Визитка

ИВАН КОРОБКО, сертифицированный специалист MCP, автор более 50 статей и двух книг. Занимается созданием различных приложений для Active Directory

Использование IIS
в корпоративных сетях

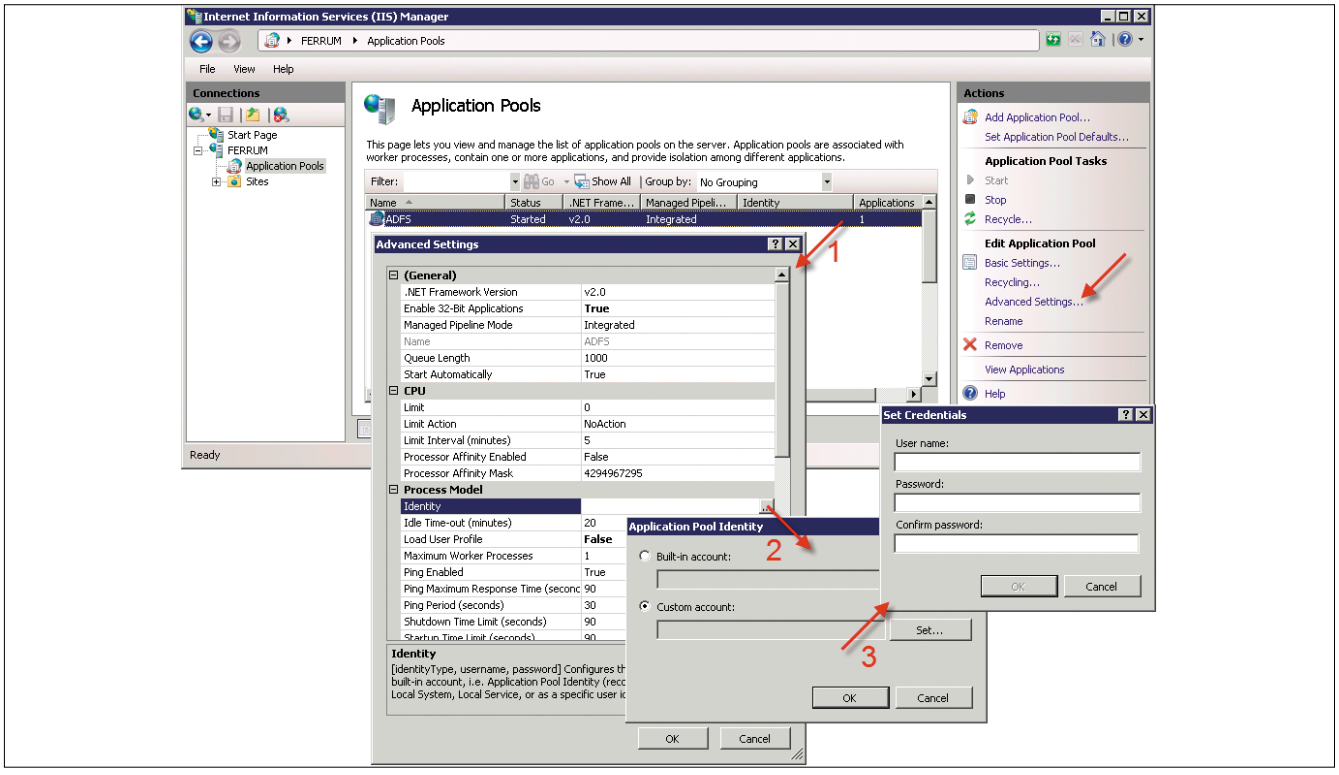
Использование олицетворения (имперсонализации), реализованного в IIS, многократно увеличивает функционал приложений, обеспечивающих автоматизацию различных процессов

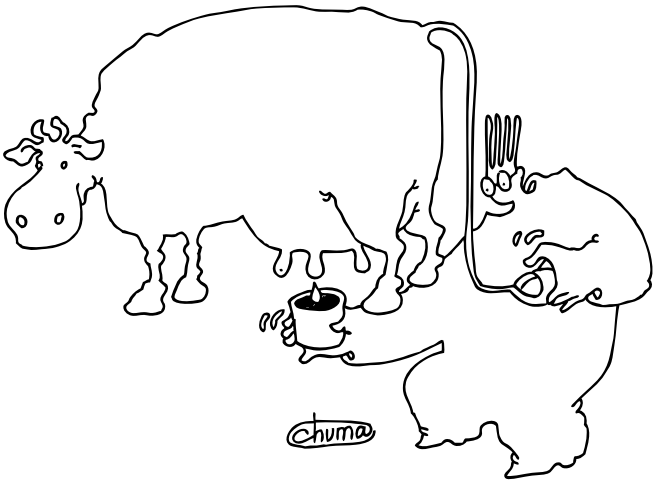
Достаточно часто перед администратором стоит задача предоставить административные права на очень узком участке для пользователей, которые их не имеют. Одним из ярких примеров является сценарий регистрации пользователей, который последовательно выполняет запуск веб-приложений. Сценарий регистрации пользователя в сети представляет собой совокупность файлов, выполняющих какие-либо действия на рабочей станции клиента от его имени. Многие функции для этого сценария не могут быть выполнены из-за ограничений статуса пользователя, поскольку большинство из них – обычные пользователи. С помощью системы олицетворения, реализованной в IIS, на локальной рабочей станции можно выполнить ряд изменений в файловой системе, некоторых ветвях реестра, управлять сервисами и т.д.

Олицетворение в IIS

Internet Information Service (IIS) – яркий пример трехзвенной системы. В качестве клиента выступает любой браузер – Internet Explorer, Mozila, Opera, в качестве сервера – IIS, в качестве сервера базы данных – этот или иной сервер.

Рисунок 1. Настройка олицетворения приложения. Пул приложений





Предоставим административные права на очень узком участке пользователям, которые их не имеют

Трехзвенная система работает следующим образом: пользователь с рабочей станции (локальной) формирует запрос от своего имени, вызывая веб-страницу, хранящуюся на IIS-сервере. На этом сервере осуществляется проверка полученных данных. В случае удачной проверки происходит их подмена на заранее определенные параметры безопасности, урезанной в целях безопасности учетной записи системного администратора. От имени привилегированного пользователя осуществляется доступ к третьему серверу и выполнение заданных действий, на которые пользователь не имеет права.

Настройка олицетворения осуществляется через пул приложений (Application Pool), который по умолчанию создается для каждого веб-приложения. Практика показывает, что иметь несколько идентичных пулов приложений не имеет смысла – достаточно двух или трех разных пулов. Установив курсор на изменяемый пул в контекстном меню, размещенном в панели справа от списка пула приложений, выберите пункт Advanced Settings… (см. рис. 1). В появившемся диалоговом окне найдите свойство identity и кликните по значку с терм точками справа от предполагаемого значения. В новом окне необходимо выбрать тип учетной записи (custom account), от имени которой будет осуществляться доступ на другие серверы. На заключительном шаге необходимо ввести имя пользователя и дважды его пароль.

Сделанные изменения записываются в файл web.config. Фрагмент файла приведен в листинге 1.

Листинг 1. Фрагмент файла web.config

```
<?xml version="1.0"?>
<configuration>
    ….
        The <authentication> section enables configuration
        of the security authentication mode used by
        ASP.NET to identify an incoming user.
 -->
            <authentication mode="Windows"/>
            <identity impersonate="true" ↵
                 userName="domain\userName" ↵
                 password="123456789"/>
…
</configuration>
```

Для повышения безопасности рекомендуется значения имени и пароля хранить в реестре [1].

В этом случае значение параметров username и password следующие:

```
userName="registry:HKLM\Software\AspNetProcess,Name"
password="registry:HKLM\Software\AspNetProcess,Pwd"
```

Веб-сайт, в котором должен использоваться уже настроенный пул приложений, также требует изменений в настройке по умолчанию. Чтобы образовать ассоциацию созданного пула сайту, необходимо войти в диалоговое окно «Расширенные настройки» (Advanced settings) и изменить в поле Application Pull значение – имя ассоциированного пула (см. рис. 2).

Второе обязательное условие – изменить тип аутентификации (см. рис. 2). По умолчанию Windows Authentication выключена, а Anonymous включена. Сделанные изменения также фиксируются в файле web.config (см. листинг 1).

Использование IIS в сценариях регистрации пользователей

Сценарий регистрации пользователей в сети, как правило, имеет блочную структуру [2]. Каждый из них решает какую-либо узкую задачу: инвентаризации, подключения сетевых дисков, сетевых принтеров, настройки рабочего окружения пользователей.

Настройка рабочего окружения сводится в конечном счете к записи данных в реестр. Поскольку сценарий выполняется от имени обычного пользователя, который лишен административных привилегий, ему доступен для записи только куст HKEY_CURRENT_USER, который формируется индивидуально для каждого пользователя на основе данных ветви HKEY_USERS.

Как ни странно, большинство настроек, позволяющих унифицировать рабочее пространство пользователя, сосредоточено в ветвях HKEY_LOCAL_MACHINE и HKEY_CLASSES_ROOT, которые недоступны для изменения обычным пользователям.

Чтобы внести нужные изменения в реестр, необходимо

запустить страницу веб-сайта, в которой реализовано олицетворение. Сценарий регистрации пользователя в сети, созданный в PowerShell, обеспечивает последовательный запуск сайтов в фоновом режиме. Пример такого сценария приведен в листинге 2.

Листинг 2. Сценарий регистрации пользователей в сети (PowerShell)

```
$ServerLogon=(dir env:logonserver).value # Сервер загрузки
cd $ServerLogon\NETLOGON\PowerShell
 [xml] $obj=gc .\IIS.xml # Чтение XML-файла

# Чтение раздела ROOT - PART - IIS, параметра ENABLE
if ([int32]$obj.root.part.iis.enable -eq 1 )
{
# Чтение списка сайтов из ROOT - IIS - SITE,
# параметры VISIBLE и URL
$obj.root.iis.site | % {
# Вызов версии Internet Explorer
$ie = New-Object -com InternetExplorer.Application
while ($ie.busy -eq $true ) {}
$ie.visible =[int32] $_.visible
$ie.navigate($_.url)
# Назначение времени ожидания
$timeUntil =(Get-Date).AddMinutes($obj.root.part.iis.timeout)
# Контроль времени ожидания
while ($ie.busy -eq $true) {If($timeUntil ↵
        -lt (Get-Date)){$ie.quit();Write-Host ↵
        "windows was closed"} }
# Обеспечение последовательного выполнения сценариев
switch ($ie.visible){
$true {while ($ie.visible -eq $true) {}}
$false {$ie.quit()}
}
}
}
```

Для реализации отказоустойчивости каждый модуль имеет конфигурационный файл, в котором сосредоточены внешние настройки (см. листинг 3). Все файлы построены по шаблону [3]. В нем присутствует обязательная и индивидуальная части. В данном случае ею является блок, в котором перечислены сайты, которые будут последовательно выполняться. В случае необходимости можно отображать какой либо-сайт. Для этого используется свойство visible.

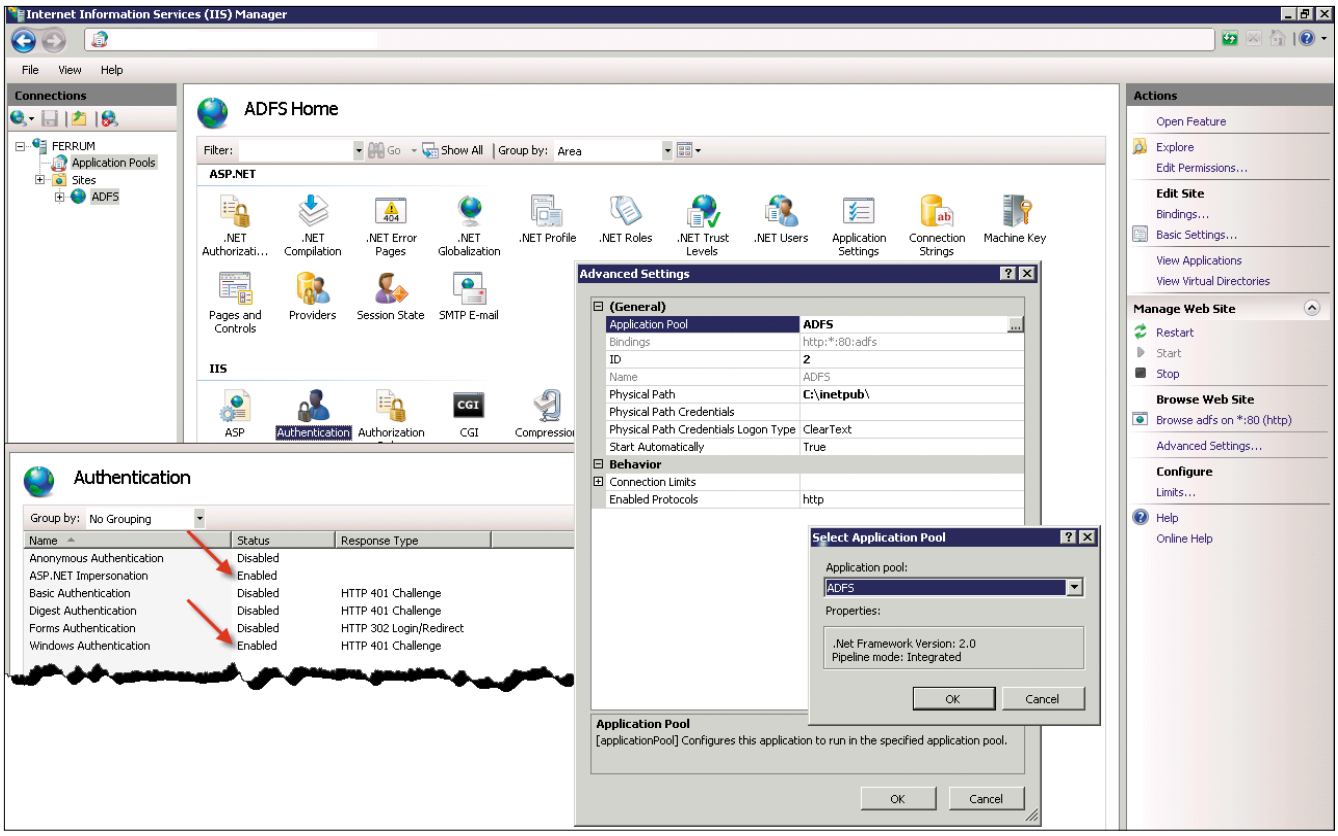
Листинг 3. Конфигурационный файл для сценария регистрации пользователей в сети (PowerShell)

```
<?xml version="1.0" encoding="utf-8" ?>
<Root>

<Part description="Настройка папки Мой Компьютер">
<iis enable="1" visual="1" timeout="2"/>
</Part>
  <IIS>
    <site visible="0" url="http://mycomputer" />
    <site visible="0" url="http://personaltlf"/>
    <site visible="0" url="http://addin"/>
  </IIS>
 </Root>
```

После того как сценарий вызвал сайт, например http://mycomputer, на IIS-сервере запускается страница от имени пользователя. Каждая из таких страниц должна обеспечивать удаленный доступ к реестру. Для этого необходимо определить имя компьютера в сети. В листинге 4 приведен шаблон получения удаленного доступа к ветви HKLM и создания в ней нового раздела и параметра.

Рисунок 2. Настройка олицетворения приложения. Веб-сайт



Листинг 4. Подключение к кусту удаленного реестра (VB.NET)

```
Partial Class _Default
Public hklm As Microsoft.Win32.RegistryKey
Public hccr As Microsoft.Win32.RegistryKey
Public pcname As String
Public Key As String = "SOFTWARE\Microsoft\Windows\ ↵
       CurrentVersion\..."

Protected Sub form1_Load(ByVal sender As Object, ↵
       ByVal e As System.EventArgs) Handles form1.Load
pcname = Request.UserHostName
    Try
       hklm = Microsoft.Win32.RegistryKey. ↵
           OpenRemoteBaseKey(Microsoft.Win32. ↵
           RegistryHive.LocalMachine, pcname)
       hklm.OpenSubKey(Key1, True).CreateSubKey(SubKeyName)
       hklm.OpenSubKey(Key1, True). ↵
           CreateSubKey(SubKeyName).SetValue(FlagKey, ↵
           FlagValue, Microsoft.Win32.RegistryValueKind. ↵
           String)
       Catch ex As Exception
           ErrorMessage("Remote Registry Access: " + ↵
                ex.Message)
       End Try
…
End Sub
```

Внимание! Удаленная запись в реестр возможна при условии запуска службы«Remote Registry. Используйте групповые политики для ее включения на всех рабочих станциях домена.

Динамическое управление ярлыками приложений в папке «Мой Компьютер»

На основе описанной технологии реализуется динамическое управление ярлыками приложений в папке «Мой Компьютер» на основе членства в группе безопасности каталога Active Directory (см. рис. 3). Такое решение позволяет централизованно управлять предоставлением доступа к сетевым, portable-приложениям, расположенным в сети, и локальным приложениям.

Создав ярлыки пользователей в папке «Мой Компьютер», можно избавиться от нескольких лишних сетевых дисков – теперь они просто не нужны. Ускорить доступ к ресурсу, сократив количество кликов. С точки зрения безопасности это решение также предпочтительно: сотрудник не сможет увидеть путь к приложению, удалив ярлык.

Создание ярлыка – комплексная задача, требующая знаний реестра. Процедура создания ярлыка для приложения состоит из двух частей. В первой части описывается местоположение объекта, а во второй – его свойства.

Для описания местоположения объекта (папка «Мой компьютер»), достаточно в ветви HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\MyComputer\NameSpace создать папку. Имя папки – уникальный CLSID, например {C7AF0CFE-D0C4-11DC-B55C-F6B756D89593}.

Для его создания можно воспользоваться любой утилитой, генерирующей GUID, например, стандартной утилитой uuidgen.exe, входящей в состав Microsoft SDK. После установки пакеты утилита находится в папке C:\Program Files\Microsoft SDK\Bin.

Описание свойств нового объекта находится в кусте CLSID раздела HCCR. В нем необходимо создать раздел, имя которого – сгенерированный CLSID. Внутренняя структура подпапок, которую необходимо воспроизвести, приведена на рис. 4.

Назначение папок, содержащиеся в них ключи и соответствующие им значения описаны в таблице 1.

Персонализация ярлыка определяется членством учетной записи пользователя в соответствующей группе безопасности. В каталоге Active Directory присутствует несколько групп, каждая из которых соответствует какому-либо ресурсу.

В свойствах этой группы описаны характеристики ярлыка (их описание приведено на рис. 5).

Чтение данных из Active Directory осуществляется с помощью стандартной .NET FrameWork библиотеки System.DirectoryServices, пространство имен которой необходимо импортировать в проект. Алгоритм работы этой части сайта следующий:

- определение имени текущего домена;
- поиск групп безопасности с фильтра;
- чтение характеристик группы.

Определение имени домена осуществляется с помощью виртуального объекта RootDSE. Этот объект присутствует во всех доменах. Считывая значение свойства DefaultNamingContext, получают имя текущего домена (см. листинг 5).

Для поиска групп безопасности используют объект DirectoryEntry [4]. В качестве фильтра указывается имя группы, значением которого является атрибут cn (см. рис. 6) и тип объекта.

После этого осуществляются чтение значений атрибутов description и info и, наконец, запись данных в удаленный реестр.

Рисунок 3. Внешний вид папки «Мой Компьютер»

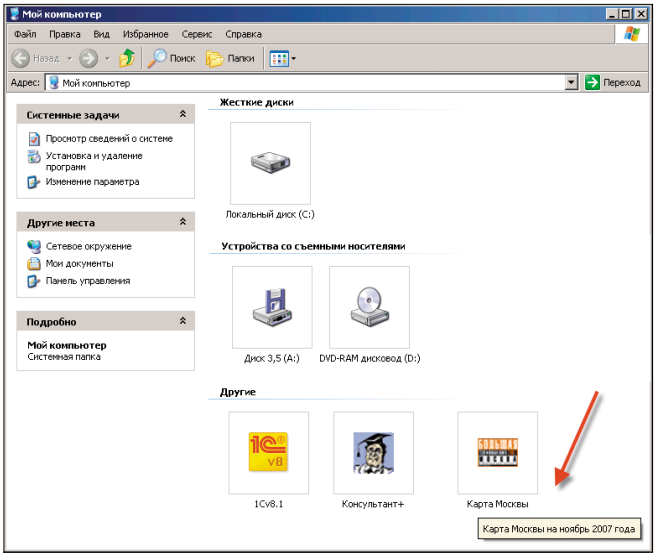
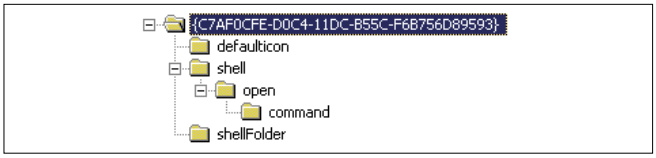


Рисунок 4. Структура раздела в HCCR\CLSID\{GUID}



Листинг 5. Поиск групп в Active Directory (VB.NET)

```
Imports System.DirectoryServices
…
Public Domain As String = ""
…
Dim obj As New DirectoryEntry("LDAP://RootDSE")
Domain = "LDAP://" + ↵
       obj.Properties("DefaultNamingContext").Value

Dim obj As New DirectorySearcher()
obj.SearchRoot = New DirectoryEntry("LDAP://" + Domain)
Dim query As String = ↵
       "(&(objectclass =group)(cn=" + Prefix + "*))"
obj.Filter = query
Dim bb As SearchResultCollection
bb = search.FindAll
    For Each b As SearchResult In bb
        Dim path As String = b.GetDirectoryEntry. ↵
            Properties("distinguishedName").Value.ToString()
        Dim read = GetObject("LDAP://" + path)
        ' чтение поля INFO
        For Each t As String In read.info
            Response.Write (t + "<br>")
        Next
        ' чтение поля DESCRIPTION
            Response.Write read.description
    Next
```

***

Другой яркий пример использования IIS – управление AD обычным пользователем. Звучит конечно странно, но это позволяет максимально повысить эффективность работы службы поддержки и др. Например, телефонный справочник организации, построенный на базе информации из каталога AD. Используя олицетворение, обеспечиваем возможность коррекции данных справочника при помощи пользователей, не наделяя специалиста службы поддержки административными полномочиями для коррекции изменений. EOF

1. Элемент identity (схема параметров ASP.NET) – http://msdn.microsoft.com/ru-ru/library/72wdk8cc.aspx.
2. Коробко И. Практика внедрения сценария регистрации пользователей в сети на PowerShell. //«Системный администратор», №3, 2010 г. – С. 46-52.
3. Коробко И. На языке PowerShell. Сценарий регистрации пользователей в сети. Часть 1. //«Системный администратор», №9, 2010 г. – С. 43-45.
4. System.DirectoryServices – http://msdn.microsoft.com/library/default.asp?url=/library/en-us/sds/sds/directoryservices_directorysearcher_filter.asp.

Таблица 1. Описание свойств ярлыка в HCCR\CLSID

Раздел реестра	Ключ	Тип данных	Значение	Комментарий
HCCR\CLSID	@	REG_SZ	Карта г. Москвы	Название ярлыка, отображаемое в папке «Мой Компьютер»
HCCR\CLSID	infotip	REG_SZ	Карта г.Москвы за ноябрь 2007 года	Подробное описание ярлыка. Отображается, если навести курсор на ярлык и подождать 1-2 секунды (см. рис. 1, указано красной стрелкой)
HCCR\CLSID\defaulticon	@	REG_SZ	\\Server\Folder$\MoscowMap\Btk2007.exe,0 или \\Server\Folder$\MoscowMap\Map.ico	Путь к иконке, которую увидит пользователь
HCCR\CLSID\defaulticon\shell\open\command	@	REG_SZ	\\Server\Folder$\MoscowMap\Btk2007.exe	Путь к приложению, которое будет запускаться при нажатии на иконку
HCCR\CLSID\shellfolder	Attributes	REG_BINARY	hex:00,01,00,a0	Благодаря этому ключу созданный ярлык нельзя переименовать, удалить и т.д.
HCCR\CLSID\shellfolder	Attributes	REG_BINARY	hex:00,01,00,a0	Благодаря этому ключу созданный ярлык нельзя переименовать, удалить и т.д.

Рисунок 5. Параметры ярлыка в группе безопасности Active Directory

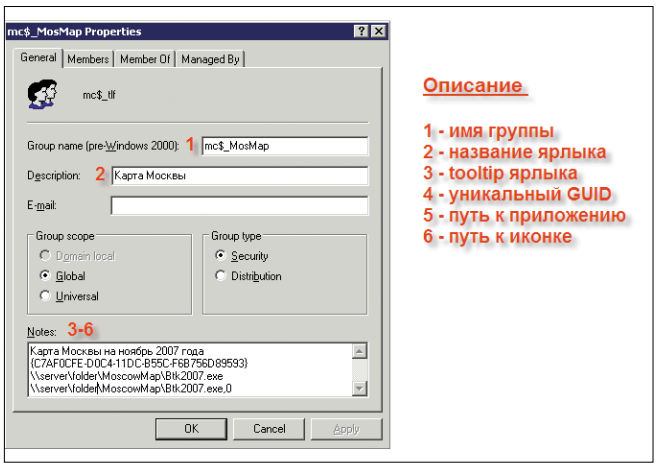


Рисунок 6. Чтение полей группы безопасности в Active Directory