

FSMO-роль Schema Master - как работает, за что отвечает, и для чего нужна. Мифы и их разоблачение - в комплекте.

 atraining.ru/active-directory-fsmo-schema-master

2016-04-30T03:24:19+08:00

CN=Schema,CN=Configuration,DC=atraining,DC=z P... ? X

Attribute Editor Security

Group or user names:

- SYSTEM
- Enterprise Read-only Domain Controllers (ATRaining\Enterpris...
- Schema Admins (ATRaining\Schema Admins)**
- Administrators (ATRaining\Administrators)
- ENTERPRISE DOMAIN CONTROLLERS

Add... Remove

Permissions for Schema Admins

	Allow	Deny
Full control	<input type="checkbox"/>	<input type="checkbox"/>
Read	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Write	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Create all child objects	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Delete all child objects	<input type="checkbox"/>	<input type="checkbox"/>

For special permissions or advanced settings, click Advanced.

Advanced

OK Cancel Apply Help

Привет.

Этот рассказ про FSMO-роли в Active Directory будет про самую могучую, легендарную, и плотнее всех покрытую мифами роль – Schema Master. Большого количества “мудрых” советов вида “схема мастер заглянул – ну вот и всё... конец... теперь директория глючить будет... только переставлять”, чем про Schema Master, просто нет в природе.

Разберёмся – как всегда, для начала предполагая, что вы знаете материал хотя бы на уровне курса [Microsoft 20410](#), читаемого в обзорно-упрощённом формате в авторизованных учебных центрах Microsoft – увы, детали работы Schema 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.

Schema Master

- Зачем нужен Schema Master
- Как работает Schema Master
- Как перенести владельца FSMO-роли Schema Master?
- Где хранится информация, кто сейчас Schema Master?
- Где располагать в лесу AD владельца FSMO-роли Schema Master?
- Как повысить надёжность работы Schema Master?
- Как повысить безопасность работы Schema Master?
- “Если Schema Master упал, то всё”
- “Если схему расширили неправильно, то всё”
- Зависит ли скорость работы доменов и репликации от расположения Schema Master?
- Нужен ли Schema Master’у отдельный бэкап?

Начнём.

Зачем нужен Schema Master

В статье про [Domain Naming Master](#) мы уже обсудили, что Active Directory логически разделена на т.н. naming context’ы (NC), одним из которых – и технически самым важным, является вложенный в **CN=Configuration** специфичный раздел **CN=Schema**.

Этот раздел содержит в себе одинокий контейнер класса **dMD** (уникальный и штучный в каждом лесу Active Directory), а хранится в этом контейнере всего два разных вида объектов – **classSchema** и **attributeSchema**.

Эти две сущности обеспечивают одно из ключевых достоинств Active Directory – возможность создавать новые объекты и расширять/изменять атрибуты у существующих объектов. В домене Windows NT такой возможности не было – перечень объектов и их полей/атрибутов был жёстко задан, и, например, если вы устанавливали новое приложение, которому для работы надо было хранить дополнительные данные у пользователя (например, сервис телефонии, который присваивал каждому пользователю SIP-номер), то такому приложению надо было

самостоятельно придумывать, как хранить эти данные, как привязывать к логинам пользователей, как обеспечивать доступность этих данных в любой точке домена и так далее. В Active Directory же ситуация упростилась – есть глобальный описатель всех существующих объектов, который можно модифицировать. Нужно добавить у пользователя новый атрибут – создаём этот атрибут, указываем его тип (строка, дата, число, и так далее), добавляем его в класс `user`, т.е. “расширяем схему”, и всё готово.

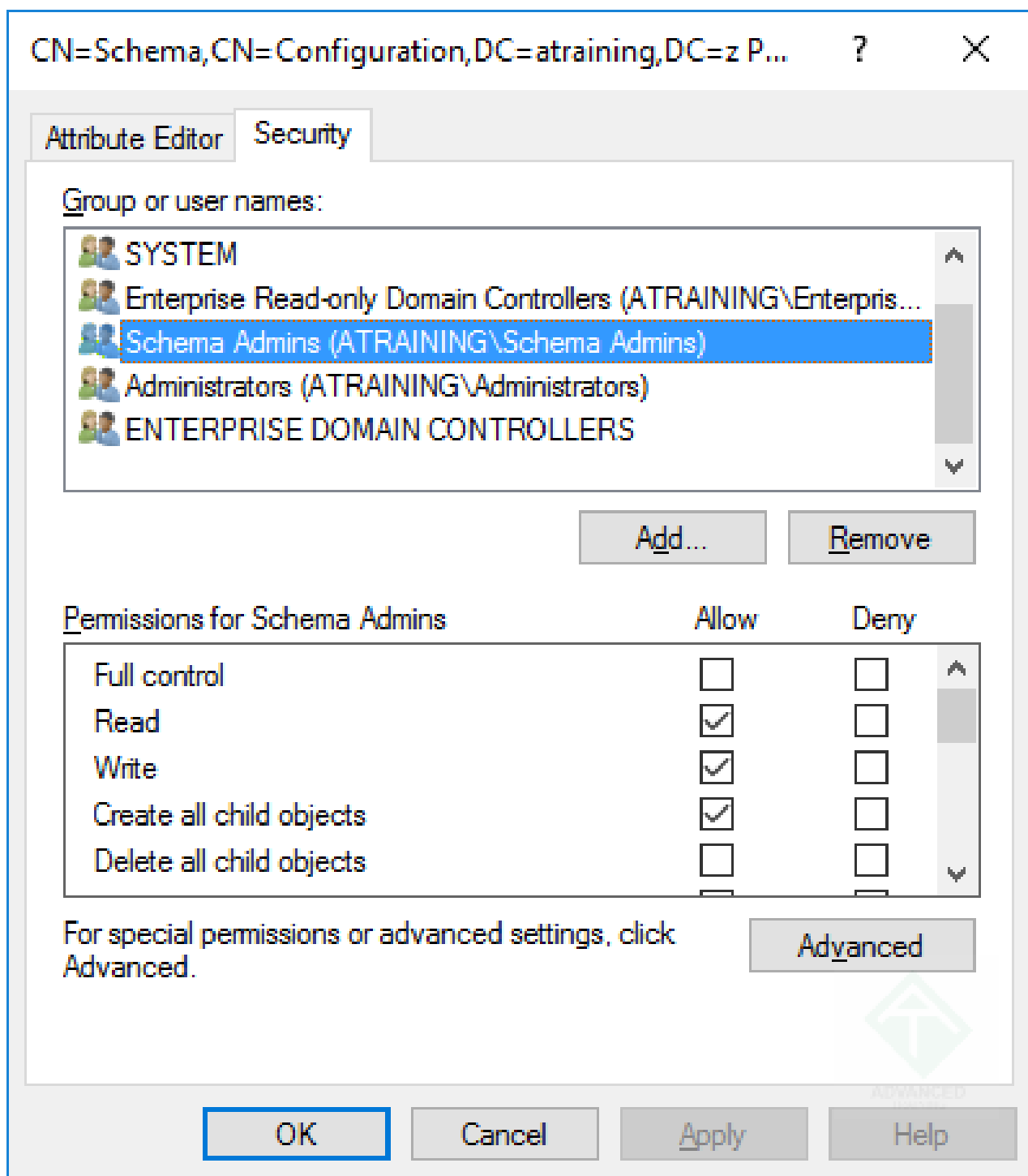
Но эта новая функциональность породила и новую проблему. Теперь надо обеспечивать единообразие описания всех объектов и их атрибутов на всех контроллерах леса.

Ведь если два контроллера начнут проводить репликацию, и будут иметь разные точки зрения на то, что именно называется, допустим, объектом класса `user`, то репликации не получится. Допустим, с точки зрения одного DC, у объекта `user` – 57 атрибутов, а с точки зрения второго DC – 58 атрибутов. При репликации последнего атрибута возникнет ситуация “что ты мне присылаешь-то, нет такого у `user`”, либо “что ты закончил-то атрибуты перечислять, там ещё один остался”. Соответственно, необходимо обеспечить идентичность этих данных – а значит процесс изменения схемы должен быть очень предсказуемым, очень чётким, очень безопасным и очень быстрым.

Скорость этого процесса обеспечивается нулевой задержкой (на уровне настроек naming context) репликации раздела схемы, а также компактностью раздела – в нём только два типа объектов, и они не особо большие в плане данных.

Остальные пункты – по части предсказуемости и безопасности – реализуются выделением во всём лесу Active Directory единственного DC, на котором в раздел `CN=Schema` можно вести запись. Это и будет Schema Master. Все остальные DC смогут только читать этот раздел.

Безопасность модификации `CN=Schema` будет достигаться специфичным ACL, где права на добавление есть только у специальной группы `Schema Admins`, да и то – даже у неё нет Full Control:



[Корневой ACL у раздела Schema](#)

[\(кликните для увеличения до 400 px на 455 px\)](#)

А также специальным параметром **Schema Update Allowed**, настройка которого ранее (в Windows 2000 Server) была доступна через графический интерфейс, а теперь скрыт от администратора. При этом параметр продолжает существовать (включить-выключить его можно на каждом DC в лесу Active Directory, но приниматься во внимание он будет только на владельце FSMO-роли Schema Master):

```
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 307
(c) 2011-2016 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

SERVER-A(config)#ntds
SERVER-A(config-ntds)#schema
SERVER-A(config-ntds-schema)#?

  sua                Enable schema update on this DC
  end                Exit to global configuration mode
  exit              Return to previous configuration mode
  no                Disable something or return value to default
  quit              Quit

SERVER-A(config-ntds-schema)#sua

  Schema update, if this DC become Schema Master, allowed

SERVER-A(config-ntds-schema)#_
```

[Разрешение на запись в Schema через параметр Schema Update Allowed \(кликните для увеличения до 979 px на 514 px\)](#)

Таким образом, модифицировать схему можно только через один-единственный DC во всём лесу Active Directory, при наличии прав (по умолчанию для этой операции есть группа Schema Admins, можно получить права как добавившись в неё, так и создав новую и выдав ей права на контейнер **emd**) и при отсутствии запрета на Schema Update Allowed (по умолчанию этот параметр не задан, поэтому изменения вносить можно).

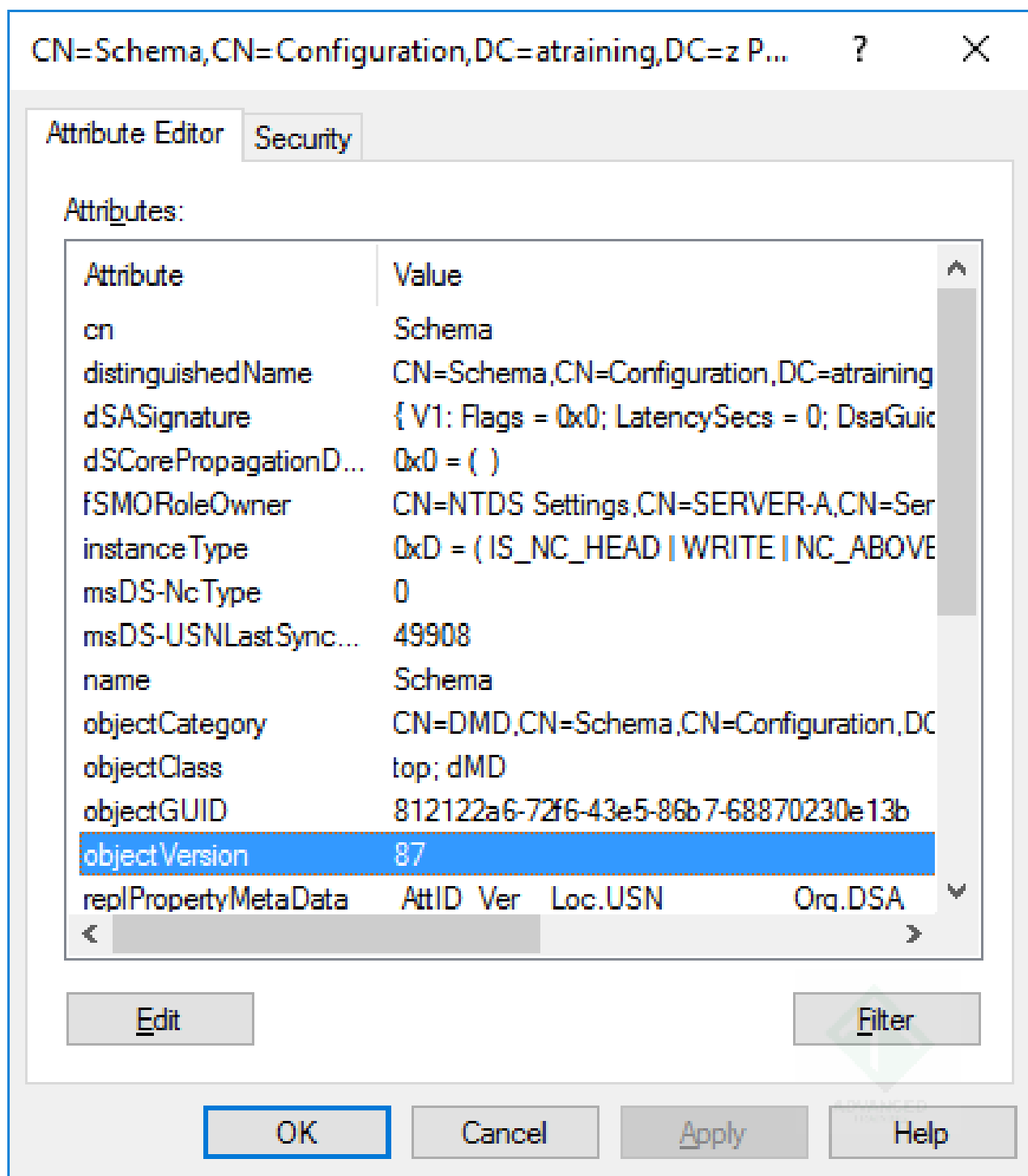
Теперь про то, какие операции этот специальный DC умеет делать.

Как работает Schema Master

Когда модификация схемы происходит – т.е. к данному контроллеру успешно подключились, имеют нужные права, редактирование схемы не отключено, и внесены какие-либо изменения – владелец роли Schema Master ждёт краткий промежуток времени, после которого, опираясь на текущие настройки (это отдельная тема, которая напрямую к FSMO-роли не относится, поэтому тут не углубляемся – про репликацию, её настройку и оптимизацию, я напишу отдельно), будет инициировать репликацию раздела **CN=Schema**. Это делается не мгновенно, чтобы в случае внесения пачки последовательных изменений не дёргать репликацию ради каждого из них, а запустить разово, по окончании процесса редактирования.

По сути, после внесения изменений в схему, лес Active Directory переходит в состояние “lack of convergence”, и остаётся в нём до момента, когда новый вариант схемы будет у всех контроллеров. В этот переходный период может возникнуть штатная ситуация, когда два контроллера начнут репликацию какого-либо раздела

(например, доменного), и будут иметь при этом разные схемы. Что в этом случае произойдёт? Всё просто – при попытке произвести репликацию любого раздела (вообще любого) вначале сверяется атрибут **objectVersion** у объекта **dMD**:



[Версия схемы Active Directory.](#)

[\(кликните для увеличения до 400 px на 455 px\)](#)

Если этот атрибут имеет разные значения, то запрос на репликацию раздела отменяется – схема ведь точно разная, надо дождаться, пока её репликация произойдёт. Вы можете легко посмотреть версию схемы на всех контроллерах леса, используя стандартную утилиту **repadmin**:

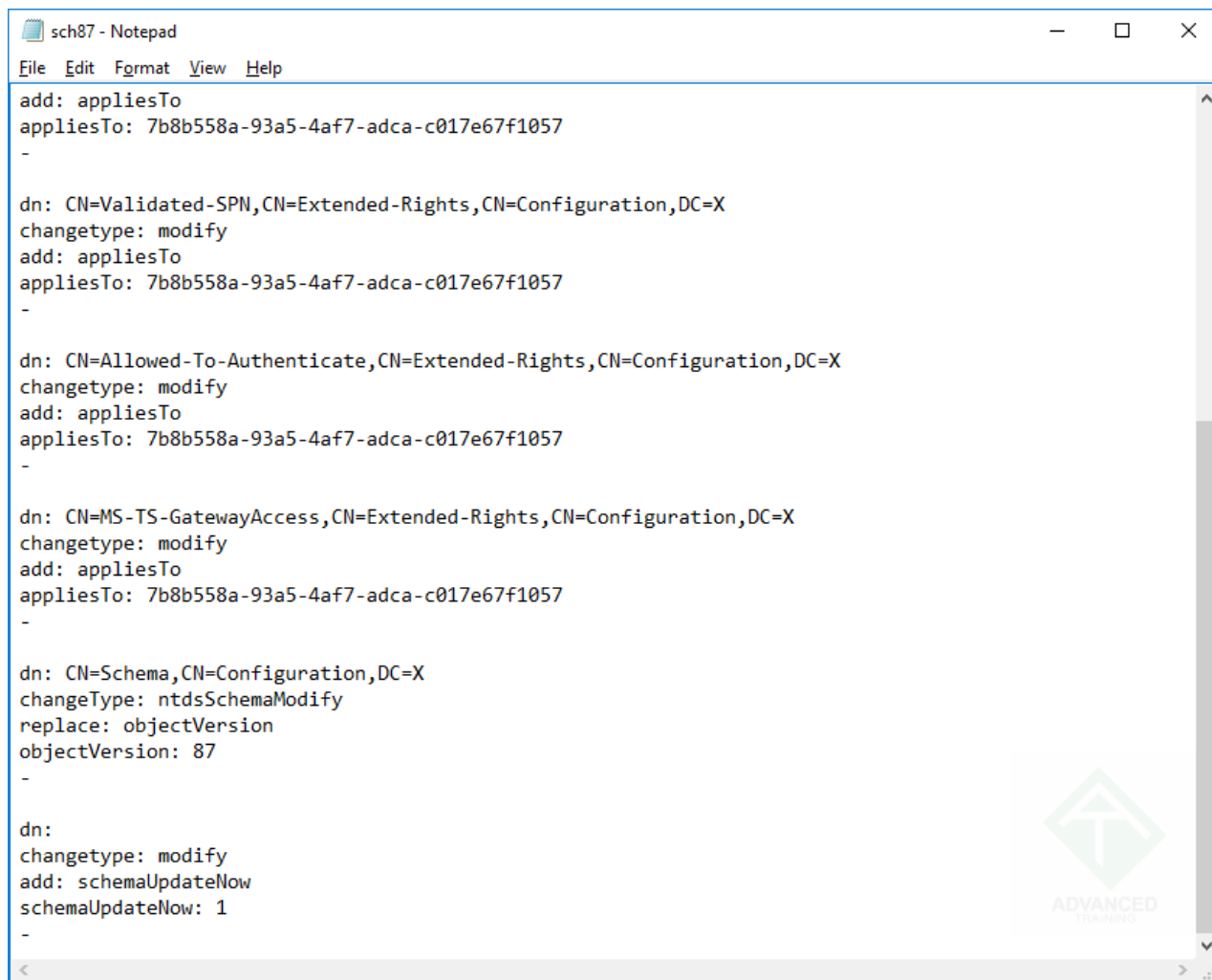
```
repadmin /ShowAttr * "CN=Schema,CN=Configuration,DC=atraining,DC=z"  
/atts:ObjectVersion
```

Прикольнее то, что даже если этот атрибут имеет одинаковые значения и у source DC и у destination DC, схема может быть разной. Суть атрибута `objectVersion` – т.н. “базовый уровень” схемы, указывающий на то, какая максимальная версия Windows Server может быть DC в этом домене. Данные уровни выглядят так:

- 13 – Windows 2000 Server
- 30 – Windows Server 2003
- 31 – Windows Server 2003 R2
- 44 – Windows Server 2008
- 47 – Windows Server 2008 R2
- 56 – Windows Server 2012
- 69 – Windows Server 2012 R2
- 87 – Windows Server 2016

и отражают только “основной поток” добавления новых объектов, без которых будет невозможно само функционирование Active Directory.

Простой пример – допустим, у вас есть лес Active Directory, в котором максимальный DC – с Windows Server 2003. Выходит Windows Server 2008, у которого есть возможность назначать на группы и пользователей отдельные политики паролей – используя новый тип объектов, `PSO`. То есть, в контейнере `System` в доменном NC появляется новый контейнер для объектов `PSO`. Следовательно, надо, чтобы все DC знали, что это за объект такой, иначе репликация доменного раздела будет технически невозможной. Поэтому на диске с Windows Server идёт утилита `adprep.exe`, которая берёт и мужественно, имитируя, что работает сама, запрягает утилиту `ldifde.exe` (сейчас уже существующую для этого применения в специальном варианте dll-файла, чтобы было необязательно иметь Active Directory command line tools установленными там, где запускается `adprep.exe`), последовательно скармливая ей LDIF-файлы с расширением `ldf`:



```
sch87 - Notepad
File Edit Format View Help

add: appliesTo
appliesTo: 7b8b558a-93a5-4af7-adca-c017e67f1057
-

dn: CN=Validated-SPN,CN=Extended-Rights,CN=Configuration,DC=X
changetype: modify
add: appliesTo
appliesTo: 7b8b558a-93a5-4af7-adca-c017e67f1057
-

dn: CN=Allowed-To-Authenticate,CN=Extended-Rights,CN=Configuration,DC=X
changetype: modify
add: appliesTo
appliesTo: 7b8b558a-93a5-4af7-adca-c017e67f1057
-

dn: CN=MS-TS-GatewayAccess,CN=Extended-Rights,CN=Configuration,DC=X
changetype: modify
add: appliesTo
appliesTo: 7b8b558a-93a5-4af7-adca-c017e67f1057
-

dn: CN=Schema,CN=Configuration,DC=X
changeType: ntdsSchemaModify
replace: objectVersion
objectVersion: 87
-

dn:
changetype: modify
add: schemaUpdateNow
schemaUpdateNow: 1
-
< > ...
```

[Пример LDF-файла с модификацией схемы Active Directory](#)
([кликните для увеличения до 841 px на 676 px](#))

На картинке – пример такого файла, sch87.ldf, последнего из комплекта LDF-файлов с информацией о схеме Active Directory у Windows Server 2016 TP5. Я специально сделал скриншот финальной части файла, чтобы показать две штуки – во-первых, как видно, атрибут **objectVersion** выставляется в 87 вручную, а не автоматически – т.е. никакого “автоучёта и автопересчёта номера версии схемы после модификации” нет, вы можете в неё напихать новых объектов сколько угодно и номер версии никуда сам не убежит меняться, а во-вторых в финале, после применения пачки таких файлов с описаниями добавлений-модификаций, вручную вызывается функция обновления кэша схемы. Этот кэш – это копия схемы из файла **ntds.dit**, полностью лежащая в оперативной памяти. Нужно это для ускорения работы контроллера, потому что обращений к схеме будет предсказуемо много. Этот кэш автообновляется по умолчанию раз в 5 минут, поэтому если не сделать эту операцию, то после добавления в схему новых классов объектов, их экземпляры можно будет создавать лишь через некоторое время – а если вручную поставить атрибуту-триггеру **schemaUpdateNow** единичку, то обновление произойдёт сразу.

После того, как все эти файлы будут применены, в схеме появятся все описания новых классов объектов, описания новых атрибутов, из которых объекты состоят, а также применятся модификации к уже существующим объектам, и после этого

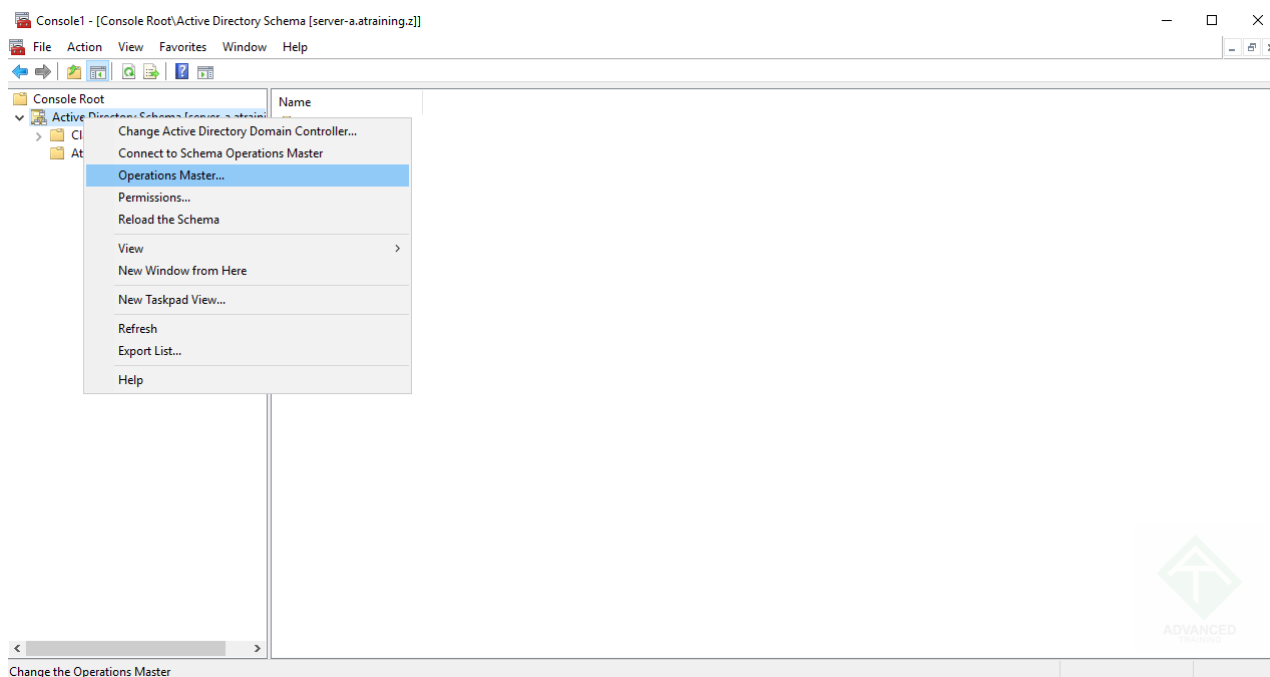
можно добавлять DC на базе нового Windows Server в лес – теперь все его новые объекты, которые он приносит с собой изначально, не вызовут у других DC паники и непонимания.

Как понятно, помимо модификации схемы, вызванной появлением новой версии Windows Server, есть случаи модификации, нужной для установки конкретного продукта (например, Microsoft Exchange или Skype For Business). В этом случае номер версии схемы Active Directory не меняется, а вот объекты – да, изменяются. У таких программных продуктов обычно есть свой атрибут для хранения версии схемы – у того же Microsoft Exchange это будет `rangeUpper` у объекта `CN=ms-Exch-Schema-Version-Pt,CN=Schema,CN=Configuration,DC=atraining,DC=z`, но, как понятно, Schema Master никакого особого подхода или дополнительного учёта этого не ведёт – для него это просто некие атрибуты, такие же, как и все другие – а вот инсталлятор Microsoft Exchange, как понятно, умеет и проверять этот атрибут, и модифицировать его, обновляя схему Active Directory своими объектами и атрибутами.

ОК, так как статья у нас всё ж именно про задачи Schema Master, а не про саму работу со схемой в Active Directory, перечень технического функционала, специфичного для FSMO Schema Master, можно заканчивать. Потому что в принципе, кроме единой точки внесения изменений в схему и их распространения по директории, всё остальное у него такое же, как у других DC – тот же кэш схемы, например, есть на каждом. Перейдём к вопросам эксплуатации этой ценной боевой единицы.

Как перенести владельца FSMO-поли Schema Master?

Изначально Schema Master'ом назначается первый DC в первом домене леса – это штатно изменяемо как утилитой `ntdsutil`, так и через оснастку Active Directory Schema. Эту оснастку надо предварительно зарегистрировать – она, в рамках логики “не надо без лишней необходимости давать слишком много продвинутых административных инструментов” лежит в Windows Server в виде `dll`-файла, но не зарегистрирована. Надо выполнить с правами уровня локального администратора команду `regsvr32 schmmgmt.dll`, после чего открыть `mmc` и добавить эту оснастку. После – правой кнопкой по корню Active Directory Schema и выбираете Operations Master:

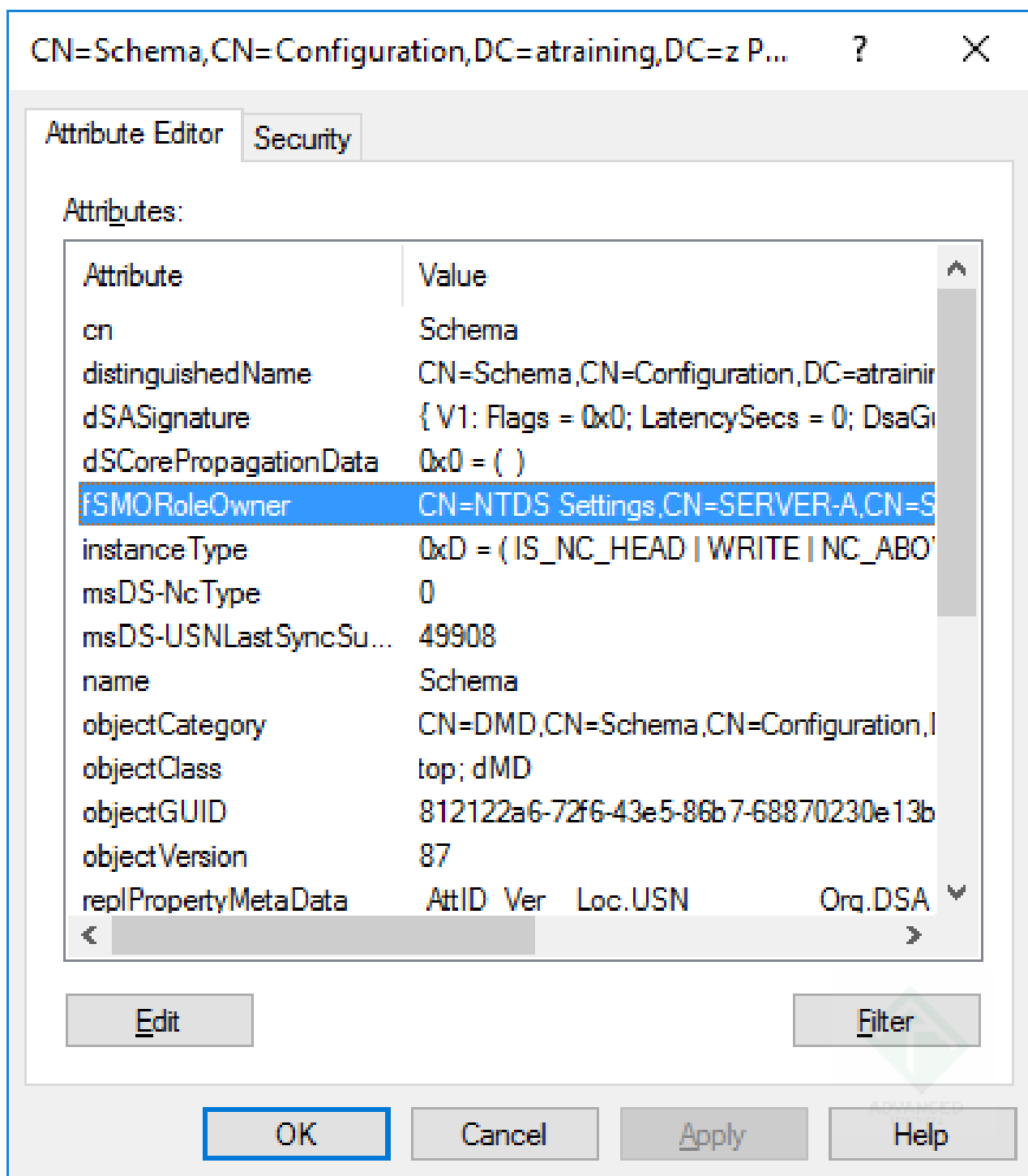


[Перенос роли Schema Master](#)

[\(кликните для увеличения до 1269 px на 673 px\)](#)

Где хранится информация, кто сейчас Schema Master?

Данные о том, кто сейчас в домене держит FSMO-роль Schema Master, содержатся в атрибуте корневого объекта **CN=Schema** – **fSMORoleOwner**:



[Как определить, кто сейчас Schema Master в домене](#)
(кликните для увеличения до 400 px на 455 px)

Где располагать в лесу AD владельца FSMO-роли Schema Master?

Учитывая, что каждая операция по модификации схемы – это отдельное событие, то имеет смысл перед каждым таким случаем смотреть текущую топологию леса Active Directory и, при необходимости, делать Schema Master'ом тот контроллер, который находится в центре топологии. Если вы хотя бы немножко разбираетесь в сетях, то подсказка проста – представьте себе сайты Active Directory как коммутаторы, между которыми работает классический алгоритм STP – вот надо

сделать схема-мастером того, кто будет STP Root. Расположение Schema Master после того, как очередные изменения в схеме разойдутся по всем контроллерам, уже абсолютно некритично, т.к. у него нет никаких задач, требующих обращения к нему в ситуации “полный convergence”.

Как повысить надёжность работы Schema Master?

Никак, его работа – разово инициировать раздачу обновлённой схемы. Если это событие прошло, то повышать нечего.

Как повысить безопасность работы Schema Master?

Самый простой способ – никого не добавлять в группу Schema Admins. Если нужно произвести модификацию схемы – добавить учётку, залогиниться ей, провести операцию, удалить учётку из группы, выйти из сеанса. Данная группа нужна только для модификации схемы, ни на что другое она дополнительные права не выдаёт. В неё не надо добавлять ген.директора и главбуха с аргументом “на всякий случай, а то у них что-нибудь не откроется, потому что прав маловато”. В неё вообще никого не надо добавлять на постоянной основе – можно, к примеру, для пущей безопасности настроить Restricted Groups на периодическую очистку этой группы, т.е. раздавать “пустой” список участников – тогда даже если кто-то случайно или злонамеренно добавится, сделав себе лазейку для атаки на будущее, его вычистит регулярное применение групповых политик.

“Если Schema Master упал, то всё”

Боже, какое количество херни на эту тему ходит в форумах спившихся сисадминов категории “не трожь если работает”, самоназванных “энтерпрайзных архитекторов” и “сопричастных к известным вендорам и крупным проектам”! В аду есть отдельный котёл для тех, у кого “схема начала подглючивать”, “атрибут недореплицировался один раз и теперь весь домен подглючивает”, свидетелей “Вася тут докупил себе MCSE, прочитал на форуме для экспертов один совет, что-то в реестре у схемы подправил, домен реально летать начал, раза в три быстрее”, а также готического сообщества “Поставил в боевой домен диск с Windows Server 2012 R2, но не RTM, а бетой, расширил схему – пиши пропало, теперь у тебя бета-схема, ничего уже не вернуть, лучше сразу новый лес делать, попутно разрезая запястья отточенным CD-диском”. Отдельно радуется закрытый клуб умственно богатых “бета-схема конфликтует с RTM-схемой” – лично наблюдал как-то дискуссию нескольких MVP по Exchange, которые всерьёз обсуждали ситуацию “в production-домен поставили новый CU на Exchange – а версия схемы не поменялась, наверное надо было ждать нового продукта, потому что апгрейд явно с глюком прошёл” – т.е. люди не знают даже азов работы Active Directory, но фантазируют на тему неизвестного им функционала.

Если владелец роли Schema Master куда-то упал – например, его стёрли – целиком, виртуалку – то надо просто назначить новым Schema Master’ом кого-нибудь другого. Ну а перед выполнением расширения схемы в очередной раз – просто оценить, там ли, с точки зрения текущей ситуации с топологией, расположен Schema Master, и перенести его при необходимости.

“Если схему расширили неправильно, то всё”

В этой мысли есть доля правды – но относится она к Windows 2000 Server; начиная с Windows Server 2003, такой ситуации уже быть не может. В Windows 2000 Server не было возможности “выключать” атрибуты и классы в схеме – поэтому можно было сделать так:

- Телепатически проникнуть в мозг разработчиков Active Directory (город Редмонд);
- Выудить у них сведения о том, какие объекты и атрибуты добавятся в новой версии Windows Server;
- Открыть редактор схемы и создать какой-нибудь объект или атрибут с таким же именем, но другим типом (например, известно, что в новом Windows Server к атрибуту `drink` у класса `user` добавится атрибут `bristolScale`, числовой – значит, добавляем атрибут `bristolScale` вручную, но, например, типа `unicode string`);
- Дело сделано – теперь в этот лес Active Directory не получится добавить новый Windows Server, когда он выйдет – ведь перед добавлением первого DC надо будет сделать `adprep`, а оный споткнётся о ситуацию “такой атрибут уже есть, только неправильного типа, и непонятно, что с этим делать”;

В Windows Server 2003 стало возможно “выключать” атрибуты и классы, ставя в их описании в схеме атрибут `isDefunct` в TRUE, поэтому данная ситуация перестала быть безвыходной.

Неправильно расширить схему и “загубить всю Active Directory глючным расширением схемы”, говоря проще, сейчас нельзя. Более того, её расширение в интересах какого-либо ПО никак не задевает другое ПО. То есть надо тому же Skype for Business добавить к классу `user` новые атрибуты – ОК, про это никак не узнает тот софт, который их не читает и не пишет. Если в Active Directory появится возможность создания нового типа объектов – аналогичная ситуация; на существующие задачи это никак не повлияет. Все раздувания модификации схемы до уровня “очень сложное и рискованное событие” не имеют под собой никакой основы, кроме разве что освоения бабла консалтерами и системными интеграторами с формулировкой “наши эксперты осторожно и без потерь проведут эту сложнейшую процедуру на уровне всей сети предприятия”. Тут всё логично, чем больше мистики и запугивания клиента, тем он больше человеко-часов на эту задачу подпишет.

Зависит ли скорость работы доменов и репликации от расположения Schema Master?

Нет. Чисто в теории, можно напихать в схему столько объектов, что репликация схемы с владельца FSMO-роли Schema Master станет отдельным, долгим и масштабным событием, которое будет тормозить работу домена, т.к. пока схема не “устаканится” во всём лесу, попытки репликации доменного раздела будут откладываться. Но сомнительно, что это достижимо на практике.

Нужен ли Schema Master’у отдельный бэкап?

Нечего бэкапить, содержимое схемы идентично на всех контроллерах леса, у владельца FSMO-роли нет локально хранимой уникальной информации.

В заключение

Всё совсем не страшно, не содержит никакой мистики и тайн – хотя упавшим Schema Master’ом и таинственной Глючной Схемой пугают людей до сих пор. Не обращайтесь внимания – качественные знания всегда побеждают мистические фантазии и предрассудки. На очереди на препарирование – другие FSMO-роли.

До встречи!