

Сборки

Приложения .NET создаются за счет использования вместе некоторого количества сборок.

Сборка представляет собой

- поддерживающий версии
- самоописываемый
- двоичный файл, обслуживаемый CLR.

Сборки .NET имеют точно такие же файловые расширения (*.exe или *.dll), как и «старые» двоичные файлы Windows, внутри они устроены совсем иначе.

Преимущества предлагает сборок:

- Сборки обеспечивают повторное использование кода
- Сборки определяют границы типов;
- Сборки являются единицами, поддерживающими версии;
- Сборкам .NET присваивается состоящий из четырех частей числовой (номер версии в формате <старший номер>.<младший номер>.<номер сборки>.<номер редакции>)
- Сборки являются самоописываемыми (содержат информацию о каждой из внешнихборок, к которой им нужно иметь доступ, чтобы функционировать надлежащим образом);
- Сборки поддаются конфигурированию (могут развертываться как «приватные» (private) или как «разделяемые» (shared)).

Формат сборки .NET

Сборка .NET (*.dll или *.exe) включает в себя следующие элементы:

- заголовок файла Windows;
- заголовок файла CLR;
- CIL-код;
- метаданные типов;
- манифест сборки;
- дополнительные встроенные ресурсы.

Формат сборки .NET

Заголовок файла Windows

Заголовок файла Windows указывает на тот факт, что сборка может загружаться и использоваться в ОС семейства Windows. Также заголовок идентифицирует тип приложения (консольное, приложение с GUI или библиотека кода *.dll).

Заголовок файла CLR

Представляет собой блок данных, который должны обязательно поддерживать все сборки .NET для того, чтобы они могли обслуживаться в CLR-среде.

Заголовок содержит флаги, которые позволяют исполняющей среде разобраться в компоновке управляемого файла, - флаги указывающие, где внутри файла находятся метаданные и ресурсы, версия исполняющей среды, на которую ориентировалась данная сборка, значение открытого ключа, и т.д.

Формат сборки .NET

CIL-код, метаданные типов и манифест сборки

В своей основе любая сборка содержит **код на языке CIL**, который, представляет собой промежуточный язык, не зависящий ни от платформы, ни от процессора.

В сборке содержатся **метаданные**, которые полностью описывают формат находящихся внутри нее типов, а также внешних типов, на которые она ссылается.

В сборке должен обязательно содержаться ассоциируемый с ней **манифест** (по-другому еще называемый метаданными сборки). В этом манифесте описан каждый входящий в состав сборки модуль, версия сборки, а также любые внешние сборки, на которые ссылается текущая сборка

Формат сборки .NET

Необязательные ресурсы сборки

В любой сборке .NET может также содержаться любое количество вложенных ресурсов, таких как значки приложения, графические файлы, звуковые фрагменты или таблицы строк.

Однофайловые и многофайловые сборки

Сборка может состоять из нескольких *модулей* (двоичных файлов .NET).

В однофайловых сборках все необходимые элементы (заголовки Windows и CIL, метаданные типов, манифест и необходимые ресурсы) размещаются внутри единственного пакета *.exe или *.dll.

Многофайловая сборка состоит из набора модулей .NET, которые развертываются в виде одной логической единицы и снабжаются одним номером версии. Один из этих модулей называется *главным модулем* и содержит *манифест сборки*, в котором описаны все остальные связанные модули, от которых зависит его работа.

Преимущества:

- предоставляют очень эффективный способ для выполнения загрузки содержимого – возможность выборочной загрузки модуля, а всей сборки.
- позволяют создавать модули на различных языках программирования .NET

По способу взаимодействия с другими сборками и приложениями сборки можно разделить на две категории:

- закрытые
- разделяемые

Закрытое развертывание (private deployment) - сборки, предназначенные исключительно для одного приложения, помещают в базовый каталог приложения или в его подкаталог.

Глобально развертываемое развертывание - сборки могут быть совместно использованы несколькими приложениями

Разделяемые сборки находятся в глобальном кэше сборок (**Global Assembly Cache**).

Местоположение GAC отличается в зависимости от версии .NET, установленной на локальной машине.

до .NET 4.0

C:\Windows\assembly

с версии .NET 4.0

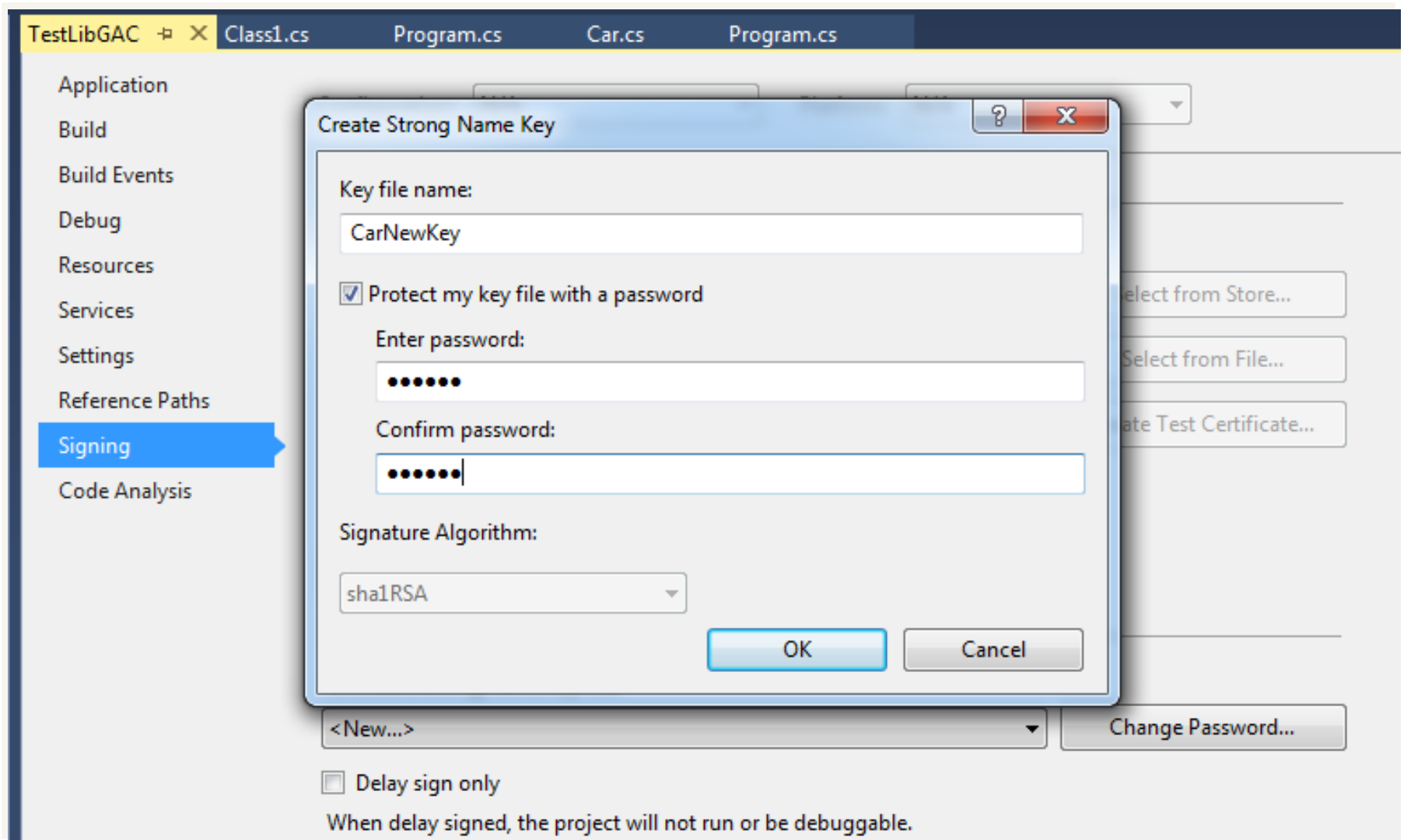
C:\Windows\Microsoft.NET\assembly\GAC_MSIL

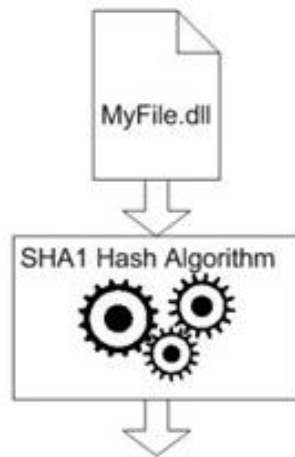
Чтобы поместить сборку в GAC сборка должна обладать **строгим именем**.

В состав строгого имени входят следующие компоненты:

- имя сборки без расширения
- номер версии. Благодаря разграничению по версии можно использовать разные версии одной и той же сборки
- открытый ключ
- необязательное значение для культуры (при локализации сборки)
- цифровая подпись, которая создается с помощью хэш-значения содержимого сборки и значения секретного ключа. Секретный ключ представляет собой файл с расширением **.snk*.

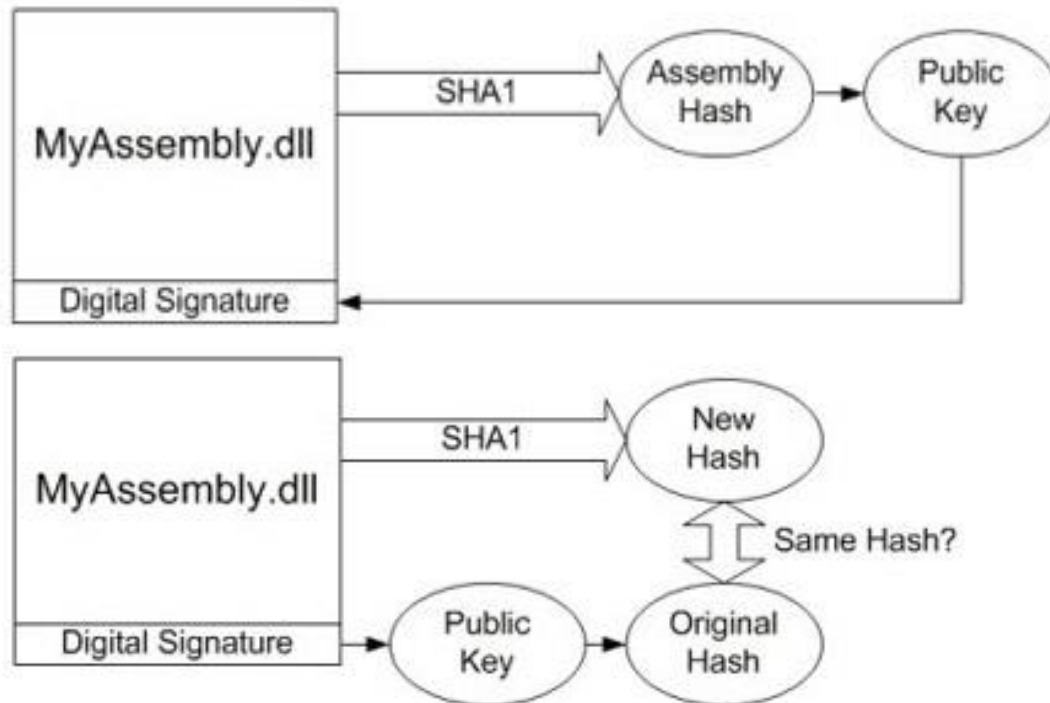
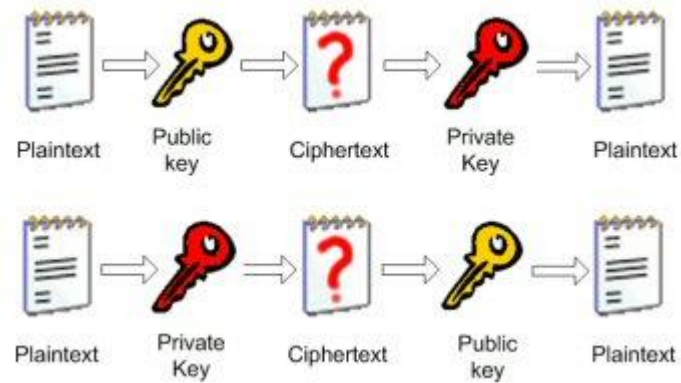
Благодаря строгому имени гарантируется уникальность сборки в глобальном кэше.

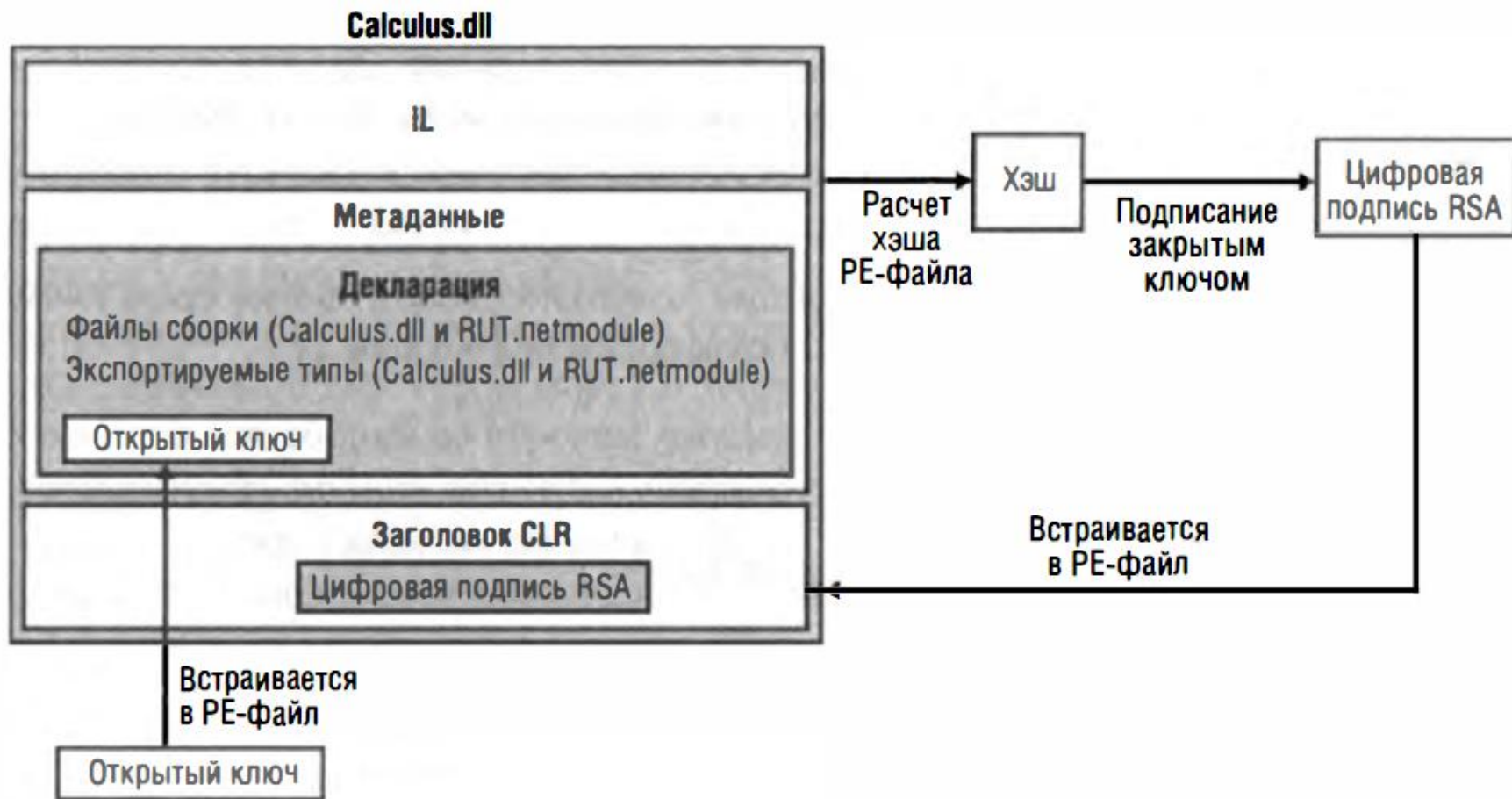




Hash Value:

559E1659B19E0EF7A034C9FC4E7F426CC0BB3B5F





переносимый исполняемый (portable executable, PE)

Получение списка всех сборок в GAC

```
"C:\Program Files (x86)\Microsoft  
SDKs\Windows\v8.1A\bin\NETFX 4.5.1  
Tools\gacutil.exe"/l (буква "l")
```

Добавление сборки в GAC

```
"C:\Program Files (x86)\Microsoft  
SDKs\Windows\v8.1A\bin\NETFX 4.5.1  
Tools\gacutil.exe"/i <Полное_имя_сборки>
```

Удаление сборки из GAC

```
"C:\Program Files (x86)\Microsoft  
SDKs\Windows\v8.1A\bin\NETFX 4.5.1  
Tools\gacutil.exe"/u <Параметры_сборки>
```

"C:\Program Files (x86)\Microsoft
SDKs\Windows\v8.1A\bin\NETFX 4.5.1 Tools\gacutil.exe"/I

"C:\Program Files (x86)\Microsoft
SDKs\Windows\v8.1A\bin\NETFX 4.5.1 Tools\gacutil.exe"/i
"TestLibGAC.dll«

(в некоторых случаях нужно указать путь к сборке)

"C:\Program Files (x86)\Microsoft
SDKs\Windows\v8.1A\bin\NETFX 4.5.1 Tools\gacutil.exe"/u
"TestLibGAC,Version=1.0.0.0, Culture=neutral,
PublicKeyToken=07a1b63e5000576f"

