

# LDAP-фильтры для поиска в Active Directory

windowsnotes.ru/activedirectory/ldap-filtry-dlya-poiska-obektov-v-active-directory-chast-3

30 января 2020 г.

## LDAP-фильтры для поиска объектов в Active Directory. Часть 3

30.01.2020

Рубрики: [Active Directory](#).

Продолжаем тему LDAP-фильтров. В третьей, завершающей статье рассмотрим некоторые инструменты для работы с фильтрами, а также примеры наиболее часто используемых фильтров. Начнем как всегда с

### PowerShell

У командлетов Get-ADUser, Get-ADComputer, Get-ADGroup и Get-ADObject имеется параметр **LdapFilter**, специально предназначенный для использования LDAP-фильтров при поиске объектов в Active Directory. Для примера найдем всех пользователей с именем Vasya с помощью такой команды:

```
Get-ADUser -LdapFilter "(cn=Vasya)"
```

```
PS C:\> Get-ADUser -LdapFilter "(cn=vasya)"

DistinguishedName : CN=Vasya,OU=Employees,DC=test,DC=local
Enabled           : True
GivenName        : Василий
Name             : Vasya
ObjectClass       : user
ObjectGUID        : c3f5f243-7d80-469e-9158-7174ea8324ea
SamAccountName    : vasya
SID              : S-1-5-21-3162703999-3679795140-219171969-2108
Surname          : Пупкин
UserPrincipalName : vasya@test.local
```

Большинство командлетов для поиска являются узкоспециализированными, т.е. предназначены только для определенного типа объектов (пользователи, компьютеры и т.п.). Исключение составляет командлет Get-ADObject, который может искать любые объекты. Например:

```
Get-ADObject -LdapFilter "(cn=Vasya)"
```

```
PS C:\> Get-ADObject -LdapFilter "(cn=vasya)"

DistinguishedName      Name ObjectClass ObjectGUID
-----
CN=Vasya,CN=Computers,DC=test,DC=local Vasya computer 34cb3515-83ad-40a3-8b6b-cb2262234774
CN=Vasya,OU=Employees,DC=test,DC=local Vasya user      c3f5f243-7d80-469e-9158-7174ea8324ea
```

При использовании Get-ADObject мы получили не только пользователей, а все объекты с указанным в фильтре именем. Если требуется найти только пользователей, то надо в фильтре добавлять дополнительные параметры. Например:

```
Get-ADObject -LdapFilter "(&(objectCategory=person)(objectClass=user)
(cn=Vasya))"
```

```
PS C:\> Get-ADObject -LDAPFilter "(&(objectCategory=person)(objectClass=user)(cn=vasya))"

DistinguishedName      Name ObjectClass ObjectGUID
-----
CN=Vasya,OU=Employees,DC=test,DC=local Vasya user c3f5f243-7d80-469e-9158-7174ea8324ea
```

При помощи LDAP-фильтра нельзя указать область поиска. Для примера найдем всех пользователей, у которых в описании (Description) имеется слово Руководитель:

```
Get-ADObject -LdapFilter "(Title=Руководитель*)" -Properties * | ft -a
DisplayName,Title
```

Эта команда произведет поиск по всему домену и выдаст всех найденных пользователей, вне зависимости от их местоположения.

```
PS C:\> Get-ADUser -LDAPFilter "(Title=Руководитель*)" -Properties * | ft -a DisplayName,Title

DisplayName      Title
-----
Павел Васильевич Петров  Руководитель отдела маркетинга
Василий Иванович Пупкин  Руководитель отдела ИТ
Василий Сергеевич Пупкин  Руководитель отдела логистики
```

Для уточнения области поиска есть параметр **SearchBase**, с помощью которого можно указать для поиска конкретное подразделение (OU), например:

```
Get-ADObject -LdapFilter "(Title=Руководитель*)" -Properties * -SearchBase
"OU=Employees,DC=Test,DC=local | ft -a DisplayName,Title
```

```
PS C:\> Get-ADUser -LDAPFilter "(Title=Руководитель*)" -Properties * -SearchBase "OU=Employees,DC=test,DC=local" | ft -a
DisplayName,Title

DisplayName      Title
-----
Павел Васильевич Петров  Руководитель отдела маркетинга
Василий Иванович Пупкин  Руководитель отдела ИТ
```

Еще один полезный параметр **Subtree**, с помощью которого можно ограничить глубину поиска. Этот параметр может принимать 3 значения:

- Base (0) — поиск только по указанному в запросе объекту. В результате поиска возвращается либо один объект, либо ничего. Данная область, как правило, используется для проверки наличия объекта.
- One level (1) — поиск только по дочерним объектам указанного объекта. Поиск по вложенным объектам не производится, также в результаты поиск не попадает сам базовый объект.
- Subtree (2) — поиск по всем дочерним объектам, включая вложенные. Сам базовый объект в поиск не попадает. Это значение используется по умолчанию.

Для примера возьмем предыдущую команду и ограничим поиск верхним уровнем (OneLevel):

```
Get-ADObject -LdapFilter "(Title=Руководитель*)" -Properties * -SearchBase "OU=Employees,DC=Test,DC=local -SearcScope OneLevel | ft -a DisplayName,Title,DistinguishedName
```

```
PS C:\> Get-ADUser -LDAPFilter "(Title=Руководитель*)" -Properties * -SearchBase "OU=Employees,DC=test,DC=local" -SearchScope OneLevel | ft -a DisplayName,Title,distinguishedname
```

DisplayName	Title	distinguishedname
Павел Васильевич Петров	Руководитель отдела маркетинга	CN=pavel,OU=Employees,DC=test,DC=local

А затем зададим поиск по всем вложенным объектам (Subtree):

```
Get-ADObject -LdapFilter "(Title=Руководитель*)" -Properties * -SearchBase "OU=Employees,DC=Test,DC=local -SearcScope SubTree | ft -a DisplayName,Title,DistinguishedName
```

Как видите, разница очевидна.

```
PS C:\> Get-ADUser -LDAPFilter "(Title=Руководитель*)" -Properties * -SearchBase "OU=Employees,DC=test,DC=local" -SearchScope Subtree | ft -a DisplayName,Title,distinguishedname
```

DisplayName	Title	distinguishedname
Павел Васильевич Петров	Руководитель отдела маркетинга	CN=pavel,OU=Employees,DC=test,DC=local
Василий Иванович Пупкин	Руководитель отдела ИТ	CN=Vasya,OU=IT,OU=Employees,DC=test,DC=local

## CMD

Для работы с Active Directory из командной строки существует великое множество различных утилит. Мы рассмотрим две наиболее часто используемые для поиска — **dsquery** и **dsget**.

Утилита **dsquery** возвращает различающееся имя (Distinguished Name) объекта, подходящего под заданные параметры, а для LDAP-фильтров у нее имеется параметр **filter**. К примеру, предыдущий запрос с использованием **dsquery** будет выглядеть так:

```
dsquery * OU=Employees,DC=test,DC=local -filter "(&(objectCategory=person)(objectClass=user)(Title=Руководитель*))" -Scope OneLevel
```

```
PS C:\> dsquery * OU=Employees,DC=test,DC=local -filter "(&(objectCategory=person)(objectClass=user)(Title=Руководитель*))" -Scope Onelevel
"CN=pavel,OU=Employees,DC=test,DC=local"
PS C:\>
```

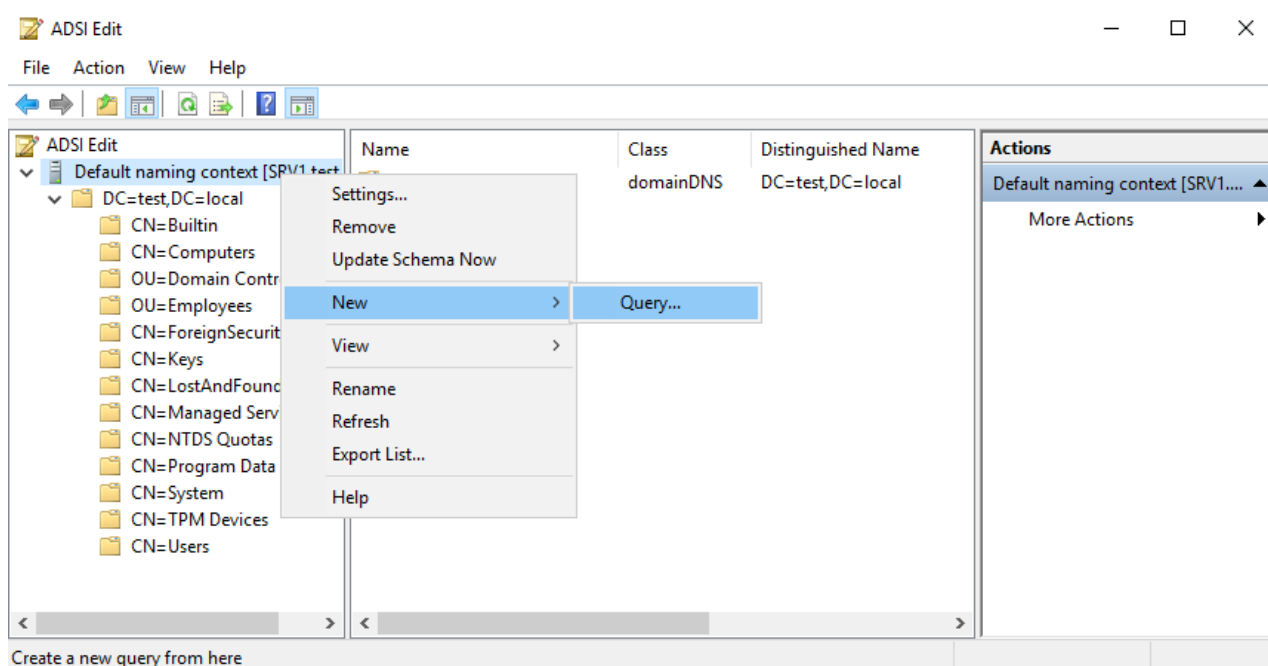
Утилита **dsget** получает на входе различающееся имя объекта и выдает для него значение указанного атрибута или атрибутов. Зачастую обе эти утилиты используются совместно, например:

```
dsquery * OU=Employees,DC=test,DC=local -filter "(&(objectCategory=person)(objectClass=user)(Title=Руководитель*))" | dsget user -display -title -dn
```

```
PS C:\> dsquery * OU=Employees,DC=test,DC=local -filter "(&(objectCategory=person)(objectClass=user)(Title=Руководитель*))" | dsget user -display -title -dn
dn                                display                                title
CN=pavel,OU=Employees,DC=test,DC=local Павел Васильевич Петров    Руководитель отдела маркетинга
CN=Vasya,OU=IT,OU=Employees,DC=test,DC=local Василий Иванович Пупкин   Руководитель отдела ИТ
dsget succeeded
PS C:\>
```

## ADSIEdit

Переходим к графическим утилитам. Оснастка ADSIEdit поддерживает использование LDAP-фильтров. Для добавления фильтра надо кликнуть по выбранному контексту именования (NC) и в контекстном меню выбрать пункт New — Query...



В открывшемся окне указываем имя запроса, выбираем область поиска (Root of Search) и в поле Query String добавляем нужный фильтр. Для примера отберем всех отключенных пользователей. Дополнительно можно выбрать глубину поиска Subtree или One level.

В результате получим что то вроде этого.

**New Query** [X]

Name:

Root of Search:  
 Browse...

Query String:

Query Scope  
☒ Subtree search  
☐ One level search Edit Query...

OK Cancel

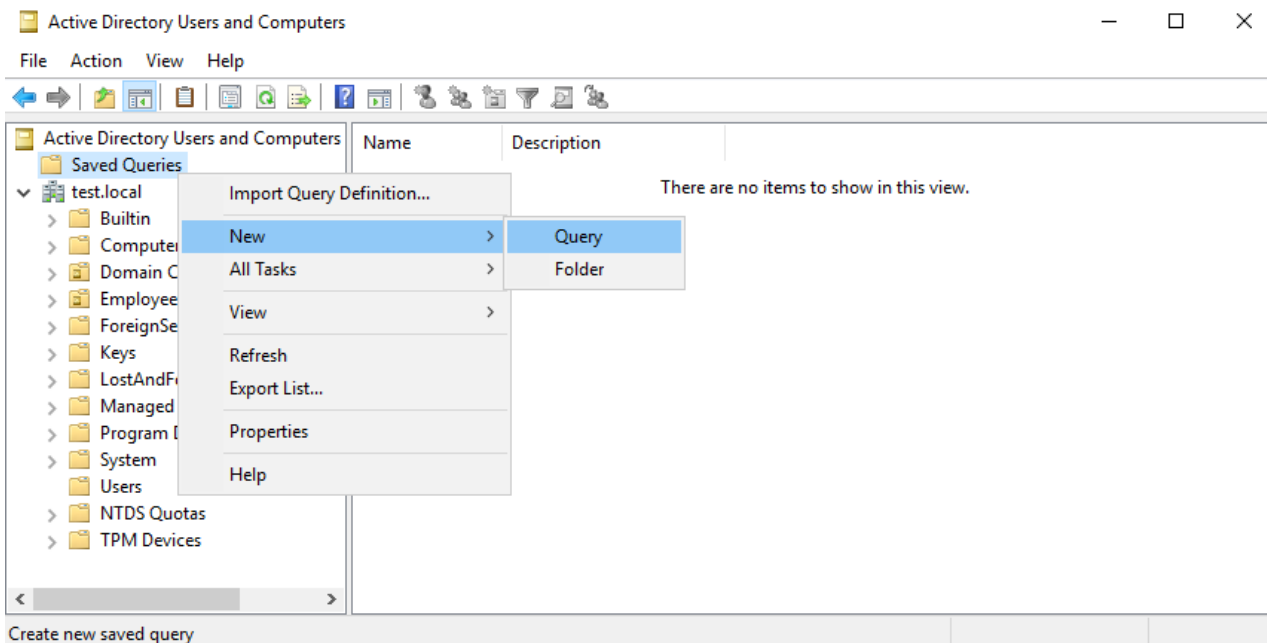
**ADSI Edit** [Min] [Max] [Close]

File Action View Help

Name	Class	Distinguished Name	Actions
CN=Builtin			
CN=Computers			
OU=Domain Controllers			
OU=Employees			
CN=ForeignSecurityPrincipals			
CN=Keys			
CN=LostAndFound			
CN=Managed Service Accounts			
CN=NTDS Quotas			
CN=Program Data			
CN=System			
CN=TPM Devices			
CN=Users			
All disabled users [DC=test,DC=local]			All disabled users ... ▲ More Actions ►
CN=Guest	user	CN=Guest,CN=Users,DC=test,DC=local	
CN=krbtgt	user	CN=krbtgt,CN=Users,DC=test,DC=local	
CN=ivan	user	CN=ivan,OU=Employees,DC=test,DC=local	

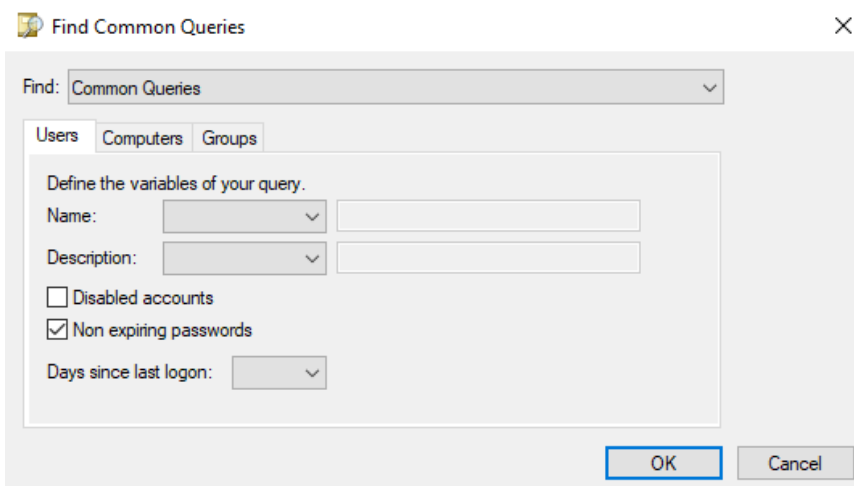
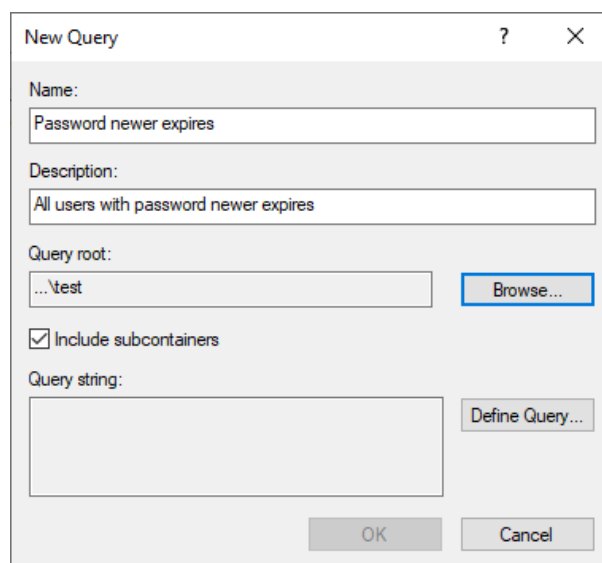
## Active Directory Users and Computers

В оснастке Active Directory Users and Computers (ADUC) имеется функция Saved Queries, которая представляет из себя сохраненные LDAP-фильтры. Для создания сохраненного запроса кликаем правой клавишей мыши и выбираем New — Query.



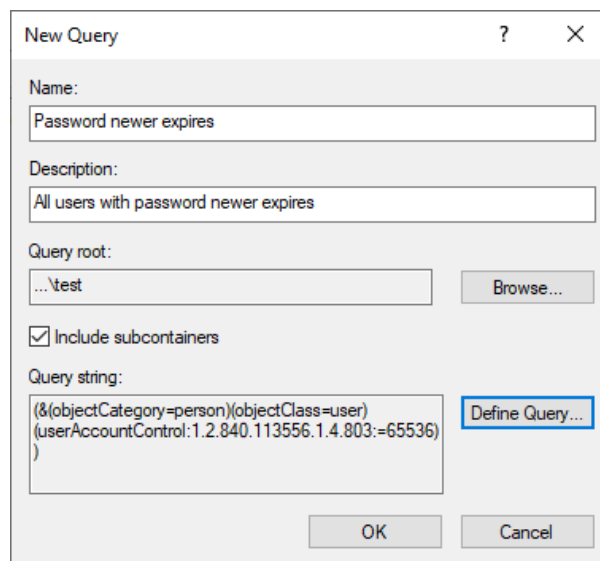
Даем запросу понятное имя и описание, в Query root выбираем область поиска и отмечаем пункт Include subcontainers (аналог Subtree) для поиска по всем вложенным объектам. Затем жмем кнопку Define Query.

Оснастка поддерживает несколько режимов и в принципе вовсе не обязательно писать текст фильтра вручную. Для наиболее часто встречающихся ситуаций достаточно выбрать нужный пункт и/или поставить галочку. Например, для показа всех пользователей с бессрочным паролем можно просто отметить чекбокс Non expiring passwords



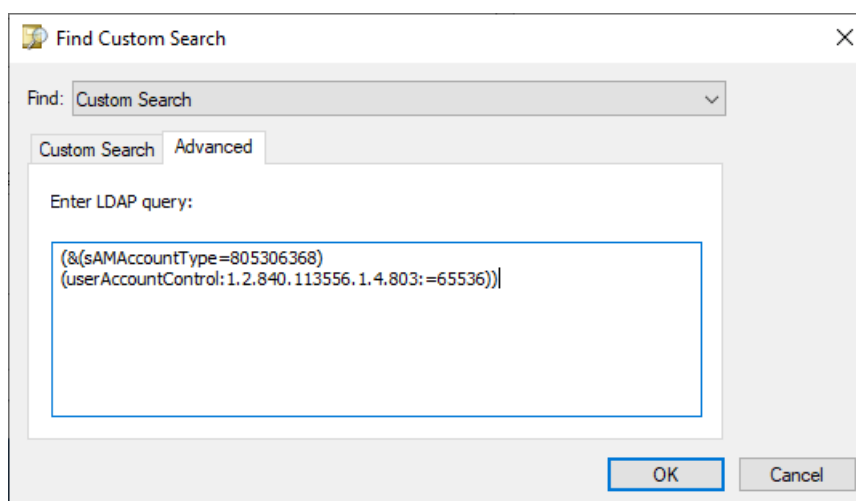
и нажать ОК.

Но мы не ищем легких путей, поэтому зададим фильтр руками. Для этого выбираем режим Custom Search и на вкладке Advanced вводим текст фильтра.



The 'New Query' dialog box is shown with the following fields and controls:

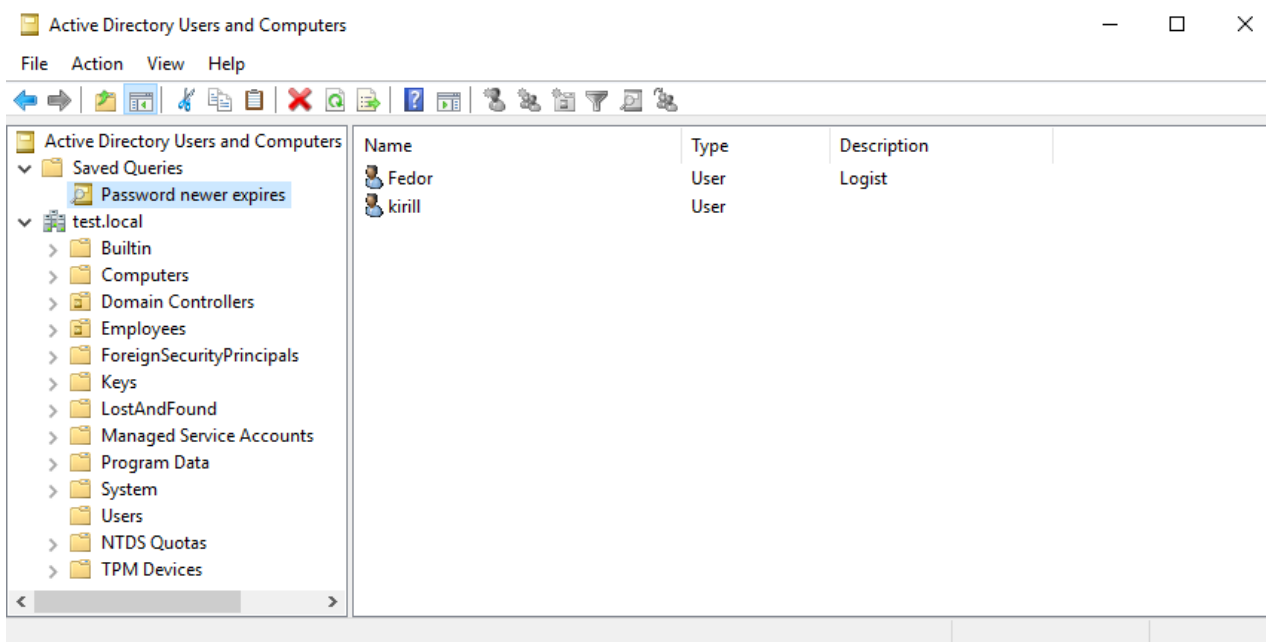
- Name:** Password newer expires
- Description:** All users with password newer expires
- Query root:** ...test (with a 'Browse...' button next to it)
- Include subcontainers:** ☒
- Query string:** (&(objectCategory=person)(objectClass=user)(userAccountControl:1.2.840.113556.1.4.803:=65536)) (with a 'Define Query...' button next to it)
- Buttons:** OK, Cancel



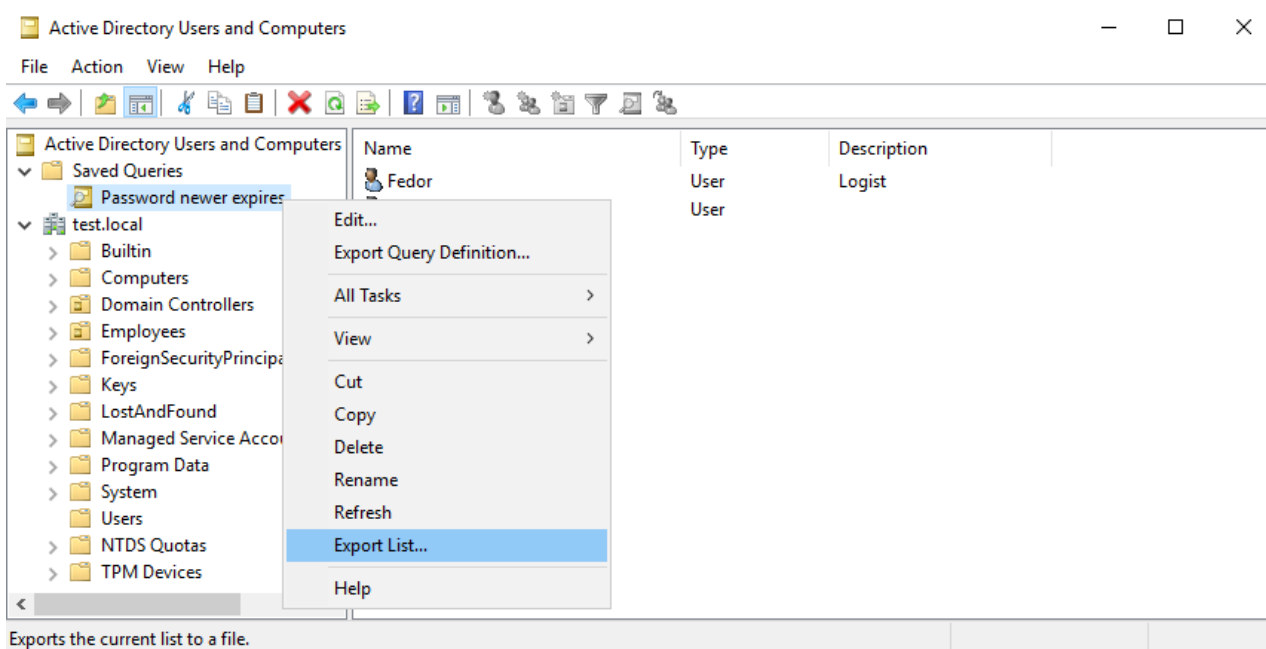
The 'Find Custom Search' dialog box is shown with the following fields and controls:

- Find:** Custom Search (dropdown menu)
- Tabs:** Custom Search, Advanced (the 'Advanced' tab is selected)
- Enter LDAP query:** (&(sAMAccountType=805306368)(userAccountControl:1.2.840.113556.1.4.803:=65536))
- Buttons:** OK, Cancel

В результате в папке Saved Queries появляется список пользователей, у которых пароль не истекает. Таким же образом можно отслеживать множество различных параметров пользователей, компьютеров и т.д.



А полученные результаты можно экспортировать в виде списка. Очень удобный и полезный функционал.



## Примеры LDAP-фильтров

В заключение приведу примеры наиболее часто используемых LDAP-фильтров. Для удобства фильтры сгруппированы по типу объектов (пользователи, компьютеры, группы и прочие непонятные сущности).

### Все пользователи:

`(&(objectCategory=person)(objectClass=user))`

или:

`(sAMAccountType=805306368)`



Фильтр с использованием sAMAccountType более эффективен для объекта пользователь.

#### **Все отключенные (Disabled) пользователи:**

```
(&(sAMAccountType=805306368)(useraccountcontrol:1.2.840.113556.1.4.803:=2))
```

#### **Все пользователи кроме отключенных:**

```
(&(sAMAccountType=805306368)  
(!useraccountcontrol:1.2.840.113556.1.4.803:=2))
```

#### **Заблокированные (Locked) пользователи:**

```
(&(sAMAccountType=805306368)  
(useraccountcontrol:1.2.840.113556.1.4.803:=16))
```

или:

```
(&(sAMAccountType=805306368)(badPwdCount>=4))
```

Здесь используется атрибут badPwdCount, в котором хранится количество неудачных попыток ввода пароля пользователем. Значение нужно указывать в соответствии с политиками безопасности вашего домена.

#### **Пользователи, не менявшие пароль более 3 месяцев:**

```
(&(sAMAccountType=805306368)(pwdLastSet<=132161330597286610))
```

Атрибут pwdlastSet содержит в себе дату и время последней смены пароля пользователем. Он имеет тип Integer8 и представляет собой число временных интервалов длительностью 100 наносекунд, прошедших с 12:00 01.01.1601 (UTC). Получить требуемое значение можно с помощью PowerShell, например:

```
(Get-Date).AddMonths(-3).ToFileTimeUtc()
```

#### **Пользователи, у которых пароль не истекает (Password never expires):**

```
(&(sAMAccountType=805306368)  
(userAccountControl:1.2.840.113556.1.4.803:=65536))
```

#### **Пользователи, обязанные сменить пароль при следующем входе в систему:**

```
(&(sAMAccountType=805306368)(pwdLastSet=0))
```

Если значение атрибута pwdlastSet равно 0 и при этом в свойствах учетной записи не отмечен пункт Password never expires, то пользователь должен сменить пароль при следующем входе.

#### **Пользователи, у которых не требуется пароль:**

```
(&(sAMAccountType=805306368)
(userAccountControl:1.2.840.113556.1.4.803:=32))
```

### **Пользователи с ограниченным сроком действия учетной записи:**

```
(&(sAMAccountType=805306368)(|(accountExpires>=1)
(accountExpires<=9223372036854775806)))
```

### **Пользователи, у которых срок действия учетной записи не ограничен:**

```
(&(sAMAccountType=805306368)(|(accountExpires=0)
(accountExpires=9223372036854775807)))
```

Атрибут accountExpires содержит дату истечения срока действия учетной записи и тоже представляет из себя число 100нс интервалов, прошедших с 01.01.1601 (UTC). Значение **0** или **9223372036854775807** (максимально возможное 64-битное число) означает, что срок действия учетной записи никогда не истечет.

### **Пользователи, созданные за определенный период:**

```
(&(sAMAccountType=805306368)(whenCreated>=20200101000000.0Z))
```

или:

```
(&(sAMAccountType=805306368)
(whenCreated>=20200101000000.0Z<=20200201000000.0Z))
```

В первом случае мы указываем только начальный период, во втором ограничиваем и начальный и конечный. Формат даты используется следующий:

YYYY MM DD HH mm ss.s Z

2020 01 01 00 00 00.0 Z

Кстати, таким образом можно искать не только пользователей, но и любые другие объекты в AD (компьютеры, группы и т.п.).

### **Пользователи, не заходившие в систему более чем 3 месяца:**

```
(&(sAMAccountType=805306368)(lastLogon<=132161330597286610))
```

или

```
(&(sAMAccountType=805306368)(lastLogonTimeStamp<=132161330597286610))
```

Атрибуты lastLogon и lastLogonTimeStamp имеют такой же ~~квивалент~~ тип, как pwdLastSet, и вычисляются таким же образом. Но между ними есть кардинальные различия, о которых необходимо знать.

lastLogon изменяется при входе пользователя только на том контроллере домена, на котором происходила аутентификация, на другие контроллеры он не реплицируется. Поэтому для получения точной информации необходимо

произвести поиск на всех контроллерах домена, а потом сравнить полученные данные.

lastLogonTimeStamp также изменяется при входе пользователя, при этом он реплицируется на все контроллеры домена. Но сама репликация происходит с большой задержкой, порядка 9-14 дней. Поэтому полученные с его помощью данные могут быть не очень актуальны.

#### **Пользователи, никогда не заходившие в систему:**

```
(&(SAMAccountType=805306368)(lastLogon=0))
```

или:

```
(&(SAMAccountType=805306368)(!lastLogonTimeStamp=*))
```

#### **Пользователи с почтовыми ящиками:**

```
(&(SAMAccountType=805306368)(|(proxyAddresses=*)(mail=*)))
```

#### **Пользователи, скрытые из адресной книги:**

```
(&(SAMAccountType=805306368)(msExchHideFromAddressLists=TRUE))
```

#### **Все компьютеры:**

---

```
(objectCategory=computer)
```

#### **Все компьютеры с определенной ОС:**

```
(&(objectCategory=computer)(operatingSystem=Windows 7*))
```

Вместо Windows 7 можно поставить любую требуемую ОС.

#### **Все серверы (компьютеры с серверной ОС):**

```
(&(objectCategory=computer)(operatingSystem=*server*))
```

#### **Все контроллеры домена:**

```
(&(objectCategory=computer)  
(userAccountControl:1.2.840.113556.1.4.803:=8192))
```

или:

```
(&(objectCategory=computer)(primaryGroupID=516))
```

#### **Контроллеры домена, доступные только для чтения:**

```
(&(objectCategory=computer)(userAccountControl:1.2.840.113556.1.4.803:=67))
```

#### **Все серверы, не являющиеся контроллерами домена:**

```
(&(objectCategory=computer)(operatingSystem=*server*)  
(!userAccountControl:1.2.840.113556.1.4.803:=8192))
```

### **SQL серверы:**

```
(&(objectCategory=computer)(servicePrincipalName=MSSQLSvc*))
```

### **Exchange серверы:**

```
(&(objectCategory=computer)(servicePrincipalName=exchangeMDB*))
```

или:

```
(&(objectCategory=computer)(objectCategory=msExchExchangeServer))
```

### **Все группы:**

---

```
(objectCategory=group)
```

### **Все локальные (Domain local) группы:**

```
(&(objectCategory=group)(groupType:1.2.840.113556.1.4.803:=4))
```

### **Все глобальные (Global) группы:**

```
(&(objectCategory=group)(groupType:1.2.840.113556.1.4.803:=2))
```

### **Все универсальные (Universal) группы:**

```
(&(objectCategory=group)(groupType:1.2.840.113556.1.4.803:=8))
```

### **Все группы безопасности (Security):**

```
(&(objectCategory=group)(groupType:1.2.840.113556.1.4.803:=2147483648))
```

### **Все группы рассылки (Distribution):**

```
(&(objectCategory=group)(!groupType:1.2.840.113556.1.4.803:=2147483648))
```

или:

```
(&(objectClass=group)(proxyAddresses=*))
```

Во втором случае ищутся все группы, имеющие почтовый адрес. Напомню, что группы безопасности также могут иметь почтовый адрес и использоваться для рассылки.

### **Все локальные группы безопасности:**

```
(&(objectCategory=group)(groupType=-2147483644))
```

или:

`(&(objectCategory=group)(groupType=2147483652))`

Атрибуты userAccountControl и groupType принимают целочисленные 32-битные значения, т.е. лежат в диапазоне от  $-2^{31}$  (-2147483648) до  $2^{31}$  (2147483647). Значение атрибутов является результатом операции побитового ИЛИ. Например, значение groupType для локальной группы безопасности определяется путем применения операции ИЛИ к маске локальной группы (4) и группы безопасности (2147483648). В результате получается число **2147483652**. Поскольку данное значение превышает максимально возможное для 32-битного числа, то оно конвертируется в отрицательное путем вычитания из него  $2^{32}$  (4294967296). Получается  $2147483652 - 4294967296 = -2\ 147\ 483\ 644$ .

Теоретически правильно использовать отрицательное значение, на практике работают оба.

**Все глобальные группы безопасности:**

`(&(objectCategory=group)(groupType=-2147483646))`

**Все универсальные группы безопасности:**

`(&(objectCategory=group)(groupType=-2147483640))`

**Все встроенные группы (BuiltIn):**

`(&(objectCategory=group)(groupType:1.2.840.113556.1.4.803:=1))`

или:

`(&(objectCategory=group)(groupType=-2147483643))`

**Все глобальные группы рассылки:**

`(&(objectCategory=group)(groupType=2))`

У группы безопасности восьмой бит должен быть установлен в 1, соответственно значение маски равно 2147483648. У группы рассылки этот бит установлен в 0, соответственно и значение маски равно 0. Отсюда получаем значение 0 (Distribution) + 2 (Global) = 2 (Global Distribution Group).

**Все локальные группы рассылки:**

`(&(objectCategory=group)(groupType=4))`

**Все универсальные группы рассылки:**

`(&(objectCategory=group)(groupType=8))`

**Все члены группы (без учета вложенности):**

`(memberOf=CN=BackOffice,CN=Users,DC=test,DC=local)`

### **Все члены группы с учетом вложенности:**

```
(memberOf:1.2.840.113556.1.4.1941:=CN=BackOffice,CN=Users,DC=test,DC=local)
```

### **Все пользователи, не являющиеся членами группы (с учетом вложенности):**

```
(&(objectCategory=person)(objectClass=user)  
(!memberOf:1.2.840.113556.1.4.1941:=CN=BackOffice,CN=Users,DC=test,DC=local  
)
```

### **Все группы, в которые входит пользователь:**

```
(&(objectCategory=group)(member=CN=Vasya,OU=Employees,DC=test,DC=local))
```

### **Все группы, в которые входит пользователь (с учетом вложенности):**

```
(&(objectCategory=group)  
(member:1.2.840.113556.1.4.1941:=CN=Vasya,OU=Employees,DC=test,DC=local))
```

### **Все пустые группы:**

```
(&(objectCategory=group)(!member=*))
```

### **Все подразделения (OU):**

---

```
(objectCategory=organizationalUnit)
```

### **Все контейнеры (CN):**

```
(objectCategory=container)
```

### **Все встроенные контейнеры:**

```
(objectCategory=builtinDomain)
```

### **Все объекты групповой политики:**

```
(objectCategory=groupPolicyContainer)
```

### **Все отношения доверия:**

```
(objectClass=trustedDomain)
```

### **Все связи между сайтами в контейнере конфигурации:**

```
(objectClass=siteLink)
```

Для запросов к атрибутам конфигурации нужно использовать поиск по контейнеру Configuration (напр. cn=Configuration,dc=test,dc=local).

### **Объекты, защищенные AdminSDHolder:**

```
(adminCount=1)
```

AdminSDHolder — это механизм защиты административных учетных записей. Если пользователь является членом защищенной группы (Domain Admins, Enterprise Admins и т.п.) то его учетной записи назначаются разрешения, установленные в объекте AdminSDHolder, а атрибуту adminCount пользователя присваивается значение 1.

Обратите внимание, что при удалении пользователя из защищенной группы его атрибут adminCount не возвращается к прежнему значению, соответственно запрос выдаст всех, кто когда либо входил в одну из этих групп.

#### **Объекты, которые не могут быть удалены:**

```
(systemFlags:1.2.840.113556.1.4.803:=2147483648)
```

Атрибут systemFlags определяет дополнительные свойства объекта и представляет из себя битовую маску.

#### **Объекты, которые не могут быть перенесены:**

```
(systemFlags:1.2.840.113556.1.4.803:=67108864)
```

#### **Объекты, которые не могут быть переименованы:**

```
(systemFlags:1.2.840.113556.1.4.803:=134217728)
```

#### **Атрибуты, помеченные в схеме как конфиденциальные:**

```
(searchFlags:1.2.840.113556.1.4.803:=128)
```

Атрибут searchFlags определяет правила поиска и индексации для атрибута и представляет из себя битовую маску.

#### **Атрибуты, сохраняемые в объекте захоронении (tombstone) при удалении объекта:**

```
(searchFlags:1.2.840.113556.1.4.803:=8)
```

#### **Объекты nTDSDSA связанные с глобальным каталогом:**

```
(&(objectCategory=nTDSDSA)(options:1.2.840.113556.1.4.803:=1))
```

С помощью этого запроса можно найти серверы глобального каталога. Поиск нужно проводить по контейнеру конфигурации. Подробнее об объектах nTDSDSA и их атрибутах.

#### **Объекты nTDSDSA связанные с ролями FSMO:**

```
(fSMORoleOwner=*)
```

Для ролей PDC Emulator, RID Master и Infrastructure Master нужно опрашивать домен. Владельца роли Schema Master нужно искать в контейнере Schema (напр. cn=Schema,cn=Configuration,dc=test,dc=local), а владельца Domain Naming Master — в контейнере Configuration (напр. cn=Configuration,dc=test,dc=local).

Можно искать каждую роль по отдельности.

#### **PDC Emulator:**

```
&(objectClass=domainDNS)(fSMORoleOwner=*)
```

#### **RID Master:**

```
(&(objectClass=rIDManager)(fSMORoleOwner=*))
```

#### **Infrastructure Master:**

```
(&(objectClass=infrastructureUpdate)(fSMORoleOwner=*))
```

#### **Schema Master:**

```
(&(objectClass=dMD)(fSMORoleOwner=*))
```

#### **Domain Naming Master:**

```
(&(objectClass=crossRefContainer)(fSMORoleOwner=*))
```