[2. Пакет компиляторов Visual C++.](#_Toc311650967)

[2.1. Рекомендуемое оборудование.](#_Toc311650968)

[2.1.1. Минимальные требования к аппаратному и программному обеспечению.](#_Toc311650969)

[2.2. Рекомендуемое аппаратное и программное обеспечение.](#_Toc311650970)

[2.3. Выбор правильных параметров установки.](#_Toc311650971)

[2.3.1. Какую конфигурацию выбрать?](#_Toc311650972)

[2.4. Обычная установка под Windows.](#_Toc311650973)

[2.4.1. Каталоги.](#_Toc311650974)

[2.5. Документация.](#_Toc311650975)

[2.6. Система разработки.](#_Toc311650976)

[2.6.1. Новый встроенный отладчик.](#_Toc311650977)

[2.6.2. Новые встроенные редакторы ресурсов.](#_Toc311650978)

[2.6.3. Дополнительное средство TestContainer.](#_Toc311650979)

[2.6.4. Инструменты, не вошедшие в интегрированную среду.](#_Toc311650980)

[2.6.5. ProcessViewer (PView).](#_Toc311650981)

[2.6.5.1. WinDiff.](#_Toc311650982)

[2.7. Важные возможности компилятора.](#_Toc311650983)

[2.7.1. P-код.](#_Toc311650984)

[2.7.2. Предварительно откомпилированные заголовки и типы.](#_Toc311650985)

[2.7.3. Библиотека MicrosoftFoundationClass.](#_Toc311650986)

[2.7.4. Встраивание функций.](#_Toc311650987)

[2.8. Параметры компилятора.](#_Toc311650988)

[2.8.1. General.](#_Toc311650989)

[2.8.2. Debug.](#_Toc311650990)

[2.8.3. CustomBuild.](#_Toc311650991)

[2.8.4. C/С++.](#_Toc311650992)

[2.8.4.1. C++ Language.](#_Toc311650993)

[2.8.4.2. CodeGeneration.](#_Toc311650994)

[2.8.4.3. Customization.](#_Toc311650995)

[2.8.4.4. ListingFiles.](#_Toc311650996)

[2.8.4.5. Optimizations.](#_Toc311650997)

[2.8.4.6. PrecompiledHeaders.](#_Toc311650998)

[2.8.4.7. Preprocessor.](#_Toc311650999)

[2.8.5. Link.](#_Toc311651000)

[2.8.5.1. General.](#_Toc311651001)

[2.8.5.2. Customization.](#_Toc311651002)

[2.8.5.3. Debug.](#_Toc311651003)

[2.8.5.4. Input.](#_Toc311651004)

[2.8.5.5. Output.](#_Toc311651005)

[2.8.6. Resources.](#_Toc311651006)

[2.8.7. OLE Types.](#_Toc311651007)

[2.8.8. BrowseInfo.](#_Toc311651008)

1. Пакет компиляторов Visual C++.

Пакет компиляторов фирмы Microsoft предоставляет вам полную, современную среду разработки промышленного уровня для любых приложений Windows.Visual C++ версии 4.0 содержит мощный 32-разрядный компилятор, рассчитанный на разработку приложений Win32 для операционных систем Windows 95 и Windows NT. Visual C++ версии 2.0 содержит 16-разрядный компилятор для разработки 16-разрядных приложений для среды Windows 3.x и MS-DOS.

Новейшие компиляторы Visual C++ имеют много новых и улучшенных возможностей. Наиболее важные усовершенствования касаются поддержки стандарта AT&T C++ 2.1, предварительно откомпилированных заголовков, автоматического встраивания и р-кода (упакованный код).

Пакет MicrosoftVisual C++ включает в себя средства для построения программ для Windows 3.x, Windows 95 и Windows NT. Кроме этого, ваш код можетбыть адаптирован для таких аппаратных платформ, как AppleMacintosh и машин с RISC-процессорами. В пакете C++ имеются все необходимые заголовочные файлы, библиотеки, редакторы окон диалога и ресурсов, необходимые для создания действительно надежного приложения Windows. Microsoft также включила непосредственно в среду разработки редакторы ресурсов для пиктограмм, растровых изображений, курсоров, меню и окон диалога. Что касается взаимодействия приложений, то новые Мастера классов помогут вам быстро создать приложения OLE (связь и внедрение объектов), пользуясь библиотекой классов MicrosoftFoundationClasses (MFC).

Здесь вы узнаете о различных компонентах компилятора C++, о требованиях к системе и об установке среды разработки. В этой теме рассматривается пакет MicrosoftVisual C++ и рассказывается о том, как настроить его в соответствии с вашими запросами.

Если это специально не оговорено, программы могут транслироваться компилятором любой из версий. В основном рассматриваемые программы — это стандартные С или C++ приложения командной строки, которые можно запускать под MS-DOS или в окне DOS Windows 95 или Windows NT.

## Рекомендуемое оборудование.

В этом разделе даются рекомендации по аппаратному и программному обеспечению, которые помогут вам получить наилучшие результаты при использовании компилятора Visual C++. Некоторые предложения нацелены на увеличение общей производительности системы, а другие призваны сделать вашу работу более приятной.

### Минимальные требования к аппаратному и программному обеспечению.

Стандартный пакет компиляторов MicrosoftVisual C++ работает на обширном семействе компьютеров с процессорами Intel.

**ПРИМЕЧАНИЕ.**

Существуют специальные версии Visual C++ для компьютеров Macintosh, MIPS и DEC Alpha AXP.

Ниже перечислены минимальные требования Microsoft к оборудованию и программному обеспечению для работы 16- и 32-разрядной версииMicrosoftVisual C++:

* ПК с процессором 80386
* 16MB ОЗУ (минимум)
* 10МВ дискового пространства для минимальной установки
* MicrosoftWindows 95 или Windows NT для разработки приложений Win32
* Дисплей VGA
* Дисковод для дискет высокой плотности
* Привод CD-ROM (для электронной документации)
* Мышь Microsoft

## Рекомендуемое аппаратное и программное обеспечение.

Для оптимизации цикла разработки программ на С и C++ мы рекомендуем следующую конфигурацию системы:

* ПК с процессором Pentium, тактовая частота 90 МГц (или выше)
* 20МВ ОЗУ
* 100МВ свободного дискового пространства
* MicrosoftWindows 95 или Windows NT для разработки приложений Win32
* Дисплей Super VGA
* Дисковод для 3.5-дюймовых дискет высокой плотности
* Привод CD-ROM (для электронной документации)
* Мышь Microsoft

Вам понадобится процессор, который мог бы справиться с объемом и сложностью нетривиальных Windows-приложений. Большой объем памяти повышает общее быстродействие и MicrosoftVisual C++, и среды Windows. (Выигрыш в быстродействии можно также получить, имея больше свободного места на диске.)

Для 32-разрядных ПК становятся стандартом две операционные системы: Windows 95 и ее более надежный родственник Windows NT. Если у вас нет особых причин разрабатывать приложения именно под MS-DOS или Windows 3.x, мы рекомендуем отказаться от этих платформ при разработке новых приложений. Если вы еще не приобрели Windows 95 или Windows NT, сделайте это до установки пакета Visual C++ версии 4.0.

Усовершенствования, внесенные в Windows 95 и Windows NT, предоставляют вам функциональную полноту и производительность, необходимые для создания высококлассных приложений Windows. Поскольку вы будете разрабатывать эти приложения в графической среде, то лучше работать на дисплее с высокой разрешающей способностью super VGA. Покупайте монитор с большим, по возможности, экраном.

## Выбор правильных параметров установки.

В этом разделе описывается основная процедура установки системы разработки Visual C++ и прилагаемых файлов Win16 и/или Win32. Программа настройки, входящая в состав Visual C++, выполняет все необходимое для установки нужных компонент Visual C++.

Visual C++ версии 2.0 включает в себя 16-разрядный и 32-разрядный компилятор. Новая версия Visual C++, 4.0, включает только 32-разрядную среду разработки. Делайте выводы!

Программой настройки Visual C++ можно также пользоваться для установкисредств поддержки OLE и компонент Win32 для приложений Windows.

### Какую конфигурацию выбрать?

Если вы разрабатываете приложения только под 32-разрядные операционные системы, то есть Windows 95 и NT, вам понадобится лишь 32-разрядная версия компилятора. Если вы разрабатываете приложения для 16-разрядной среды Windows 3.x и MS-DOS, вам потребуется установить только 16-разрядный компилятор версии 2.0. Если же вы разрабатываете приложения для обеих программных платформ, вам придется иметь и 16-, и 32-разрядную версии компилятора.

## Обычная установка под Windows.

Установка пакета MicrosoftVisual C++ происходит почти полностью автоматически. Тем не менее, вам придется ответить на несколько вопросов. В этом разделе мы рассмотрим типичную установку 32-разрядной версии Visual C++ 4.0. Установка 16-разрядной версии компилятора происходит аналогично. Ниже приведено пошаговое описание обычной установки:

1. Из Windows 95 или Windows NT запустите программу SETUP.EXE с первой дискеты Visual C++ или CD-ROM. Если операционная система работает, версия на CD-ROM начнет установку сразу же, как только он будет вставлен в привод.
2. Вам будет представлен список возможных конфигураций, а именно Typical (Обычная), Custom (Произвольная), Minimum (Минимальная), и CD-ROM. От того, какой вариант вы выберете, зависит количество требуемого дискового пространства. Мы рекомендуем Обычную конфигурацию.
3. В этот момент начнется копирование файлов с дискет или CD-ROM на жесткий диск. Вы можете следить за процессом, наблюдая окно диалога FileCopyProcess (Процесс копирования файлов). Установка занимает более 15 минут на машине с Pentium 90МГц.
4. Когда все файлы установлены, появится запрос о том, изменить ли конфигурацию сейчас или позже. Мы рекомендуем выбрать "Makechangesnowandbackupcurrentversion" ("Внести изменения сейчас и сделать резервную копию текущей версии").
5. Вы получите запрос об установке переменных среды. Он будет дан в форме помеченного галочкой флажка. Ответьте утвердительно, и переменные среды будут установлены. Через переменные среды вы обеспечиваете компилятор важной информацией о системе.
6. Когда установка будет закончена, перезапустите всю систему, чтобы измененные установки были введены в действие.

### Каталоги.

В таблице показана типичная группа подкаталогов для Visual C++ 4.0, установленного в каталог MSdev:

Таблица. Подкаталоги Visual C++

|  |  |
| --- | --- |
| **Название** | **Назначение** |
| BIN | Исполняемые файлы и средства для построения 32-разрядных приложений |
| HELP | Файлы справки |
| INCLUDE | Заголовочные файлы C++ |
| LIB | Библиотеки языка C++ и Win32 SDK |
| MFC | Файлыбиблиотеки Microsoft Foundation Classes (MFS) |
| OLE | Файлы для построения приложений OLE |
| SAMPLES | Примеры программ |

Вы также найдете несколько файлов README, расположенных в каталоге MSdev. Эти файлы содержат самую последнюю информацию о компиляторе (и об ошибках в нем).

## Документация.

Документация по Visual C++ состоит из QuickReference (Быстрой справки) и BooksOnline (Электронных книг). Быстрая справка позволит вам быстро отыскать информацию в процессе программирования. Электронные книги представляют собой полный набор документации по Visual C++ в компьютерном формате. Каждая тема из Быстрой справки содержит ссылку на Электронные книги, где находится полная информация. Visual C++ установит Быструю справку на ваш жесткий диск, в то время как сами Электронные книги останутся на CD-ROM. Вы можете указать в настройках, где размещать файлы или откуда получать информацию, либо обращаться к Электронным книгам непосредственно из контекстно-зависимой справки (F1).

В документации содержатся, в частности, следующие разделы:

* Как пользоваться Электронными книгами
* Различные Руководства пользователя
* MicrosoftFoundationClasses (MFC)
* Программирование с использованием библиотеки MFC
* Справочник по библиотеке MFC
* Примеры по библиотеке MFC
* Технические указания по библиотеке MFC/C++

Методы программирования:

* Справочник по языку С
* Справочник по языку C++
* Справочник по стандартной библиотеке
* Справочник по классам iostream
* Справочник по препроцессору
* Примеры на C/C++
* Набор разработчика Win32 программ (Win32 SDK)
* Функции API 32
* Функции OpenGL
* Справочник программиста по Win32s
* Интерфейс Windowssockets
* Набор разработчика OLE-программ (OLE SDK)
* Справочник программиста по OLE, том 1
* Справочник программиста по OLE Automation, том 2

## Система разработки.

32-разрядный компилятор MicrosoftVisual C++ для Windows NT и Windows 95 вобрал в себя новейшие средства разработки для Windows, тесно связанные друг с другом и снабженные наглядным интерфейсом. Например, возможности отладчика CodeView теперь непосредственно доступны из встроенного редактора Visual C++. В следующих разделах перечислены отдельные утилиты, которые теперь включены прямо в компилятор MicrosoftVisual C++.

### Новый встроенный отладчик.

Microsoft использовала всю мощь своего отладчика CodeView непосредственно в среде Visual C++. Отладчик доступен из меню Build (сборка). Встроенный отладчик позволит вам выполнять программу в пошаговом режиме, считывать и изменять содержимое переменных, и даже двигаться назад по коду. Вы увидите, что это очень помогает в тех случаях, когда программа после компиляции ведет себя не так, как ожидалось.

### Новые встроенные редакторы ресурсов.

Доступ к этим редакторам осуществляется из меню Insert (вставить). Редакторы ресурсов позволяют проектировать и создавать ресурсы Windows: растровые изображения, курсоры, значки, меню и окна диалога. Ресурсы дают возможность создать для ваших программ привлекательный пользовательский интерфейс.

### Дополнительное средство TestContainer.

TestContainer(тестовый контейнер) — это спроектированное Microsoft приложение, которое позволяет быстро тестировать ваши управляющие элементы. Можно изменять свойства и возможности элемента, находящегося в тестовом контейнере. Вы узнаете, как использовать его в работе, из Приложения С.

### Инструменты, не вошедшие в интегрированную среду.

Отдельные инструменты расположены вне интегрированной среды разработки (IntegratedDevelopmentEnvironment, IDE). Некоторые из них, как например Spy++ и MFC Tracer, доступны и внутри интегрированной среды, и вне ее.

### ProcessViewer (PView).

ProcessViewer (наблюдение за процессами) позволяет быстро устанавливать параметры, необходимые для отслеживания выполняемых процессов, потоков и квантования процессорного времени. Чтобы запустить ProcessViewer, просто дважды щелкните на значке PView в группе Visual C++. ProcessViewer поможет вам ответить на следующие вопросы:

* Сколько памяти используется при выполнении различных фрагментов кода?
* Сколько памяти временно перемещено на диск?
* Какие процессы и потоки занимают большую часть процессорного времени?
* Как программа выполняется при различных значениях системных приоритетов?
* Что случается, если процесс или поток перестает отвечать на DDE, OLE или конвейерный ввод-вывод?
* Какой процент времени тратится на выполнение вызовов функций API?

WinDiff.

Утилита WinDiff также находится в группе Visual C++. Этот средство позволит вам в графическом виде сравнивать и изменять два файла или два каталога. Все возможности WinDiff работают очень сходно с соответствующими командами Windows 95 Explorer или Windows NT FileManager.

## Важные возможности компилятора.

Пакет компилятора Visual C++ включает множество усовершенствований, новых возможностей и дополнений. Следующие разделы представят вам эти улучшения и кратко пояснят их применение.

### P-код.

P-код (сокращение от packedcode — упакованный код)нацелен на оптимизацию размера и быстродействия кода. Р-код может существенно уменьшить размер программы и повысить скорость ее выполнения на величину до 60 процентов. Более того, все это достигается простым включением определенного режима компиляции. Это означает, что любая программа на С или C++ может компилироваться как обычным способом, так и с применением р-кода.

По этой технологии, исходный текст программы транслируется в "интерпретируемый объектный код", который является более высокоуровневым и компактным представлением объектного кода. В конце этого процесса в приложение встраивается небольшой интерпретирующий модуль.

Наиболее эффективное использование этой методики все же требует определенных знаний. Поскольку интерпретатор генерирует объектный код во время выполнения программы, р-код работает более медленно, чем готовый объектный код. Путем умелого использования директивы #pragma можно обеспечить генерацию р-кода в тех местах, где особенно важен размер кода, и переключиться обратно на генерацию обычного объектного кода там, где важнее скорость.

Наилучшими кандидатами на применение р-кода являются те процедуры, которые имеют дело с пользовательским интерфейсом, а поскольку многие приложения Windows тратят 50 процентов времени на работу с ним, р-код обеспечивает оптимальные показатели производительности.

### Предварительно откомпилированные заголовки и типы.

Visual C++ помещает родовые типы, прототипы функций, внешние ссылки и объявления функций-членов класса в специальные файлы, называемые заголовочными. Эти файлы содержат много важных определений, необходимых многочисленным исходным файлам, которые связываются воедино для создания исполняемой программы. Части заголовочных файлов, как правило, компилируются заново для каждого модуля, в который он включается. К сожалению, необходимость повторной компиляции участков текста приводит к снижению общей производительности компилятора. Visual C++ ускоряет процесс компиляции, позволяя вам заранее откомпилировать ваши заголовочные файлы. Хотя принцип предварительной компиляции не нов, Microsoft использует действительно новый подход. Предварительная компиляция сохраняет на определенном этапе состояние компилируемой программы и вводит соотношение между файлами с текстом программы и заранее откомпилированным заголовком. Можно создать и больше одного откомпилированного заголовочного файла на каждый из файлов с текстом программы.

Одним из лучших примеров применения этой технологии является цикл Разработки приложения, в котором происходят частые изменения исходных текстов, но определения базовых классов относительно стабильны. Если заголовочный файл откомпилирован заранее, компилятор может сосредоточиться на изменениях в текстах. Откомпилированные заголовки также снижают время компиляции тех приложений, в которых в этих файлах содержатся большие участки исходного кода данного модуля, как это часто случается в программах на C++.

Компилятор Visual C++ полагает, что текущее состояние компиляции такое же, как тогда, когда были откомпилированы какие-либо заголовки. Если компилятор заметит несоответствия, он выдаст предупреждение. Такие несоответствия могут возникать вследствие изменения модели памяти, изменения значений определенных констант, или смены параметров отладки или генерации кода.

В отличие от большинства распространенных компиляторов C++, компилятор фирмы Microsoft не ограничивает предварительную компиляцию только заголовочными файлами. Поскольку он позволяет предварительно откомпилировать программу до заданной точки, вы можете также иметь и заранее откомпилированный исходный текст. Это чрезвычайно существенно для тех программ на C++, которые содержат в заголовочных файлах большую часть определений функций-членов класса. Вообще, предварительная компиляция предназначена для тех участков текста программы, которые можно считать неизменными; она служит для уменьшения времени компиляции тех частей программы, которые находятся в процессе разработки.

### Библиотека MicrosoftFoundationClass.

Windows-программы легко использовать, однако их нелегко разрабатывать.

Многих программистов обескураживает необходимость осваивать сотнифункций Windows API, необходимых для написания приложений.

Решение Microsoft, облегчающее программисту восхождение к вершинам мастерства, состоит в применении объектно-ориентированной библиотеки базовых классов. Классы C++ гораздо легче изучить и использовать, чем функции API. Библиотека MicrosoftFoundationClass (MFC) активно пользуется достоинствами абстракции данных, предоставляемой языком C++, поэтому ее применение упрощает программирование для Windows. Начинающие программисты могут использовать классы как "кулинарную книгу", а опытные программисты могут расширить эти классы или встроить их в свою иерархию классов.

Библиотека MFC предоставляет классы для управления объектами Windows и обладает рядом классов общего назначения, которые могут быть использованы и в приложениях MS-DOS, и Windows. Например, есть классы для управления файлами, строками, временем, памятью и обработки исключительных ситуаций.

В сущности, библиотека классов MFC обладает возможностями практически каждой из функций Windows API и содержит интеллектуальный код, который упрощает обработку сообщений, диагностику ошибок и другие мелочи, обязательно входящие в каждое приложение Windows. Такое логичное сочетание и расширение функций Windows API имеет десять основных достоинств:

* Инкапсуляция Windows API логична и полна. Библиотека MFC обеспечивает поддержку всех часто используемых функций Windows API, включая управление окнами, сообщения, элементы управления, меню, окна диалога, объекты интерфейса графического устройства (шрифты, кисти, перья и растровые изображения), связь объектов и многодокументный интерфейс.
* Библиотека MFC легка для изучения.Microsoftпредпринимала целенаправленные усилия для того, чтобы сделать имена функций MFC и их параметров как можно более схожими с именами их прототипов из Windows API. Это снизит возможность путаницы для опытных программистов Windows, которые захотят воспользоваться платформой MFC. Это также позволит начинающему программисту более легко перейти на Windows API, если он к этому готов или того требует программа.
* Текст на C++ более эффективен. Если компилировать MFC в малой модели памяти, то использующее ее приложение потребует лишь немногим больше ОЗУ, чем обычное. Скорость выполнения MFC-приложения практически равна скорости аналогичного приложения на С, использующего стандартный Windows API.
* Библиотека MFC дает возможность автоматической обработки сообщений. Библиотека MFC ликвидирует один из источников частых ошибок — цикл обработки сообщений Windows API. Классы MFC спроектированы для автоматической обработки каждого из сообщений Windows. Вместо использования стандартного оператора switch/case, каждое сообщение непосредственно направляется методу класса, который предпринимает соответствующие действия.
* Библиотека MFC дает возможность самодиагностики. Более того, возможность самодиагностики туда встроена. Это означает, что вы можете сбросить информацию о различных объектах в файл, а затем проанализировать состояние внутренних переменных объекта в удобном для вас виде.
* Библиотека MFC имеет надежную архитектуру. Опережая официальное включение в стандарт ANSI С конструкции throw/catch, библиотека MFC уже основывается на архитектуре, широко использующей обработку исключительных ситуаций. Это позволяет объекту MFC изящно выходить из таких стандартных ошибочных ситуаций, как недостаток памяти, неверные параметры или трудности с загрузкой файла или ресурса. Все составляющие этой архитектуры совместимы с предлагаемыми рекомендациями ANSI С
* Библиотека MFC дает возможность использовать динамические типы данных. Это чрезвычайно мощное средство позволяет не заботиться о типе динамически размещенного объекта до момента выполнения программы. Вам дается возможность манипулировать объектом, не зная о лежащей в его основе структуре данных. Поскольку информация о типе объекта доступна во время выполнения, программист освобождается от одного лишнего уровня детализации.
* Библиотека MFC может гармонично сосуществовать с Windows-программами, написанными на С. Наиболее важное свойство MFC — ее способность уживаться с программами, написанными на С с использованием Windows API. Программисты могут в одной и той же программе сочетать классы MFC с вызовами API-функций. Это позволяет приложению MFC легко развиться до уровня настоящей объектно-ориентированной программы, если опыт программиста или внешние условия этого требуют. Такая прозрачность среды стала возможной благодаря общим для обеих архитектур соглашениям об именах. Это означает, что заголовки, типы и глобальные определения MFC не вступают в конфликт с именами из Windows API. Еще одно важное условие столь успешного взаимодействия — прозрачное управление памятью.
* Библиотека MFC может использоваться в MS-DOS. Библиотека MFC проектировалась специально для разработки приложений Windows. Многие классы обеспечивают такие часто используемые функции, как файловый ввод-вывод и работу со строками. Эти классы общего назначения могут быть использованы при программировании как для Windows, так и для MS-DOS.
* Библиотека MFC и Мастера. ClassWizard (Мастер классов) и ControlWizard (Мастер элементов управления) генерируют коды, предназначенные для работы только с использованием MFC. Эти средства быстрой разработки в особенности облегчают создание приложений OLE.

### Встраивание функций.

КомпиляторMicrosoftVisual C++ обеспечивает полную поддержку встраиваемых функций. Это означает, что функция, содержащая комбинацию команд любого типа, может быть встроена в программу в месте ее вызова. Многие распространенные компиляторы C++ не допускают встраивание определенных операторов или выражений, — например, встраивание не применяется к функциям, содержащим оператор switch, while или for.Компилятор Visual C++ позволяет встраивать наиболее критичные по времени функции (включая редко используемые функции-члены классов или конструкторы) без ограничений на их содержание. Эту возможность можно включить через меню Project (проект), выбрав Settings (параметры), затем папку C/C++ и, наконец, Optimizations (оптимизация) из списка Category (категории).

## Параметры компилятора.

Рассматриваемые здесь компиляторы MicrosoftVisual C++ относятся к глобально оптимизирующим, которые позволят вам пользоваться различными возможностями оптимизации по скорости и размеру кода для всех типов разработки. В этом разделе мы рассмотрим параметры, непосредственно относящиеся к 32-разрядному компилятору Microsoft C++ версии 4.0. Если вы используете 16-разрядный или 32-разрядный компилятор версии 2.0, то параметры будут схожими, но могут располагаться на других страницах окон диалога.

Рассматриваемые ниже параметры позволяют оптимизировать скорость, размер исполняемого модуля или время компиляции и сборки. Если вы не наблюдаете заметного скачка производительности, то, возможно, ваше приложение содержит недостаточно кода. Все параметры устанавливаются путем выбора пункта Settings (параметры) из меню Build (сборка).

### General.

Закладка General(общие параметры) позволяет указать, следует или нет использовать библиотеку MicrosoftFoundationClass. Можно также задать каталоги, куда будут помещены промежуточные и окончательные результаты компиляции.

### Debug.

Закладка Debug(отладка) позволяет указать местоположение исполняемого файла и рабочий каталог, дополнительные параметры для программы, а также путь и имя файла для дистанционной отладки. Кроме того, можно задать дополнительные динамически подключаемые библиотеки (DLL) при помощи списка Category.

### CustomBuild.

При помощи закладки CustomBuild (дополнительные средства) вы можете задать дополнительные инструменты для построения программы. Сюда входят, например, программы, обрабатывающие выходной файл конфигурации проекта.

### C/С++.

Закладка C/C++позволяет выбрать одну из следующих категорий:

* General
* ++ Language
* CodeGeneration
* Customization
* ListingFiles
* Optimizations
* PrecompiledHeaders
* Preprocessor

C++ Language.

Категория General(общие параметры) позволяет установить порог выдачи предупреждающих сообщений, указать отладочную информацию, установить оптимизацию компиляции и перечислить дополнительные параметры проекта.

CodeGeneration.

Категория CodeGeneration(генерация кода) позволяет задать целевой микропроцессор (от 80386 до Pentium), выбрать соглашение о вызовах, указать стандартную библиотеку и задать метод выравнивания элементов структуры. Опять же, перечисляются дополнительные параметры.

Customization.

Категория Customization(адаптация) позволяет включить или отключить следующие возможности:

* расширения языка
* компоновка на уровне функций
* идентичные строки
* минимальная перекомпиляция
* инкрементная компиляция
* отмена заставки и информационных сообщений

ListingFiles.

Категория ListingFiles (файлы листинга) позволяет включить генерацию информации для браузера. Кроме того, можно указать местоположение файла для браузера. Можно разрешить включение локальных переменных в информацию браузера. Дополнительно можно задать типы файлов. Перечислены параметры проекта.

Optimizations.

Категория Optimizations(параметры оптимизации) позволяет установить различные варианты оптимизации, например, по скорости или по размеру.

Задается также возможность встраивания функций. Перечислены параметры проекта.

PrecompiledHeaders.

Категория PrecompiledHeaders (предварительно откомпилированные заголовки) позволяет использовать заранее откомпилированные заголовочные файлы. Это файлы с расширением РСН. Предварительная компиляция заголовочных файлов ускоряет процесс компиляции и сборки, но по завершении проекта их следует удалить, так как они занимают много места. Перечислены также параметры проекта.

Preprocessor.

Категория Preprocessor(препроцессор) позволяет задать макроопределения для препроцессора. Можно также указать дополнительные каталоги и отменить стандартные пути. Перечислены также параметры проекта.

### Link.

Закладка Link(компоновка) позволяет выбрать одну из следующих категорий: - General -Customization - Debug -Input -Output

General.

В категории General(общие параметры) можно задать имя и расширение файла. Чаще всего расширение файла будет .ЕХЕ, однако вы еще узнаете о разработке приложений с расширениями .DLL и .SCR. Можно также ввести дополнительные объектные модули и библиотеки. Это может потребоваться для приложений мультимедиа, поскольку некоторые необходимые библиотеки не включаются в компоновку автоматически. Можно также включить следующие возможности:

* отладочная информация
* инкрементная компоновка
* профилирование
* отмена библиотек, включаемых по умолчанию
* генерация карты компоновки

Customization.

Категория Customization(адаптация) позволяет включить следующие возможности:

* инкрементная компоновка
* база данных программы
* имя выходного файла
* вывод сообщений процесса
* заставка

Debug.

Категория Debug(отладка) позволяет задать различные форматы для создания карты компоновки и отладочной информации.

Input.

Категория Input(ввод) позволяет указать объектные модули и библиотеки. Кроме того, можно задать имена файла перекрестных ссылок и файла, где находится DOS-заголовок программы.

Output.

Категория Output(вывод) позволяет установить базовый адрес, точку входа, размер стека и информацию о версии проекта.

### Resources.

Закладка Resources(ресурсы) позволяет задать имя файла ресурсов (обычно он имеет расширение .RES). Можно дополнительно указать язык ресурсов, каталоги для ресурсов и задать препроцессорные определения.

### OLE Types.

Закладка OLE Types(типы OLE) позволяет задать имя выходного файла, имя выходного заголовочного файла, определения для препроцессора и заставку.

### BrowseInfo.

Закладка BrowseInfo(информация браузера) дает возможность указать имя файла с информацией браузера. Кроме того, можно включить возможность генерации информации браузера и заставки.