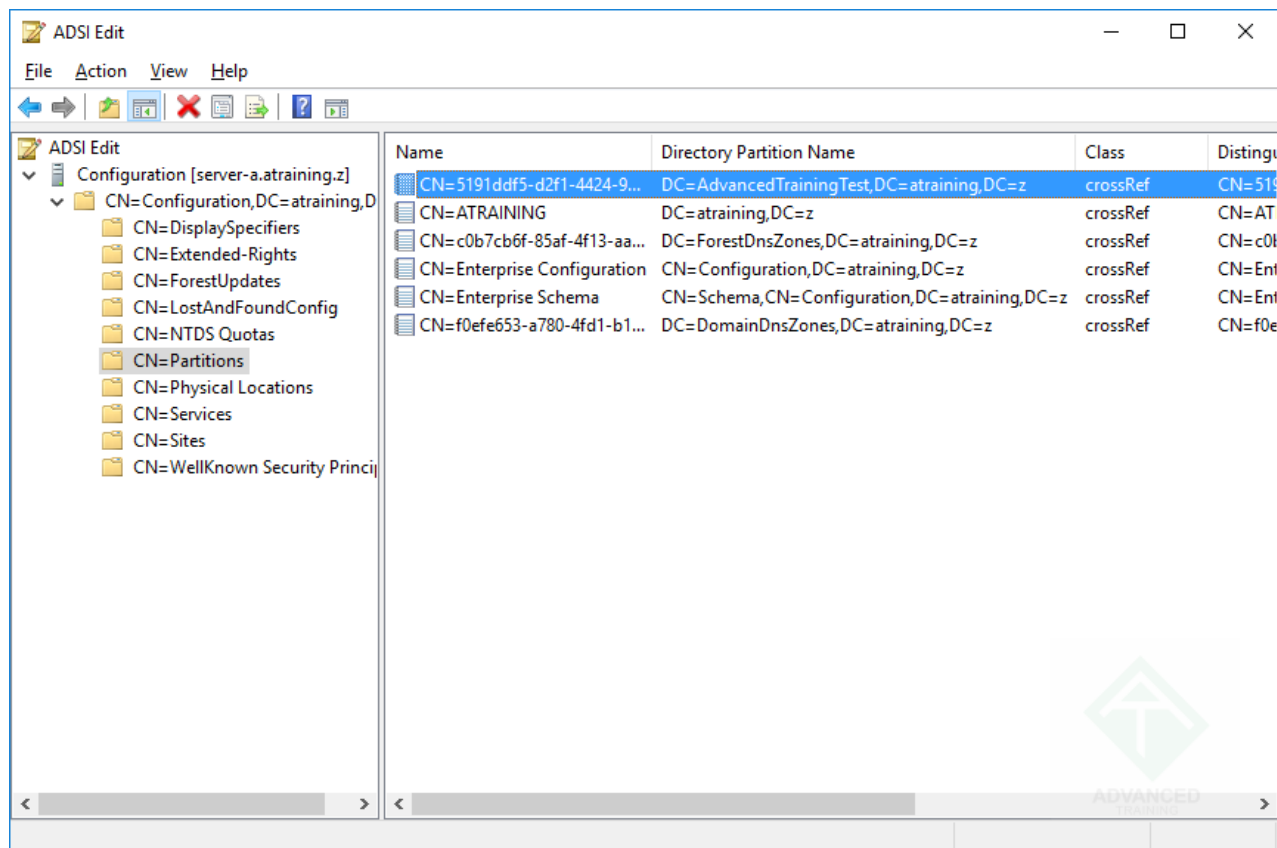


# FSMO-роли - без секретов, тайн, и до деталей. Эта статья - про DNM (Domain Naming Master).

 atraining.ru/active-directory-fsmo-domain-naming-master

2016-04-28T13:46:57+08:00



Привет.

Наш мини-цикл рассказов про FSMO-роли в Active Directory расширяется.

Повторюсь – 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.

Мы пробуем разобраться с каждым из FSMO-мастеров отдельно. Стараясь не сильно углубляться в сторонние темы (хотя возможностей будет масса), и 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 Domain Naming 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.

# Domain Naming Master

---

Начнём.

## Базовые задачи Domain Naming Master

---

Для того, чтобы лучше разобраться, сделаем краткий экскурс в мат. часть про устройство Active Directory.

Все данные в Active Directory хранятся физически на диске в специфичной БД, которая реализована в формате с интересным названием – Microsoft Jet Blue. Голубой он по каким-то личным причинам, но нам главное, что это упоминание цвета важно – потому что формат Microsoft Jet бывает также варианта Red – и этот вариант используется в Microsoft Access. Нас же интересует “серверный” Jet Blue, чаще упоминаемый под именем Extensible Storage Engine (ESE) и использующийся во множестве применений – от базы DHCP Server и уже экзотического WINS до Active Directory и Exchange.

Данный формат БД удобен тем, что объекты находятся не в таблицах, а в виде отдельных объектов, расположенных внутри БД в логических блоках-“страницах” (размером от 2х до 32х килобайт), доступ к которым реализован посредством бинарного дерева. Благодаря этому чтение произвольного объекта происходит быстро и не влечёт за собой необходимость линейного перебора БД, выборки SQL-style, равно как и предварительной индексации, создания “ключевой колонки” и подгрузки всей БД в память. То есть формат ESE адресно адаптирован под задачу “быстрое произвольное чтение объектов” – в отличие от обычной SQL-базы, которая адаптирована под логику “быстро и много дописывать в конец таблицы, а потом хитрыми запросами выгребать нужное из массива однотипных строк в этой БД”.

Основная причина такого выбора – это то, что Active Directory чаще читается, чем пишется. Т.е. вы создаёте пользователя или группу один раз, а запрашиваете атрибуты пользователя или состав группы – чаще, чем один раз. Поэтому традиционный SQL-вариант не подходит – он под другой профиль нагрузки адаптирован, обратный нашему.

Благодаря этому подходу получают дополнительные плюсы (помимо очевидной скорости доступа) – например, можно делать внутреннюю дефрагментацию базы, перенося записи из частично заполненных страниц, чтобы освободить их полностью и, таким образом, сокращать количество необходимых для чтения объектов страниц. Но остаётся одна проблема – что все объекты пишутся как попало, без структуры (на самом деле структура, подобная таблице, есть, но это тема для другого разговора).

Теперь к логической организации внутри Microsoft Jet Blue. Чтобы избежать превращения ESE-базы в варианте Active Directory в мусорник, где вперемешку лежат различные объекты и их атрибуты, база, выглядящая как файл ntds.dit и её transaction-логи (файлы с именем вида edb\*.\*, часть из которых действительно

хранит данные транзакций и отметки о процессе их выполнения, а часть просто резервирует место на случай критической нехватки пространства для расширения ntds.dit на диске), делится внутри на логические разделы – т.н. Active Directory partitions. Это выглядит со стороны использующего как аналог логических разделов на жёстком диске, но по факту внутри ESE-база всё равно мусорник. Несмотря на логическую отделённость, в ntds.dit при добавлении нового раздела никаких внутренних изменений формата не происходит – просто регистрируется новый контекст имён.

У каждого контекста имён будут свои настройки хранения (на какие сервера он будет реплицироваться), настройки технических параметров (частота и тайм-ауты репликации), и настройки безопасности (корневой ACL).

Как понятно, полный список этих контекстов имён (NC – naming context) уникален в пределах леса Active Directory. Часть из NC стандартна и есть в каждом лесу, начиная с Windows NT 5.0 – это, как минимум:

- Forest root domain – тот домен, с которого начинался лес;
- Configuration – общие для всех доменов леса параметры (то, что не привязано напрямую к NT-доменам – например, настройки PKI, или [LDAP-политики](#) или настройки Exchange Server);
- Schema – отдельные атрибуты и инструкция “как собрать из пачки атрибутов конкретный экземпляр объекта”;

Новые контексты создаются в лесу каждый раз, когда вы добавляете новый домен – ведь у него будут свои личные объекты (пользователи, контейнеры, группы, компьютеры), поэтому ему надо будет хранить их отдельно, и реплицировать тоже строго между своими контроллерами, одного домена. Начиная с Windows NT 5.1 также появляется возможность создавать свои NC – просто для задач “хранить данные на указанных DC и реплицировать по расписанию” – впрочем, этой возможностью, цензурно выражаясь, не особо пользуются.

И вот мы наконец дошли до того, зачем нам нужен Domain Naming Master. Задача этой роли – которая, как переходящий вымпел, присваивается единственному контроллеру в лесу Active Directory (сколько бы доменов в этом лесу не было) – очень проста; не допускать конфликтов имён NC.

То есть, владелец роли Domain Naming Master отвечает за:

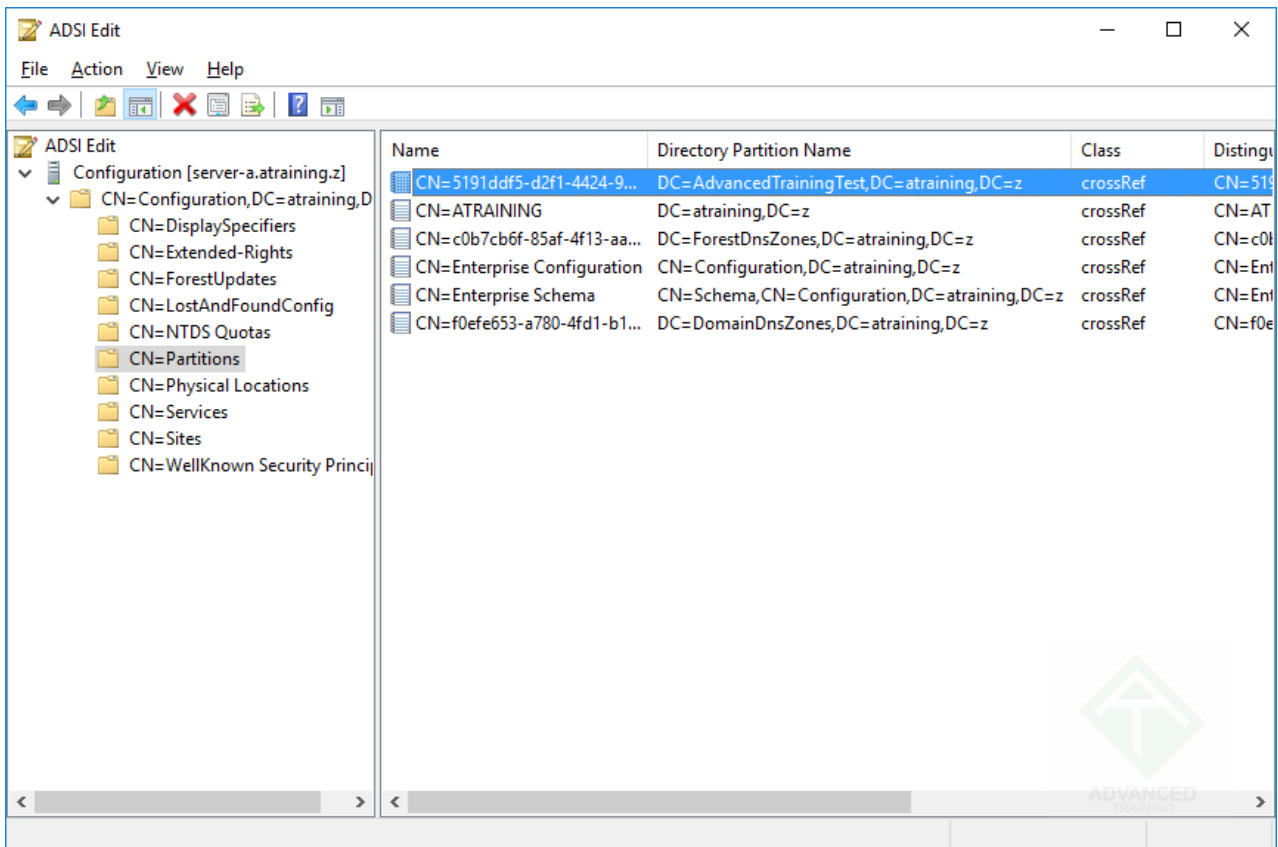
- Недопущение добавления в лес домена с именем, которое уже есть у другого домена;
- Блокировку возможности создать контекст имён, который совпадает по DN’у с уже существующим;
- Переименование домена в лесу;

Итак, данная роль, по сути, самая малоактивная из всех доступных – сервер-владелец роли DNM нужен только при операциях добавления-удаления домена из леса, да создания новых контекстов имён.

Посмотрим чуть внимательнее на техническую сторону вопроса.

## Как выглядит зона ответственности Domain Naming Master

Описания и параметры naming context'ов будут располагаться в контейнере **CN=Partitions** раздела **Configuration** – мы залезем в него, используя ADSI:



[Где в домене находятся naming context-ы](#)  
([кликните для увеличения до 862 px на 572 px](#))

Наш домен, как понятно из картинки, называется **atraining.z**, в нём:

- три стандартных NC – доменный (**DC=atraining,DC=z**), конфигурация леса (**CN=Configuration,DC=atraining,DC=z**), и схема (**CN=Schema,CN=Configuration,DC=atraining,DC=z**);
- две стандартных (с Windows Server 2003) application partition – Domain DNS Zones (**DC=DomainDNSZones,DC=atraining,DC=z**) и Forest DNS Zones (**DC=ForestDNSZones,DC=atraining,DC=z**);
- и одна просто так созданная вручную application partition с DN = **DC=AdvancedTrainingTest,DC=atraining,DC=z** – исключительно чтобы не дефолтную конфигурацию рассматривать :)

Если мы захотим сделать какие-либо изменения в этих данных – то есть или добавить новый домен в лес, или убрать домен из леса (удалив последний контроллер), или создать новый NC, или изменить параметры существующего NC – нам нужен присутствующий в онлайне владелец роли Domain Naming Master, потому что все записи в лесу в данный кусок **Configuration** идут через этот выбранный DC. Подчеркну – он не держит каких-то уникальных данных; вы можете назначить на эту роль любой контроллер, главное – что все указанные операции пойдут через него, и это уберёт целый пласт ситуаций вида “у нас большой лес, и два администратора в двух удалённых сайтах одновременно добавляют домен с одинаковым названием” – все такие операции на уровне леса будут реализовываться строго последовательно через владельца роли Domain Naming Master, поэтому конфликта в приведённом случае не будет – просто вторая попытка добавить домен с названием, которое уже есть, будет неудачной.

## Как Domain Naming Master задействуется при переименовании домена

---

Для переименования домена используется утилита `rendom.exe`, доступная с Windows Server 2003. В ходе работы утилита [rendom.exe](#) обращается к FSMO Domain Naming Master по следующим поводам:

- Создавая XML-файл `DomainList.XML` на основании информации о доменных NC;
- Забирая этот файл после редактирования, а после – помещая созданный на основании данных из файла скрипт в атрибут **msDS-UpdateScript** объекта **CN=Partitions**;
- А также помещая новое имя каждого задействованного в переименовании раздела (не забывайте, у домена – одна domain partition, но вы можете насоздавать кучу application partition, привязанных к имени конкретного домена, поэтому переименование домена в лесу может повлечь за собой изменение DN у нескольких NC) в атрибут **msDS-DnsRootAlias**, который есть у каждого объекта класса **crossRef**;

Детально процесс переименования домена вы можете посмотреть по ссылке на утилиту – нас интересует именно то, как владелец данной FSMO-роли задействуется при этой операции – и хорошо видно, что без него – никуда.

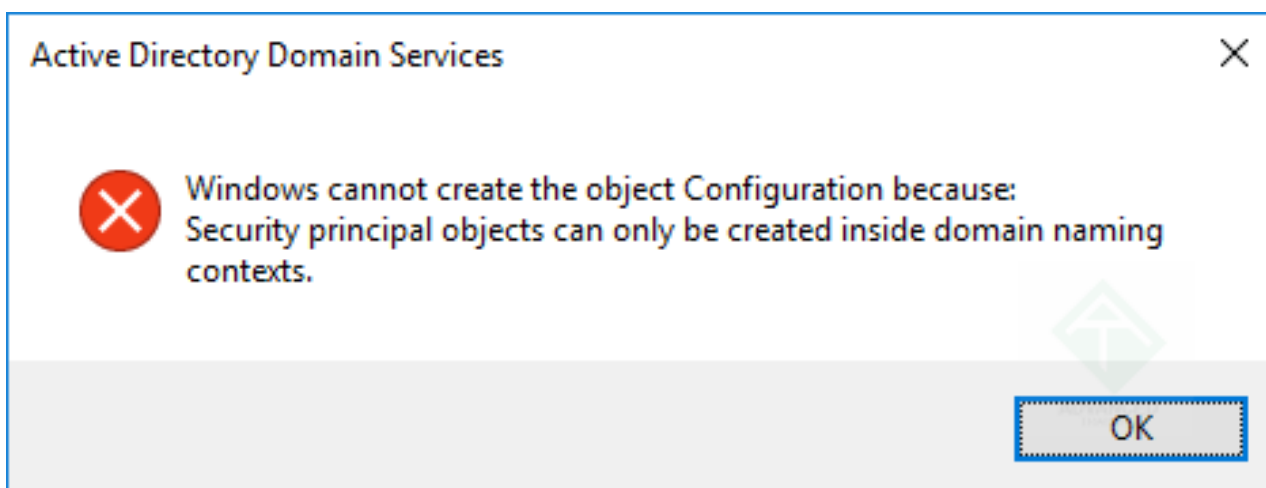
Посмотрим теперь, что из похожих операций не относится к власти Domain Naming Master’a.

## Что не является зоной ответственности Domain Naming Master

---

Хочется остановиться на том, что, несмотря на вроде бы очевидные задачи по предотвращению конфликтов, Domain Naming Master нужен не для всех подобных “конфликтных” случаев.

Например, как видно из содержимого **CN=Partitions**, раздел Configuration, несмотря на то что называется разделом (т.е. DC), является контейнером (CN=Configuration) в доменном разделе forest root domain'a. То, что он вложен, как матрёшка, а не “параллелен” (как допустим DC=DomainDNSZones, который является полноценным разделом), доказывается легко – попробуйте создать в корне forest root domain'a объект с CN=Configuration (например, компьютер или пользователя). Результат будет таким:



[Объект с именем CN=Configuration создать в корне forest root domain не получается \(кликните для увеличения до 468 px на 178 px\)](#)

NTDS немножечко лукавит, ссылаясь на то, что нельзя создавать security principal'ов вне доменных контекстов (мы там и создаём) – но попробуйте создать объект с именем, допустим, CN=Configuration2, и всё волшебным образом получится.

Это наглядно показывает, что есть полновесные контексты, DN которых целиком выглядит как DC=context,DC=domain,DC=... – они у доменов леса или создаются вручную как application partition – и специфичные “общелесные” **CN=Configuration** и **CN=Schema**, которые по сути – виртуальные контейнеры в NC первого домена леса. Различаться в плане функционала они будут минимально – у полноценных контекстов можно будет выставлять доп.параметры (время репликации, тайм-ауты), а у указанных – нет, они всегда будут реплицироваться по всему лесу, без вариантов – это и логично, потому что странно представить себе контроллер домена, который обходится без схемы леса или конфигурации.

И эту ситуацию, хоть она и про конфликт на уровне naming context'ов, отрабатывает не Domain Naming Master, а контроллер, на котором проводится эта операция.

ОК, теоретическая часть понятна – посмотрим, когда нам этот уникальный агрегат по имени DNM вообще будет нужен.

## Пример задач, при которых задействуется Domain Naming Master

---

Задача первая и самая простая – добавление нового домена в лес.

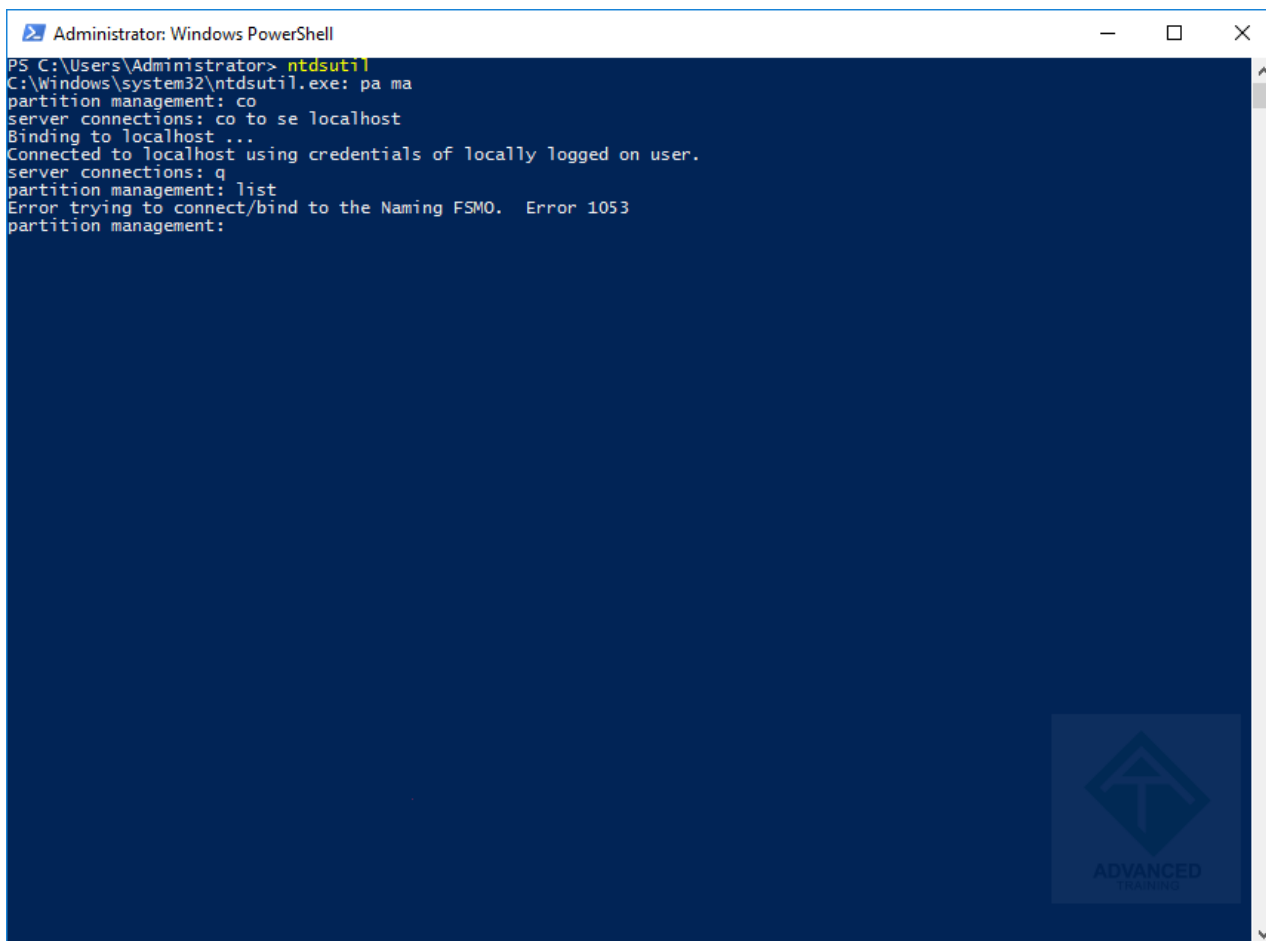
Domain Naming Master будет внимательно смотреть на то, чтобы новый домен не дублировал своим NBname-именем существующий. Понятное дело, что если вы попытаетесь добавить домен с идентичным уже существующему FQDN, то это не выйдет – однако с NBname ситуация интереснее. Дело в том, что при создании домена вы можете выбрать ему NBname, который никак не связан с его FQDN. То есть домен может иметь FQDN вида `lab.atraining.local`, а NETBIOS-имя – `CHINABRANCH`. Поэтому Domain Naming Master, получив запрос на добавление нового домена, проверит на существование и FQDN, и NBName. Это нужно, потому что возможна ситуация вида:

- В лесу есть домен с FQDN = `child.atraining.local` и NBname = `CHILD`;
- Мы пробуем добавить домен с FQDN = `child.hq.atraining.local` и NBname = `CHILD`;
- FQDN у них разные – но NBname одинаковые, поэтому такой домен добавить не получится;

Эта проверка нужна для предотвращения множества конфликтов, самый наглядный из которых – это окно входа на старенькой Windows XP, подключенной к домену – когда в третьей строке выпадающий список “в какой домен вы хотите попасть”. В случае, если в лесу Active Directory будут домены с не-уникальным названием, в этом списке придётся вывести одинаковые имена доменов, что сделает процедуру входа без UPN крайне затруднительной.

Задача вторая – добавление новой application partition и изменение её настроек.

Эти изменения (чтобы их сделать, я запущу утилиту `ntdsutil`, зайду в раздел `partition management`, зайду в настройки подключения к серверу `connections`, подключусь к текущему серверу командой `connect to server localhost`, и вернусь назад в управление разделами / naming context’ами через команду `quit`) будут проводиться только при живом Domain Naming Master’е – если он будет неактивен, то будет такая вот грустная картинка:



```
Administrator: Windows PowerShell
PS C:\Users\Administrator> ntdsutil
C:\Windows\system32\ntdsutil.exe: pa ma
partition management: co
server connections: co to se localhost
Binding to localhost ...
Connected to localhost using credentials of locally logged on user.
server connections: q
partition management: list
Error trying to connect/bind to the Naming FSMO. Error 1053
partition management:
```

[Без работающего Domain Naming Master создать новую partition в лесу Active Directory не выйдет](#)  
([кликните для увеличения до 859 px на 634 px](#))

Я попробовал всего лишь вывести список naming context'ов – казалось бы, это не запись, а всего лишь чтение, и той информации, которая есть в виде реплики на каждом DC в лесу Active Directory – но все равно, для этой операции, чтобы администратор видел самые свежие данные, без оглядки на возможные задержки при репликации, обращение идёт сразу к Domain Naming Master'у – так система страхует от ситуации “другой админ только что создал раздел с таким же именем, подключившись к Domain Naming Master'у, а у вас плохая связь и вы решили, раз уж всего лишь список разделов запрашивается, то можно и с местного DC показать”.

Как только связь с Domain Naming Master'ом восстановится, я смогу и создать раздел, и изменить его параметры.

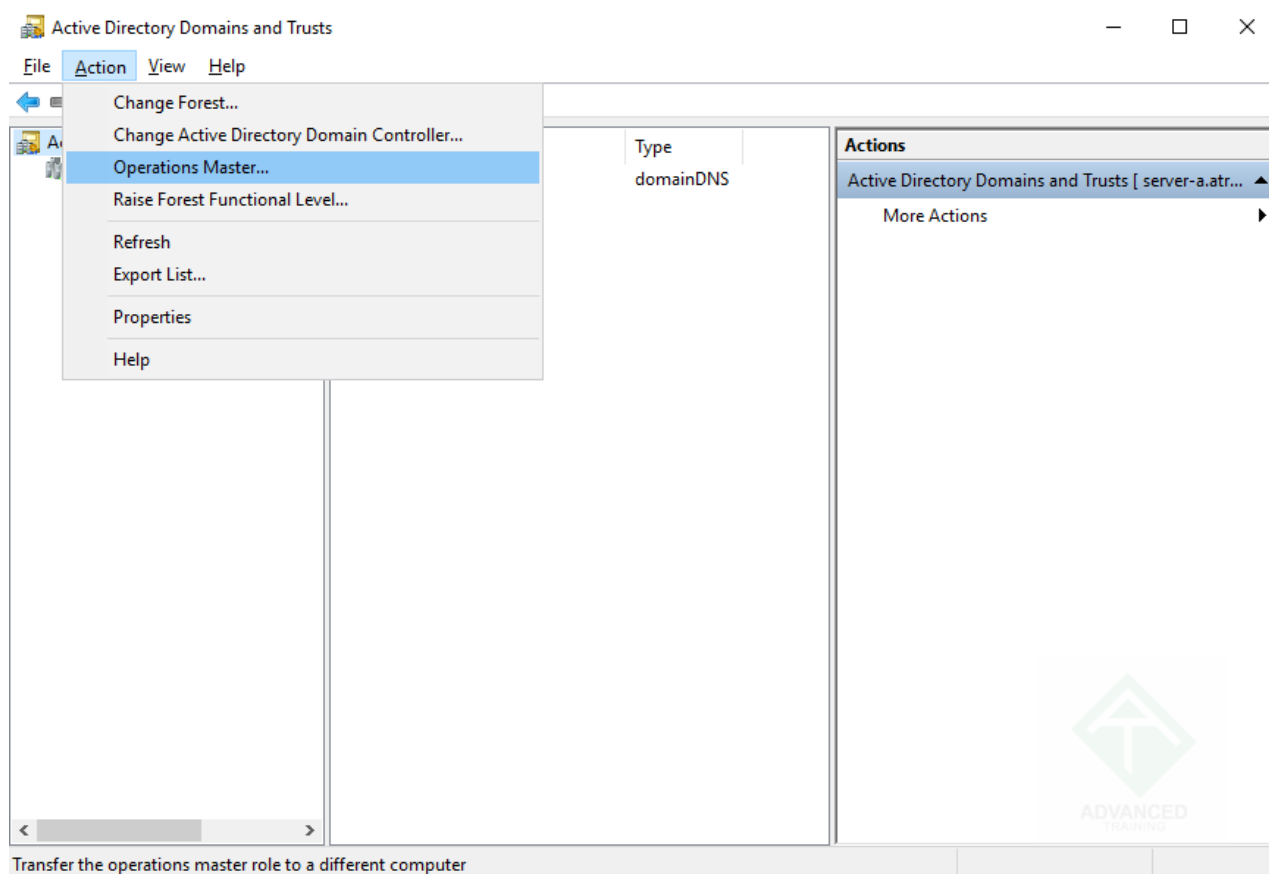
Что ж, функционал роли кончился – теперь, по традиции, поговорим об эксплуатации в рабочей среде и разберём мифы.

## Как перенести владельца FSMO-роли Domain Naming Master?

Изначально Domain Naming Master'ом назначается первый DC в первом домене леса – это штатно изменяемо как утилитой ntdsutil, так и через оснастку Active Directory Domains and Trusts. Это просто – подключаетесь оснасткой к тому DC, на



который собираетесь переносить роль, и вызываете в меню Action пункт Operations Master:

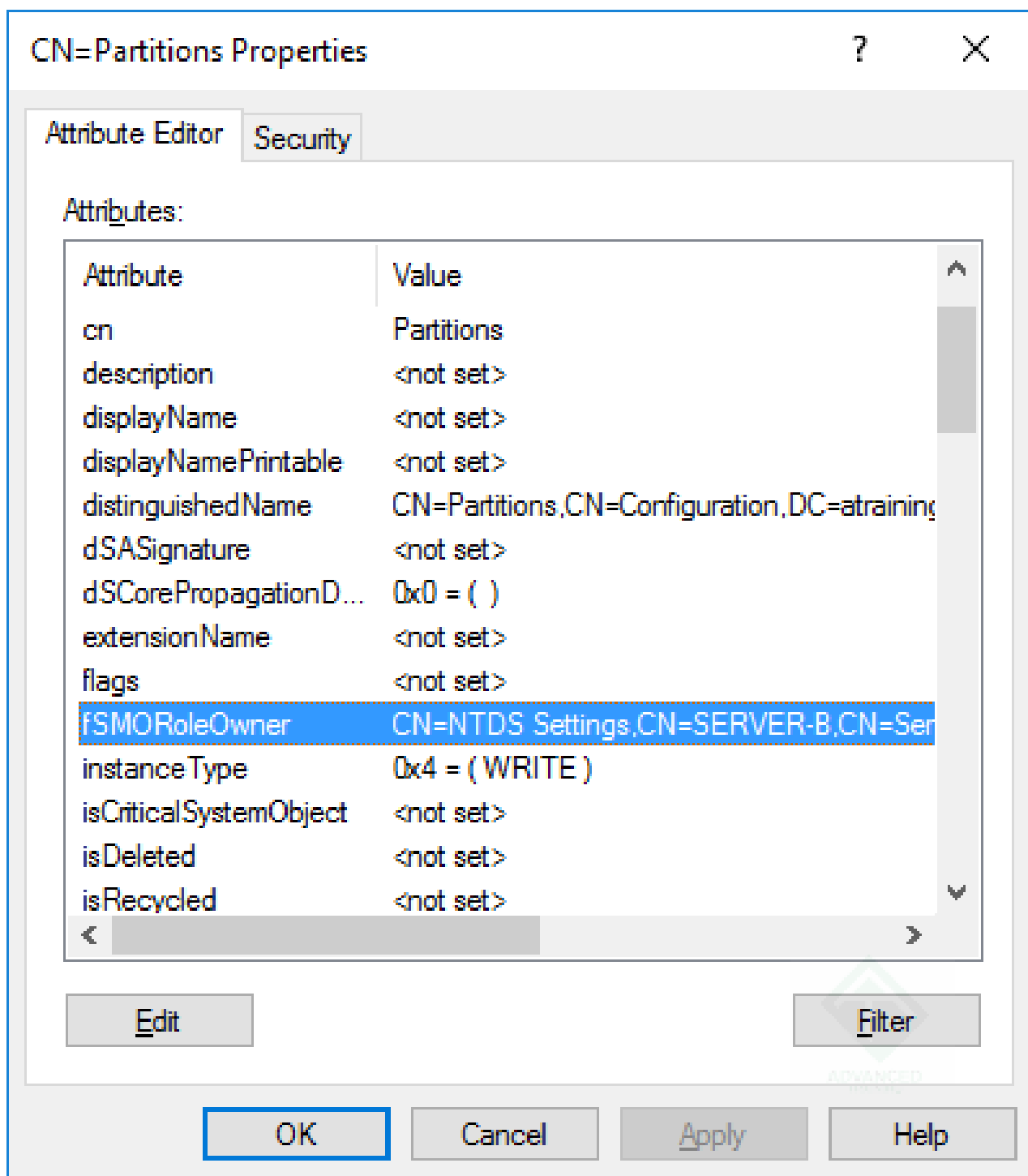


[Перенос роли Domain Naming Master](#)

[\(кликните для увеличения до 866 px на 596 px\)](#)

## Где хранится информация, кто сейчас Domain Naming Master?

Данные о том, кто сейчас в домене держит FSMO-роль Domain Naming Master, содержатся в атрибуте объекта **Partitions** – **fSMORoleOwner**:



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

Этот атрибут отображается в более удобном виде и в пункте меню Operations Master, и в ntdsutil.

## Где располагать в лесу AD владельца FSMO-роли Domain Naming Master?

Имеет смысл назначить Domain Naming Master'ом тот контроллер, который находится в центре топологии (с точки зрения скорости доступа). Также замечу, что ранее, в Windows 2000 Server, было требование, чтобы владелец FSMO-роли

Domain Naming Master всегда был GC. Начиная с Windows Server 2003 это требование не критично.

## Как повысить надёжность работы Domain Naming Master?

---

Именно повысить её – трудно, т.к. он и так надёжно работает – сервис, как видно из функционала, несложный. Постарайтесь, чтобы владелец FSMO-роли Domain Naming Master был минимально загружен, т.е. не занимался авторизацией пользователей и раздачей им настроек group policy. В очень крупных инсталляциях с высокими требованиями по надёжности можно создать специфичный сайт с двумя DC, один из которых будет Domain Naming Master, а другой будет служить точкой “быстрых внутрисайтовых репликаций” и явно заданным bridgehead’ом. К этому сайту надо будет привязать сеть, в которой не будет пользовательских машин, а у контроллера можно убрать GC. В этом сценарии Domain Naming Master будет исключительно отрабатывать свои задачи, его никто не будет беспокоить, при любом изменении он мгновенно отдаст реплику соседу по сайту, а тот уже раздаст по другим сайтам, опять же никак не нагружая DNM’a. При форс-мажоре роль можно мгновенно поднять на соседе.

## “Если Domain Naming Master упал то всё”

---

Хорошо представляя себе задачи данной роли, прекрасно понятно, что её наличие или отсутствие влияет на единичные и очень редкие операции. Так как никаких данных на себе владелец этой роли не хранит, то его исчезновение вообще ни на что не повлияет – просто назначите другой контроллер на эту роль, да и всё. Никаких “вот теперь Domain Naming Master упал и все домены в лесу начнут подглючивать” не бывает, потому что всё, что делает этот FSMO – это выдача разрешения на разовые операции “можно добавить домен с таким именем, да”. Он не поддерживает какую-то постоянную телепатическую связь со всеми доменами леса. Не надо прибегать к суициду и переставлять Active Directory.

## Как часто надо переносить FSMO-роль Domain Naming Master и куда?

---

По best practice от Microsoft, эту роль надо совмещать со второй лесной ролью – [Schema Master](#). По сути, обе они должны быть на некоем “очень устойчивом, не загруженном обработкой групповых политик и логинов пользователей, и находящимся в центре топологии леса Active Directory” контроллере. Просто так перетаскивать эту роль не имеет смысла, она используется в штучных редких операциях, скорость которых не критична (возможно, где-то существует лес Active Directory, куда спешно добавляют новые домены, но даже в этой ситуации таскать туда-сюда владельца роли Domain Naming Master не надо.

## Зависит ли скорость работы доменов и репликация от расположения Domain Naming Master?

---

Нет.

## Нужен ли Domain Naming Master'у отдельный бэкап?

---

Нечего бэкапить, содержимое контейнера **CN=Partitions** идентично на всех контроллерах леса, у владельца FSMO-роли нет локально хранимой уникальной информации.

## В заключение

---

Эта простая, но необходимая для понимания функциональности FSMO-роль разобрана – на очереди другие. Active Directory – это просто и интересно. Если видите тех, кто мистифицирует эту тему – знайте, что это просто плохо знающие матчасть люди.

До встречи!