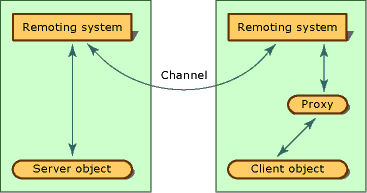
**Лекція 10. Основи технології Remoting**

Технологія Remoting розроблена для створення розподілених додатків. З її допомогою можна звертатися до екземплярів класів .Net, які перебувають за межами власного домену (application domain). Це може бути інший додаток усередині одного процесу (наприклад, звернення з одного asp.net додатки до об'єктів іншого), інший процес на тій же машині, або процес на іншій машині (у тому числі підключеної через Internet).  
Механізм Remoting працює таким чином.  
   
  
Сервер при запуску налаштовує Remoting для використання певного протоколу. При цьому вказується транспортний протокол і протокол доступу. Потім сервер реєструє всі класи, до яких він надає доступ.  
Клієнт при необхідності доступу до віддалених об'єктах також вказує протоколи доступу і посилає запит на сервер. Сервер у відповідь на цей запит створює у себе викликана об'єкт і передає його ідентифікатор клієнта. Клієнт створює у себе спеціальний проксі-клас, який потім і використовує, як якщо б це був об'єкт в його власному домені.  
Протоколи, використовувані Remoting.  
Для своєї роботи Remoting використовує два види протоколів: транспортний і доступу до об'єктів. На даний момент в .Net Framework реалізовано два перших і два других.  
Перший транспортний протокол (Channel в термінах .Net), реалізований у Remoting - це TCP. В якості протоколу доступу (formatter в .Net) TcpChannel використовує власний бінарний формат, але при необхідності можна використовувати будь-який інший.  
Другий протокол - HTTP. Для доступу він використовує SOAP (Simple Object Access Protocol) - протокол доступу, заснований на XML. Основні його переваги - SOAP стандартизований, так що теоретично можливе використання з клієнтів, створених без використання .Net Remoting, можливий доступ через стандартно налаштовані проксі-сервери, не створюється постійне з'єднання. Основний недолік - протокол текстовий, має велику надмірність.  
TcpChannel і HttpChannel - це двонаправлені канали. У випадку, якщо програма має суто клієнт-серверну архітектуру, необхідність використання саме двонапрямлених каналів відсутня. Більш того, якщо між клієнтом і сервером знаходиться проксі-сервер, функціонування двонапрямлених каналів на клієнті неможливо. Для подібних випадків існують два різновиди односпрямованих каналів. Наприклад, для протоколу HTTP це HttpClientChannel і HttpServerChannel.  
Передача об'єктів клієнту  
У Remoting об'єкти передаються за значенням (by value), або за посиланням (by reference).  
  При передачі за посиланням передається тільки ідентифікатор (точніше, спеціальний об'єкт, ObjRef). Клієнт створює спеціальний проксі-клас, який перенаправляє виклики методів на сервер і отримує повернені значення. Об'єкт, який передається по посиланню, повинен успадковуватися від MarshalByRefObject. Насправді створюються два проксі-класу - «справжній» проксі (real proxy) і «прозорий» проксі (transparent proxy). Клієнт спілкується з другим, який у свою чергу, створюється і управляється першим. «Прозорий» проксі має такий же набір методів, як і віддалений клас. При його використанні у клієнта створюється враження, що клас, з яким він працює, знаходиться в адресному просторі клієнта, але насправді виклики методів серіалізуются і передаються по каналах зв'язку екземпляру віддаленого об'єкта.  
  Необхідно відзначити одну відмінність між передачею за значенням і за посиланням. У разі передачі за значенням на клієнті повинен бути присутнім код самого об'єкта. В результаті цього клієнтський код доводиться частіше оновлювати, так як будь-яка зміна серверного коду потребують оновлення клієнта. Другий недолік - наявність серверного коду на клієнті як таке, що дає потенційному зловмиснику додаткові можливості, так як код .Net прекрасно дізассембліруется. Виходячи з цього, треба дуже обережно підходити до передачі за значенням класів, що містять критично важливі дані і логіку.  
При передачі за посиланням цілком достатньо наявності опису інтерфейсу (збірки, що містить опис інтерфейсу без його реалізації).  
Активація об'єктів  
Для об'єктів, переданих за посиланням, існує два механізми створення об'єктів: на сервері (server activated) і на клієнті (client activated).  
Актівіруемие сервером об'єкти (server activated)  
Актівіруемие сервером об'єкти не створюються на сервері у момент створення клієнтом проксі-класу. Замість цього вони створюються в момент першого виклику, або при кожному виклику методів. При цьому існують два типи подібних об'єктів - Singleton і SingleCall. Перші гарантують, що реальний примірник буде завжди один для всіх клієнтів. Другі створюють новий об'єкт для кожного виклику методу (модель, схожа на Web Services).  
Активуються клієнтом об'єкти (client activated)  
Активуються клієнтом об'єкти, на противагу активуються сервером, управляються клієнтом. Саме клієнт визначає час життя того чи іншого екземпляра.  
Існує проблема видалення об'єктів, керованих клієнтом, пов'язана з тим, що зв'язок сервер-клієнт принципово ненадійна. Класичний підхід - клієнт періодично відсилає на сервер повідомлення про те, що клієнт живий. Remoting використовує інший підхід. Кожному екземпляру призначається час життя. По закінченню цього часу сервер посилає клієнтові запит. Якщо клієнт дозволяє видалити екземпляр, або якщо відповіді від клієнта немає, сервер видаляє об'єкт.

В основі використання технології Remoting лежить клас RemotingConfiguration. який надає різні статичні методи для конфігурації інфраструктури віддаленого взаємодії.  
Простір імен: System.Runtime.Remoting  
Збірка: mscorlib (в mscorlib.dll)  
Тип RemotingConfiguration надає наступні члени.  
**Методи**

|  |  |
| --- | --- |
| Имя | Описание |
| [Configure](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.configure(v=VS.90).aspx) | Перевантажений. |
| [CustomErrorsEnabled](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.customerrorsenabled(v=VS.90).aspx) | Вказує, чи повертають канали сервера в цьому домені додатку фільтровані або повні відомості про винятки в локальні чи віддалені викликають оператори. |
| [GetRegisteredActivatedClientTypes](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.getregisteredactivatedclienttypes(v=VS.90).aspx) | Витягує масив типів об'єктів, зареєстрованих з боку клієнта як віддалено активованих типів. |
| [GetRegisteredActivatedServiceTypes](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.getregisteredactivatedservicetypes(v=VS.90).aspx) | Витягує масив типу об'єктів, зареєстрованих з боку служби, який може бути активований на запит клієнта |
| [GetRegisteredWellKnownClientTypes](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.getregisteredwellknownclienttypes(v=VS.90).aspx) | Витягує масив типів об'єктів, зареєстрованих з боку клієнта як добре відомих типів. |
| [GetRegisteredWellKnownServiceTypes](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.getregisteredwellknownservicetypes(v=VS.90).aspx) | Витягує масив типів об'єктів, зареєстрованих з боку служби як добре відомих типів. |
| [IsActivationAllowed](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.isactivationallowed(v=VS.90).aspx) | Повертає логічне значення, що показує, чи дозволено клієнтська активація зазначеного типу |
| [IsRemotelyActivatedClientType](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.isremotelyactivatedclienttype(v=VS.90).aspx) | Перевантажений. Перевіряє, чи зареєстрований тип зазначеного об'єкта як віддалено активованого типу клієнта.. |
| [IsWellKnownClientType](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.iswellknownclienttype(v=VS.90).aspx) | Перевантажений. Перевіряє, чи зареєстрований тип зазначеного об'єкта як добре відомого типу клієнта. |
| [RegisterWellKnownServiceType](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.registerwellknownservicetype(v=VS.90).aspx) | Перевантажений. Реєструє об'єкт Type з боку сервера в якості добре відомого типу (єдиного виклику або одноелементні множини |
| [RegisterWellKnownClientType](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.registerwellknownclienttype(v=VS.90).aspx) | Перевантажений. Реєструє об'єкт Type з боку клієнта як добре відомого типу (єдиного виклику або одноелементні множини). |
| [RegisterActivatedServiceType](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.registeractivatedservicetype(v=VS.90).aspx) | Перевантажений. Реєструє об'єкт Type з боку служби як об'єкт, який може активуватися за запитом клієнта. |
| [RegisterActivatedClientType](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.registeractivatedclienttype(v=VS.90).aspx) | Перевантажений. Реєструє об'єкт Type з боку клієнта як типу, який може активуватися сервером. |

# Свойства

|  |  |
| --- | --- |
| Ім’я | Опис |
| [ApplicationId](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.applicationid(v=VS.90).aspx) | Повертає ідентифікатор встановлений у справжній момент програми. |
| [ApplicationName](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.applicationname(v=VS.90).aspx) | Повертає або встановлює ім'я програми віддаленого доступу.. |
| [CustomErrorsMode](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.customerrorsmode(v=VS.90).aspx) | Повертає або задає значення, що визначає, як виконується обробка користувальницьких помилок. |
| [ProcessId](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.processid(v=VS.90).aspx) | Отримує ідентифікатор встановлений у справжній момент процесу. |

# Метод RemotingConfiguration.RegisterWellKnownServiceType (WellKnownServiceTypeEntry)

Реєструє об'єкт Type, записаний в наданому WellKnownServiceTypeEntry з боку служби як добре відомого типу.   
**Список перевантажень**

.

|  |  |
| --- | --- |
| Имя | Опис |
| [RegisterWellKnownServiceType (WellKnownServiceTypeEntry)](http://msdn.microsoft.com/ru-ru/library/3xzb9z8s(v=VS.90).aspx) | Реєструє об'єкт Type як об’єкт добре відомого типу (єдиного виклику або одноелементні множини, активізується сервером за зверненням клієнта). |
| [RegisterWellKnownServiceType](http://msdn.microsoft.com/ru-ru/library/czshbka7(v=VS.90).aspx)  [(Type, String, WellKnownObjectMode)](http://msdn.microsoft.com/ru-ru/library/czshbka7(v=VS.90).aspx) | Реєструє об'єкт Type з боку клієнта як добре відомого типу, використовуючи параметри, задані для ініціалізації нового екземпляра класу WellKnownServiceTypeEntry |

Винятки: SecurityException Принаймні у одного з викликають операторів, що знаходяться у верхній частині стека викликів, відсутній дозвіл на налаштування типів і каналів віддаленого взаємодії.

Процесом реєстрації не створюється екземпляр віддаленого об'єкта. Це відбувається, тільки коли клієнт намагається викликати метод на об'єкт або активувати об'єкт з боку клієнта.

Будь-який клієнт, що знає URI зареєстрованого добре відомого об'єкта, може отримати проксі для об'єкта, реєструючи кращий канал ChannelServices і активуючи об'єкт, викликавши метод new або Activator.GetObject. Клієнтська програма має дві можливості в для того щоб активувати добре відомий об'єкт

- Застосування методу new. Для цього спочатку необхідно зареєструвати тип добре відомого об'єкта, використовуючи метод RegisterWellKnownClientType. Виклик методу RegisterWellKnownClientType дає інфраструктурі віддаленого взаємодії місцезнаходження віддаленого об'єкта, який дозволяє зарезервоване слово new створити його.

- Застосування метод Activator.GetObject. Ветом випадку необхідно задати як аргумент методу наступні параметри: URL об'єкта, попередня реєстрація на стороні клієнта не потрібно.

Коли сервер отримує виклик,. NET Framework витягує URI з повідомлення, досліджує таблиці віддаленого доступу, щоб знайти посилання на об'єкт із заданим URI і, при необхідності, створює екземпляр цього об'єкта, передаючи виклик методу об'єкту. Якщо об'єкт зареєстрований як SingleCall, то він знищується після його завершення методу. Новий екземпляр об'єкту створюється для кожного викликаного методу. Єдина різниця між Activator.GetObject і new у тому, що перший з них дозволяє вказати URL-адресу як параметр, а останній отримує URL-адрес з конфігурації.

Примітка. URI розшифровується як Uniform Resource Identifier - "уніфікований ідентифікатор ресурсу". Іноді URI розшифровують як Universal Resource Identifier, тобто, універсальний ідентифікатор ресурсу. . URI - це деяка рядок, який дозволяє однозначно ідентифікувати якийсь ресурс, не важливо, чи є він віртуальним чи реальним. Прикладами віртуальних ресурсів є web-сторінки, електронні поштові скриньки; реальних - книги, які ідентифікуються номером ISBN.

# Метод RemotingConfiguration.RegisterActivatedServiceType (Type)

Регистрирует указанный тип объектов, зарегистрированных со стороны службы, в качестве типа, который может быть активирован по запросу клиента.

**Список перегрузки**

|  |  |
| --- | --- |
| Имя | Описание |
| [RegisterActivatedServiceType](http://msdn.microsoft.com/ru-ru/library/bsz90thf.aspx)  [(ActivatedServiceTypeEntry)](http://msdn.microsoft.com/ru-ru/library/bsz90thf.aspx) | Регистрирует тип объекта в предоставленном со стороны службы [ActivatedServiceTypeEntry](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.activatedservicetypeentry.aspx) в качестве типа, который может активироваться по запросу клиента. |
| [RegisterActivatedServiceType (Type)](http://msdn.microsoft.com/ru-ru/library/93z65zdc.aspx) | Регистрирует указанный тип объектов, зарегистрированных со стороны службы, в качестве типа, который может быть активирован по запросу клиента. |

Параметры type Тип: [System.Type](http://msdn.microsoft.com/ru-ru/library/system.type.aspx) тип объекта для регистрации.

Исключения : [SecurityException](http://msdn.microsoft.com/ru-ru/library/system.security.securityexception.aspx) По крайней мере у одного из вызывающих операторов, находящихся в верхней части стека вызовов, отсутствует разрешение на настройку типов и каналов удаленного взаимодействия.

Чтобы создать на сервере экземпляр объекта, активированный клиентом, необходимо знать его [Type](http://msdn.microsoft.com/ru-ru/library/system.type.aspx). Этот объект также должен быть зарегистрирован на сервере с использованием метода [RegisterActivatedServiceType](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.registeractivatedservicetype.aspx). Чтобы получить прокси для экземпляра объекта, активированного клиентом, клиенту необходимо сначала зарегистрировать канал службой [ChannelServices](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.channels.channelservices.aspx), а затем активировать объект, вызвав new или [Activator.CreateInstance](http://msdn.microsoft.com/ru-ru/library/system.activator.createinstance.aspx).

Чтобы активировать тип активируемого клиентом объекта ключевым словом new, необходимо сначала зарегистрировать тип объекта со стороны клиента с помощью метода [RegisterActivatedClientType](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.registeractivatedclienttype.aspx). Вызов метода [RegisterActivatedClientType](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.registeractivatedclienttype.aspx) дает инфраструктуре удаленного взаимодействия местонахождение удаленного приложения, где new пытается его создать. Если для создания нового экземпляра объекта, активированного клиентом, используется метод [CreateInstance](http://msdn.microsoft.com/ru-ru/library/system.activator.createinstance.aspx), то необходимо задать в качестве параметра URL-адрес удаленного приложения. Предварительная регистрация на стороне клиента не требуется. Чтобы предоставить методу [CreateInstance](http://msdn.microsoft.com/ru-ru/library/system.activator.createinstance.aspx) URL-адрес сервера, на котором нужно создать объект, необходимо включить URL-адрес в экземпляр класса [UrlAttribute](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.activation.urlattribute.aspx).

В приведенном ниже примере кода показана регистрация типа объекта со стороны сервера в качестве типа, который может активироваться клиентом. Для кода клиента, который соответствует представленному коду сервера, см. пример для метода [RegisterActivatedClientType](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.registeractivatedclienttype.aspx).

using System;

using System.Runtime.Remoting;

using System.Runtime.Remoting.Channels;

using System.Runtime.Remoting.Channels.Tcp;

public class ServerClass {

public static void Main() {

ChannelServices.RegisterChannel(new TcpChannel(8082));

RemotingConfiguration.RegisterActivatedServiceType(typeof(HelloServiceClass));

Console.WriteLine("Press enter to stop this process.");

Console.ReadLine();

}

}

В приведенном ниже примере кода показан объект службы, зарегистрированный в приведенном выше примере кода.

using System;

public class HelloServiceClass : MarshalByRefObject {

static int n\_instance;

public HelloServiceClass() {

n\_instance++;

Console.WriteLine(this.GetType().Name + " has been created. Instance # = {0}", n\_instance);

}

~HelloServiceClass() {

Console.WriteLine("Destroyed instance {0} of HelloServiceClass.", n\_instance);

n\_instance --;

}

public String HelloMethod(String name) {

// Reports that the method was called.

Console.WriteLine();

Console.WriteLine("Called HelloMethod on instance {0} with the '{1}' parameter.",

n\_instance, name);

// Calculates and returns the result to the client.

return "Hi there " + name + ".";

}

}

# Метод RemotingConfiguration.RegisterWellKnownClientType

Регистрирует объект [**Type**](http://msdn.microsoft.com/ru-ru/library/system.type.aspx) со стороны клиента в качестве хорошо известного типа (единственного вызова или одноэлементного множества).

**Список перегрузки**

|  |  |
| --- | --- |
| Имя | Описание |
| [RegisterWellKnownClientType](http://msdn.microsoft.com/ru-ru/library/zx2by16x.aspx)  [(WellKnownClientTypeEntry)](http://msdn.microsoft.com/ru-ru/library/zx2by16x.aspx) | Регистрирует объект [Type](http://msdn.microsoft.com/ru-ru/library/system.type.aspx), записанный в предоставленный [WellKnownClientTypeEntry](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.wellknownclienttypeentry.aspx) со стороны клиента в качестве хорошо известного типа, который может быть активирован сервером. |
| [RegisterWellKnownClientType](http://msdn.microsoft.com/ru-ru/library/hcd5d6y7.aspx)  [(Type, String)](http://msdn.microsoft.com/ru-ru/library/hcd5d6y7.aspx) | Регистрирует объект [Type](http://msdn.microsoft.com/ru-ru/library/system.type.aspx) со стороны клиента в качестве хорошо известного объекта, который может быть активирован на сервере, используя данные параметры для инициализации нового экземпляра класса [WellKnownClientTypeEntry](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.wellknownclienttypeentry.aspx). |

Любой клиент, знающий URI зарегистрированного хорошо известного объекта, может получить прокси для объекта, регистрируя предпочтительный канал [ChannelServices](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.channels.channelservices.aspx) и активируя объект, вызвав new или [Activator.GetObject](http://msdn.microsoft.com/ru-ru/library/system.activator.getobject.aspx).

Чтобы активировать хорошо известный объект new, сначала необходимо зарегистрировать тип хорошо известного объекта, используя метод [RegisterWellKnownClientType](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.registerwellknownclienttype.aspx). Вызов метода [RegisterWellKnownClientType](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.registerwellknownclienttype.aspx) дает инфраструктуре удаленного взаимодействия местонахождение удаленного объекта, который позволяет зарезервированному слову new создать его.

Если для активации хорошо известного объекта используется метод [Activator.GetObject](http://msdn.microsoft.com/ru-ru/library/system.activator.getobject.aspx), то необходимо задать в качестве аргумента URL объекта, предварительная регистрация на стороне клиента не требуется.

# Метод RemotingConfiguration.RegisterActivatedClientType (ActivatedClientTypeEntry)

Регистрирует объект [Type](http://msdn.microsoft.com/ru-ru/library/system.type(v=VS.90).aspx) со стороны клиента в качестве типа, который может активироваться сервером.

# Список перегрузки

|  |  |
| --- | --- |
| Имя | Описание |
| [RegisterActivatedClientType](http://msdn.microsoft.com/ru-ru/library/8kz5f4f8(v=VS.90).aspx)  [(ActivatedClientTypeEntry)](http://msdn.microsoft.com/ru-ru/library/8kz5f4f8(v=VS.90).aspx) | Регистрирует объект [Type](http://msdn.microsoft.com/ru-ru/library/system.type(v=VS.90).aspx), записанный в предоставленный [ActivatedClientTypeEntry](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.activatedclienttypeentry(v=VS.90).aspx) на стороне клиента в качестве типа, который может быть активирован сервером. |
| [RegisterActivatedClientType(Type, String)](http://msdn.microsoft.com/ru-ru/library/hkzabx8b(v=VS.90).aspx) | Регистрирует объект [Type](http://msdn.microsoft.com/ru-ru/library/system.type(v=VS.90).aspx) со стороны клиента в качестве типа, который может активироваться сервером, используя данные параметры для инициализации нового экземпляра класса [ActivatedClientTypeEntry](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.activatedclienttypeentry(v=VS.90).aspx). |

Регистрирует объект [Type](http://msdn.microsoft.com/ru-ru/library/system.type(v=VS.90).aspx), записанный в предоставленный [ActivatedClientTypeEntry](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.activatedclienttypeentry(v=VS.90).aspx) на стороне клиента в качестве типа, который может быть активирован сервером.

Параметры entry Тип: [System.Runtime.Remoting.ActivatedClientTypeEntry](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.activatedclienttypeentry(v=VS.90).aspx)  
Параметры конфигурации для активированного клиентом типа.

Исключения : [SecurityException](http://msdn.microsoft.com/ru-ru/library/system.security.securityexception(v=VS.90).aspx) По крайней мере у одного из вызывающих операторов, находящихся в верхней части стека вызовов, отсутствует разрешение на настройку типов и каналов удаленного взаимодействия.

Чтобы создать на сервере экземпляр объекта, активированный клиентом, необходимо знать его [Type](http://msdn.microsoft.com/ru-ru/library/system.type(v=VS.90).aspx). Этот объект также должен быть зарегистрирован на сервере с использованием метода [RegisterActivatedServiceType](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.registeractivatedservicetype(v=VS.90).aspx). Чтобы получить прокси для экземпляра объекта, активированного клиентом, клиенту необходимо сначала зарегистрировать канал службой [ChannelServices](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.channels.channelservices(v=VS.90).aspx), а затем активировать объект, вызвав new.

Чтобы активировать тип активируемого клиентом объекта ключевым словом new, необходимо сначала зарегистрировать тип объекта со стороны клиента с помощью метода [RegisterActivatedClientType](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.registeractivatedclienttype(v=VS.90).aspx). Вызов метода [RegisterActivatedClientType](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.registeractivatedclienttype(v=VS.90).aspx) дает инфраструктуре удаленного взаимодействия местонахождение удаленного приложения, где new пытается его создать. Если для создания нового экземпляра объекта, активированного клиентом, используется метод [Activator.CreateInstance](http://msdn.microsoft.com/ru-ru/library/system.activator.createinstance(v=VS.90).aspx), то необходимо задать в качестве параметра URL-адрес удаленного приложения. Предварительная регистрация на стороне клиента не требуется. Чтобы предоставить методу [Activator.CreateInstance](http://msdn.microsoft.com/ru-ru/library/system.activator.createinstance(v=VS.90).aspx) URL-адрес сервера, на котором нужно создать объект, необходимо включить URL-адрес в экземпляр класса [UrlAttribute](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.activation.urlattribute(v=VS.90).aspx).

Подробное описание объектов, активированных клиентом, см. в разделе [Client Activation](http://msdn.microsoft.com/ru-ru/library/w93betdk(v=VS.90).aspx).

Разрешения : для конфигурации инфраструктуры удаленного взаимодействия. Значение по требованию: [SecurityAction.Demand](http://msdn.microsoft.com/ru-ru/library/system.security.permissions.securityaction.demand(v=VS.90).aspx). Значение по разрешению: [SecurityPermissionFlag.RemotingConfiguration](http://msdn.microsoft.com/ru-ru/library/system.security.permissions.securitypermissionflag.remotingconfiguration(v=VS.90).aspx).

# Активация сервером

Объекты, активируемые сервером, — это объекты, время существования которых напрямую контролируется сервером. Домен серверного приложения создает эти объекты только тогда, когда клиент вызывает метод такого объекта, а не когда клиент использует оператор **new** (**New()** в Visual Basic) или вызывает метод [Activator.GetObject](http://msdn.microsoft.com/ru-ru/library/system.activator.getobject.aspx); это позволяет избежать отправки запроса по сети и получения ответа только с целью создания экземпляра. Когда клиент запрашивает экземпляр типа, активируемого сервером, в домене клиентского приложения создается только прокси. Это значит, что для типов, активируемых сервером, можно использовать только конструкторы по умолчанию. Для публикации типа, экземпляры которого будут создаваться с помощью особых конструкторов, принимающих аргументы, должна использоваться [Активация клиента](http://msdn.microsoft.com/ru-ru/library/w93betdk.aspx) или динамическая публикация конкретного экземпляра.

## Режимы активации сервером

Имеется два режима активации (или значения [WellKnownObjectMode](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.wellknownobjectmode.aspx)) для активируемых сервером объектов: **Singleton** и **SingleCall**.

У типов **Singleton** никогда не может одновременно быть более одного экземпляра. Если экземпляр есть, все запросы клиентов обслуживаются этим экземпляром. Если экземпляра нет, сервер создает его и все последующие запросы клиентов обслуживаются этим экземпляром. Поскольку для типов **Singleton** задается время существования по умолчанию, клиенты не будут всегда получать ссылку на один и тот же экземпляр удаленного типа, даже если в каждый конкретный момент времени доступно не более одного экземпляра.

У типов **SingleCall** всегда создается один экземпляр на каждый клиентский запрос. Очередной вызов метода будет обслуживаться другим серверным экземпляром, даже если предыдущий экземпляр еще не был уничтожен системой. Типы **SingleCall** не участвуют в процессах аренды времени существования.

Чтобы создать экземпляр типа, активируемого сервером, клиентские приложения должны быть настроены программным образом или с помощью файла конфигурации. В случае настройки приложения программным образом для создания на клиенте экземпляра объекта, активируемого сервером, используется метод Activator.GetObject. В случае настройки приложения с помощью файла конфигурации для создания на клиенте экземпляра объекта, активируемого сервером, используется метод Activator.GetObject или оператор new.

**y0h540a7.note(ru-ru,VS.100).gifПримечание.**

Возможно, регистрировать канал на стороне клиента не потребуется. Если клиент не регистрирует канал, служба удаленного взаимодействия автоматически создает канал для исходящих запросов, используя один из каналов по умолчанию, заданных в файле Machine.config. При таком автоматическом выборе каналов на клиенте этот канал не регистрируется для ожидания функций обратного вызова с сервера и, если только в файл machine.config не добавлен пользовательский канал, не регистрируются никакие пользовательские реализации канала. В этих случаях нужно зарегистрировать тип канала, который требуется использовать, в домене клиентского приложения.

В следующем примере кода показан вызов метода **Activator.GetObject** в предположении, что канал [TcpChannel](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.channels.tcp.tcpchannel.aspx) зарегистрирован для связи через порт 8080. Если клиенту известен только факт, что в серверном объекте реализован определенный интерфейс, необходимо вызывать метод **Activator.GetObject**, поскольку для создания экземпляра класса можно использовать только оператор **new** (**New** в Visual Basic).

RemoteObjectClass MyRemoteClass = (RemoteObjectClass)Activator.GetObject(

typeof(RemoteObjectClass),

"tcp://computername:8080/RemoteObjectUri " );

Следует помнить, что на данном этапе никакого взаимодействия с сервером не происходит и удаленный объект не создается. Создается только экземпляр прокси-объекта на стороне клиента. Теперь клиент может использовать объект MyRemoteClass, как если бы у него была прямая ссылка на удаленный объект. Экземпляр RemoteObjectClass, который клиент фактически использует для взаимодействия при различных вызовах методов, зависит от того, к какому типу относится серверный объект — **Singleton** или **SingleCall**. Независимо от того, раскрывает ли эту информацию издатель серверного объекта, клиент использует имеющуюся у него ссылку на объект совершенно одинаково.

## Объекты Singleton

В модели COM тип "singleton" означал, что объект хранится в памяти, пока у клиентов имеются ссылки на этот объект. Однако в системе удаленного взаимодействия .NET на объекты **Singleton** распространяется действие заданного для них времени аренды, поэтому такие объекты могут быть уничтожены, даже если у клиентов еще имеются ссылки на них. Можно создать объект типа **Singleton**, соответствующий старым правилам, переопределив метод [InitializeLifetimeService](http://msdn.microsoft.com/ru-ru/library/system.marshalbyrefobject.initializelifetimeservice.aspx) класса [MarshalByRefObject](http://msdn.microsoft.com/ru-ru/library/system.marshalbyrefobject.aspx) таким образом, чтобы он возвращал пустую ссылку (**Nothing** в Visual Basic). В результате объект будет оставаться в памяти, пока существует домен основного приложения. Дополнительные сведения см. в разделе [Аренда времени существования](http://msdn.microsoft.com/ru-ru/library/23bk23zc.aspx). Объект **Singleton** последнего типа можно создать, настроив время начальной аренды в файле конфигурации удаленного взаимодействия.

# Метод Activator.GetObject (Type, String, Object)

Создает прокси для хорошо известного объекта, который идентифицируется по заданному типу, URL и данным канала.

**Параметры** type Тип: [System.Type](http://msdn.microsoft.com/ru-ru/library/system.type(v=VS.90).aspx) Тип хорошо известных объектов, к которым нужно подключиться.

url Тип: [System.String](http://msdn.microsoft.com/ru-ru/library/system.string(v=VS.90).aspx) URL-адрес хорошо известного объекта.

state Тип: [System.Object](http://msdn.microsoft.com/ru-ru/library/system.object(v=VS.90).aspx) Данные, зависящие от канала, или значение null.

**Возвращаемое значение** Тип: [System.Object](http://msdn.microsoft.com/ru-ru/library/system.object(v=VS.90).aspx). Прокси, который указывает на конечную точку, используемую требуемым хорошо известным объектом.

Исключения

[ArgumentNullException](http://msdn.microsoft.com/ru-ru/library/system.argumentnullexception(v=VS.90).aspx) Параметр type или url имеет значение null.

[RemotingException](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingexception(v=VS.90).aspx) Параметр type не маршалируется и не передается по ссылке.

-или-Параметр type является интерфейсом.

[MemberAccessException](http://msdn.microsoft.com/ru-ru/library/system.memberaccessexception(v=VS.90).aspx) Этот член был вызван с помощью механизма с поздней привязкой.

Для отправки сообщения удаленному объекту необходимо обратиться к прокси. Пока метод не будет вызван для прокси, сообщения по сети не отправятся.

Разрешения

* [ReflectionPermission](http://msdn.microsoft.com/ru-ru/library/system.security.permissions.reflectionpermission(v=VS.90).aspx)   для обеспечения возможности вызова операций для всех членов типа. Связанное перечисление: [ReflectionPermissionFlag.MemberAccess](http://msdn.microsoft.com/ru-ru/library/system.security.permissions.reflectionpermissionflag.memberaccess(v=VS.90).aspx)
* [SecurityPermission](http://msdn.microsoft.com/ru-ru/library/system.security.permissions.securitypermission(v=VS.90).aspx)   для обеспечения возможности настраивать типы и каналы удаленного взаимодействия. Связанное перечисление: [SecurityPermissionFlag.RemotingConfiguration](http://msdn.microsoft.com/ru-ru/library/system.security.permissions.securitypermissionflag.remotingconfiguration(v=VS.90).aspx)