

# Управляем объектами в Active Directory

## Часть 2

*Иван Коробко*

Залог успешного чтения параметров программным способом – верное определение типа данных считываемой переменной и использование соответствующего алгоритма чтения.

В предыдущей статье [6] мы рассмотрели типы объектов, существующие в Active Directory; было рассказано о методике создания объектов на примере учетной записи пользователя, узнали, что происходит в Active Directory в процессе создания учетной записи. В этой статье уделим внимание типам данных объектов в Active Directory и методике их чтения.

Чтение/изменение свойств объекта осуществляется одним из следующих способов:

- программным способом;
- с помощью мастера MMC-консоли Active Directory.

В этой статье будет описан программный способ чтения параметров учетной записи. Второй способ будет рассмотрен в следующий раз.

## Изменение свойств объекта программным способом

Каждое свойство в каталоге в Active Directory хранится в определенном формате, поэтому чтение данных осуществляется по алгоритму, пригодному для чтения каждого из типов данных. Одной из важнейших задач является определение типа данных объекта, значение которого необходимо получить или изменить.

## Определение типа данных объекта

Информация в Active Directory хранится в объектах. Для определения типа объекта в VBScript обычно используют функцию VarType(), которая имеет следующий синтаксис:

```
Value=VarType (VarName)
```

где:

- **VarName** – переменная, тип которой необходимо определить;
- **Value** – возвращаемое функцией значение.

Функция VarType() возвращает числовое значение, по которому из **таблицы 1** можно определить тип данных объекта. Шаблон сценария, в котором реализовано определение типов данных переменной с помощью функции VarType(), приведен в **листинге 1**.

Особое внимание стоит уделить чтению данных из массивов. Во время определения типа данных массива функция VarType() возвращает сумму, состоящую из двух слагаемых. С помощью первого слагаемого, равного 8192, идентифицируется массив данных. С помощью второго – элементы массива. Например, если массив состоит из чисел (Integer), то функция VarType() возвратит значение  $8192 + 2$ . Поскольку элементами массива не могут быть пустые значения (vbEmpty), то функция VarType(Array) никогда не возвратит значение 8192.

Листинг 1. Шаблон использования функции VarType()

```
'Создание ADODB-соединения
Set objConn = CreateObject("ADODB.Connection")
Set objCom = CreateObject("ADODB.Command")
objConn.CommandTimeout = 120
objConn.Provider = "AdsDSOObject"
objConn.Open "Active Directory Provider"
Set objCom.ActiveConnection=objConn
```

```
'Определение имени домена
Set oRoot = GetObject("LDAP://rootDSE")
LdapDomain = "LDAP://" & oRoot.Get("DefaultNamingContext")

'составление и обработка SQL-запроса
Query = "SELECT ПЕРЕМЕННАЯ FROM '" & LdapDomain & "' _
        WHERE objectclass='group' and name='Test'"
Set st = objConn.Execute(Query)

'Вывод результата типа данных переменной
wscript.echo VarType(st.Fields("ПЕРЕМЕННАЯ").Value)
```

## Методика чтения данных из Active Directory

Рассмотрим методику чтения параметров следующих типов данных:

- **Long (VarType = 3)** – длинное число;
- **Date (VarType = 7)** – дата-время;
- **String (VarType = 8)** – строка;
- **Object (VarType = 9)** – объект;
- **Array (VarType = 8192+x)** – массив элементов.

## Тип данных Long (VarType = 3)

Длинному числу в интерпретации VBScript соответствует тип данных Long. В **таблице 2** приведены характерные для учетной записи пользователя параметры, соответствующие

Таблица 1. Значения, возвращаемые функцией VarType()

| Константа    | Значение | Описание                                  |
|--------------|----------|---|
| vbEmpty      | 0        | Empty (пустое значение)                   |
| vbNull       | 1        | Null (не содержит данных)                 |
| vbInteger    | 2        | Integer                                   |
| vbLong       | 3        | Long                                      |
| vbSingle     | 4        | Single (число с плавающей точкой)         |
| vbDouble     | 5        | Double (число с плавающей точкой)         |
| vbCurrency   | 6        | Currency                                  |
| vbDate       | 7        | Date                                      |
| vbString     | 8        | String                                    |
| vbObject     | 9        | Object                                    |
| vbError      | 10       | Error                                     |
| vbBoolean    | 11       | Boolean                                   |
| vbVariant    | 12       | Variant (используется только с массивами) |
| vbDataObject | 13       | Object                                    |
| vbByte       | 17       | Byte                                      |
| vbArray      | 8192     | Array                                     |

Таблица 2. Параметры объектов типа Long для учетной записи пользователя

| Переменная         | Комментарий   | Пример    |
|--------------------|---|-----------|
| codePage           | Код страны. Используется для переключения языка интерфейса. Появился в Windows 2003. В настоящее время не используется  | 0         |
| countryCode        |   | 0         |
| instanceType       | Значением параметра описывается статус объекта на сервере. Возможные значения – 1, 2, 4, 8, 16, 32. По умолчанию принимает значение 4                                 | 4         |
| primaryGroupID     | Идентификатор группы безопасности, назначенной по умолчанию (см. вкладку MemberOf свойств пользователя). RID группы Domain Users, назначаемой по умолчанию, равен 513 | 513       |
| userAccountControl | Параметр, значение которого – сумма, складывающаяся из различных настроек безопасности учетной записи пользователя  | 805306368 |

Таблица 3. Параметры типа Date всех типов объектов

| Переменная      | Комментарий  | Пример            |
|-----------------|--|-------------------|
| createTimeStamp | Дата и время создания объекта. При попытке получить значение с помощью функции GetObject() выдает ошибку   | 20070222080007.0Z |
| modifyTimeStamp | Дата и время последнего изменения свойств объекта. При попытке получить значение с помощью функции GetObject() выдает ошибку   | 20080507120605.0Z |
| whenCreated     | Дата и время создания объекта. Значение совпадает со значением параметра createTimeStamp. Используется для совместимости с доменами Windows NT. Чтение данных с помощью функции GetObject() осуществляется корректно                     | 20070222080007.0Z |
| whenChanged     | Дата и время последнего изменения свойств объекта. Значение совпадает со значением параметра modifyTimeStamp. Используется для совместимости с доменами Windows NT. Чтение данных с помощью функции GetObject() осуществляется корректно | 20080507120605.0Z |

| Name            | Value             | Type                  | Size |
|-----------------|-------------------|-----------------------|------|
| cn              | Ivan, Petrov      | text attribute        | 12   |
| createTimeStamp | 20080410090637.0Z | operational attribute | 17   |
| modifyTimeStamp | 20080410090637.0Z | operational attribute | 17   |
| whenChanged     | 20080410090637.0Z | text attribute        | 17   |
| whenCreated     | 20080410090637.0Z | text attribute        | 17   |

Рисунок 1. Параметры в формате Generalized-Time

этому типу данных. При чтении данных значение типа Long автоматически преобразуется к строке (String).

Чтение данных осуществляется с помощью ADODB-соединения или функции GetObject(). В том случае, когда известен путь к объекту, стоит отдать предпочтение функции GetObject(), поскольку этот способ значительно проще (см. **листинг 2а**).

Листинг 2а. Чтение значений типа Long с помощью функции GetObject()

```
Set obj = GetObject("LDAP://CN=Test,OU=Group,DC=msk,DC=ru")
Wscript.Echo obj.sAMAccountType
```

Когда точное местоположение объекта неизвестно, то необходимо выполнить поиск объекта в Active Directory по заданным критериям, однако в запросе можно задать путь к объекту явным образом (см. **листинг 2б**) – как в функции GetObject(). Для реализации функции поиска используют ADODB-соединение провайдера AdsDSOObject.

Листинг 2б. Чтение значения строковых параметров с помощью ADODB-соединения

```
'Создание ADODB-соединения
Set objConn = CreateObject("ADODB.Connection")
Set objCom = CreateObject("ADODB.Command")
objConn.CommandTimeout = 120
objConn.Provider = "AdsDSOObject"
objConn.Open "Active Directory Provider"
Set objCom.ActiveConnection=objConn
'Составление и обработка SQL-запроса
Query = "SELECT sAMAccountType
FROM 'LDAP://CN=Test,OU=Group,DC=msk,DC=ru'"
Set st = objConn.Execute(Query)
'Вывод результата
wscript.echo st.Fields("sAMAccountType").Value
```

## Тип данных Date (VarType = 7)

В Active Directory присутствует ряд обязательных для всех объектов параметров, к их числу относятся параметры ти-

па Date, характеризующие дату и время создания и изменения объектов (см. **таблицу 3**). Данные в формате дата-время в процессе чтения автоматически преобразуются в строку. Несмотря на это, для получения понятного всем значения необходимо преобразовать это значение в соответствии с алгоритмом (см. **листинг 3а** и **3б**).

В Active Directory эти параметры хранятся в формате Generalized-Time. Значения параметров этого типа имеют следующую структуру (см. **рис. 1, таблицу 3**):

YYYYMMDDHHMMSS.MS Z

где: YYYY – год, MM – месяц, DD – день, HH – час, MM – минута, SS – секунда, MS – доля секунды, Z (от Zero) – обозначает нулевой меридиан.

Чтение параметров в формате Generalized-Time рекомендуется осуществлять с помощью ADODB-соединения (**листинг 3а**), поскольку значения параметров createTimeStamp и modifyTimeStamp невозможно получить с помощью функции GetObject(). Это связано с тем, что функция первоначально использовалась для чтения данных через провайдер WinNT (**листинг 3б**). К ним относятся параметры whenCreated и whenChanged.

Листинг 3а. Чтение параметров в формате UTC-Time

```
'Создание ADODB-соединения
Set objConn = CreateObject("ADODB.Connection")
Set objCom = CreateObject("ADODB.Command")
objConn.CommandTimeout = 120
objConn.Provider = "AdsDSOObject"
objConn.Open "Active Directory Provider"
Set objCom.ActiveConnection=objConn
'Составление и обработка SQL-запроса
Query = "SELECT createTimeStamp
FROM 'LDAP://CN=Ivan\, Petrov,CN=Users,DC=msk,DC=ru'"
Set st = objConn.Execute(Query)
'Вывод результата
wscript.echo vartype(st.Fields("createTimeStamp").Value)
wscript.echo (st.Fields("createTimeStamp").Value)
```

Результатом работы сценария является дата и время в привычном для человека формате (см. **рис. 2**).

Листинг 3б. Чтение параметров в формате UTC-Time

```
' Только для параметров whenCreated, whenChanged
set objUser = GetObject("LDAP://CN=Ivan\, Petrov,CN=Users,DC=msk,DC=ru")
WScript.echo objUser.Get("whenCreated")
```

## Тип данных string (VarType = 8)

Большинство данных в каталоге Active Directory хранится в строковом виде. В **таблице 4** приведено описание полей, которые чаще всего используются в сценариях.

Чтение данных осуществляется с помощью функции GetObject() или универсального способа – с помощью ADODB-соединения. Первый способ (см. **листинг 4а**) хорош своей лаконичностью: всего две строки. Его основной недостаток – необходимо знать LDAP URL-путь к объекту.

Когда точное местоположение объекта неизвестно, используют

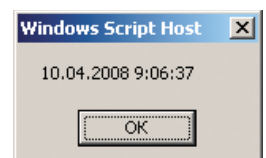


Рисунок 2. Чтение параметра в формате Generalized-Time

возможность поиска ADODB-соединения (см. **листинг 4б**). В приведенном примере механизм поиска объекта не использован. (Поиск объекта осуществляется на основе SQL-запроса, в котором указан путь к корневой папке домена.) Путь к объекту задан в явном виде. Во всех трех примерах продемонстрировано чтения одного и того же параметра.

Листинг 4а. Чтение значения строковых параметров

```
Set obj = GetObject("LDAP://CN=Test,OU=Group,DC=msk, DC=ru")
Wscript.Echo obj.cn
```

Листинг 4б. Чтение значения строковых параметров с помощью ADODB-соединения

```
'Создание ADODB-соединения
Set objConn = CreateObject("ADODB.Connection")
Set objCom = CreateObject("ADODB.Command")
objConn.CommandTimeout = 120
objConn.Provider = "AdsDSOObject"
objConn.Open "Active Directory Provider"
Set objCom.ActiveConnection=objConn
'Составление и обработка SQL-запроса
Query = "SELECT cn
FROM 'LDAP://CN=Test,OU=Group,DC=msk,DC=ru'"
Set st = objConn.Execute(Query)
'Вывод результата
wscript.echo st.Fields("cn").Value
```

## Тип данных object (VarType = 9)

Хранящаяся в Active Directory информация, соответствующая типу данных Large Integer, воспринимается интерпретатором VBScript как объект (см. **таблицу 5**).

Все данные типа Object, хранящиеся в Active Directory, считываются с помощью функции Get(). Впоследствии полученное значение складывают из двух: HighPart (Дата) и LowPart (Время). Пример чтения в **листинге 5**.

Листинг 5. Получение значения типа Object

```
Set obj = GetObject("LDAP://CN=Ivan\, CN=Petrov,CN=Users,DC=msk,DC=ru")
Set objUSN = obj.Get("uSNChanged")
Wscript.Echo Abs(objUSN.HighPart * 2^32 + objUSN.LowPart)
```

Часть параметров, например pwdLastSet (см. **таблицу 5**) имеет несколько другую методику чтения, которая также используется для полей accountExpires, uSNChanged, lastLogonTimestamp, lastLogon, badPasswordTime. Значение параметра pwdLastSet – промежуток времени от 1 января 1601 года по настоящее время с точностью 100 миллисекунд. Преобразование данных осуществляется с помощью функции (см. **листинг 6**).

Листинг 6. Чтение параметра lastLogonTimestamp (accountExpires, lastLogon, pwdLastSet)

```
Set obj = GetObject("LDAP://CN=Ivan\, CN=Petrov,CN=Users,DC=msk,DC=ru")
Set objUSN = obj.Get("lastLogonTimestamp")
Wscript.Echo ConvertTime(objUSN)

Function ConvertTime(objDate)

Set objShell = CreateObject("Wscript.Shell")
Key="HKLM\System\CurrentControlSet\Control\ TimeZoneInformation\ActiveTimeBias"
TempKey = objShell.RegRead(Key)

If (VarType(TempKey) = 3) Then
Temp = TempKey
ElseIf (VarType(TempKey)=12) Then
```

```
Temp = 0
For k = 0 To UBound(TempKey)
Temp = Temp + (TempKey(k) * 256^k)
Next

End If

lngHigh = objDate.HighPart
lngLow = objDate.LowPart
If (lngLow < 0) Then
lngHigh = lngHigh + 1
End If
If (lngHigh = 0) And (lngLow = 0 ) Then
dtmDate = #1/1/1601#
Else
dtmDate = #1/1/1601# + (((lngHigh * 2 ^ 32)) + lngLow)/6/10^8 - Temp)/1440
End If
ConvertTime=dtmDate
End Function
```

## Тип данных array (VarType = 8192+x)

Возвращаемое значение функции VarType складывается из двух составляющих: 8192 (array) + x. В данном выражении x – число, обозначающее тип данных элементов массива. Так, массив, элементы которого являются строками, возвращает значение 8200 = 8192 + 8; если массив состоит из бинарных элементов, то возвращаемое значение – 8209 = 8192 + 17 (см. **таблицу 6**).

Массивы, возвращаемые Active Directory, как правило, состоят из неопределенных элементов (Variant = 12). Функция в этом случае возвращает значение 8204 = 8192 + 12. Элементами массива в данном случае является либо строка, либо число, которое при получении значений элементов автоматически преобразуется в строку (см. **листинг 7**). Чтение массива осуществляется с помощью цикла For...Each.

Таблица 4. Параметры типа String для учетной записи пользователя

| Переменная        | Комментарий   | Пример                                  |
|-------------------|---|---|
| cn                | Составное имя пользователя (cn – сокращение от Canonical Name), формирующееся из фамилии (поле givenName), имени (поле sn) и необязательного инициала (поле initials) | Ivan, Petrov                            |
| displayName       | Отображаемое имя. Его значение идентично составному имени пользователя (поле cn). Используется для совместимости с доменами Windows NT                                | Ivan, Petrov                            |
| distinguishedName | Составной путь (DC) к объекту в соответствии с RFC 1779, 2247   | CN=Ivan\,Petrov, CN=Users,DC=msk, DC=ru |
| mail              | Адрес электронной почты в формате UPN. Заполняется после создания ящика электронной почты для учетной записи пользователя   | IPetrov@msk.ru                          |
| sAMAccountName    | Имя пользователя для входа в сеть. Используется для совместимости пространств имен Active Directory с доменами на основе Windows NT                                   | IPetrov                                 |
| sn                | Фамилия пользователя (sn – сокращение от Second Name)   | Petrov                                  |
| userPrincipalName | UPN-имя пользователя, описанное в RFC 822. Это основной формат имен, используемый в Active Directory  | IPetrov@msk.ru                          |
| givenName         | Имя учетной записи пользователя. Является обязательным параметром   | Ivan                                    |
| initials          | Инициалы пользователя. Необязательный параметр длиной до 6 символов   | V                                       |

Листинг 7. Чтение массива строк с помощью ADODB-соединения

```
'Создание ADODB-соединения
Set objConn = CreateObject("ADODB.Connection")
Set objCom = CreateObject("ADODB.Command")
objConn.CommandTimeout = 120
objConn.Provider = "AdDSOObject"
objConn.Open "Active Directory Provider"
Set objCom.ActiveConnection=objConn
'составление и обработка SQL-запроса
Query = "SELECT OBJECTCLASS
FROM 'LDAP://CN=Test,OU=Group,DC=msk,DC=ru'"
Set st = objConn.Execute(Query)
'Вывод результата
For each el in st.Fields("OBJECTCLASS").Value
wscript.echo el
Next
```

В Active Directory присутствует два массива бинарных элементов, которые необходимо выделить отдельно: SID (параметр objectSID) и GUID (параметр objectGUID) объекта. В связи с этим функция VarType() возвращает значение 8209 = 8192 + 17. После того как элементы массива считаны, их необходимо преобразовать в понятный всем вид (см. рис. 3) с помощью функции HEX.

## Шаблон чтения данных

Суммируя сказанное, заметим, что грамотно написанный сценарий чтения каких-либо данных из Active Directory име-

Таблица 5. Параметры типа Object для учетной записи пользователя

| Переменная     | Комментарий   | Пример              |
|----------------|---|---------------------|
| uSNChanged     | Порядковый номер изменений свойств объекта, отсчитывающийся с момента создания объекта. Количество изменений, произошедших с объектом, равно разности uSNChanged – uSNCreated | 794175              |
| uSNCreated     | Значением является порядковый номер изменений, сделанных в Active Directory на момент создания объекта  | 794169              |
| accountExpires | Дата и время, по истечении которых учетная запись пользователя будет автоматически заблокирована. Если значение равно 0 – функция деактивирована                              | 9223372036854775807 |
| lastLogon      | Дата и время последней регистрации пользователя в сети. При попытке получить значение с помощью функции GetObject() выдает ошибку   | 128555016510096179  |
| pwdLastSet     | Дата и время последней смены пароля   | 128527377084375000  |

Таблица 6. Параметры типа Array для учетной записи пользователя

| Переменная  | Комментарий  | Пример   |
|-------------|--|--|
| description | Комментарий, описание объекта. Как ни странно – это не строка, а массив символов | Тестовая учетная запись                            |
| memberOf    | Список групп, членом которой является пользователь                               | Domain Users<br>Test                               |
| objectClass | Список идентификаторов класса, к которому относится объект                       | Top<br>Person                                      |
| objectGUID  | Глобальный идентификатор объекта   | 93 C5 2A A1 1D AF 10 42 AA<br>3E 7A 33 0F A3 E3 96 |
| objectSID   | Идентификатор безопасности объектов  | 01 05 00 00 00 00 05 15<br>00 00 00 2C 8F EC FB    |

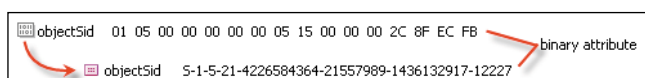


Рисунок 3. Преобразование SID-объекта

ет следующую структуру. Сначала с помощью ADODB-соединения получают доступ к каталогу Active Directory, указывая в запросе названия полей, значения которых необходимо получить. Затем определяется тип данных объекта и в зависимости от этого – считывание объекта по указанному алгоритму.

Конечно, возникает вопрос: «Если я знаю название поля и я точно знаю его тип данных – зачем использовать функцию VarType(). Ведь это усложняет сценарий, делает его громоздким?» Да, вопрос правомерный. Ответ на него очень прост: «Если всем параметрам присвоены значения, то все именно так – в функции VarType() нет необходимости. Однако на практике это не так. Более того. Часто создают сценарии, чтобы выявить незаполненные поля. Если поле не заполнено, например поле description, то функция VarType() возвратит значение NULL (см. таблицу 1).» В листинге 8 приведен пример чтения данных с проверкой функцией VarType().

Листинг 8. Рекомендуемый сценарий чтения данных из AD

```
'Создание ADODB-соединения
Set objConn = CreateObject("ADODB.Connection")
Set objCom = CreateObject("ADODB.Command")
objConn.CommandTimeout = 120
objConn.Provider = "AdDSOObject"
objConn.Open "Active Directory Provider"
Set objCom.ActiveConnection=objConn
'Составление и обработка SQL-запроса
Query = "SELECT description, cn
FROM 'LDAP://CN=Test,OU=Group,DC=msk,DC=ru'"
Set st = objConn.Execute(Query)
'Вывод результата
wscript.echo st.Fields("cn").Value
temp = ""
if VarType(st.Fields("description").Value)>8192 then
For Each description In st.Fields("description").Value
temp = temp + description
Next
Else
temp = "Поле description пусто"
End If
Wscrit.Echo temp
```

Обратите внимание, что в сценарии осуществляется проверка типа данных только для переменной description. Это связано с тем, что параметру cn всегда присвоено значение, поскольку оно является обязательным. Объект без этого параметра не может существовать.

## Заключение

В следующей статье будут рассмотрены основные вкладки учетной записи пользователя в MMC-консоли Active Directory. Поговорим о правилах назначения типов данных и исключениях из этих правил, а также о соответствиях полей мастера и параметров в Active Directory.

1. Коробко И. Администрирование сетей Windows с помощью сценариев. //СПб.: БХВ-Петербург, 2007. – 368 с.: ил. – (Системный администратор). ISBN 978-5-9775-0140-8.
2. RFC 822. Standard for the format of ARPA internet text messages.
3. RFC 1779. A string representation of distinguished names.
4. RFC 2247. Using Domains in LDAP/X.500 Distinguished Names.
5. RFC 2251. Lightweight Directory Access Protocol (v3).
6. Коробко И. Управляем объектами в Active Directory. Часть 1. //Системный администратор, №5, 2008 г. – С. 4-9.