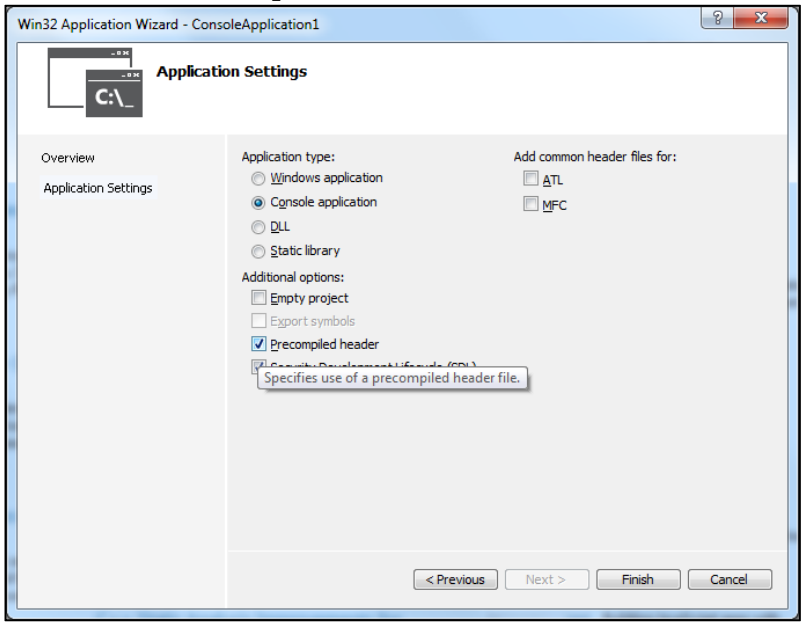
ФАЙЛ STDAFX.H

При разработке проектов на языке C/C++ в *Visual Studio* можно использовать предварительно откомпилированные заголовки (*precompiled headers*) для ускорения компиляции. Один из заголовочных файлов включает в себя другие заголовочные файлы, часто используемые в проекте и при этом редко подвергающиеся правкам. Этот файл чаще всего имеет имя stdafx.h.

Механизм предварительно откомпилированные заголовков достаточно неочевиден и имеет массу нюансов. Например, файл *stdafx.h* должен быть первым файлов, включенным в **.cpp* файл. Рассмотрим механизм использования предварительно откомпилированных заголовков подробно.

В незнакомой среде всё кажется странным и непонятным. Особенно начинающих разраьотчиков раздражает файл *stdafx.h,* из-за которого возникают странные ошибки во время компиляции. Очень часто всё заканчивается тем, что начинающий разработчик долгое время везде старательно отключает *Precompiled Headers*.



ДЛЯ ЧЕГО НУЖНЫ PRECOMPILED HEADERS

***Precompiled headers* предназначены для ускорения сборки проектов.** Обычно программисты начинают знакомиться с *C++*, используя небольшие проекты. На них сложно заметить выигрыш от *precompiled headers*. Что с ними, что без них, программа компилируется одинаковое время. Это добавляет путаницы. Человек не видит для себя пользы от этого механизма и решает, что *precompiled headers* нужны для специфичных задач и ему никогда не понадобится. И иногда считает так многие годы.

На самом деле, *precompiled headers* очень полезная технология. Пользу от них можно заметить, даже если в проекте всего несколько десятков файлов. Особенно выигрыш становится заметен, если используются такие тяжёлые библиотеки как *boost*.

Если посмотреть **.cpp* файлы в проекте, то можно заметить, что во многие включаются одни и теже наборы заголовочных файлов. Например, <stdio.h>,<locale.h>,<vector>, <string>, <algorithm> и т.д. В свою очередь, эти файлы включают другие заголовочные файлы и так далее.

Всё это приводит к тому, что препроцессор в компиляторе вновь и вновь выполняет идентичную работу. Он должен читать одни и те же файлы, вставлять их друг в друга, выбирать *#ifdef* ветки и подставлять значения макросов. **Происходит колоссальное дублирование одних и тех же операций.**

Можно существенно сократить объем работы, которую должен проделать препроцессор при компиляции проекта. Идея в том, чтобы заранее препроцессировать группу файлов и затем просто подставлять готовый фрагмент текста.

В файле "stdafx.h" находится директивы препроцессора, включающие заголовочные файлы, которые будут подключаться к проекту. Например,

```
#pragma warning(push)
#pragma warning(disable : 4820)
#pragma warning(disable : 4619)
#pragma warning(disable : 4548)
#pragma warning(disable : 4668)
#pragma warning(disable : 4365)
#pragma warning(disable : 4710)
#pragma warning(disable : 4371)
#pragma warning(disable : 4826)
#pragma warning(disable : 4061)
#pragma warning(disable : 4640)
#include <stdio.h>
#include <string>
#include <vector>
#include <iostream>
#include <fstream>
#include <algorithm>
#include <set>
#include <map>
#include <list>
#include <deque>
#include <memory>
#include <stdio.h>
```

```
#include <locale.h>
#pragma warning(pop)
```

Директивы "#pragma warning" нужны, чтобы избавиться от предупреждений, выдаваемых на стандартные библиотеки.

Теперь во все файлы **.c/*.cpp* следует включить "*stdafx.h*". Заодно стоит удалить из этих файлов заголовки, которые уже включаются с помощью *"stdafx.h"*.

При создании нового проекта *Wizard* в *Visual Studio* создаёт два файла: *stdafx.h* и *stdafx.cpp*. Именно с помощью них и реализуется механизм *precompiled headers.*

На самом деле, эти файлы могут называться, как угодно. Важно не название, а параметры компиляции в настройках проекта.

Итак, если вы воспользовались *wizard*-ом, то у вас уже есть файлы *stdafx.h* и *stdafx.cpp*. Плюс выставлены все необходимые ключи компиляции.

Заголовочный файл *"stdafx.h"* должен включаться в **.c/*.cpp* файл самым первым. Обязательно! Иначе возникнут ошибки компиляции.

Правило. Включайте в *"stdafx.h"* только те файлы, которые никогда не изменяются или меняются ОЧЕНЬ редко. Хорошими кандидатами являются заголовочные файлы системных и сторонних библиотек.

Если включаете в *"stdafx.h"* собственные файлы из проекта, соблюдайте двойную бдительность. Включайте только те файлы, которые меняются очень-очень редко.