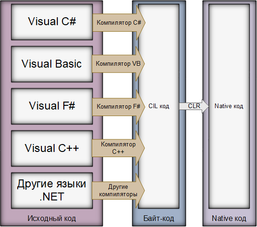
.NET framework -- программная платформа, обеспечивает совместное использование разных языков программирования, а также безопасность, переносимость программ и общую модель программирования для платформы Windows. Языки, используемые на платформе: C++/CLI, C#, Visual Basic, F#, Iron Pithon, Iron Ruby, ассемблер IL.  
Узнать о частях из которых состоит .NET framework можно рассмотрев процесс выполнения программы.   
1) Код, написанный на языке из семейства поддерживающего .NET, скачала переводится в промежуточный байт-код CIL (common intermediated Language ) (раньше назывался MSIL ). Получается СБОРКА   
2) Сборка транслируется виртуальной машиной CLR common language runtime в исполняемый код для процессора. Трансляция происходит посредством встроенного в CLR JIT-компилятор. 

**.NET FRAMEWORK – решение следующих проблем   
1. Интеграция языков программирования.** CLS (Common Language Specification) – общеязыковая спецификация, предназначенная для разработчиков компиляторов.   
CTS (Common Type Systems)- спецификацию типов, которые должны поддерживаться всеми языками ориентированными на CLR. Microsoft выпустил несколько компиляторов соответствующих этой спецификации: С++/CLI (C++ с управляемыми расширениями), С#, VB .NET, JScript.  **2. Работа на многих платформах.**При компиляции кода компиляторы NET Framework генерируют код на промежуточном языке (CIL, Common Intermediate Language). При исполнении CLR транслирует CIL-код в команды соответствующего процессора. В принципе, однажды .NET Framewor kприложение должно работать везде, где установлены и работают CLR и FCL.  
**3. Упрощенное повторное использование кода.** CLR позволяет типы разработанные на одном языке использовать в других языках.  
**4. Автоматическое управление памятью.**CLR автоматически отслеживает использование ресурсов. Сборщик мусора.  
**5. Проверка безопасности типов.** При работе в CLR практически исключена возможность записать (стереть) данные в область памяти, которая для этого не предназначена. Нет возможности передать управление в произвольную точку. **6. Единый принцип обработки сбоев.** Один из самых неприятных моментов в Window программирование – это отсутствие единой системы обработки ошибок и сбоев: возврат функций, коды состояний, HRESULT, исключения и т.п. Для обработки ошибок и сбоев в CLR используется только механизм исключений.

**7. Взаимодействие с существующим кодом.** ► Поддерживаются функции Win32 DLL – библиотек.

**8. Проблемы с версиями.** ► В Windows возникают проблемы связанные с совместимостью DLL-библиотек. .NET Framework приложение всегда работает с компонентами с которыми компилировалось и тестировалось приложение.

Метаданные - двоичный набор таблиц данных: типы и их члены портируемые типы и их члены

Назначение:   
1)устраняют необходимость в заголовочных файлах (прототипы);   
2) используются в VS для подсказок;   
3) используется при верификации кода на предмет безопасных операций;   
4) можно сериализовать объект на одной машине и восстановить состояние объекта на другой машине;   
5) используются при сборке мусора.

**Манифест**   
 набор таблиц метаданных файлы, которые входят в сборку, общедоступные экспортируемые типы,   
 файлы ресурсов или данных

►**Сборка (assembly)** —   
1) это абстрактное понятие, для логической группировки одного или нескольких управляемых модулей или файлов ресурсов.   
2) дискретная единица многократно используемого кода внутри CLR (Exe, dll)

**Исполнение сборки JIT-компилятор (Just-In-Time)**

1) CLR ищет типы данных и загружает во внутренние структуры

2) Для каждого метода CLR заносит адрес внутренней CLR функции JITCompiler

3) JITCompiler ищет в метаданных соответствующей сборки IL-код вызываемого метода, проверяет и компилирует IL-код в машинные команды

4) Они хранятся в динамически выделенном блоке памяти.

5) JITCompiler заменяет адрес вызываемого метода адресом блока памяти, содержащего готовые машинные команды

6) JITCompiler передает управление коду в этом блоке памяти.

**Типы сборок:**

►с нестрогими именами (weakly named assemblies)

► со строгими именами (strongly named assemblies).

 подписаны при помощи пары ключей,

уникально идентифицирующей издателя

сборки (безопасность, управление ее

версиями, развертывание в любом месте

пользовательского жесткого диска или в

Интернете)

 атрибуты: имя файла (без расширения), номер

версии, идентификатор регионального стандарта и

открытый ключ.

**IL является стековым языком**; это означает, что все его инструкции заносят операнды в исполнительный стек и извлекают результаты из стека. IL не содержит инструкций для работы с регистрами, и это упрощает создание новых языков и компиляторов, генерирующих код для CLR. IL-код обеспечивает безопасность приложения и его устойчивость перед ошибками. В процессе компиляции IL в машинные инструкции CLR выполняется процедура, называемая верификацией — анализ высокоуровневого кода IL и проверка безопасности всех операций.

Поскольку типы занимают центральное место в CLR, компания Microsoft разработала формальную спецификацию **CTS (Common Type System),** которая описывает способ определения и поведение типов. CTS также задает правила видимости типов и доступа к членам типа.

компания Microsoft определила спецификацию **CLS (Common Language Speciication);** в ней перечислен минимальный набор возможностей, которые должны поддерживаться компилятором для генерирования типов, совместимых с другими компонентами, написанными на других CLS-совместимых языках на базе CLR.

