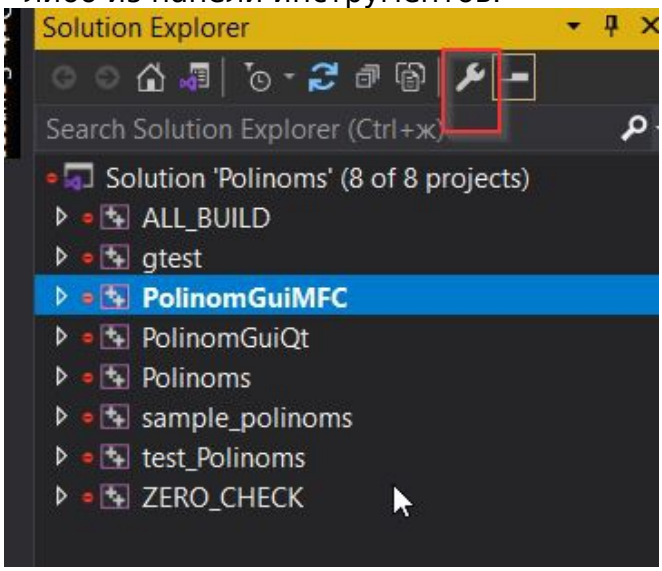
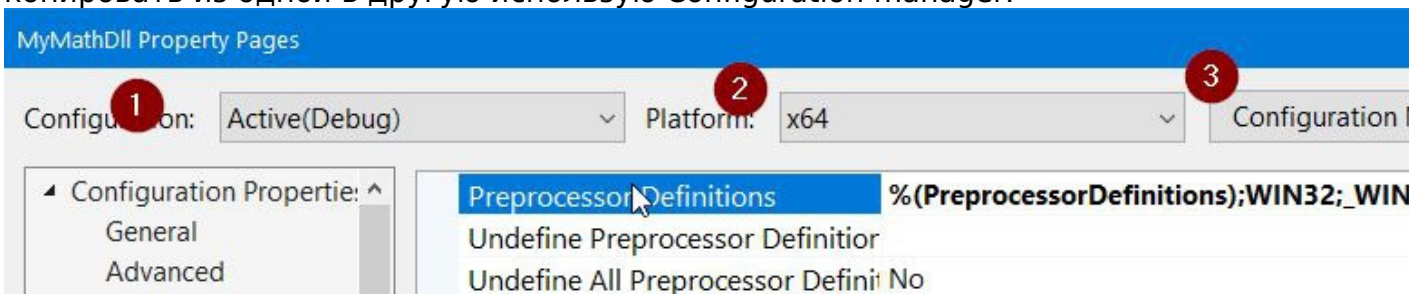
Настройка проекта в среде Visual studio 2019 C++

Вызов настроект проекта и правил сборки проекта (конфигураций сборки)
Меню настроек проекта и конфигураций вызывается из Solution explorer находясь на ветке конкретного проекта
- с клавиатуры Alt+Enter.
- из меню правой по правой кнопке на проекте -> свойства.
- либо из панели инструментов.



1. Конфигурация сборки и платформа
Это всего лишь возможность указывать группы настроек проекта и организовавать группы параметров что бы между ними было удобно переключаться. После создания конфигурации нам все равно надо проверить и установить для выбранной конфигурации все настройки проекта, которые будут указаны ниже. Так же конфигурации в момент создания можно копировать из одной в другую использую Configuration manager.

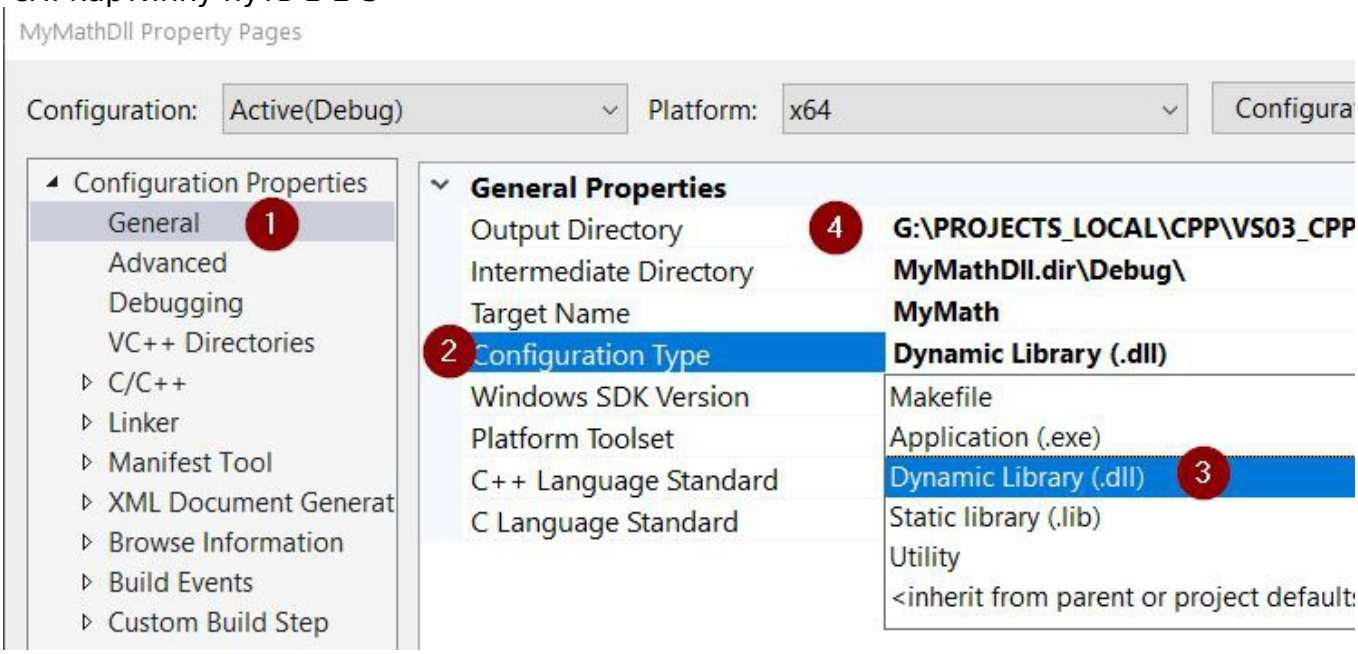


1 - Выбор конфигурации Release/Debug and other (названия могут быть любыми которые будут удобнее и понятнее)
2 - Выбор выходной платформы для которой будет производится сборка. Система предложит только те, для которых есть компиляторы и toolchain.
3 - Менеджер конфигураций где можно создать другие конфигурации (можно с другими define или библиотеками)

Важно понимать что создав конфигурацию это не гарантирует ни чего, это всего лишь навание группы настроек проекта, все параметры после создания надо настраивать. Выбираем конфигурацию и настраиваем. После настройки активной конфигурации можно переключиться на другую через пункты 1, 2 и опять настроить. Настройка конфигурации есть настройка конфигурации текущего активного проекта. После того как все проверено и настроено мы

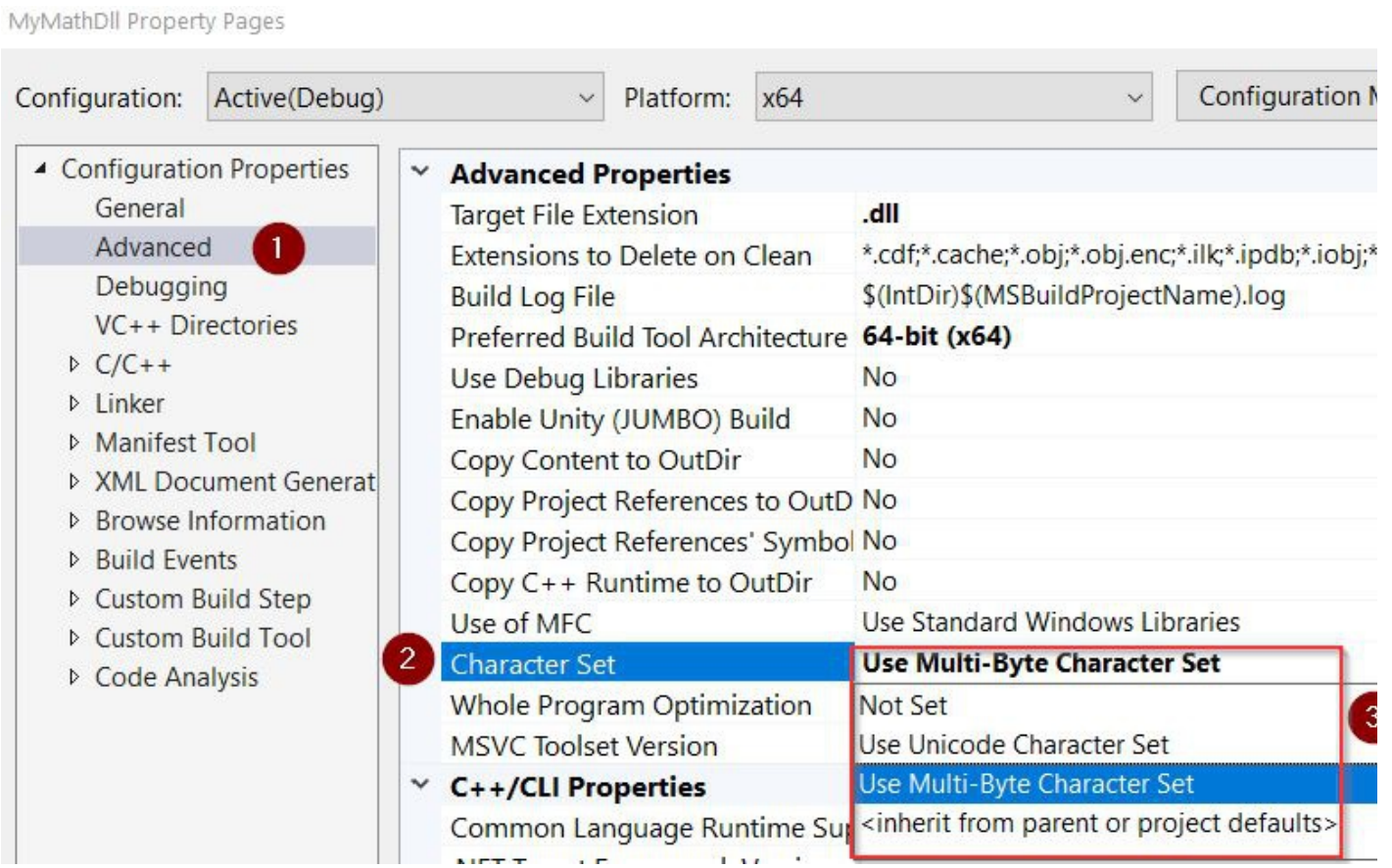
уже сможем переключаясь между конфигурациями получать нужный нам результат. Каждый проект настраивается отдельно.

2. Тип проекта
определяет что будет данный проект на выходе получать exe, lib, dll.
см. картинку путь 1-2-3

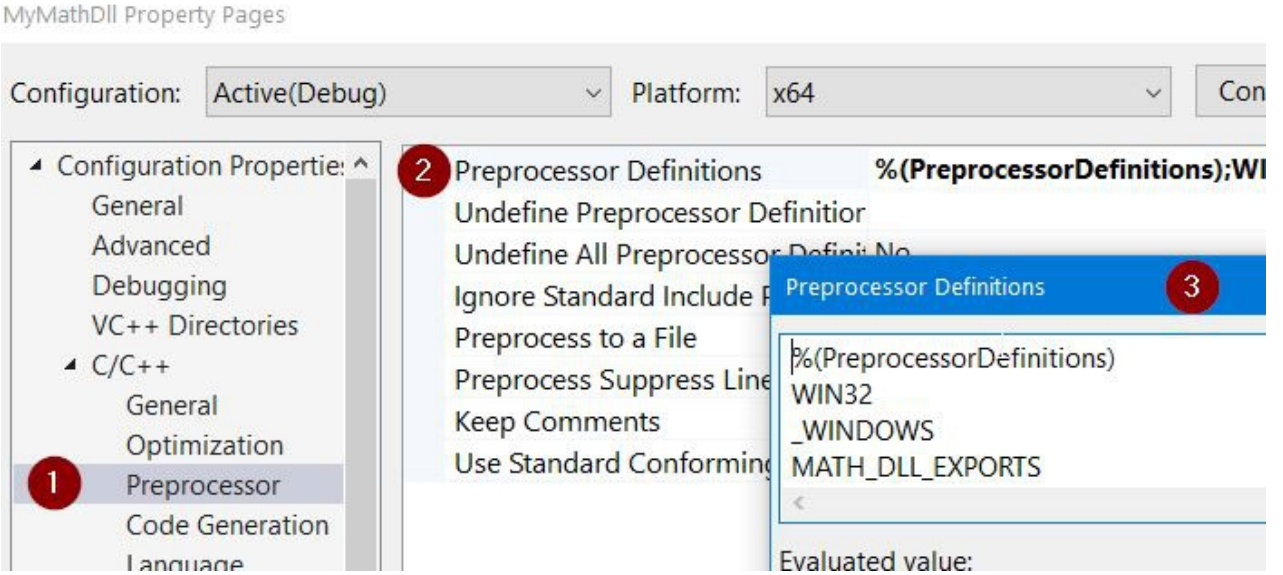


3. Output directory
Определяет путь (выходную директорию) куда будут помещены готовые бинарные файлы проекта
смотри предудущую картинку, путь 1-4.

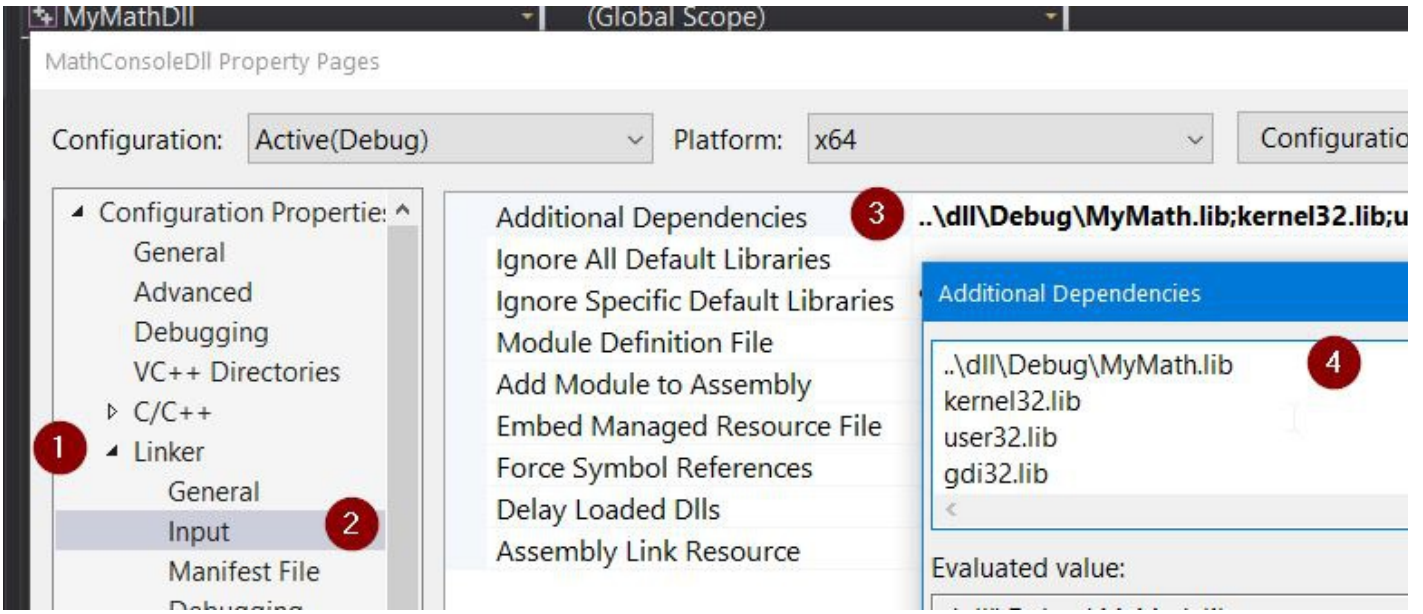
4. Правила работы с массивами строк в проекте
Может потребоваться при работе с строками си и массивами символов при передаче данных из DLL.



5. Константы препроцессора
Это так называемые #define установленные для всего проекта в рамках выбранной конфигурации.

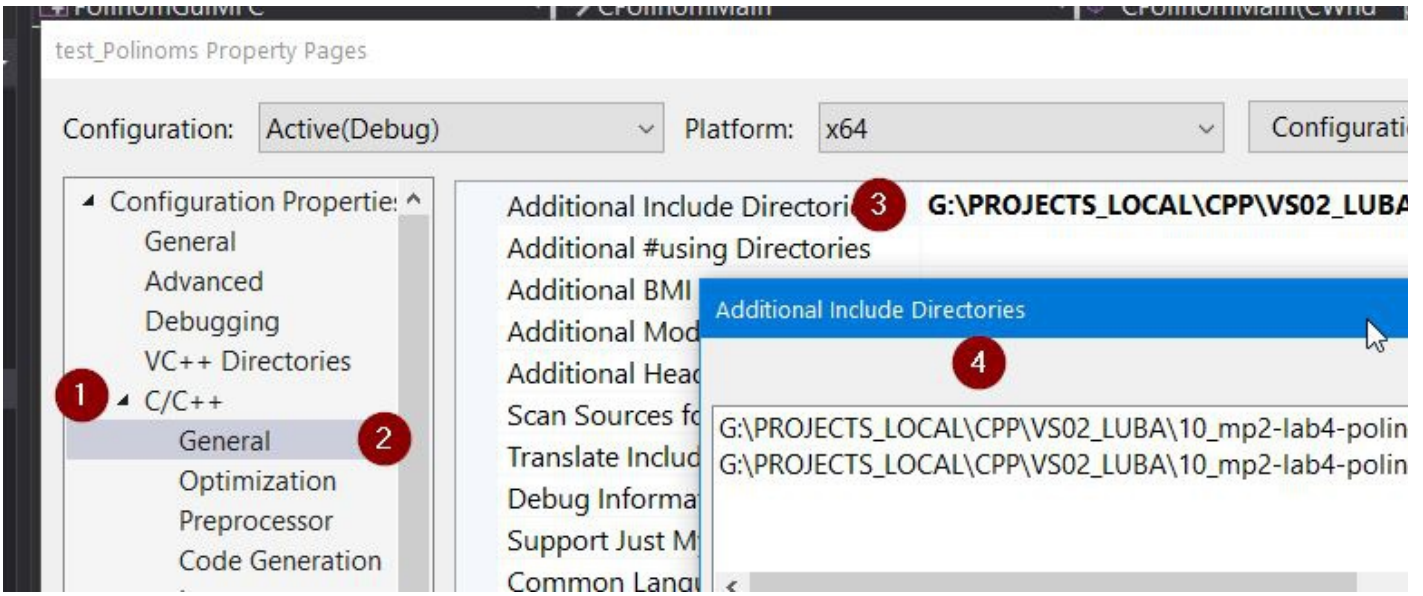


6. Расположение подключаемых к проекту библиотек LIB.
В частности библиотека gtest подключается как Static library

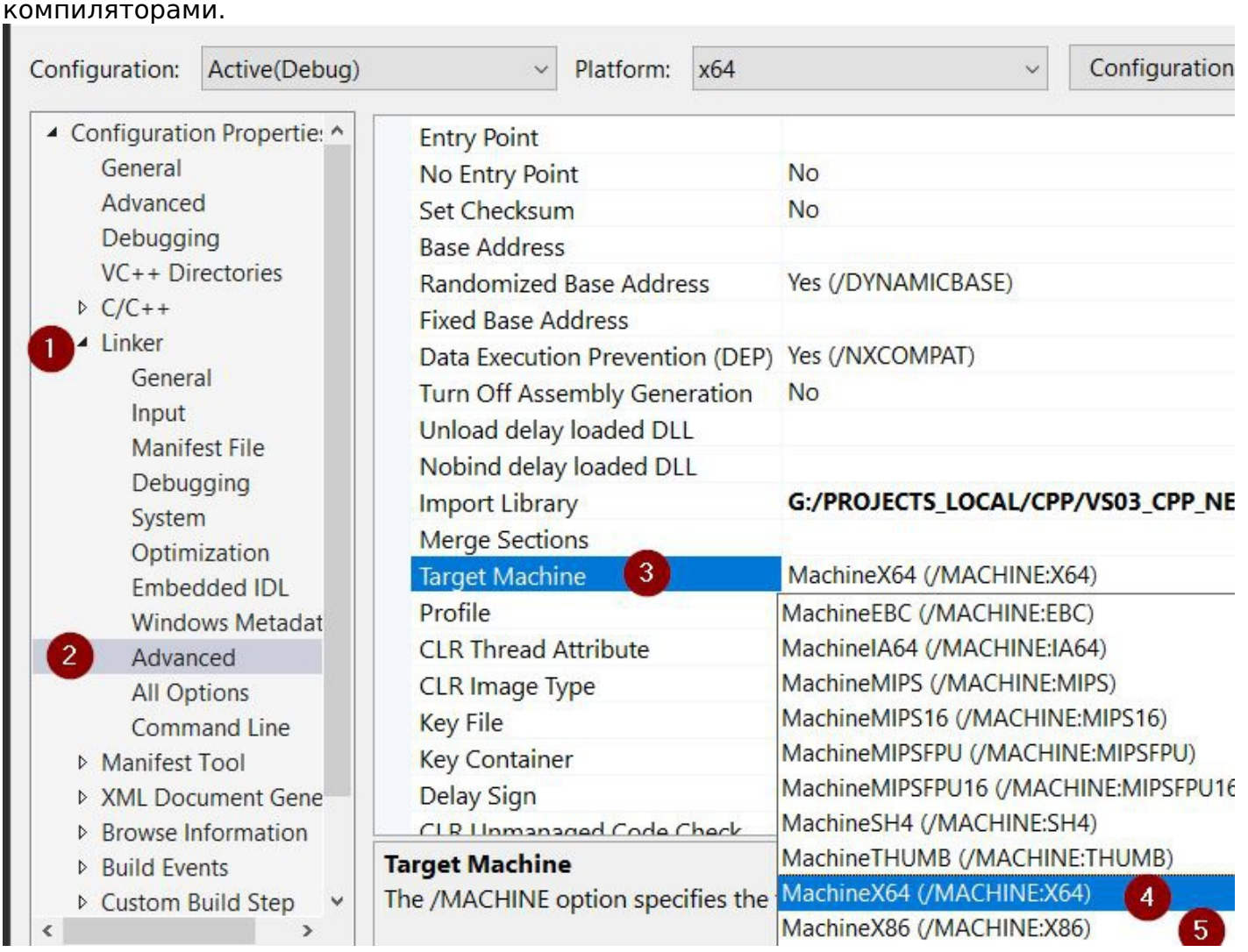


6.1.Статическая библиотека LIB
Подключается и включается внутрь проекта при сборке.
Для подкючения к проекту надо указать путь к библиотеке как на картинке выше и сделать возможным поиск .h файла описания библиотеки по одному из путей.
6.2.Динамическая библиотека DLL
Загружается для работы в память явно или неявно в момент старта программы.
- Для подключения DLL к проекту и работы в режиме автозагрузки надо иметь так же LIB файл (данный файл генерируется вместе с DLL), но в данном случае он будет выполнять роль бинарной заглушки для линкера, а в момент старта будет загружаться реальная DLL используя пути поиска. Для сборки проекта надо указать путь к библиотеке LIB как на картинке выше и сделать возможным поиск .h файла описания библиотеки по одному из путей. Так же в h файле должна быть указана спецификация вызова функций находящихся внутри библиоотеки, будет описано отдельно.

7. Additional include directory
Пути по которым будут разыскиваться .h файлы для библиотек которые мы включаем в проект.



8. Целевая машина (процессор)
Это процессор для которого производится целевая сборка, нас интересует либо x86 or x64, наличие в списке определяется установленными компиляторами.



Настройка конфигурации DEBUG и настройка отладки
Для конфигурации Debug включают возможность отладки, выключают

оптимизацию, вводят дополнительные defines. Естественно все define определенные в настройках конфигурации можно использовать в своем коде вставля #ifdef _DEBUG or #indef NDEBUG.