Лекция 5. Динамически связываемые библиотеки

Архитектура ОС Windows

22 октября 2014 г.

Виды использования библиотек

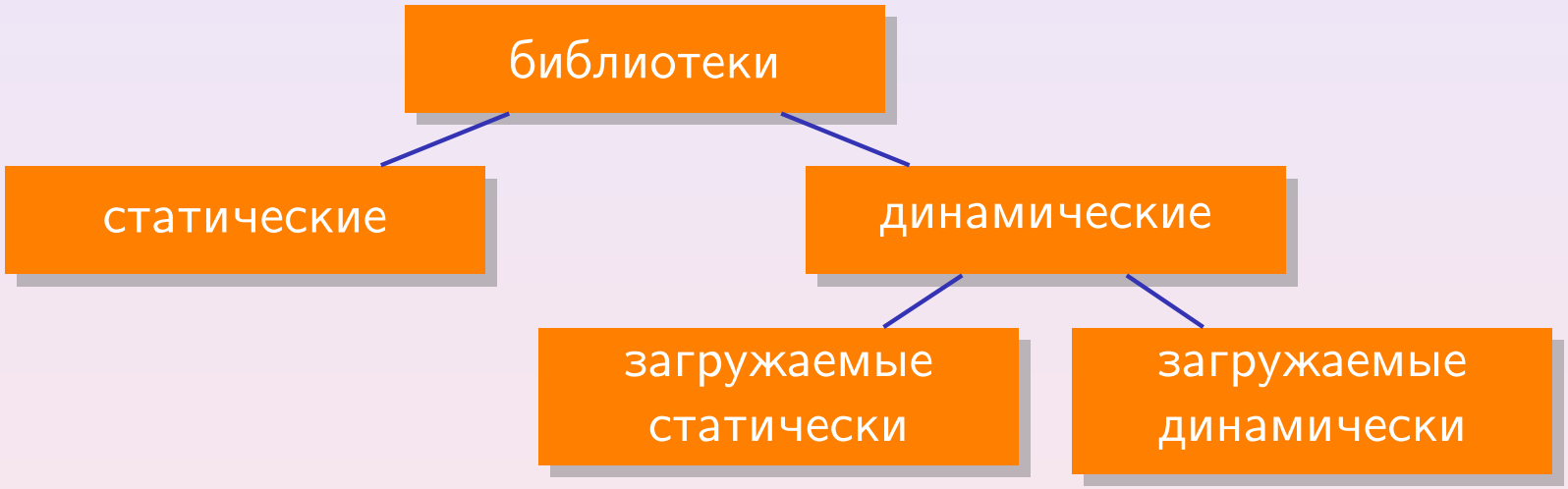

Рис. 1: виды использования библиотек

Модульный подход к проектированию ПО

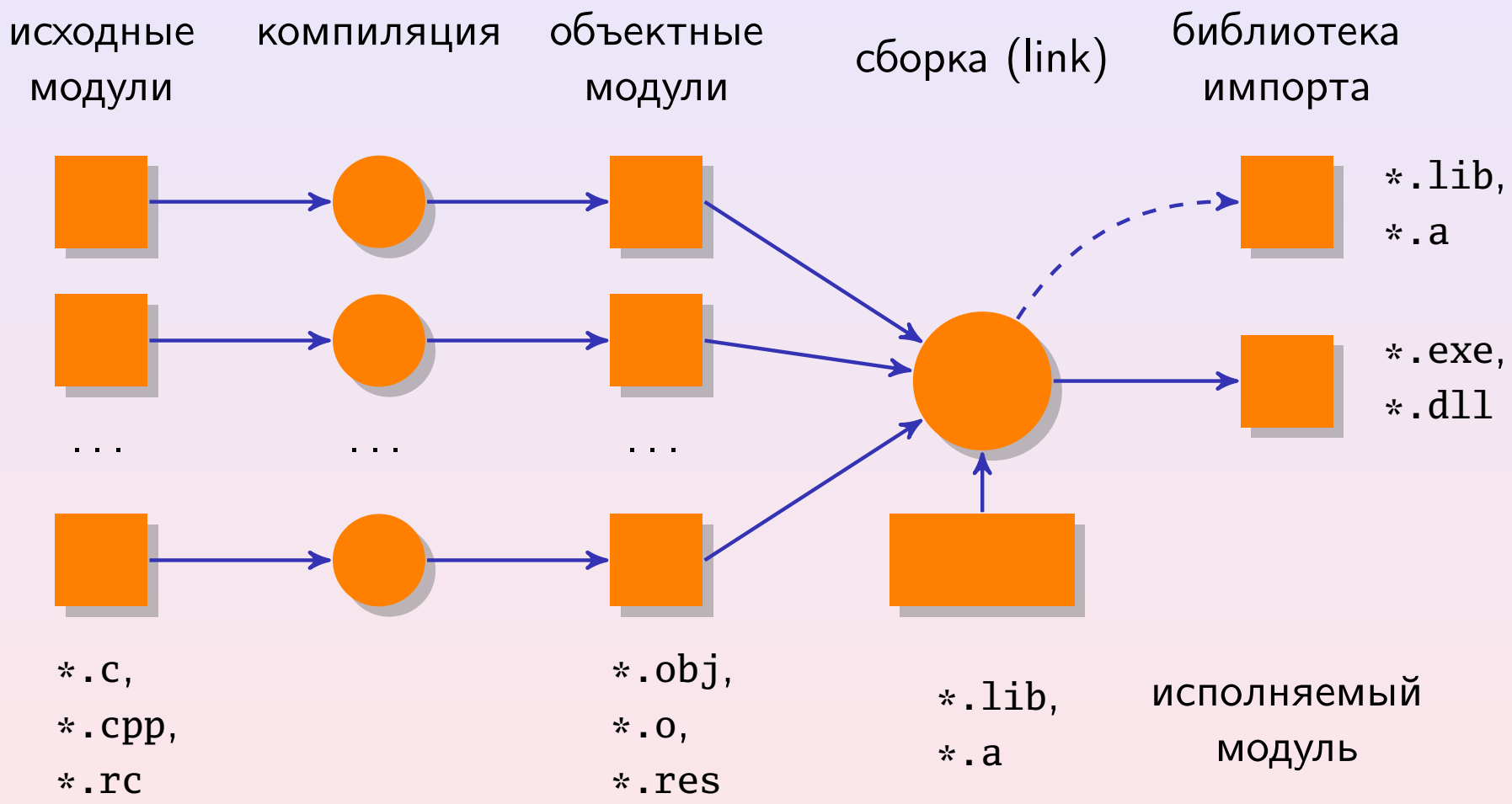

Рис. 2: схема модульной сборки программы

Определение зависимостей

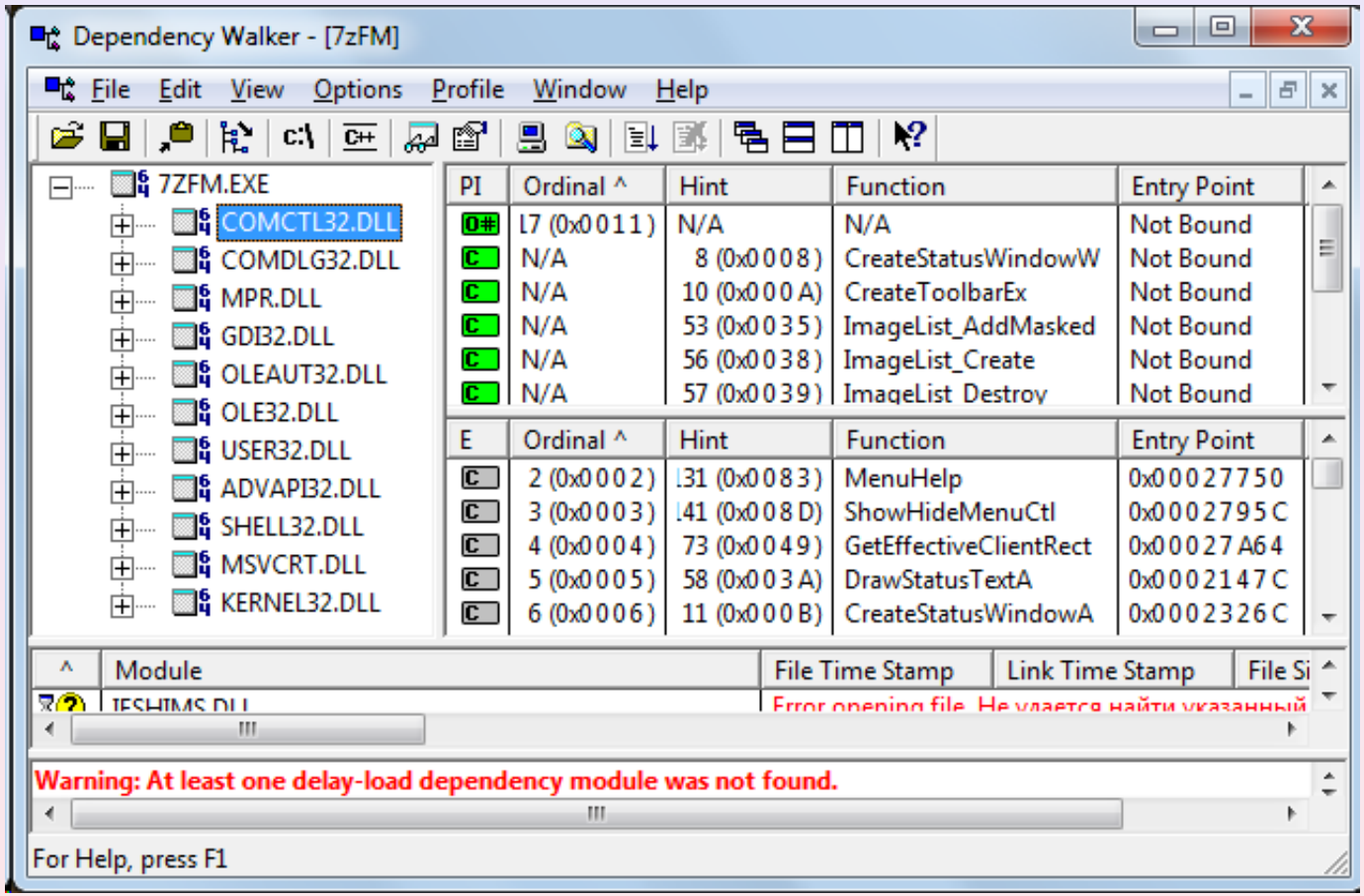

Рис. 3: Окно программы Dependency Walker

Преимущества динамических библиотек

Преимущества

- Расширяемость функциональности приложений.
- Упрощение управления разработкой проекта.
- Экономия оперативной памяти.
- Упрощение разделения ресурсов.
- Упрощение локализации.
- Устранение различий в платформах.
- Специальные функции (ловушки — hooks, COM, . . .)

Пример

Пример (with_def.h)

```
#ifndef WITH_DEF_H__
#define WITH_DEF_H__

#include <windows.h>

#ifdef __cplusplus
extern "C"
{
#endif

extern int g_n;
```

Пример (окончание)

```
void SomeFunction(
  const LPCTSTR lpctszText);
int Increment(int n);
int YetAnotherFunction(int n);

#ifdef __cplusplus
}
#endif

#endif    // WITH_DEF_H__
```

Пример (продолжение)

Пример (with_def.cpp)

```
#include "with_def.h"

int g_n = 10;

void SomeFunction(const LPCTSTR lpctszText)
{
  MessageBox(
    NULL, lpctszText, TEXT("DLL Message"),
    MB_OK | MB_ICONINFORMATION);
}
```

Пример (продолжение)

Пример (with_def.cpp, окончание)

```
extern "C" int AnotherFunction(int n)
{
  return (n + 1);
}

int YetAnotherFunction(int n)
{
  return (n - 1);
}
```

Пример (продолжение)

Пример (with_def.def)

```
LIBRARY with_def BASE=0x11000000

VERSION 1.0

EXPORTS
  SomeFunction
  Increment = AnotherFunction
  YetAnotherFunction @3 NONAME
  g_n DATA
```

Пример (окончание)

Пример (using_dll.cpp)

```
#include "with_def.h"

#include <iostream>

int main()
{
  SomeFunction(TEXT("Hello world!"));
  std::cout <<
    Increment(1) << ' ' <<
    YetAnotherFunction(1) << ' ' <<
    g_n << std::endl;
}
```

Пример

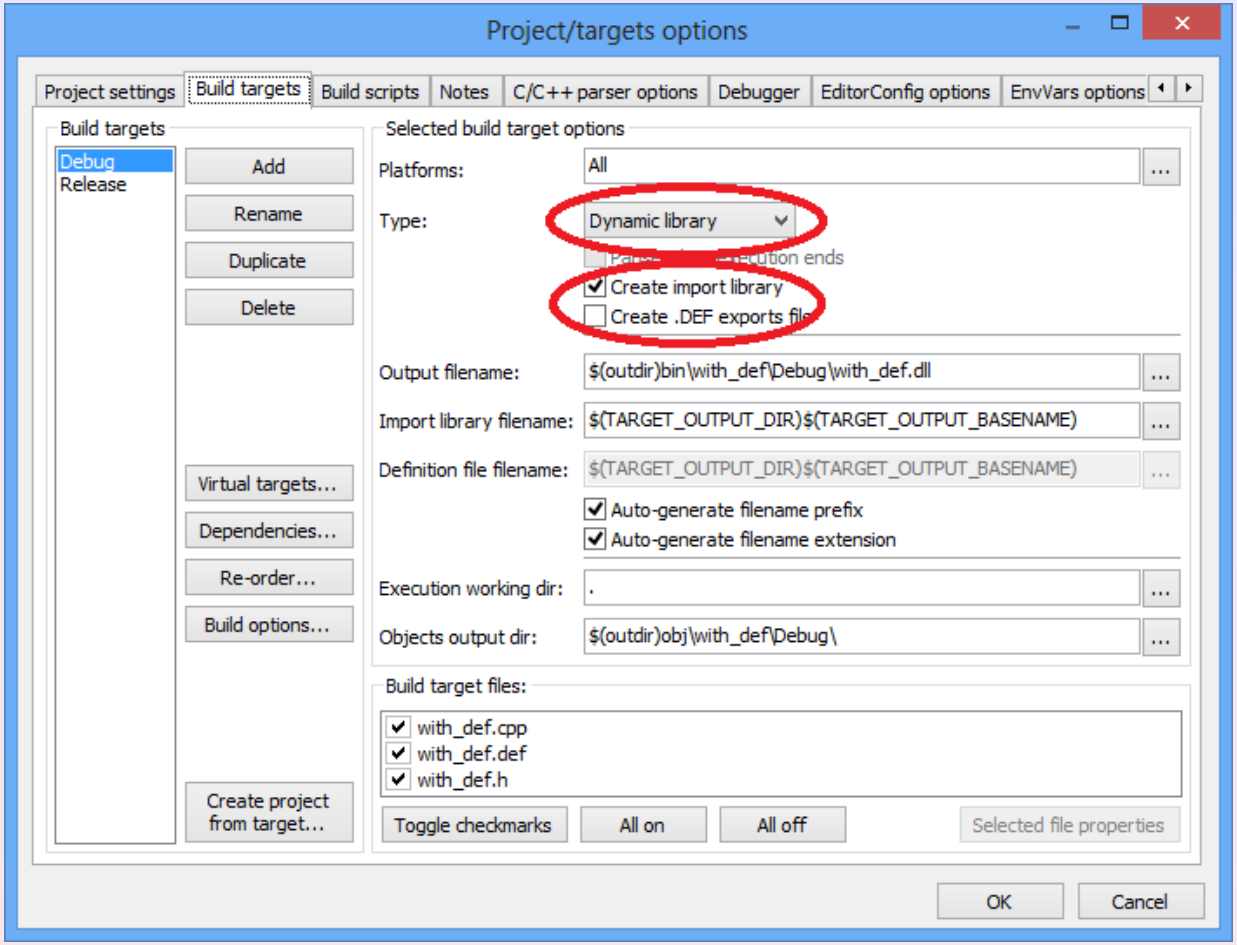

Рис. 4: настройка вывода компилятора при создании библиотеки

Пример (продолжение)

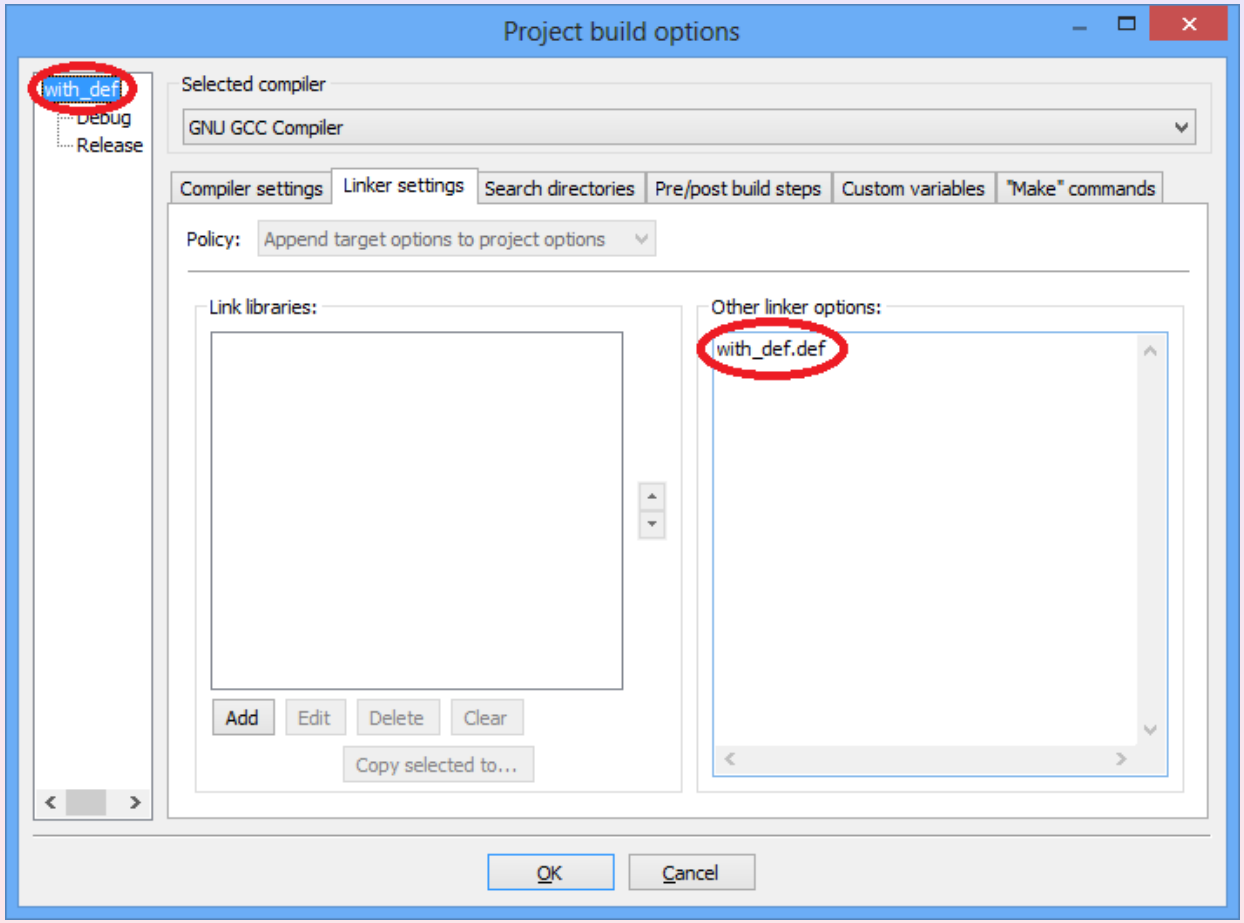

Рис. 5: настройка использования DEF-файла при сборке

Пример (продолжение)

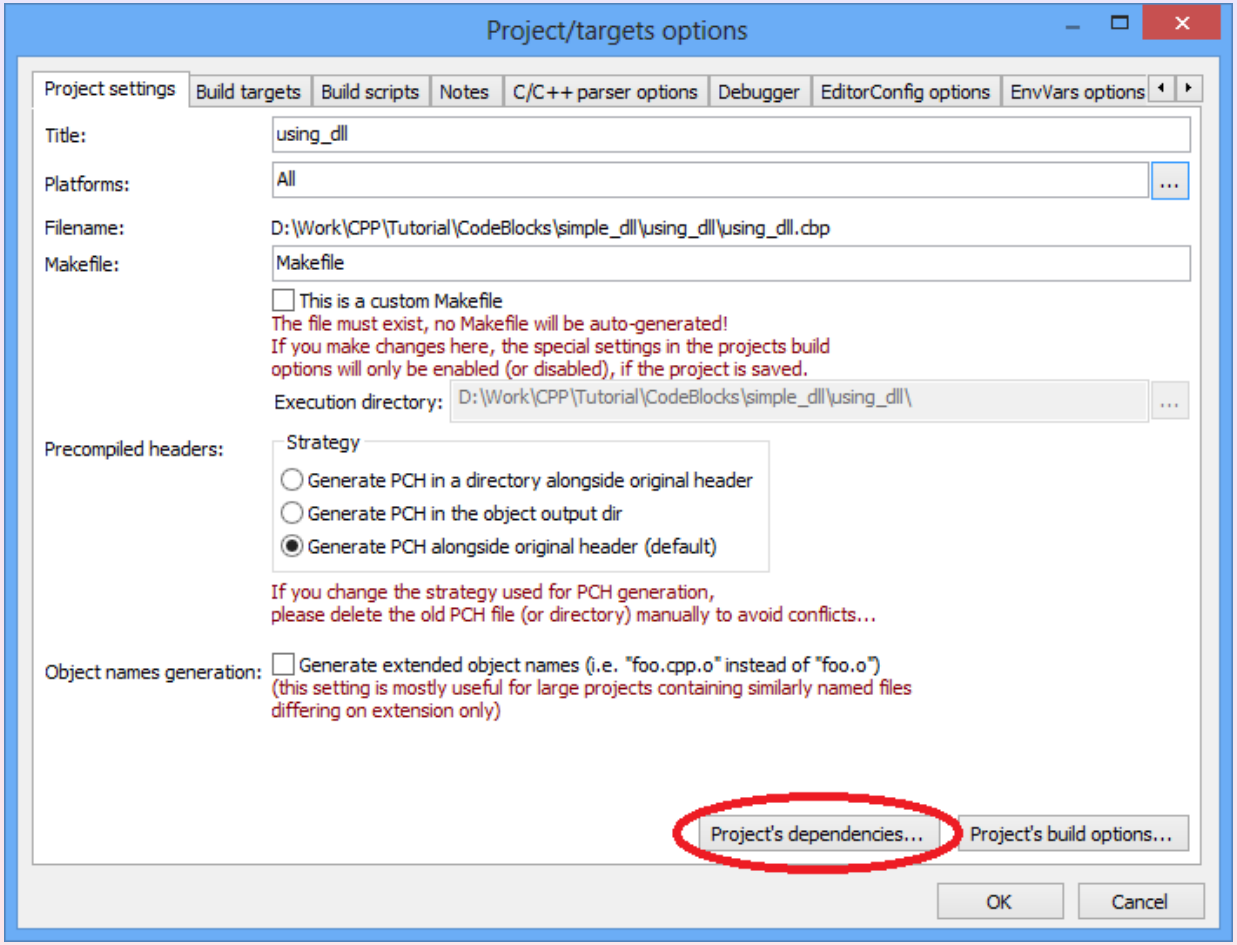

Рис. 6: установка зависимостей проекта приложения от библиотеки

Пример (продолжение)

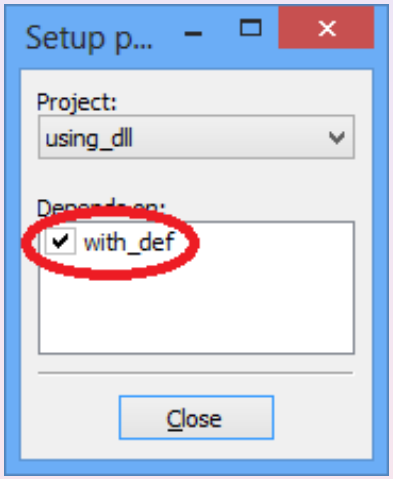

Рис. 7: настройка зависимостей

Пример (продолжение)

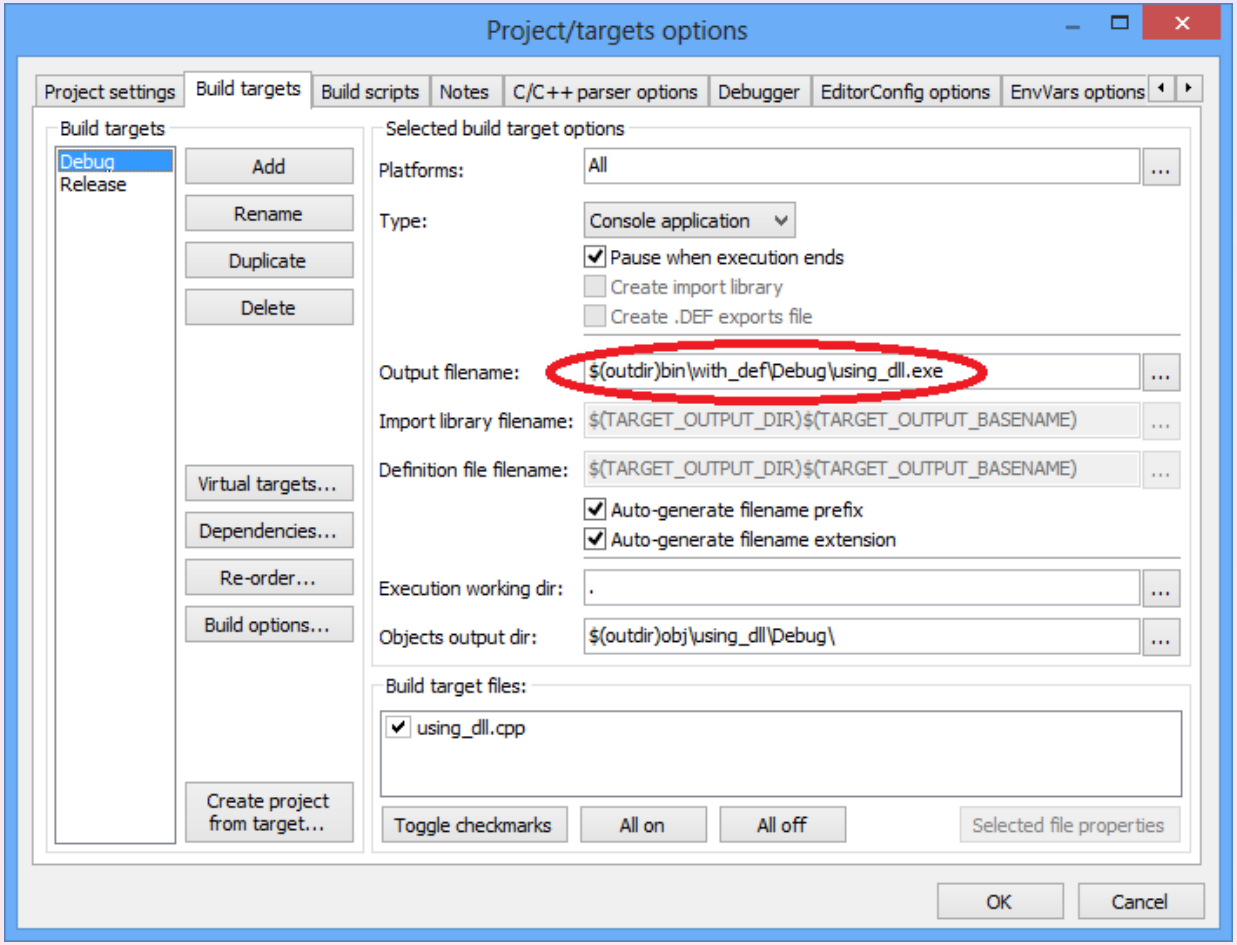

Рис. 8: настройка вывода компилятора при создании приложения

Пример (продолжение)

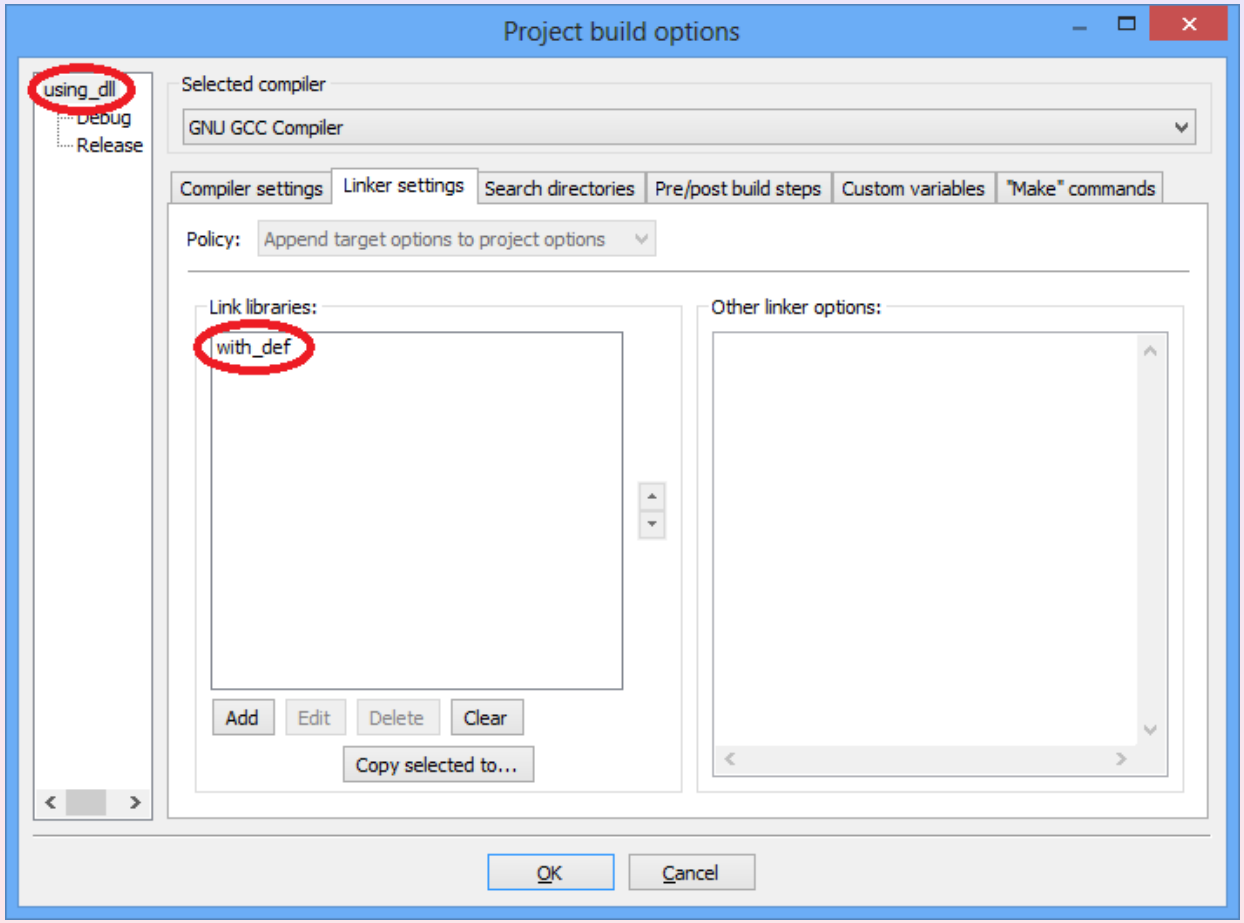

Рис. 9: настройка библиотек для сборки

Пример (продолжение)

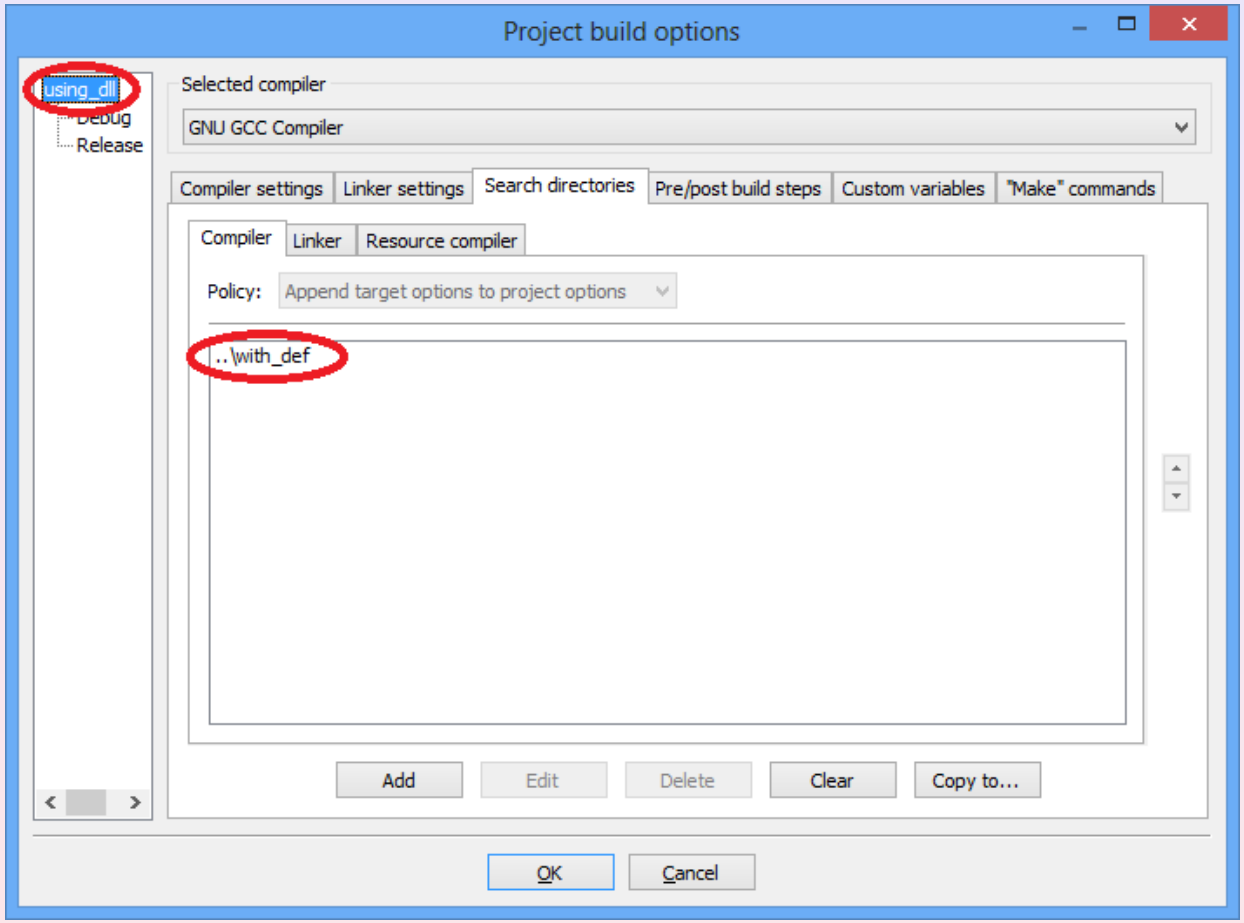

Рис. 10: настройка путей препроцессора

Пример (окончание)

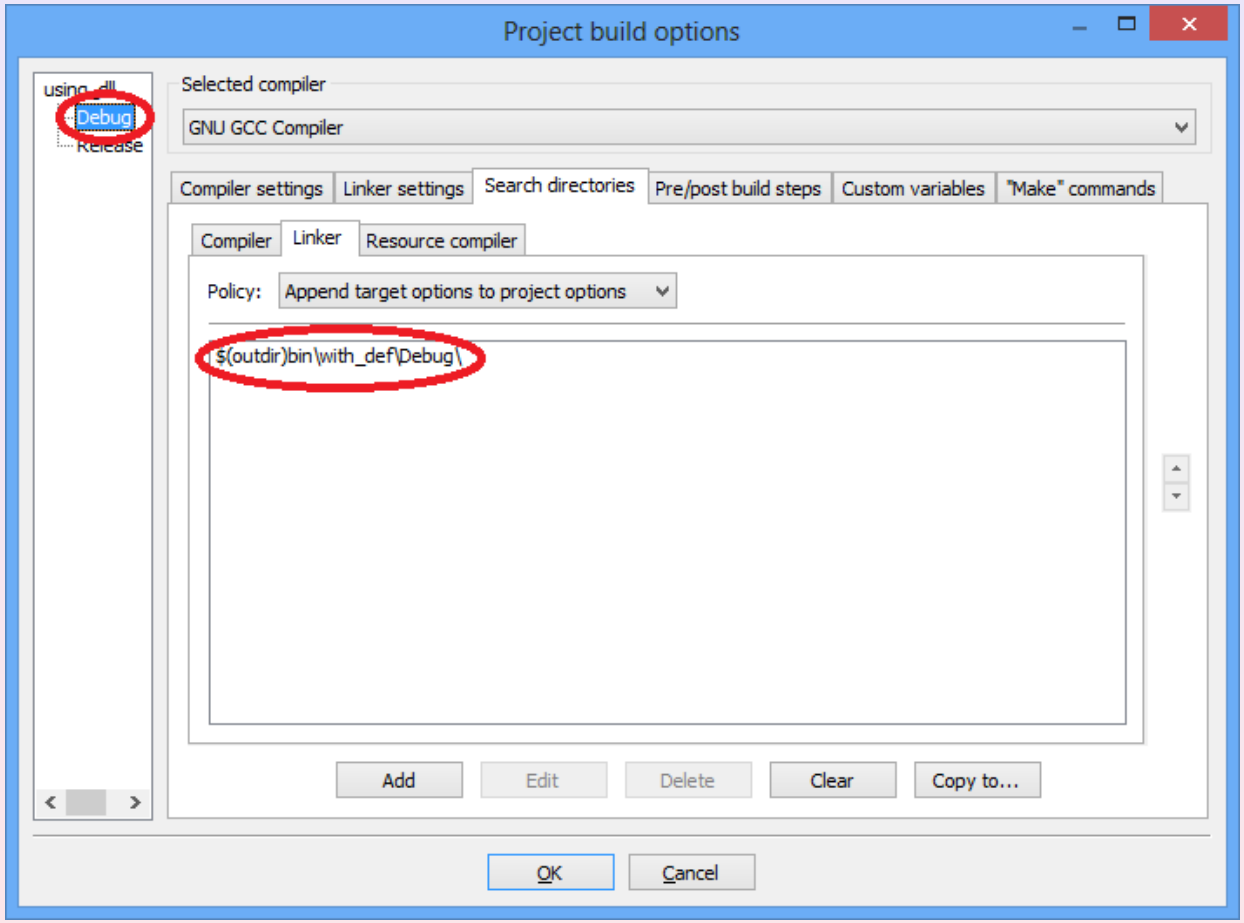

Рис. 11: настройка путей сборки

Пример

Пример (вывод утилиты DumpBin)

```
D:\_compiled\bin\with_def\Debug>dumpbin /exports with_def.dll
...
        1.00 version
           1 ordinal base
           4 number of functions
           3 number of names

    ordinal hint RVA      name

          1    0 00001242 Increment
          2    1 00001214 SomeFunction
          4    2 00002000 g_n
          3      0000124B [NONAME]
```

Пример

Пример (сборка вручную при помощи gcc-MinGW)

```
D:\with_def>g++ -c -o with_def.o with_def.cpp

D:\with_def>g++ -shared -Wl,--out-implib=libwith_def.a -Wl,--dll
  -o with_def.dll with_def.o with_def.def
Creating library file: libwith_def.a

D:\with_def>cd ..\using_dll

D:\using_dll>g++ -c -I ..\with_def -o using_dll.o using_dll.cpp

D:\using_dll>g++ -L ..\with_def -o using_dll.exe using_dll.o -l with_def
```

Пример

Пример (сборка вручную при помощи MSVC)

```
D:\with_def>cl /c with_def.cpp

D:\with_def>link /DLL /DEF:with_def.def with_def.obj user32.lib

D:\with_def>cd ..\using_dll

D:\using_dll>cl /c /EHsc /I ..\with_def using_dll.cpp

D:\using_dll>link /NOLOGO using_dll.obj ..\with_def\with_def.lib
```

Пример

Пример (with_export.h)

```
#ifndef WITH_EXPORT_H__
#define WITH_EXPORT_H__

#ifdef WITH_EXPORT_BUILD_DLL
  #define WITH_EXPORT_EXPORT __declspec(dllexport)
#else
  #define WITH_EXPORT_EXPORT __declspec(dllimport)
#endif

int WITH_EXPORT_EXPORT ExportFunction(int n1, int n2);
```

Пример (продолжение)

Пример (with_export.h, окончание)

```
class WITH_EXPORT_EXPORT ExportClass
{
public:
  //
  int GetData() const;
  void SetData(int n);
  //
private:
  //
  int m_n;
};

#endif    // WITH_EXPORT_H__
```

Пример (продолжение)

Пример (with_export.cpp)

```
#include "with_export.h"

int ExportFunction(int n1, int n2)
{
  return (n1 + n2);
}
```

Пример (окончание)

```
int ExportClass::GetData() const
{
  return m_n;
}

void ExportClass::SetData(int n)
{
  m_n = n;
}
```

Пример (окончание)

Пример (using_export.cpp)

```
#include "with_export.h"

#include <iostream>

int main()
{
  ExportClass c;
  c.SetData(99);
  std::cout <<
    ExportFunction(2, 3) << ' ' <<
    c.GetData() << std::endl;
}
```

Пример

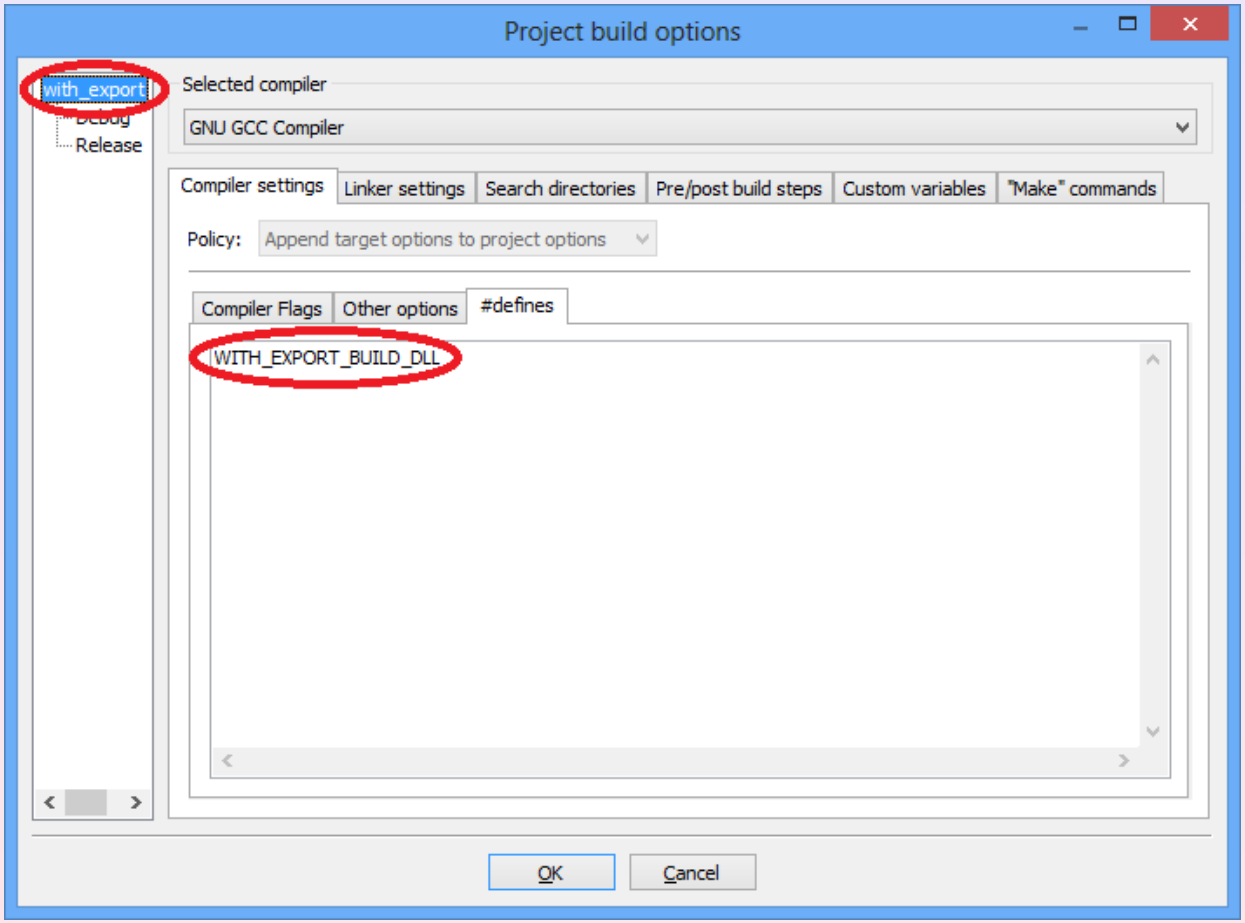

Рис. 12: настройка препроцессора

Пример

Пример (вывод утилиты DumpBin)

```
D:\_compiled\bin\with_export\Debug>dumpbin /exports with_export.dll
...
        0.00 version
           1 ordinal base
           3 number of functions
           3 number of names

    ordinal hint RVA      name

          1    0 00001214 _Z14ExportFunctionii
          2    1 00001232 _ZN11ExportClass7SetDataEi
          3    2 00001222 _ZNK11ExportClass7GetDataEv
...
```

Функции LoadLibrary() и FreeLibrary()

Определение LoadLibrary(Ex)()

```
HMODULE WINAPI LoadLibrary(
  _In_        LPCTSTR lpctszFileName
);

HMODULE WINAPI LoadLibraryEx(
  _In_        LPCTSTR lpctszFileName,
  _Reserved_ HANDLE  hFile,
  _In_        DWORD   dwFlags
);
```

Определение FreeLibrary()

```
BOOL WINAPI FreeLibrary(
  _In_        HMODULE hModule
);
```

Определение GetProcAddress()

```
FARPROC WINAPI GetProcAddress(
  _In_        HMODULE hModule,
  _In_        LPCSTR  lpcszName
);
```

Пример

Пример (using_dynamic.cpp)

```
#include <iostream>

#include <windows.h>

int main()
{
  HMODULE hLib = LoadLibrary(TEXT("with_def.dll"));
  if (hLib == NULL)
  {
    std::cerr << "Error loading with_def" << std::endl;
    return 1;
  }
```

Пример (продолжение)

Пример (using_dynamic.cpp, продолжение)

```
//
typedef void (*PFN_SOME_FUNC)(LPCTSTR);
typedef int (*PFN_YET_FUNC)(int);
//
PFN_SOME_FUNC pfnSomeFunc =
  (PFN_SOME_FUNC) GetProcAddress(hLib, "SomeFunction");
if (pfnSomeFunc == NULL)
{
  std::cerr << "Could not find SomeFunction" << std::endl;
  FreeLibrary(hLib);
  return 1;
}
```

Пример (продолжение)

Пример (using_dynamic.cpp, продолжение)

```
//
PFN_YET_FUNC pfnYetFunc =
  (PFN_YET_FUNC) GetProcAddress(hLib, MAKEINTRESOURCE(3));
if (pfnYetFunc == NULL)
{
  std::cerr << "Could not find 3-rd function" << std::endl;
  FreeLibrary(hLib);
  return 1;
}
//
```

Пример (окончание)

Пример (using_dynamic.cpp, окончание)

```
  (*pfnSomeFunc)(TEXT("Test dynamic loading"));
  //
  int n = (*pfnYetFunc)(4);
  std::cout << n << std::endl;
  //
  FreeLibrary(hLib);
}
```

Пример

Пример (DllMain())

```
BOOL WINAPI DllMain(
  HINSTANCE hInstDLL, DWORD fdwReason, LPVOID lpvReserved)
{
  switch (fdwReason)
  {
    case DLL_PROCESS_ATTACH:
      //
      // Отображение в адресное пространство процесса
      //
      break;
      //
```

Пример (продолжение)

Пример (DllMain(), продолжение)

```
case DLL_THREAD_ATTACH:
  //
  // Создание потока
  //
  break;
  //
case DLL_THREAD_DETACH:
  //
  // Корректное завершение потока
  //
  break;
  //
```

Пример (окончание)

Пример (DllMain(), окончание)

```
    case DLL_PROCESS_DETACH:
      //
      // Удаление из адресного пространства процесса
      //
      break;
      //
  }    // switch (fdwReason)
  //
  // Проверяется только при вызове с DLL_PROCESS_ATTACH
  //
  return TRUE;
}    // DllMain()
```

Пример запуска потока в DllMain()

Пример (DllMain())

```
BOOL WINAPI DllMain(
  HINSTANCE hInstDLL, DWORD fdwReason, LPVOID lpvReserved)
{
  HANDLE hThread;
  switch (fdwReason)
  {
    case DLL_PROCESS_ATTACH:
      //
      DisableThreadLibraryCalls(hInstDLL);
      hThread = CreateThread(NULL, 0, MyThreadFunction, NULL, 0, NULL);
      WaitForSingleObject(hThread, INFINITE);
      // ...
```

Загрузчик образа (`Ntdll.dll`)

Структуры данных

- Таблица импорта (IAT);
- Таблица экспорта (EAT);
- База данных модулей (module database, часть process environment block, PEB).

Порядок поиска библиотек

Безопасный режим поиска

1. Каталог, из которого было запущено приложение;
2. Системный каталог Windows (`C:\Windows\System32`);
3. 16-битный системный каталог Windows (`C:\Windows\System`);
4. Каталог Windows (`C:\Windows`);
5. Текущий каталог родительского процесса при запуске;
6. Каталоги, перечисленные в системной переменной (`%PATH%`).

Перенаправления библиотек

Перенаправления

- Наборов API (MinWin);
- Локальные:
  `MyLib.dll.local`, `C:\Program Files\My App\.LOCAL\MyLib.dll`;
- Fusion (Side-by-Side, SxS);
- Известных библиотек (`\KnownDlls32`, `\KnownDlls`).

Состав базы данных модулей

Поле	Значение
`BaseDllName`	Имя модуля без пути.
`DllBase`	Адрес, по которому модуль загружен.
`EntryPoint`	Адрес точки входа (`DllMain()` и т. п.)
`FullDllName`	Полный путь к модулю.
`LoadCount`	Счётчик загрузок (локально для процесса).
`OriginalBase`	Исходный базовый адрес модуля.

Таблица 1: основные поля базы данных модулей (`LDR_DATA_TABLE_ENTRY`)