

Домены приложений. NT-службы

№ урока: 17 Курс: C# Professional

Средства обучения: Компьютер с установленной Visual Studio

Обзор, цель и назначение урока

Разработчикам часто приходится запускать внешние сборки. Однако запуск внешней сборки может привести к неэффективному расходованию ресурсов и возникновению уязвимостей в защите. Лучший способ держать эти риски под контролем – создать домен приложения и запустить сборку внутри него, под надежной защитой.

Изучив материал данного занятия, учащийся сможет:

- Понимать назначение доменов приложений
- Создавать домены приложений
- Запускать сборку внутри домена приложения
- Выгружать домен приложения
- Понимать назначение служб
- Создавать собственные службы и устанавливать их в системе

Содержание урока

1. Введение в домены приложения
2. Класс System.AppDomain
3. Создание доменов
4. Выполнение сборок в доменах
5. Выгрузка доменов
6. Конфигурирование доменов приложений
7. Запуск сборок с ограниченными привилегиями
8. Введение в службы Windows
9. Создание служб
10. Установка служб

Резюме

- Домен приложения – логический контейнер, позволяющий выполнять внутри одного процесса несколько сборок, не предоставляя им прямого доступа к памяти внутри других сборок. Домены поддерживают многие из функций, стандартных для процессов, включая разграничение областей памяти и доступа к ресурсам. Однако домены приложений эффективнее процессов, поскольку в одном домене приложений могут работать несколько сборок, но без издержек на запуск отдельных процессов.
- Ключевое различие между доменом и процессом: Доменами приложения управляет исполняющая среда .NET Framework, а процессами – операционная система.
- Обычно домены приложений для сборок автоматически создает хост исполняющей среды. Примерами хостов в таком случае могут быть, например, ASP .NET, Internet Explorer, сама операционная система.
- Программист может создавать собственные домены приложений, чтобы вызывать сборки, не опасаясь некорректных действий со стороны сборки или доступа к ресурсам, запрещенным для нее.
- Сборки изолируют не только в целях защиты, но и для повышения надежности и производительности:
 - Повышение надежности – используя домены приложений для изоляции задач, которые могут вызвать аварийное завершение процесса. Если домен приложения становится нестабильным, его можно выгрузить, не затрагивая процесс в целом. Это особенно важно, когда процесс должен длительное время работать без перезапуска.

- Повышение производительности – если сборка загружена в домен приложения по умолчанию, ее нельзя выгрузить из памяти, пока выполняется процесс. Но если открыть второй домен приложения, загрузить в нем сборку, эта сборка может быть выгружена вместе с доменом приложения. Эта методика позволяет сократить рабочий набор процессов, работающих длительное время и использующих громоздкие динамически подключаемые библиотеки (DLL).
- В .NET Framework домены приложений реализуют при помощи класса `System.AppDomain`. Для создания домена приложения вызывается один из перегруженных методов `AppDomain.CreateDomain`. Получить домен, в котором выполняет сборка в текущий момент можно, обратившись к свойству `AppDomain.CurrentDomain`.
- Для выполнения сборки в отдельном домене, необходимо получить доступ к нужному домену (или создать его) и вызвать метод `ExecuteAssembly`.
- Вместо указания полного пути к сборке, можно добавить ссылку на нее и вызывать сборку по имени методом `ExecuteAssemblyByName`. Такой способ вызова изолирует сборку, но не позволяет в полной мере воспользоваться преимуществами доменов приложений.
- Одним из преимуществ загрузки сборок в новые домены приложений является возможность в любой момент освободить ресурсы, выгрузив домен приложения. Чтобы выгрузить домен приложения со всеми его сборками, достаточно вызвать статический метод `AppDomain.Unload`.
- Хорошо сконфигурированный домен приложения не только изолирует сборки, но и ограничит ущерб, который может быть нанесен злоумышленником, взломавшим защиту сборки.
- Чтобы назначить сборке удостоверение, следует создать объект `System.Security.Policy.Evidence`, а затем передать его как параметр перегруженному методу `ExecuteAssembly`. Проще всего управлять разрешениями, назначенными сборке в домене приложения, передав удостоверение зоны безопасности при помощи объекта `System.Security.Policy.Zone` и перечислимого `System.Security.SecurityZone`.
- Создание сборки как службы позволяет запускать ее в фоновом режиме, без взаимодействия с пользователем. Службы идеально подходят для непрерывного мониторинга каких-либо событий, прослушивания сетевых подключений и запуска сборок до входа пользователя в систему.
- Службы Windows – это процессы, выполняющиеся в фоновом режиме, не имеющие пользовательского интерфейса и работающие в отдельном сеансе. Службы могут запускаться автоматически при загрузке компьютера, даже если пользователь не вошел в систему.

Закрепление материала

1. Что такое домен приложения?
2. Перечислите основные преимущества, обеспечиваемые использованием доменов приложений.
3. Как создать домен приложения и запустить в ней сборку?
4. Каким образом можно управлять привилегиями домена/сборки?
5. Что такое службы Windows?
6. Как установить службу в систему?

Дополнительное задание

Создайте приложение `WindowsForms`, которое позволит безопасно запускать сборки. В интерфейсе предусмотрите возможность выбора ограничения привилегий для запускаемой сборки.

Самостоятельная деятельность учащегося

Задание 1

Выучите основные конструкции и понятия, рассмотренные на уроке.

Задание 2

Создайте службу Windows, которая будет мониторить жесткие диски и при удалении из этих дисков файла записывать информацию (полный путь) в текстовый файл.

Создайте WPF приложение. Разместите в нем TextBox, в который с определенной периодичностью будет считываться информация из текстового файла (в который пишет сервис). Также разместите четыре кнопки, которые будут отвечать за установку, деинсталляцию, старт и остановку сервиса.

Задание 3

Зайдите на сайт MSDN.

Используя поисковые механизмы MSDN, найдите самостоятельно описание темы по каждому примеру, который был рассмотрен на уроке, так, как это представлено ниже, в разделе «Рекомендуемые ресурсы», описания данного урока. Сохраните ссылки и дайте им короткое описание.

Рекомендуемые ресурсы

MSDN: Класс `AppDomain`

<http://msdn.microsoft.com/ru-ru/library/system.appdomain.aspx>

MSDN: Создание домена приложения

[http://msdn.microsoft.com/ru-ru/library/6s0z09xw\(v=vs.90\).aspx](http://msdn.microsoft.com/ru-ru/library/6s0z09xw(v=vs.90).aspx)

MSDN: Загрузка сборок в домен приложения

[http://msdn.microsoft.com/ru-ru/library/25y1ya39\(v=vs.90\).aspx](http://msdn.microsoft.com/ru-ru/library/25y1ya39(v=vs.90).aspx)

MSDN: Настройка домена приложения

[http://msdn.microsoft.com/ru-ru/library/c8hk0245\(v=vs.90\).aspx](http://msdn.microsoft.com/ru-ru/library/c8hk0245(v=vs.90).aspx)

MSDN: Выгрузка домена приложения

[http://msdn.microsoft.com/ru-ru/library/c5b8a8f9\(v=vs.90\).aspx](http://msdn.microsoft.com/ru-ru/library/c5b8a8f9(v=vs.90).aspx)

MSDN: Создание приложения служб Windows в конструкторе компонентов

<http://msdn.microsoft.com/ru-ru/library/zt39148a.aspx>