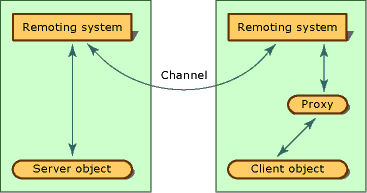
# Лекция10. Основы технологии Remoting

Технология Remoting разработана для создания распределенных приложений. С ее помощью можно обращаться к экземплярам классов .Net, находящимся за пределами собственного домена (application domain). Это может быть другое приложение внутри одного процесса (например, обращение из одного asp.net приложения к объектам другого), другой процесс на той же машине, или процесс на другой машине (в том числе подключенной через Internet).

Механизм Remoting работает следующим образом.



Сервер при запуске настраивает Remoting для использования определенного протокола. При этом указывается транспортный протокол и протокол доступа. Затем сервер регистрирует все классы, к которым он предоставляет доступ.

Клиент при необходимости доступа к удаленным объектам также указывает протоколы доступа и посылает запрос на сервер. Сервер в ответ на этот запрос создает у себя затребованный объект и передает его идентификатор клиенту. Клиент создает у себя специальный прокси-класс, который затем и использует, как если бы это был объект в его собственном домене.

## Протоколы, используемые Remoting.

Для своей работы Remoting использует два вида протоколов: транспортный и доступа к объектам. На данный момент в .Net Framework реализовано два первых и два вторых.

Первый транспортный протокол (Channel в терминах .Net), реализованный в Remoting – это TCP. В качестве протокола доступа (formatter в .Net) TcpChannel использует собственный бинарный формат, но при необходимости можно использовать любой другой.

Второй протокол – HTTP. Для доступа он использует SOAP (Simple Object Access Protocol) – протокол доступа, основанный на XML. Основные его преимущества – SOAP стандартизован, так что теоретически возможно использование из клиентов, созданных без использования .Net Remoting, возможен доступ через стандартно настроенные прокси-серверы, не создается постоянное соединение. Основной недостаток – протокол текстовый, имеет большую избыточность.

TcpChannel и HttpChannel – это двунаправленные каналы. В случае, если приложение имеет чисто клиент-серверную архитектуру, необходимость использования именно двунаправленных каналов отсутствует. Более того, если между клиентом и сервером находится прокси-сервер, функционирование двунаправленных каналов на клиенте невозможно. Для подобных случаев существуют две разновидности однонаправленных каналов. К примеру, для протокола HTTP это HttpClientChannel и HttpServerChannel.

## Передача объектов клиенту

В Remoting объекты передаются по значению (by value), либо по ссылке (by reference).

При передаче по ссылке передается только идентификатор (точнее, специальный объект, ObjRef). Клиент создает специальный прокси-класс, который перенаправляет вызовы методов на сервер и получает возвращаемые значения. Объект, передаваемый по ссылке, должен наследоваться от MarshalByRefObject. На самом деле создаются два прокси-класса – «настоящий» прокси (real proxy) и «прозрачный» прокси (transparent proxy). Клиент общается со вторым, который в свою очередь, создается и управляется первым. «Прозрачный» прокси имеет такой же набор методов, как и удаленный класс. При его использовании у клиента создается впечатление, что класс, с которым он работает, находится в адресном пространстве клиента, но на самом деле вызовы методов сериализуются и передаются по каналам связи экземпляру удаленного объекта.

Необходимо отметить одно отличие между передачей по значению и по ссылке. В случае передачи по значению на клиенте должен присутствовать код самого объекта. В результате этого клиентский код приходится чаще обновлять, так как любое изменение серверного кода потребует обновления клиента. Второй недостаток – наличие серверного кода на клиенте как таковое, дающее потенциальному злоумышленнику дополнительные возможности, так как код .Net прекрасно дизассемблируется. Исходя из этого, надо очень осторожно подходить к передаче по значению классов, содержащих критически важные данные и логику.

При передаче по ссылке вполне достаточно наличия описания интерфейса (сборки, содержащей описание интерфейса без его реализации).

## Активация объектов

Для объектов, передаваемых по ссылке, существует два механизма создания объектов: на сервере (server activated) и на клиенте (client activated).

### Активируемые сервером объекты (server activated)

Активируемые сервером объекты не создаются на сервере в момент создания клиентом прокси-класса. Вместо этого они создаются в момент первого вызова, или при каждом вызове методов. При этом существуют два типа подобных объектов – Singleton и SingleCall. Первые гарантируют, что реальный экземпляр будет всегда один для всех клиентов. Вторые создают новый объект для каждого вызова метода (модель, похожая на Web Services).

**Активируемые клиентом объекты (client activated)**

Активируемые клиентом объекты, в противоположность активируемым сервером, управляются клиентом. Именно клиент определяет время жизни того или иного экземпляра.

Существует проблема удаления объектов, управляемых клиентом, связанная с тем, что связь сервер-клиент принципиально ненадежна. Классический подход – клиент периодически отсылает на сервер уведомления о том, что клиент жив. Remoting использует иной подход. Каждому экземпляру назначается время жизни. По истечению этого времени сервер посылает клиенту запрос. Если клиент разрешает удалить экземпляр, или если ответа от клиента нет, сервер удаляет объект.

В основе использования технологии Remoting лежит класс RemotingConfiguration. который предоставляет различные статические методы для конфигурации инфраструктуры удаленного взаимодействия.

**Пространство имен:**  [System.Runtime.Remoting](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting(v=VS.90).aspx)  
**Сборка:**  mscorlib (в mscorlib.dll)

# Тип [RemotingConfiguration](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration(v=VS.90).aspx) предоставляет следующие члены.

# Методы

|  |  |
| --- | --- |
| Имя | Описание |
| [Configure](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.configure(v=VS.90).aspx) | Перегружен. |
| [CustomErrorsEnabled](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.customerrorsenabled(v=VS.90).aspx) | Указывает, возвращают ли каналы сервера в этом домене приложения фильтрованные или полные сведения об исключениях в локальные или удаленные вызывающие операторы. |
| [GetRegisteredActivatedClientTypes](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.getregisteredactivatedclienttypes(v=VS.90).aspx) | Извлекает массив типов объектов, зарегистрированных со стороны клиента в качестве удаленно активированных типов. |
| [GetRegisteredActivatedServiceTypes](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.getregisteredactivatedservicetypes(v=VS.90).aspx) | Извлекает массив типа объектов, зарегистрированных со стороны службы, который может быть активирован по запросу клиента. |
| [GetRegisteredWellKnownClientTypes](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.getregisteredwellknownclienttypes(v=VS.90).aspx) | Извлекает массив типов объектов, зарегистрированных со стороны клиента в качестве хорошо известных типов. |
| [GetRegisteredWellKnownServiceTypes](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.getregisteredwellknownservicetypes(v=VS.90).aspx) | Извлекает массив типов объектов, зарегистрированных со стороны службы в качестве хорошо известных типов. |
| [IsActivationAllowed](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.isactivationallowed(v=VS.90).aspx) | Возвращает логическое значение, показывающее, разрешена ли клиентская активация указанного типа [Type](http://msdn.microsoft.com/ru-ru/library/system.type(v=VS.90).aspx). |
| [IsRemotelyActivatedClientType](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.isremotelyactivatedclienttype(v=VS.90).aspx) | Перегружен. Проверяет, зарегистрирован ли тип указанного объекта в качестве удаленно активированного типа клиента. |
| [IsWellKnownClientType](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.iswellknownclienttype(v=VS.90).aspx) | Перегружен. Проверяет, зарегистрирован ли тип указанного объекта в качестве хорошо известного типа клиента. |
| [RegisterWellKnownServiceType](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.registerwellknownservicetype(v=VS.90).aspx) | Перегружен. Регистрирует объект [Type](http://msdn.microsoft.com/ru-ru/library/system.type(v=VS.90).aspx) со стороны сервера в качестве хорошо известного типа (единственного вызова или одноэлементного множества). |
| [RegisterWellKnownClientType](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.registerwellknownclienttype(v=VS.90).aspx) | Перегружен. Регистрирует объект [Type](http://msdn.microsoft.com/ru-ru/library/system.type(v=VS.90).aspx) со стороны клиента в качестве хорошо известного типа (единственного вызова или одноэлементного множества). |
| [RegisterActivatedServiceType](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.registeractivatedservicetype(v=VS.90).aspx) | Перегружен. Регистрирует объект [Type](http://msdn.microsoft.com/ru-ru/library/system.type(v=VS.90).aspx) со стороны службы в качестве объекта, который может активироваться по запросу клиента. |
| [RegisterActivatedClientType](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.registeractivatedclienttype(v=VS.90).aspx) | Перегружен. 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. |

# Свойства

|  |  |
| --- | --- |
| Имя | Описание |
| [ApplicationId](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.applicationid(v=VS.90).aspx) | Возвращает идентификатор выполняемого в настоящий момент приложения. |
| [ApplicationName](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.applicationname(v=VS.90).aspx) | Возвращает или устанавливает имя приложения удаленного доступа. |
| [CustomErrorsMode](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.customerrorsmode(v=VS.90).aspx) | Возвращает или задает значение, определяющее, как выполняется обработка пользовательских ошибок. |
| [ProcessId](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.remotingconfiguration.processid(v=VS.90).aspx) | Получает идентификатор выполняемого в настоящий момент процесса. |

# Метод RemotingConfiguration.RegisterWellKnownServiceType (WellKnownServiceTypeEntry)

Регистрирует объект [Type](http://msdn.microsoft.com/ru-ru/library/system.type(v=VS.90).aspx) в качестве хорошо известного типа (единственного вызова или одноэлементного множества, активизируется сервером по обращению клиента).

# Список перегрузки

|  |  |
| --- | --- |
| Имя | Описание |
| [RegisterWellKnownServiceType (WellKnownServiceTypeEntry)](http://msdn.microsoft.com/ru-ru/library/3xzb9z8s(v=VS.90).aspx) | Регистрирует объект [Type](http://msdn.microsoft.com/ru-ru/library/system.type(v=VS.90).aspx), записанный в предоставленном [WellKnownServiceTypeEntry](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.wellknownservicetypeentry(v=VS.90).aspx) со стороны службы в качестве хорошо известного типа. |
| [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) | Регистрирует объект [Type](http://msdn.microsoft.com/ru-ru/library/system.type(v=VS.90).aspx) со стороны клиента в качестве хорошо известного типа, используя параметры, заданные для инициализации нового экземпляра класса [WellKnownServiceTypeEntry](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.wellknownservicetypeentry(v=VS.90).aspx). |

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.

Процессом регистрации не создается экземпляр удаленного объекта. Это происходит, только когда клиент пытается вызвать метод на объект или активировать объект со стороны клиента.

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 метод new или [Activator.GetObject](http://msdn.microsoft.com/ru-ru/library/system.activator.getobject.aspx). Клиентская программа имеет две возможности в для того чтобы активировать хорошо известный объект

* Применение метода new. Для этого 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.
* Применение метод [Activator.GetObject](http://msdn.microsoft.com/ru-ru/library/system.activator.getobject.aspx) . Вэтом случае необходимо задать в качестве аргумента метода следующие параметры: URL объекта, предварительная регистрация на стороне клиента не требуется.

Когда сервер получает вызов, .NET Framework извлекает URI из сообщения, исследует таблицы удаленного доступа, чтобы найти ссылку на объект с заданным URI и, при необходимости, создает экземпляр этого объекта, передавая вызов метода объекту. Если объект зарегистрирован в качестве [SingleCall](http://msdn.microsoft.com/ru-ru/library/system.runtime.remoting.wellknownobjectmode.singlecall.aspx), то он уничтожается по завершении вызова метода. Новый экземпляр объекта создается для каждого вызванного метода. Единственная разница между [Activator.GetObject](http://msdn.microsoft.com/ru-ru/library/system.activator.getobject.aspx) и new в том, что первый из них позволяет указать URL-адрес в качестве параметра, а последний получает URL-адрес из конфигурации.

**Примечание.** URI расшифровывается как Uniform Resource Identifier - "унифицированный идентификатор ресурса". Иногда URI расшифровывают как Universal Resource Identifier, то есть, универсальный идентификатор ресурса. . URI - это некоторая строка, которая позволяет однозначно идентифицировать какой-то ресурс, не важно, является он виртуальным или реальным.

# 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.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).

# 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).

# 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.

## 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.