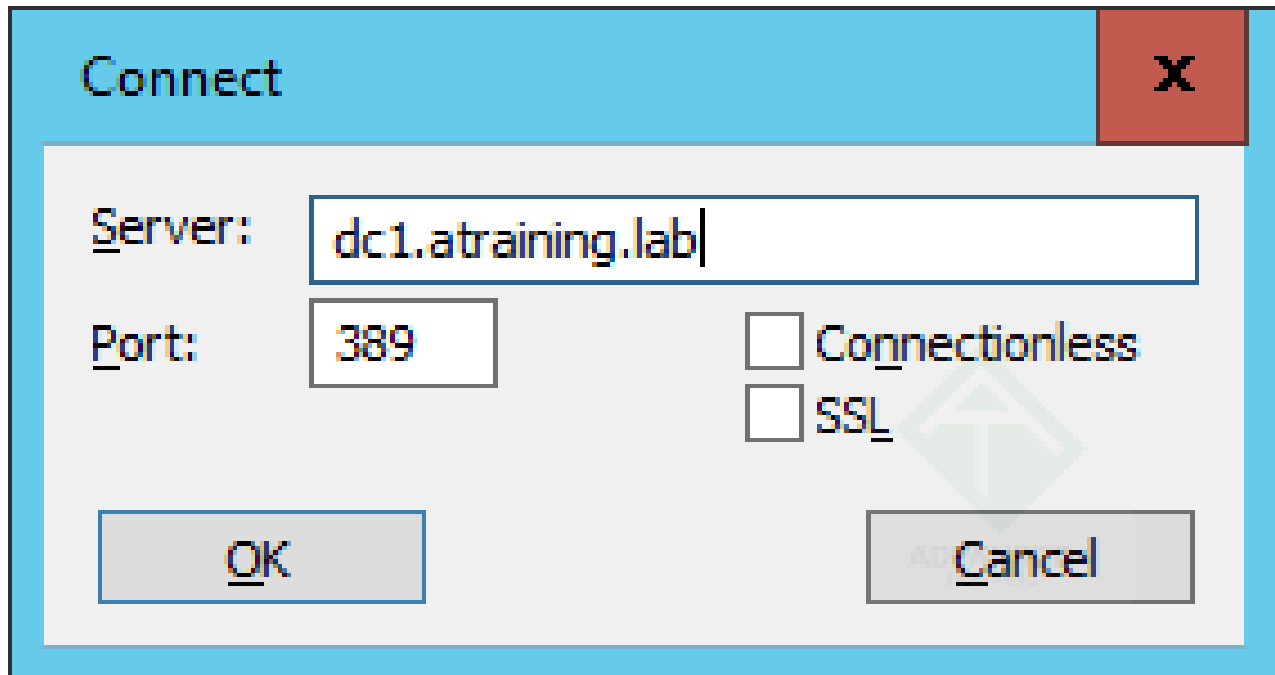


Вокруг FSMO-ролей - много мифов и легенд. Разбираемся с каждой детально. Эта статья - про RID Master.

 attraining.ru/active-directory-fsmo-rid-master

2015-08-25T13:00:51+08:00



Привет.

Этой статьёй мы стартуем мини-цикл рассказов про FSMO-роли в Active Directory. Вокруг данных ролей, выполняемых ими задач, вопросов надёжности и отказоустойчивости, нужности и не очень в разных ситуациях, написано множество всего, а также за 15 с лишним лет существования Active Directory накоплено много мифов, верований, ритуалов и прочего совершенно не нужного в данный момент. Многие задачи и функции претерпели изменения, однако устаревшие советы, актуальные для Windows Server 2000 / 2003, до сих пор активно используются, что приводит к неэффективной, а зачастую и небезопасной работе инфраструктуры.

Попробуем разобраться с каждым из FSMO отдельно. Стараясь не сильно углубляться в сторонние темы (хотя возможностей будет масса), и предполагая, что Вы знаете материал хотя бы на уровне сертифицированного курса [Microsoft 20410](#), читаемого в обзорно-упрощённом формате в авторизованных учебных центрах Microsoft – увы, детали работы RID Master там не изучаются, кроме краткого описания его работы. Если же вы проходили этот курс у нас, то часть статьи вы уже, по сути, изучили.

RID Master

- Базовые задачи RID Master

- Пространство RID'ов и пулы
 - Проверяем, расширено ли пространство RID'ов
 - Включаем расширенное пространство RID'ов в домене Active Directory
- Глобальные настройки RID Master
 - msDS-RIDPoolAllocationEnabled или RID Ceiling
 - Как отключить RID Ceiling
 - Атрибут rIDAvailablePool
 - Атрибут NextRid
- Настройки RID-пулов у DC
 - Атрибут rIDPreviousAllocationPool
 - Атрибут rIDAllocationPool
 - Атрибут rIDNextRID
 - Атрибут rIDUsedPool
 - Изменяем размер RID pool
- Работа RID Master при переносе объектов между доменами
- Как перенести держателя FSMO-роли RID Master?
- Где хранится информация, кто сейчас RID Master?
- Где располагать держателя FSMO-роли RID Master?
- Нужен ли RID Master'у отдельный бэкап?
- Как повысить надёжность работы RID Master?
- Если RID Master упал то всё
- Как часто надо переносить FSMO-роль RID Master?

Начнём.

Базовые задачи RID Master

В Active Directory объекты уникально идентифицируются атрибутом Object-Guid (при просмотре объектов штатными средствами будет называться objectGUID) – он будет у всех объектов и именно он является ключевым идентификатором в случаях переименования (т.е. смены RDN), переноса (т.е. смены DN), и подобных. Однако, хоть GUID визуально и представлен как структуризированный, по сути это строка из 128ми бит, никак не привязанная к иным задачам, кроме “быть максимально уникальной”.

Для идентификации объектов, которые могут проводить операции над другими (будем называть их security principal), используется дополнительный идентификатор, присущий только им – Security ID или SID. Этот идентификатор уже будет обладать определённой структурой, упрощающей его обработку и анализ на корректность, и, в частности, один из его компонентов – визуально это “хвост”, будет называться Relative ID, или RID. Например, в таком вот SID:

S-1-5-32-544

RID'ом будет 544.

Relative ID будет так называться, потому что однотипные security principal'ы в пределах одного единообразно управляемого пространства безопасности (security domain) будут отличаться именно RID'ами – остальные части SID'ов у них будут совпадать (в пределах однотипных security principal'ов). То есть SID логически делится на последний сегмент – RID объекта – и остальное – Security Domain ID.

Раз так, то кто-то должен отвечать за назначение RID'ов. В случае, когда security domain равен одному экземпляру ОС, дело это несложное – RID'ы назначаются с тысячи и далее, прибавляя по единице. Эта простейшая ситуация возможна по той причине, что одновременно нельзя запросить создание нескольких security principal'ов – даже если Вы сделаете многопоточное приложение и синхронизируете подачу запросов, то они уйдут в очередь и выполнятся все равно по порядку. Поэтому всё просто – создали security principal с RID = 1071, значит после создаём с RID = 1072. Линейно. Так и происходит на локальной системе.

В случае же, когда security domain – это домен NT, ситуация усложняется. Потенциальных точек создания новых security principal'ов, начиная с Windows 2000, в домене столько же, сколько DC за вычетом RODC. И выстроить все операции по созданию новых security principal'ов – которых ещё и больше стало к тому же разнообразных, чем в случае одного NT-хоста – в одну цепочку – нереально. Сотни и тысячи географически разнесённых в масштабах планеты DC, связанных неочевидной схемой каналов, никак не смогут работать эффективно, если всё создавать будет только кто-то один. Поэтому используется схема с индивидуальным назначением каждому DC диапазонов RID – или RID-пулов. Это будут диапазоны RID'ов, которые может раздавать каждый, кто их получит. Тогда никакого пересечения нет – ты раздаёшь пачку “с 1100 по 1599й”, я – “с 1600 по 2099й”. В любом темпе, как удобно – никак не пересекаясь друг с другом. Когда от пачки останется определённое количество – запрашивается новая.

Идея достаточно проста – разве что всем нужно договориться о том, кто будет раздавать эти пулы, как, какого размера будут эти пулы, как они будут запрашиваться и выделяться, как будет обеспечиваться надёжность и непрерывность всей этой схемы.

Если с первым делом из списка понятно – это будет держатель роли FSMO RID Master, один DC из домена, то про остальное пункты поговорим подробнее.

Пространство RID'ов и пулы

Глобальное теоретически доступное пространство RID-ов разделено достаточно просто. В случае домена Windows NT:

- От 0 до 499 – системные RID'ы (это не security principal'ы, это спец.идентификаторы – допустим, типов сессии (INTERACTIVE, NETWORK, подобные))

- От 500 до 1000 – встроенные security principal'ы (и группы и отдельные учётные записи – например, BUILTIN\Administrators или guest))
- Далее и до 1.073.741.822 – RID'ы под произвольные задачи создания новых security principal'ов

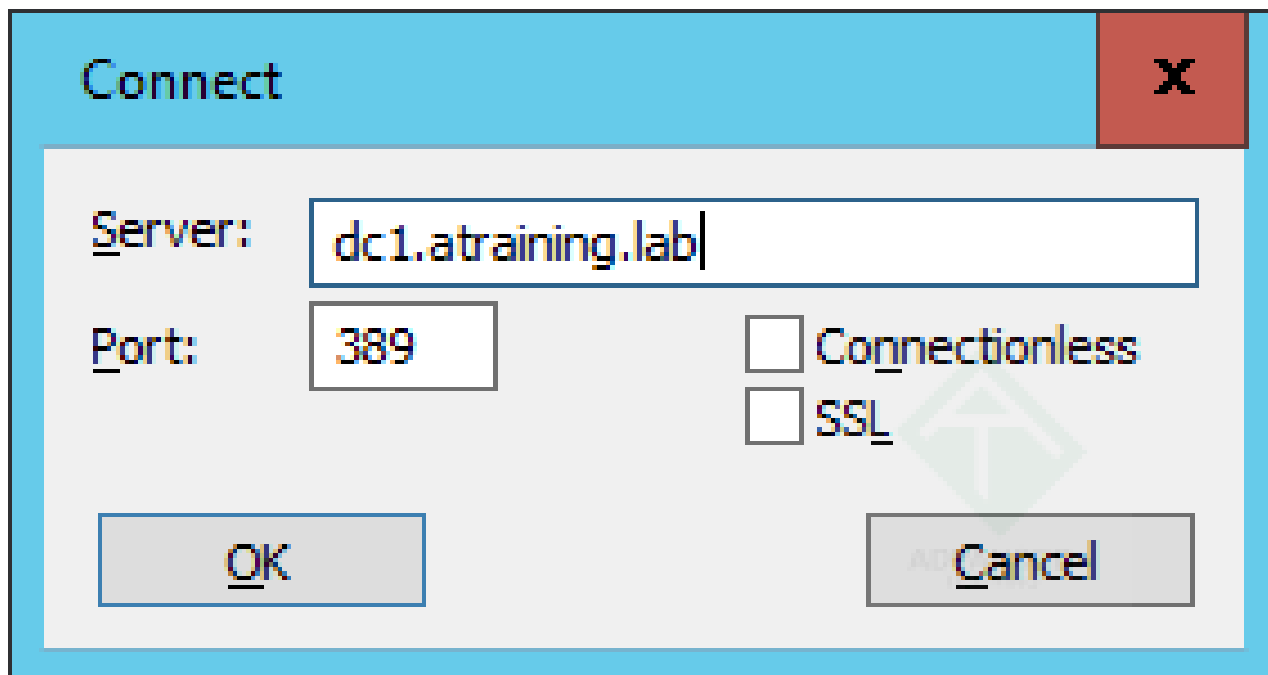
Последний пункт исторически обусловлен тем, что под RID'ы выделяется стандартный 32х битовый dword, а два верхних бита зарезервированны. Начиная с Windows Server 2012 логику этой обработки можно поменять и выделить не 30, а 31 бит. Обращу внимание, что это – экстремальная мера, её не надо делать “впрок” и “потому что круто”. Это у олигофренов “круто” и “новый левел в жизни начался”, когда вместо мобильного с экраном 4” удаётся заполучить мобильник с экраном 4.7”, а в Active Directory такой подход не работает. Реально подойти к необходимости “разлочить” второй миллиард с лишним RID'ов – практически невозможно, но на всякий случай посмотрим, как это сделать.

Первым делом – контроллеры домена на базе NT 6.0 и ниже работать с увеличенным RID-пространством не будут совсем. Они будут воспринимать его как закончившееся. Контроллеры на NT 6.1 смогут работать в домене, где будет включено увеличение RID-пространства, но только если применить к ним патч [KB 2642658](#). Учтите, нужно именно исключить ситуацию появления в домене контроллеров, не знающих этой возможности, поэтому до начала работ убедитесь, что все DC на базе 2008 R2, включая бэкапы, снапшоты виртуалок, заготовленные образы, имеют данный патч. Патч не какой-то волшебный или секретный, и, если контроллеры обладают всеми обновлениями на середину 2012 года, то он уже применён.

Проверяем и устанавливаем диапазон RID'ов в домене

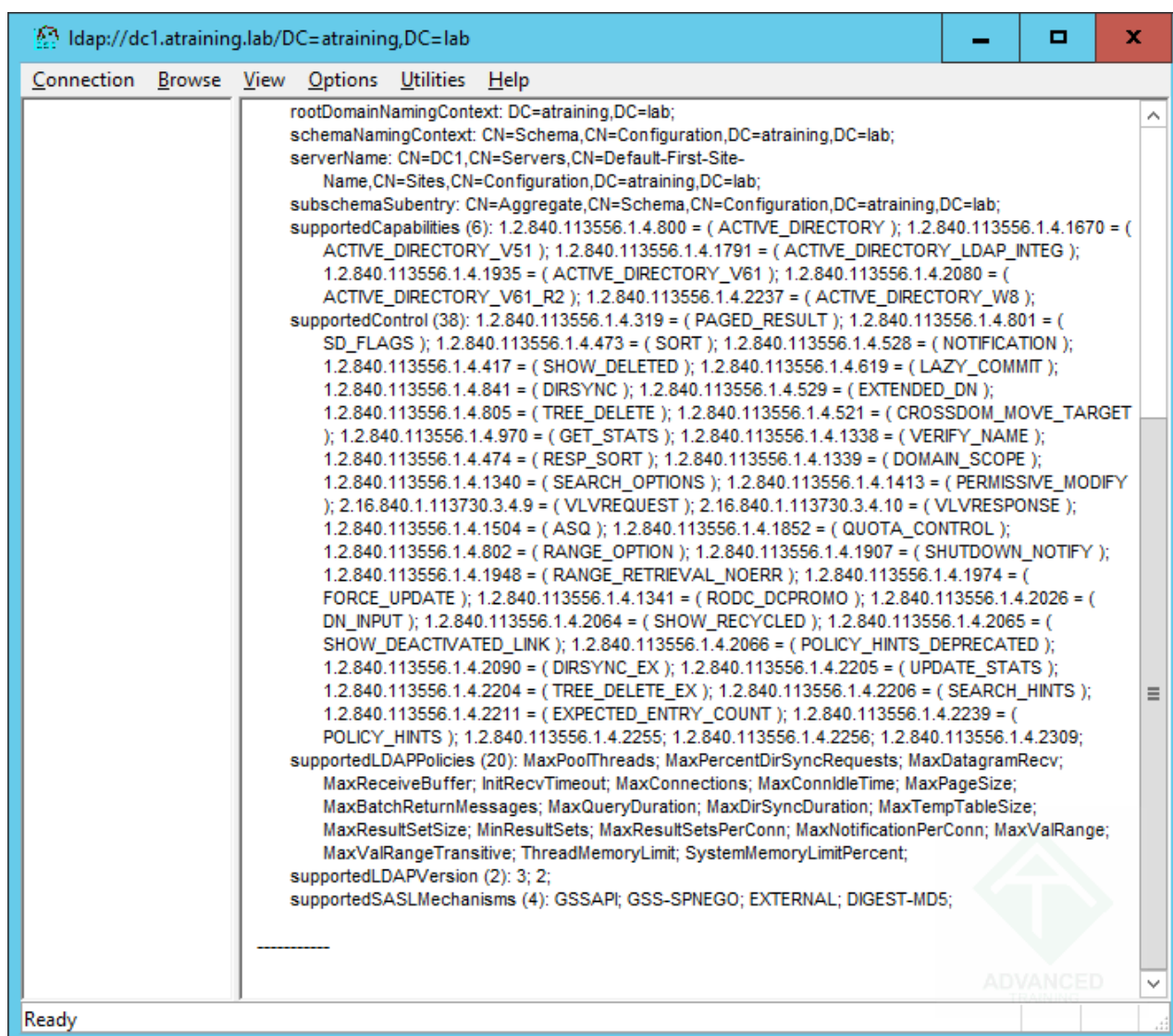
Теперь тест на то, включен ли он в домене (данный функционал – это функционал уровня домена, а не леса – каждый домен в лесу включает режим увеличения RID-пространства индивидуально). По сути, это будет видно из значения, находящееся в атрибуте **SidCompatibilityVersion** корневого объекта **RootDSE**.

Первым делом, проверяем, жив ли и доступен наш экспериментальный DC на базе Windows Server 2016 TP3. Мы будем использовать для этого ldp.exe:



[Подключаемся к Active Directory, используя ldp.exe](#)
[\(кликните для увеличения до 283 px на 150 px\)](#)

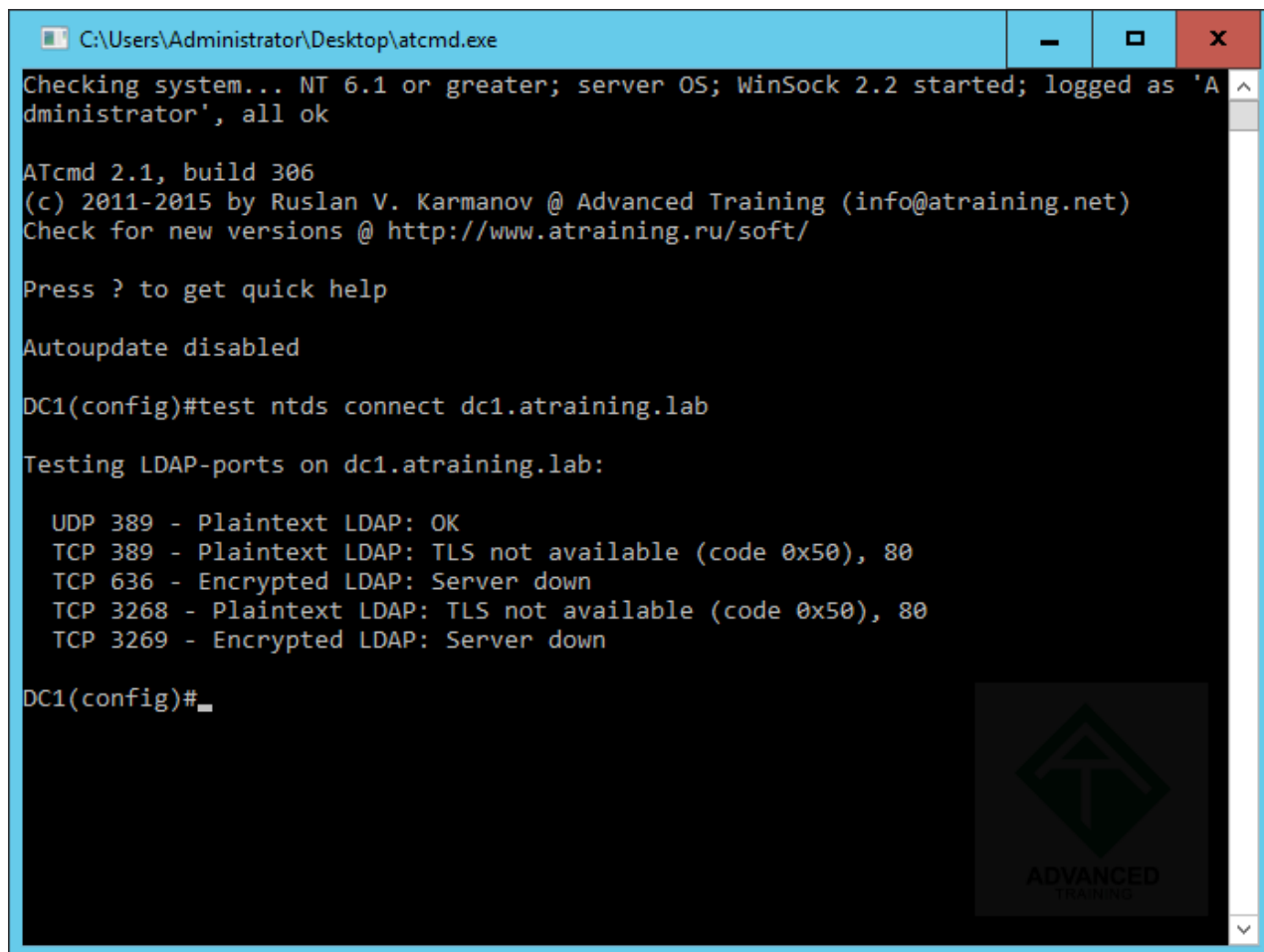
Подключение прошло успешно:



[Подключились к Active Directory, используя ldap.exe](#)

[\(кликните для увеличения до 712 px на 625 px\)](#)

Но мы использовали обычный LDAPv3, без защиты сессии и без подписи содержимого. В ряде случаев при подключении по такому, самому простому варианту, контроллер домена будет отдавать нам не всю информацию про некоторые атрибуты, поэтому нам надо перезацепиться по безопасному LDAPv3. Проверим, поддерживается ли он:



```
C:\Users\Administrator\Desktop\atcmd.exe
Checking system... NT 6.1 or greater; server OS; WinSock 2.2 started; logged as 'Administrator', all ok

ATcmd 2.1, build 306
(c) 2011-2015 by Ruslan V. Karmanov @ Advanced Training (info@atraining.net)
Check for new versions @ http://www.atraining.ru/soft/

Press ? to get quick help

Autoupdate disabled

DC1(config)#test ntds connect dc1.atraining.lab

Testing LDAP-ports on dc1.atraining.lab:

  UDP 389 - Plaintext LDAP: OK
  TCP 389 - Plaintext LDAP: TLS not available (code 0x50), 80
  TCP 636 - Encrypted LDAP: Server down
  TCP 3268 - Plaintext LDAP: TLS not available (code 0x50), 80
  TCP 3269 - Encrypted LDAP: Server down

DC1(config)#_
```

[Проверяем поддержку LDAPS и STARTTLS в LDAPv3 на контроллере домена Active Directory, используя atcmd.exe](#)

[\(кликните для увеличения до 694 px на 521 px\)](#)

Нет, т.к. это новорожденный domain controller, он без сертификата – поставим сертификат (самоподписанный, нам сейчас это не критично – но вообще, в боевом применении, надо, конечно, сделать специальный шаблон сертификата DC и автоматически, через Group Policy, его раздать) и повторим попытку.

```
C:\Users\Administrator\Desktop\atcmd.exe

ATcmd 2.1, build 306
(c) 2011-2015 by Ruslan V. Karmanov @ Advanced Training (info@atraining.net)
Check for new versions @ http://www.atraining.ru/soft/

Press ? to get quick help

Autoupdate disabled

DC1(config)#test ntds connect dc1.atraining.lab

Testing LDAP-ports on dc1.atraining.lab:

  UDP 389 - Plaintext LDAP: OK
  TCP 389 - Plaintext LDAP: TLS not available (code 0x50), 80
  TCP 636 - Encrypted LDAP: Server down
  TCP 3268 - Plaintext LDAP: TLS not available (code 0x50), 80
  TCP 3269 - Encrypted LDAP: Server down

DC1(config)#test ntds connect dc1.atraining.lab

Testing LDAP-ports on dc1.atraining.lab:

  UDP 389 - Plaintext LDAP: OK
  TCP 389 - Plaintext LDAP: TLS available, OK
  TCP 636 - Encrypted LDAP: OK
  TCP 3268 - Plaintext LDAP: TLS available, OK
  TCP 3269 - Encrypted LDAP: OK

DC1(config)#
```

[Проверяем включение поддержки LDAPS и STARTTLS в LDAPv3 на контроллере домена Active Directory, используя atcmd.exe \(кликните для увеличения до 694 px на 521 px\)](#)

Видим, что контроллер доступен во всех вариантах – и UDP, и по обычному LDAP (по портам LDAP 389 и GC 3268, притом доступна возможность отправить STARTTLS и включить TLS после установки соединения), и по LDAP поверх полноценного TLS (на портах LDAPS 636 и GCS 3269). Это хорошо. Если нужны детали по LDAP и по TLS-соединению, их тоже можно посмотреть:

```
C:\Users\Administrator\Desktop\atcmd.exe
TCP 636 - Encrypted LDAP: Server down
TCP 3268 - Plaintext LDAP: TLS not available (code 0x50), 80
TCP 3269 - Encrypted LDAP: Server down

DC1(config)#test ntds connect dc1.atraining.lab

Testing LDAP-ports on dc1.atraining.lab:

UDP 389 - Plaintext LDAP: OK
TCP 389 - Plaintext LDAP: TLS available, OK
TCP 636 - Encrypted LDAP: OK
TCP 3268 - Plaintext LDAP: TLS available, OK
TCP 3269 - Encrypted LDAP: OK

DC1(config)#test ntds serverinfo dc1.atraining.lab 636

Getting LDAP-specific info about dc1.atraining.lab , using 636 port

Server dc1.atraining.lab info
LDAP vendor 'Microsoft Corporation.', version 5.10
Top supported LDAP version 3, revision 2004, features version 1.1
LDAP referrals hop limit is 32
LDAP referrals: Enabled

TLS info
Protocol: TLS 1.2 client
Cipher: AES-256 with 256 bits
Hash: SHA-384
Key exchange: 0xae06 with 256 bits
DC1(config)#
```

[Смотрим криптографические настройки LDAPS на контроллере домена Active Directory, используя atcmd.exe](#)
(кликните для увеличения до 694 px на 521 px)

Всё в принципе очень неплохо, и TLS 1.2 по факту включен, и хэш по умолчанию очень даже взрослый, SHA-2/384, и даже обмен ключами настолько новомодный, что идентификатор ему не нашёлся (0x0000ae06, судя по первым 3+4 битам, явно обмен ключами, но какой-то ранее, до Windows Server 2016 TP3, не сильно используемый – однако, как видно, успешно согласованный между двумя Windows Server 2016 TP3). Даже дефолтные параметры безопасности, в общем, серьезнее чем у всех остальных LDAP-серверов из имеющихся – дополнительный плюс для новой Cloud OS.

Теперь проверяем, расширено ли пространство RID'ов в нашем домене – эта операция ещё называется “Global RID Space Size Unlock”.

Проверяем, расширено ли пространство RID'ов

Открываем ldr.exe уже по защищённому подключению (в данном случае это необязательно, но я всегда буду делать так для единообразия и безопасности работы, поэтому в последующих примерах это уже не будет детализироваться):

Connect

X

Server:

dc1.atraining.lab

Port:

636

☐ Connectionless

☒ SSL

OK

Cancel

[Безопасно подключаемся по ldaps.exe](#)

[\(кликните для увеличения до 283 px на 150 px\)](#)

И видим, что атрибута **SidCompatibilityVersion** у **RootDSE** не наблюдается:

ldaps://dc1.atraining.lab/DC=atraining,DC=lab

Connection

Browse

View

Options

Utilities

Help

DC=ForestDnsZones,DC=atraining,DC=lab;
 rootDomainNamingContext: DC=atraining,DC=lab;
 schemaNamingContext: CN=Schema,CN=Configuration,DC=atraining,DC=lab;
 serverName: CN=DC1,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=atraining,DC=lab;
 subschemaSubentry: CN=Aggregate,CN=Schema,CN=Configuration,DC=atraining,DC=lab;
 supportedCapabilities (6): 1.2.840.113556.1.4.800 = (ACTIVE_DIRECTORY); 1.2.840.113556.1.4.1670 = (ACTIVE_DIRECTORY_V51); 1.2.840.113556.1.4.1791 = (ACTIVE_DIRECTORY_LDAP_INTEG); 1.2.840.113556.1.4.1935 = (ACTIVE_DIRECTORY_V61); 1.2.840.113556.1.4.2080 = (ACTIVE_DIRECTORY_V61_R2); 1.2.840.113556.1.4.2237 = (ACTIVE_DIRECTORY_W8);
 supportedControl (38): 1.2.840.113556.1.4.319 = (PAGED_RESULT); 1.2.840.113556.1.4.801 = (SD_FLAGS); 1.2.840.113556.1.4.473 = (SORT); 1.2.840.113556.1.4.528 = (NOTIFICATION); 1.2.840.113556.1.4.417 = (SHOW_DELETED); 1.2.840.113556.1.4.619 = (LAZY_COMMIT); 1.2.840.113556.1.4.841 = (DIRSYNC); 1.2.840.113556.1.4.529 = (EXTENDED_DN); 1.2.840.113556.1.4.805 = (TREE_DELETE); 1.2.840.113556.1.4.521 = (CROSSDOM_MOVE_TARGET); 1.2.840.113556.1.4.970 = (GET_STATS); 1.2.840.113556.1.4.1338 = (VERIFY_NAME); 1.2.840.113556.1.4.474 = (RESP_SORT); 1.2.840.113556.1.4.1339 = (DOMAIN_SCOPE); 1.2.840.113556.1.4.1340 = (SEARCH_OPTIONS); 1.2.840.113556.1.4.1413 = (PERMISSIVE_MODIFY); 2.16.840.1.113730.3.4.9 = (VLVREQUEST); 2.16.840.1.113730.3.4.10 = (VLVRESPONSE); 1.2.840.113556.1.4.1504 = (ASQ); 1.2.840.113556.1.4.1852 = (QUOTA_CONTROL); 1.2.840.113556.1.4.802 = (RANGE_OPTION); 1.2.840.113556.1.4.1907 = (SHUTDOWN_NOTIFY); 1.2.840.113556.1.4.1948 = (RANGE_RETRIEVAL_NOERR); 1.2.840.113556.1.4.1974 = (FORCE_UPDATE); 1.2.840.113556.1.4.1341 = (RODC_DCPROMO); 1.2.840.113556.1.4.2026 = (DN_INPUT); 1.2.840.113556.1.4.2064 = (SHOW_RECYCLED); 1.2.840.113556.1.4.2065 = (SHOW_DEACTIVATED_LINK); 1.2.840.113556.1.4.2066 = (POLICY_HINTS_DEPRECATED); 1.2.840.113556.1.4.2090 = (DIRSYNC_EX); 1.2.840.113556.1.4.2205 = (UPDATE_STATS); 1.2.840.113556.1.4.2204 = (TREE_DELETE_EX); 1.2.840.113556.1.4.2206 = (SEARCH_HINTS); 1.2.840.113556.1.4.2211 = (EXPECTED_ENTRY_COUNT); 1.2.840.113556.1.4.2239 = (POLICY_HINTS); 1.2.840.113556.1.4.2255; 1.2.840.113556.1.4.2256; 1.2.840.113556.1.4.2309;
 supportedLDAPPolicies (20): MaxPoolThreads; MaxPercentDirSyncRequests; MaxDatagramRecv; MaxReceiveBuffer; InitRecvTimeout; MaxConnections; MaxConnIdleTime; MaxPageSize; MaxBatchReturnMessages; MaxQueryDuration; MaxDirSyncDuration; MaxTempTableSize; MaxResultSetSize; MinResultSets; MaxResultSetsPerConn; MaxNotificationPerConn; MaxValRange; MaxValRangeTransitive; ThreadMemoryLimit; SystemMemoryLimitPercent;
 supportedLDAPVersion (2): 3; 2;
 supportedSASLMechanisms (4): GSSAPI; GSS-SPNEGO; EXTERNAL; DIGEST-MD5;

Ready

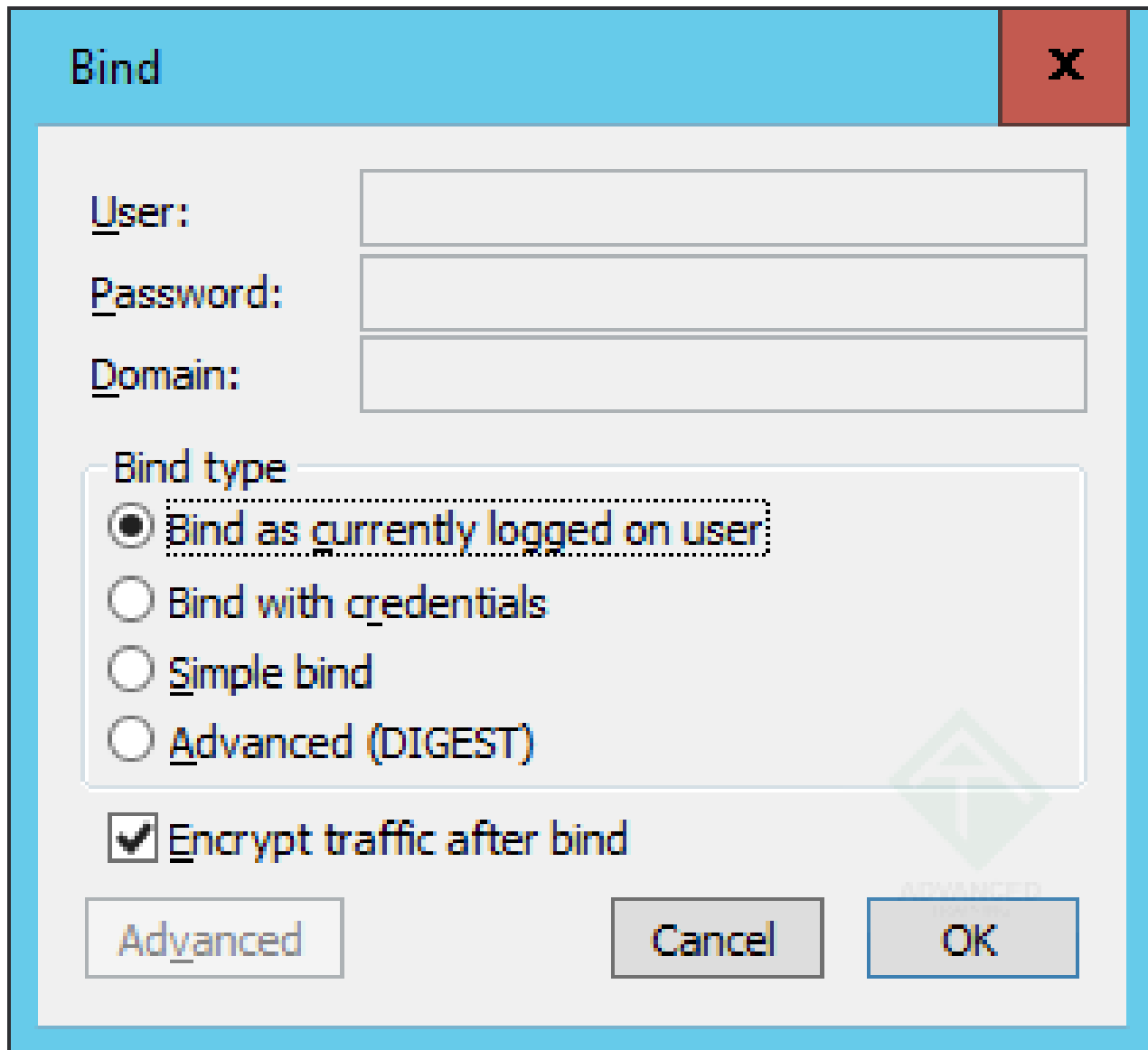
[Просмотр содержимого RootDSE](#)

[\(кликните для увеличения до 712 px на 625 px\)](#)

Т.е. этого атрибута просто нет, даже в свежеразвёрнутом домене на базе Windows Server 2016. Это как раз то, про что я писал выше – что “на всякий случай” делать это не нужно. Но если надо – включим.

Включаем расширенное пространство RID’ов в домене Active Directory

Подключаемся так же ldp.exe и не забываем авторизацию через bind:



[Авторизация LDAP-подключения к Active Directory](#)

[\(кликните для увеличения до 297 px на 271 px\)](#)

После чего заходим в Browse / Modify и говорим, что хотим создать новый атрибут, прямо в корне контекста, и задать его значение в единицу:

Modify

DN:

Edit Entry

Attribute:

Values:

Operation

☒ Add ☐ Delete ☐ Replace

Entry List

☒ Synchronous

☐ Extended

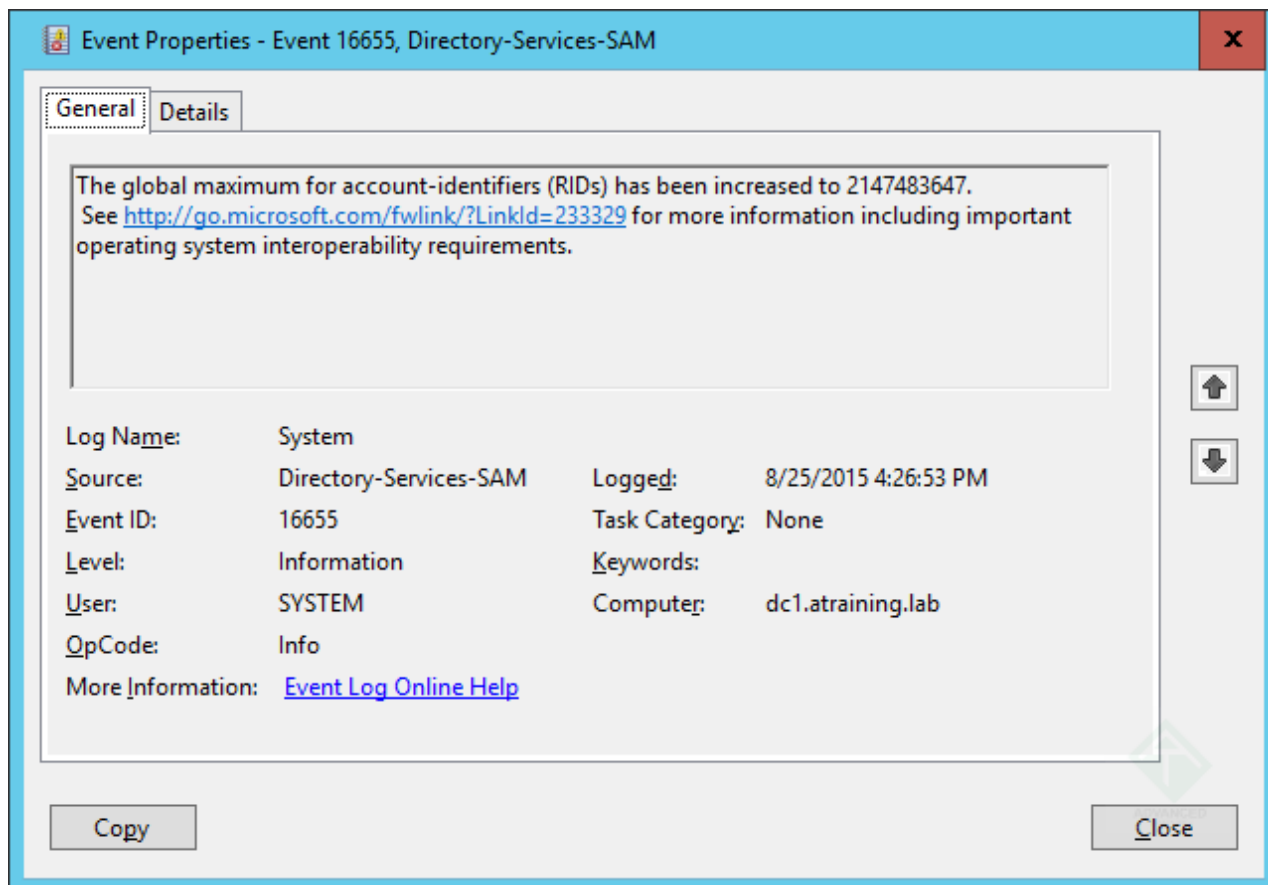
[Создаём атрибут SidCompatibilityVersion](#)

([кликните для увеличения до 358 px на 392 px](#))

Если у вас хватило прав и вы всё корректно сделали, ldp.exe отпишетс чем-то вида:

```
***Call Modify... ldap_modify_s(ld, '(null)',[1] attrs); Modified "".
```

А в журнале System появится событие 16655 о том, что расширение прошло успешно:

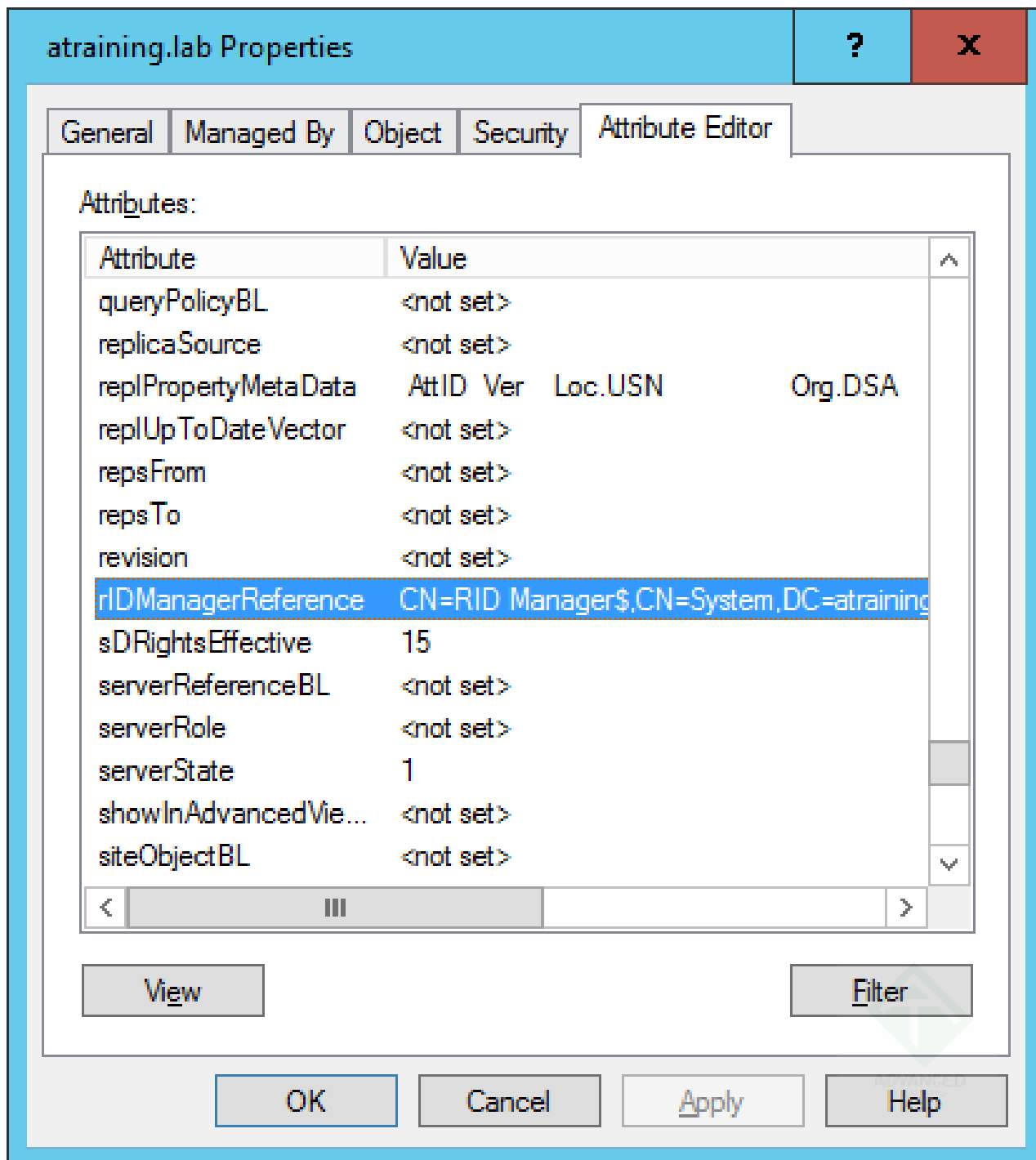


[Успешно провели Global RID Space Size Unlock в домене Active Directory.
\(кликните для увеличения до 640 px на 445 px\)](#)

ОК, с общим размером пула разобрались. Теперь посмотрим глобальные параметры RID Master на уровне домена

Глобальные настройки RID Master

Где именно у нас сейчас в домене находятся глобальные настройки RID Master, узнать просто – за это отвечает атрибут **ridManagerReference** у корневого объекта домена:



[Где в домене находятся глобальные настройки RID Master](#)

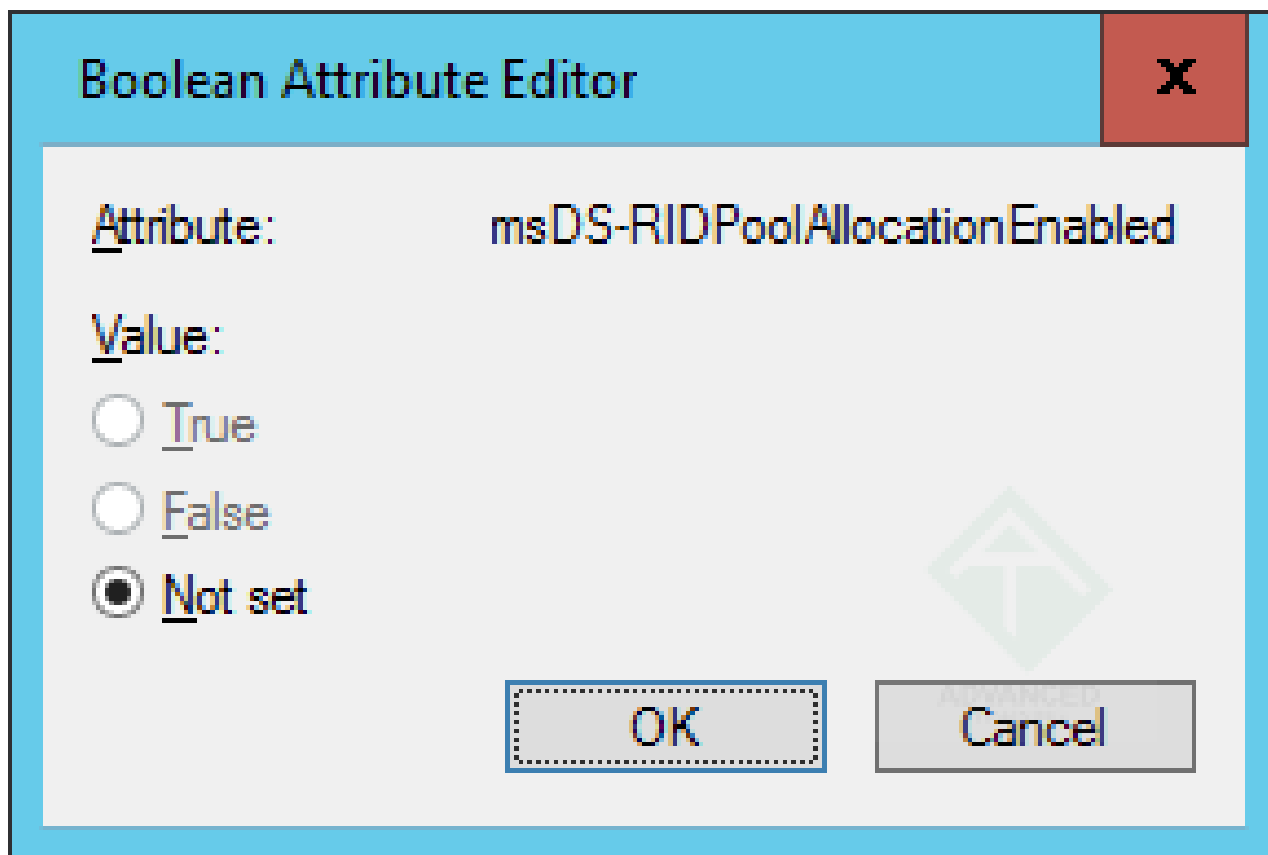
(кликните для увеличения до 414 px на 462 px)

Стандартно это объект **CN=RID Manager\$** в контейнере /System. Настроек у него много – но нам нужна только та часть, которая относится к его прямым обязанностям.

msDS-RIDPoolAllocationEnabled или RID Ceiling

Атрибут **msDS-RIDPoolAllocationEnabled** у **CN=RID Manager\$** будет указывать, включен ли предупреждающий механизм RID Ceiling. Суть проста – когда израсходовано 90% RID'ов в домене, на контроллере домена, держащего роль RID

Master, начинает генериться событие 16657 о том, что достигнут потолок и больше пачки RID'ов, пока админ не разберётся с ситуацией, выделяться не будут. Вот как выглядит этот атрибут:



[Атрибут управления RID Ceiling.y RID Master - msDS-RIDPoolAllocationEnabled](#)
(кликните для увеличения до 289 px на 193 px)

Значение FALSE указывает, что механизм включен.

Добавлю, что этот атрибут работает только в случае, если RID Master находится на DC на Windows Server 2012 и выше – если нет, то обработки данного события не происходит.

Теперь про отключение этого ограничения. Сразу же предупреждение – просто так отключать его тоже не надо. Надо, в случае срабатывания этого механизма, разобраться с тем, почему так произошло. И уже потом – найдя причину опустошения склада с RID'ами, выключить RID Ceiling.

В своё время в Windows Server 2008 R2 была проблема “утечки RIDов” – иногда DC на базе WinServer 2008 R2 начинал раз в 30 секунд запрашивать новый RID pool. Эта проблема решается патчем [KB 2618669](#) – думаю, он у вас уже установлен, но лишний раз проверить не помешает.

Как отключить RID Ceiling

Подключимся ldp.exe к контроллеру домена (любому, необязательно держателю роли RID Master – ведь объект с настройками реплицируется на все контроллеры) и выберем Modify на объекте **CN=RID Manager\$**. Нам надо модифицировать атрибут

MsDS-RIDPoolAllocationEnabled, выставив его в TRUE. Это ldap-операция Replace, поэтому итоговое окно с заготовленной операцией будет выглядеть так:

Modify

DN: CN=RID Manager\$,CN=System,DC=atraining

Edit Entry

Atttribute: MsDS-RIDPoolAllocationEnabled

Values: TRUE

Operation

☐ Add ☐ Delete ☒ Replace

Entry List

[Replace]MsDS-RIDPoolAllocationEnabled:TRUE

☒ Synchronous

☐ Extended

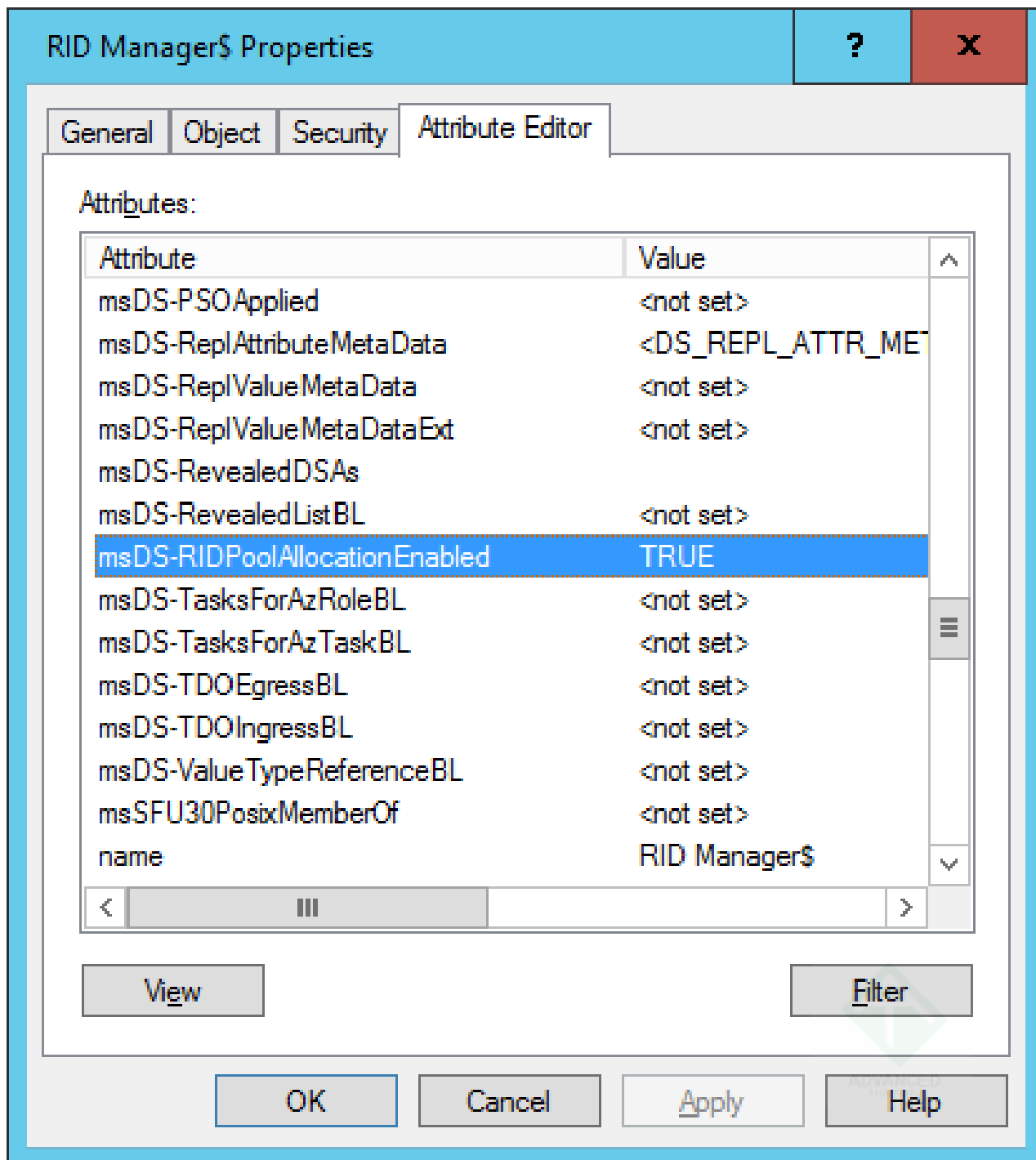
[Выключаем RID Ceiling в домене Active Directory.](#)

(кликните для увеличения до 358 px на 392 px)

В результате удачной операции ldp.exe напишет нам примерно такое:

```
----- ***Call Modify... ldap_modify_s(ld, 'CN=RID
Manager$,CN=System,DC=atraining,DC=lab',[1] attrs); Modified "CN=RID
Manager$,CN=System,DC=atraining,DC=lab". -----
```

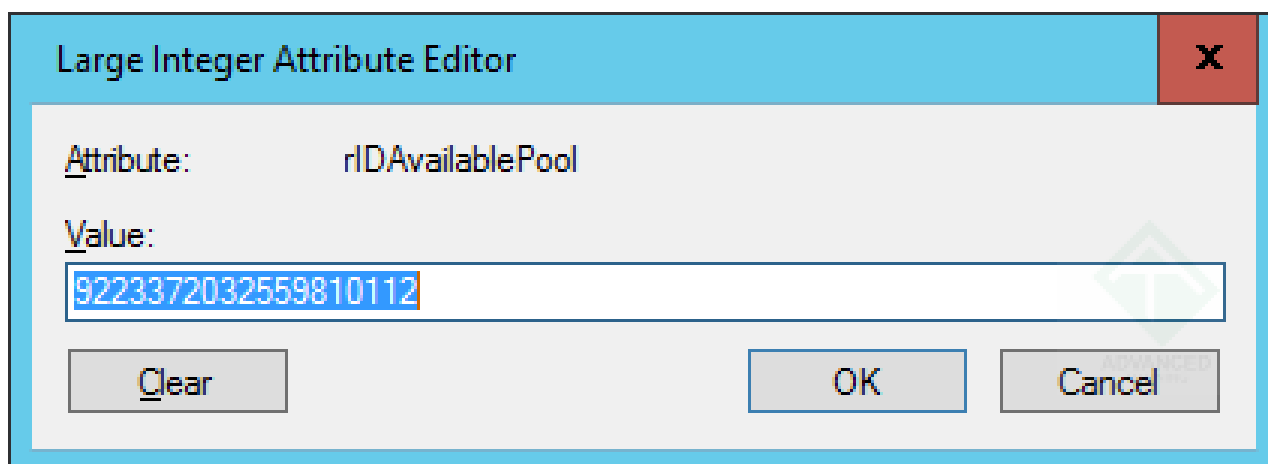
Можем проверить результат и через обычную консоль Active Directory Users and Computers:



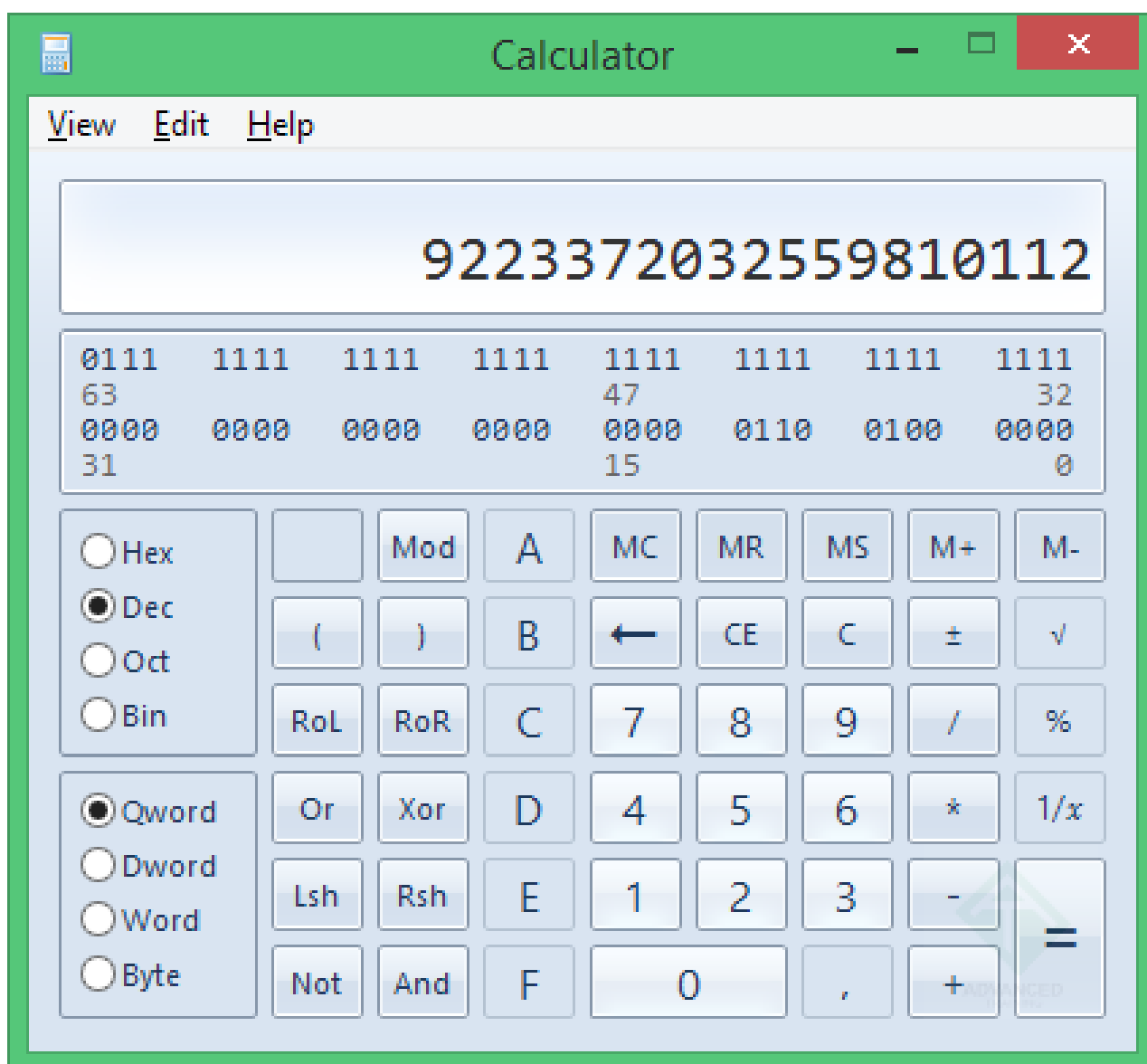
[Выключили обработку RID Ceiling в домене Active Directory.](#)
[\(кликните для увеличения до 414 px на 462 px\)](#)

Атрибут rIDAvailablePool

В этом атрибуте в виде 64х битового QWORD, составленного из двух 32х битовых DWORD, находится информация о доступном в данном домене диапазоне RID'ов – т.е. нижняя и верхняя границы. Вы можете запустить обычный калькулятор и увидеть, сколько RID'ов в вашем домене уже раздали по различным DC в виде RID-пулов. Это просто – берём значение атрибута:



[Значение атрибута rIDAvailablePool](#)
[\(кликните для увеличения до 423 px на 153 px\)](#)
 Добавляем в калькулятор:



[Обрабатываем значение атрибута rIDAvailablePool](#)
[\(кликните для увеличения до 423 px на 389 px\)](#)
 и отсекаем старший DWORD:



[Результирующий младший 32х битовый DWORD атрибута rIDAvailablePool](#)
(кликните для увеличения до 423 px на 389 px)

1600 у новорожденного домена – норма. Первый контроллер при первой загрузке выдал себе RID pool с 1100 до 1600. 1600 – это начало нового пула, который будет выдан RID Master’ом запрашивающему DC.

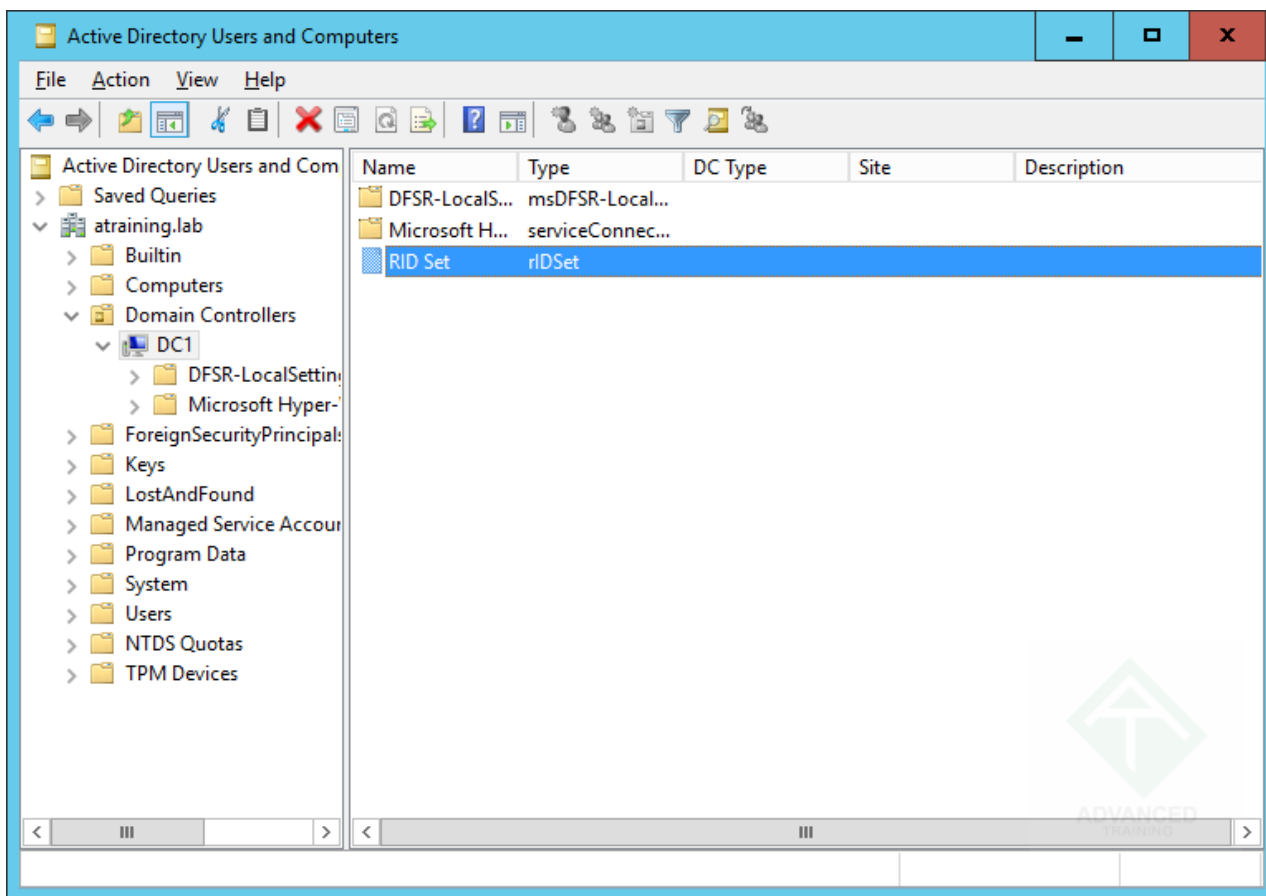
Атрибут NextRid

Данный атрибут находится в корневом объекте домена и сейчас не используется. Он был создан в стартовой версии, Windows 2000, для работы в Mixed Mode – вместе с Windows NT 4.0. Вкратце – в протоколе Directory Replication Service (DRS) Remote Protocol есть функция PerformExtendedOpRequestMsg, которая учитывает состояние этого атрибута. Сейчас его значение должно быть 0.

Теперь перейдём к настройкам конкретных DC.

Настройки RID-пулов у DC

У каждого контроллера домена будет свой объект – **RID Set**. Чтобы увидеть его, включим в оснастке Active Directory Users and Computers контейнерное отображение и зайдём в учётку контроллера домена:



[Объект RID Set у контроллера домена](#)
([кликните для увеличения до 768 px на 537 px](#))

Атрибутов у него будет чуть побольше:

RID Set Properties
?
X

General
Object
Security
Attribute Editor

Attributes:

Attribute	Value
replUpToDateVector	<not set>
repsFrom	<not set>
repsTo	<not set>
revision	<not set>
rIDAllocationPool	6867652707404
rIDNextRID	1102
rIDPreviousAllocationPool	6867652707404
rIDUsedPool	0
sDRightsEffective	15
serverReferenceBL	<not set>
showInAdvancedViewOnly	TRUE
siteObjectBL	<not set>
structuralObjectClass	top; rIDSet
subRefs	<not set>

<
|||
>

Edit
Filter

OK
Cancel
Apply
Help

[Атрибуты объекта RID Set у контроллера домена](#)

(кликните для увеличения до 414 px на 462 px)

Что же они будут обозначать?

Атрибут rIDPreviousAllocationPool

Данный атрибут будет опять же 64х битовым значением, составленным из 2х 32х битовых, и будет обозначать текущий пул RID'ов – тот, из которого идёт раздача клиентам. Разделить его на две части при помощи калькулятора вы сможете аналогично предыдущему примеру, так что повторяться не буду. Отмечу лишь, что часто возникает логичный вопрос – зачем хранить диапазон, если можно только стартовое значение? Проблема в том, что у каждого DC заложена возможность выбирать свой размер RID-пула, поэтому хранить надо именно диапазон.

Атрибут rIDAllocationPool

Это – следующий пул. Он только один и задаётся так же, диапазоном. Следующий пул будет запрашиваться в случае, когда израсходовалось 50% текущего (этот параметр ранее был 80%, но сейчас 50% и может изменяться) – т.е. контроллер домена, когда идёт активное расходование RID'ов (например, скриптом создаются учётные записи или группы), увидев в стандартной ситуации, что из пула в 500 значений израсходовано более 250, сразу, на фоне, запросит RID-мастер про новый пул – чтобы процесс не остановился и всегда был бы запас. Это даёт также дополнительную страховку от временного отсутствия RID-мастера – у каждого DC будет как минимум половина пула про запас, а то и больше (пример – из пула в 500 значений израсходовали 300, запросили новый пул – имеем $200+500=700$ RID'ов в запасе на оффлайн-работу). В нашем примере rIDAllocationPool и rIDPreviousAllocationPool равны – мы ведь ещё не израсходовали нужное количество RID'ов.

Атрибут rIDNextRID

Название этого атрибута не соответствует действительности – это последний выданный RID, а не следующий. В нашем примере он равен 1102; это обозначает, что если на этом DC создадут нового security principal'a, то его SID будет заканчиваться на 1103.

Интересная деталь – этот атрибут не реплицируется, т.е. он нужен только локальному DC, который ведёт свой личный подсчёт текущего RID'а. Это снижает “шумовую репликацию” из-за каждой мелочи (например, Вы начали создавать пользователя, и указали ему слишком короткий пароль – упс, ошибка, а SID-то уже сгенерили ему на этом DC, значит RID потратили, значит атрибут на единичку переместился, атрибут у объекта в domain partition – здравствуй, полномесная репликация всего раздела), но приводит к тому, что если пул отдали контроллеру, то никто не знает, сколько от этого пула контроллер уже израсходовал и какова текущая ситуация. В результате если DC умер или оказался в просроченном backup'e и у него уже tombstone, то пул считается израсходованным полностью – использовать неизрасходованные RID'ы не получится.

Атрибут rIDUsedPool

Атрибут не используется и ставится в нуль.

Что можно править из этих атрибутов? В общем ничего не надо править – но на некоторые параметры мы всё же сможем повлиять.

Изменяем размер RID pool

Изменить его просто – в реестре у каждого DC есть значение **HKLM \ SYSTEM \ CurrentControlSet \ Services \ NTDS \ RID Values**, у которого есть параметр **RID Block Size**. Этот параметр может изменяться от 500 до 15000 – если значения

выпадают за эти границы, то они трактуются как 500 или 15000 соответственно. Стандартное значение – 500; вы можете увеличить его, чтобы снизить число запросов к RID Master при интенсивном добавлении security principal'ов, но будьте осторожны – ведь потери RID'ов от этого только увеличатся, например, просто поправив этот параметр, вы мгновенно сделаете локальный RID Pool (и тот, который запросили следующим) некорректными и вызовете повторный запрос нового пула.

Простой сценарий, когда этот параметр надо увеличить – это далеко расположенный и редко выходящий на связь DC, на котором создаются доменные пользователи для доступа к местным ресурсам.

Просмотр локальных настроек RID Master

Локальных настроек у RID Master изначально было много – только действует сейчас одна. Вы можете просмотреть их через atcmd.exe:

RID Threshold обозначает то количество оставшихся в текущем RID pool RID'ов, при котором будет запрашиваться новый пул. Оно не изменяется с Windows 2000 Server SP4 – до него оно было 80%, теперь – 50%. Остальные настройки также неактуальны и игнорируются – нет смысла их задавать и рассчитывать на реакцию со стороны RID Master.

С главной задачей RID Master'a всё – теперь про не-основную.

Работа RID Master при переносе объектов между доменами

Перемещение объекта между разными лесами – случай тривиальный, потому что это по сути создание нового объекта. И, что логично, если это security principal, то без живого RID Master'a в destination никак не обойтись.

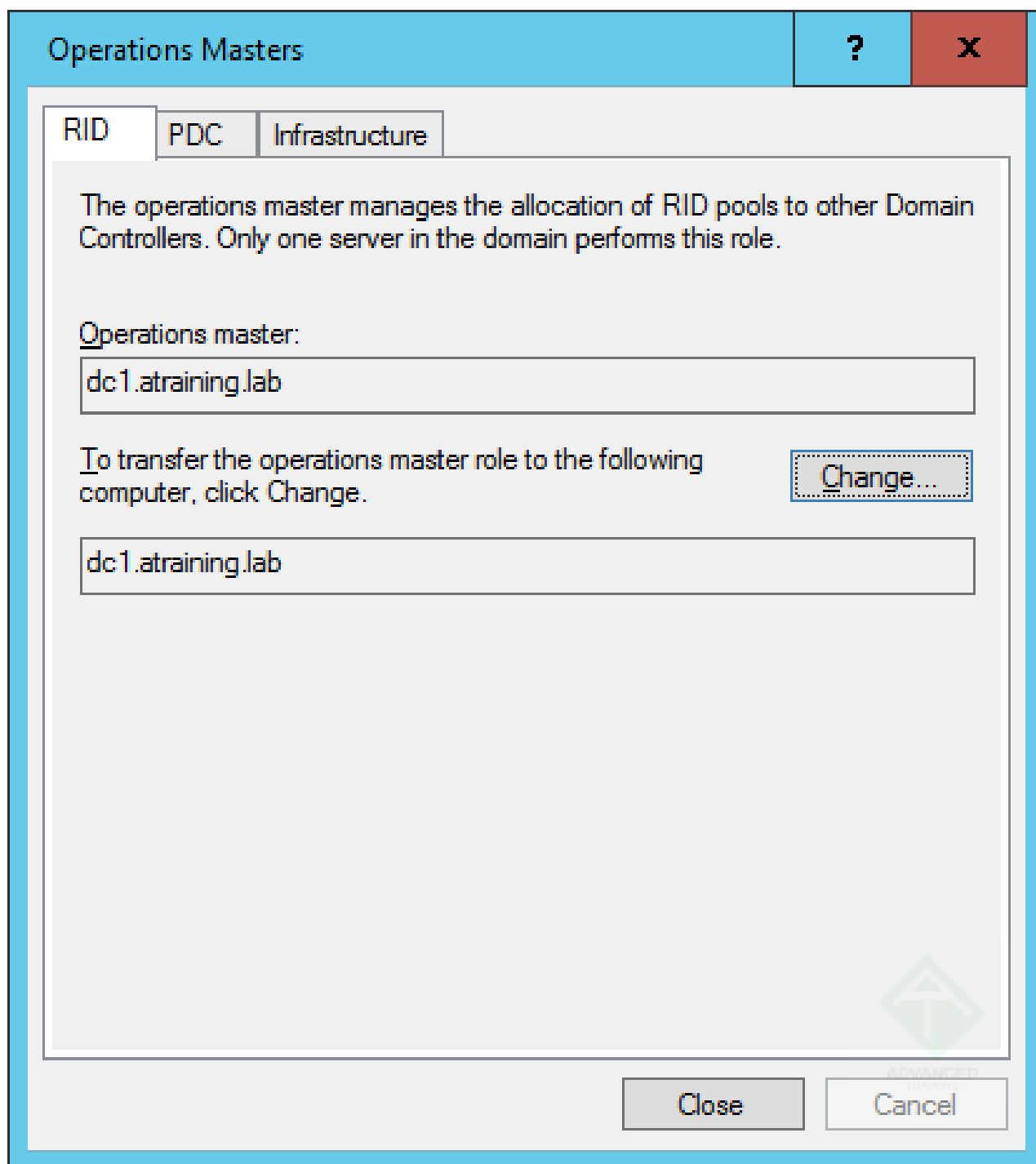
Интереснее перемещение security principal'a между разными доменами одного леса – когда GUID объекта остаётся, т.к. он не покидает своего леса, но вот SID надо сделать новый. В этой ситуации так же необходим живой RID Master – хоть, подчеркну, объект и не пересоздаётся “с нуля”, а перемещается. RID Master в этом случае нужен, потому что перемещаемым объектам надо выдавать новые SID'ы, и это нужно делать без конфликтов – иначе можно представить себе ситуацию “с двух DC одного домена параллельно запускают два ADMT, которые переносят объекты на два DC другого домена”. Так что если мигрируете, в пределах леса, один домен в другой – убедитесь, что RID Master жив и работоспособен.

С настройками и функционалом всё – теперь обсудим эксплуатацию.

Как перенести держателя FSMO-поли RID Master?

Изначально RID Master'ом назначается первый DC в домене – это штатно изменяемо как утилитой ntdsutil, так и через обычную оснастку Active Directory Users and Computers. Это просто – подключаетесь оснасткой к тому DC, на который

собираетесь переносить роль, и вызываете окно Operation Masters:

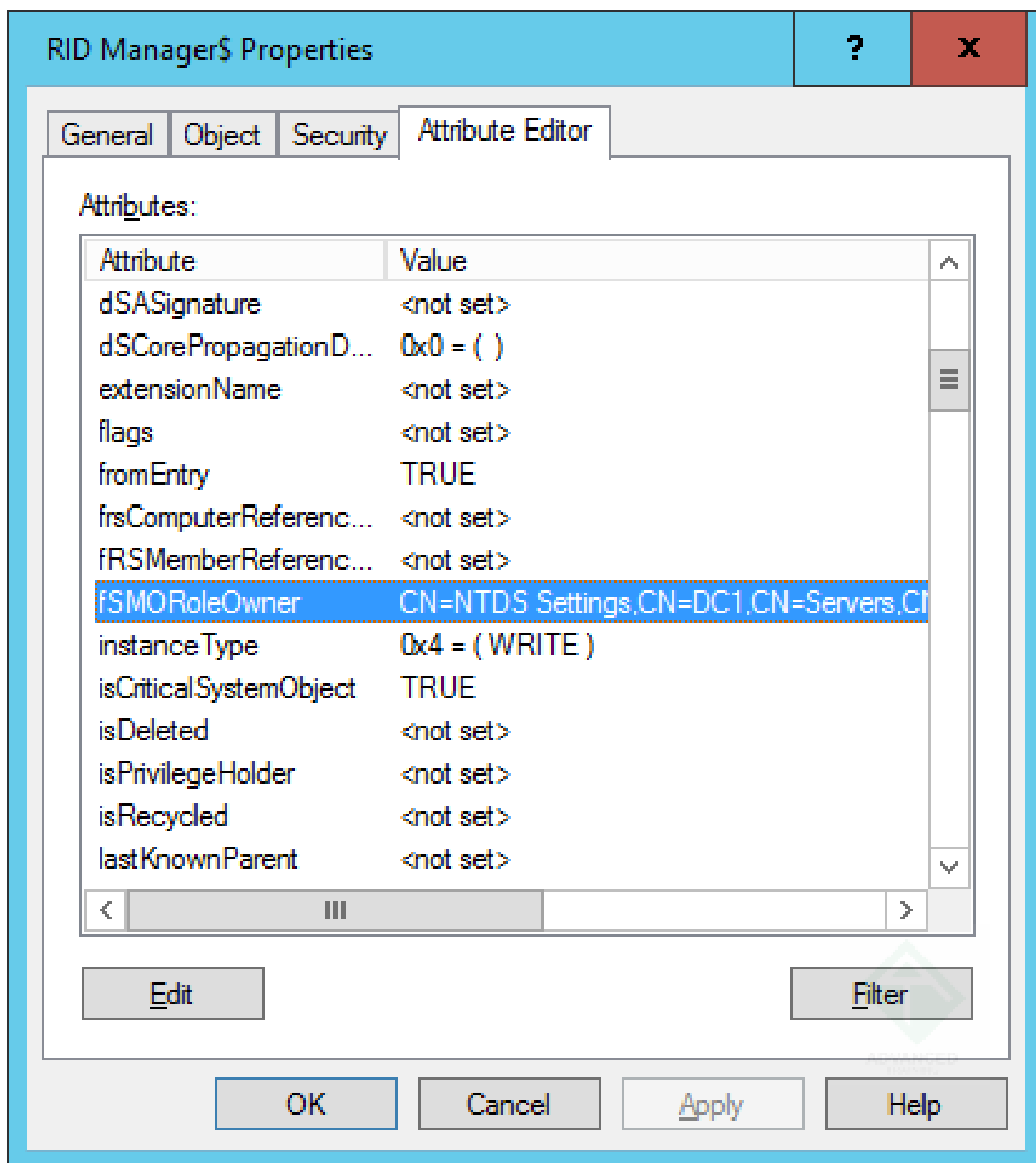


[Перенос роли RID Master](#)

[\(кликните для увеличения до 414 px на 462 px\)](#)

Где хранится информация, кто сейчас RID Master?

Данные о том, кто сейчас в домене держит FSMO-роль RID Master, содержатся в атрибуте объекта **RID Manager\$ – fsmoRoleOwner**:



[Как определить, кто сейчас RID Master в домене](#)

[\(кликните для увеличения до 414 px на 462 px\)](#)

Этот атрибут отображается в более удобном виде и в Operation Masters, и в ntdsutil.

Где располагать держателя FSMO-роли RID Master?

Так как эта роль нужна только контроллерам домена, то имеет смысл назначить RID Master'ом тот контроллер, который находится в центре топологии (с точки зрения скорости доступа). Ему не нужна какая-то особая безопасность и у него не будет явно выраженной дополнительной нагрузки даже в очень большом домене. Поэтому выделять для него, допустим, отдельную виртуалку – не надо, нет смысла.

Нужен ли RID Master'у отдельный бэкап?

Нет, у него нет своих личных данных – всё лежит в объектах Active Directory.

Как повысить надёжность работы RID Master?

Именно повысить её – трудно, т.к. он и так надёжно работает – сервис, как видно из функционала, несложный. Единственное – постарайтесь, чтобы держатель FSMO-роли RID Master не был единственным контроллером в сайте Active Directory – тогда все изменения с него примерно через 15 секунд будут на других контроллерах – в случае же, если он один, надо будет ждать межсайтовую репликацию.

С функционированием разобрались; теперь разберём мифы.

“Если RID Master упал то всё”

Первым делом надо понять, что RID Master – назначаемая роль; её обладатель не имеет никакой уникальной информации, хранимой локально – всё лежит в реплицируемых объектах Active Directory, поэтому назначение нового RID Master – операция простая и безболезненная. Ничего из данных нельзя потерять при пропадании RID Master – новый просто прочитает служебные объекты и продолжит выдавать пулы. Если он упал, то с текущими операциями ничего не случится – у всех DC есть запас RID’ов как минимум в целый пул, и они смогут даже без RID Master создавать группы и пользователей. И они не проверяют регулярно, жив ли RID Master – только, когда он им понадобится. Поэтому если он стал неработоспособным – надо просто сделать операцию seize и назначить FSMO-роль новому держателю.

Не надо прибегать к суициду и переставлять Active Directory – никто, кроме DC, RID Master’ом не интересуется вообще. Да и те – редко.

Как часто надо переносить FSMO-роль RID Master?

Только при изменении топологии домена Active Directory – т.е. таком изменении расположения DC по сайтам, чтобы текущий RID Master стал явно нерационально расположенным. Переносить надо штатно, через transfer, а не через seize – хоть при seize и будет для начала пробоваться штатный перенос роли с одного держателя на другого, если всё же получится seize, то будут заново выданы пулы RID’ов. Seize – операция только для ситуации, когда RID Master нужен вот прямо сейчас, а никакой возможности оживить предыдущий – нет.

Зависит ли скорость работы с пользователями и группами от мощности RID Master

Нет, он никак с ними не взаимодействует – он поставляет DC данные для локальной генерации одного из атрибутов – SID’a. Сам же RID Master с пользователями и группами в домене не взаимодействует.

В заключение

Надеюсь, что с одной FSMO-ролью вы теперь разобрались лучше – ну, а скоро будет и про другие. Active Directory – это мощный и хорошо настраиваемый механизм, и хотя многие “эксперты” по незнанию мистифицируют оный, представляя Active Directory как чёрный ящик, в котором что-то в зависимости от настроения и погоды “поглючивает”, всё проще – в Active Directory просто очень много всяких полезных небольших механизмов, которые надо хорошо понимать, равно как и взаимодействие одних механизмов с другими. Тогда – никакой мистики. Научный подход всегда побеждает религиозный.

До встречи!