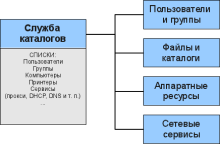
# Управление схемой службы каталогов

## Служба каталогов

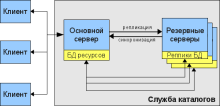
### Описание службы каталогов

**Служба каталогов** — это сетевой сервис, представляющий централизованные средства управления ресурсами автоматизированной системы. Под ресурсами подразумеваются все компоненты сетевой инфраструктуры, которые используются для выполнения функций АСУ: пользователи, файлы и каталоги, устройства, сетевые сервисы и т.д. (рис.1).

**

*Рис. 1. Служба каталогов*

Как правило, служба каталогов состоит из базы данных, в которой размещены сведения о сетевых ресурсах и серверного ПО, представляющего механизмы доступа к этой базе. Как база данных сервиса каталогов, так и ее управляющая программа могут быть распределены на несколько серверов (рис. 2).

**

*Рис. 2. Распределенная служба каталогов*

Основными функциями службы каталогов являются следующие:

* Управление пользователями и группами (создание/удаление, настройка прав доступа).
* Управление ресурсами (представление в общий доступ, установка ограничений, удаленное администрирование и т.п.).
* Разграничение прав доступа (как правило, на уровне пользователей, групп и отдельных ресурсов).

Среди дополнительных функций сервиса каталогов можно указать, например, такие:

* поиск ресурсов;
* распространение сетевых политик;
* интеграция с другими сервисами.

**Сетевая политика** — совокупность правил, определяющих методы и средства взаимодействия с общими ресурсами в корпоративной сети.

Сетевые ресурсы в службе каталогов обычно представлены в виде иерархической структуры. Такой способ наиболее близок к реальной организационной модели подавляющего большинства предприятий и организаций. Корень иерархии описывает предприятие в целом, нижележащие уровни — подразделения и отдельные элементы. Для единообразного обращения к любому элементу иерархии протокол взаимодействия представляет унифицированную схему адресации — либо собственную, либо совместимую со стандартными схемами.

Технология Active Directory (AD) является службой каталогов, созданной компанией Microsoft. Служба каталогов содержит данные в организованном формате и предоставляет к ним упорядоченный доступ. Служба Active Directory — это не изобретение компании Microsoft, а реализация существующей индустриальной модели (а именно X.500), коммуникационного протокола (LDAP — Lightweight Directory Access Protocol) и технологии поиска данных (службы DNS).

Active Directory предназначен для хранения информации о всех сетевых ресурсах. Клиенты имеют возможность отправлять запросы Active Directory для получения информации о любом объекте сети. В список возможностей Active Directory входят следующие функции.

* Безопасное хранилище данных. Каждый объект в Active Directory имеет собственный список управления доступом (ACL), который содержит список ресурсов, получивших право доступа к объекту, а также предопределенный уровень доступа к этому объекту.
* Многофункциональный механизм запросов, основанный на созданном Active Directory глобальном каталоге (GC). Все клиенты, поддерживающие Active Directory, могут обращаться к этому каталогу.
* Репликация данных каталога на все контроллеры домена упрощает доступ к информации, повышает степень ее доступности и увеличивает надежность всей службы.
* Концепция модульного расширения, которая позволяет добавлять новые типы объектов или дополнять существующие объекты. Например, к объекту “пользователь” можно добавить атрибут “зарплата”.
* Сетевое взаимодействие с использованием нескольких протоколов. Служба Active Directory основана на модели X.500, благодаря чему поддерживаются различные сетевые протоколы, например, LDAP 2, LDAP 3 и HTTP.
* Для реализации службы имен контроллеров доменов и поиска сетевых адресов вместо NetBIOS используется служба DNS.

Информация каталога распределяется по всему домену, тем самым позволяя избежать чрезмерного дублирования данных.

Хотя Active Directory распределяет информацию каталога по различным хранилищам, пользователи имеют возможность запросить Active Directory на получение информации о других доменах. Глобальный каталог содержит сведения о всех объектах леса предприятия, помогая осуществлять поиск данных в рамках всего леса.

При запуске утилиты DCPROMO (программы повышения обычного сервера до контроллера домена) на компьютере под управлением Windows для создания нового домена, утилита создает домен на сервере DNS. Затем клиент связывается с сервером DNS для получения информации о своем домене. Сервер DNS предоставляет информацию не только о домене, но и о ближайшем контроллере домена. Клиентская система, в свою очередь, подключается к базе данных домена Active Directory на ближайшем контроллере домена с целью нахождения необходимых объектов (принтеров, файловых серверов, пользователей, групп, организационных подразделений), входящих в домен. Поскольку каждый контроллер домена хранит ссылки на другие домены в дереве, клиент может выполнять поиск во всем дереве домена.

Контролер домена — это компьютер с поднятой службой Active Directory, который управляет доступом пользователей к ресурсам сети. Он хранит ресурсы (например, принтеры, папки с общим доступом), службы (например, электронная почта), людей (учетные записи пользователей и групп пользователей), компьютеры (учетные записи компьютеров). Число таких сохраненных ресурсов может достигать миллионов объектов. В качестве контролера домена могут выступать следующие версии MS Windows: Windows Server 2000/2003/2008/2012 кроме редакций Web-Edition.

Контролер домена помимо того, что является центром аутентификации сети, также является центром управления всеми компьютерами. Сразу после включения компьютер начинает обращаться к контролеру домена, задолго до появления окна аутентификации, таким образом выполняется аутентификация не только пользователя вводящего логин и пароль но и аутентификация клиентского компьютера.

Разновидность Active Directory, которая перечисляет все объекты в лесу доменов, доступна для тех случаев, когда необходимо найти данные за пределами дерева доменов клиента. Подобная версия называется глобальный каталог. Глобальный каталог можно хранить на любом контроллере домена в лесу AD.

Глобальный каталог предоставляет быстрый доступ к каждому объекту, который располагаться в лесу доменов, но при этом содержит только некоторые параметры объектов. Для получения всех атрибутов следует обратиться к службе Active Directory целевого домена (контроллеру интересующего домена). Глобальный каталог можно настроить на предоставление необходимых свойств объектов.

Для упрощения процесса создания объектов Active Directory контроллер домена содержит копию и иерархию классов для всего леса. Служба Active Directory содержит структуры классов в расширяемой схеме, в которую можно добавить новые классы.

Схема (schema) — это часть конфигурационного пространства имен Windows, которое поддерживается всеми контроллерами доменов в лесу. Конфигурационное пространство имен Windows состоит из нескольких структурных элементов, таких как физическое расположение, сайты Windows и подсети.

Сайт (site) содержится внутри леса и может объединять компьютеры из любого домена, причем все компьютеры сайта должны иметь быстрые и надежные сетевые соединения для резервирования данных контроллера домена.

Подсеть (subnet) — это группа IP-адресов, выделенная сайту. Подсети позволяют ускорить репликацию данных Active Directory между контроллерами доменов.

**Именование объектов**

Active Directory поддерживает несколько способов именования объектов:

* характерное LDAP-имя;
* глобальный уникальный идентификатор (Global Unique Identifier, GUID);
* идентификатор безопасности (Security ID, SID);
* имя NetBIOS;
* основное имя субъекта системы безопасности (Security principal name).

В Active Directory имя объекта может быть представлено в одной из двух форм: *характерное имя* (DN, Distinguished Name), либо *относительное характерное имя* (RDN, Relative Distinguished Name). Характерное имя однозначно определяет объект в дереве. Можно провести аналогию между характерным именем в Active Directory и полным путем в файловой системе. Характерное имя включает информацию обо всех узлах дерева, которые необходимо пройти, чтобы прийти от данного объекта к корню дерева. В записи характерного имени используются сокращения, имеющие следующий смысл:

* **CN (Common Name)** - общее имя;
* **DC (Domain Component)** - составная часть доменного имени;
* **OU (Organizational Unit)** - организационная единица.

В Active Directory не может быть определено несколько объектов, имеющих одинаковое характерное имя, поскольку это привело бы к неоднозначности выбора конкретного объекта.

Большинство утилит, работающих с Active Directory, позволяют использовать другой способ именования объектов - канонические имена. В этом случае опускаются сокращения, обозначающие тип объекта или контейнера, и просто перечисляются их имена. Для обозначения имени домена используется соглашение о доменных именах. В характерном имени объекта присутствует компонент, идентифицирующий непосредственно сам объект, а не содержащие его контейнеры. Этот компонент называется относительным характерным именем объекта. Контейнер, который непосредственно содержит данный объект, называется родителем объекта.

Относительное характерное имя не обязано быть уникальным. Однако в силу того, что каждый объект должен однозначно идентифицироваться в Active Directory, относительное характерное имя объекта должно быть уникально в пределах того контейнера, что является его родителем.

С целью обеспечения уникальности объектов и облегчения их поиска каждому объекту при создании ставится в соответствие 128-разрядное число - *глобальный уникальный идентификатор* (GUID). Глобальный идентификатор является обязательным атрибутом любого объекта, который не может быть изменен ни при каких обстоятельствах. Существует и другая форма именования объектов, которая используется для обеспечения безопасности системы. Эти имена используются при входе пользователя в систему, позволяя провести его аутентификацию. Для этого можно использовать:

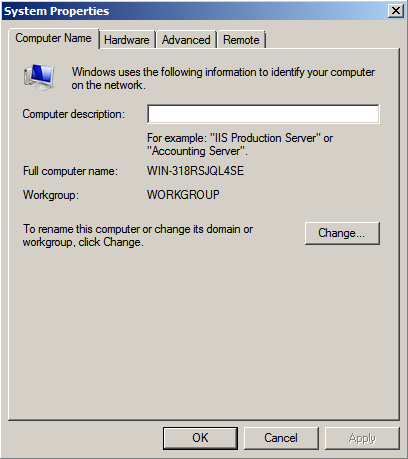
* *имена NetBIOS*.;
* *основное имя пользователя* (user principal names). В Active Directory каждый пользователь имеет основное имя пользователя в формате <пользователь>@< DNS имя домена>. Основное имя образуется из имени пользователя и имени домена, к которому этот пользователь принадлежит.

### Настройка Active Directory

В этом разделе приводится пример установки Active Directory на Windows Server 2012 R2.

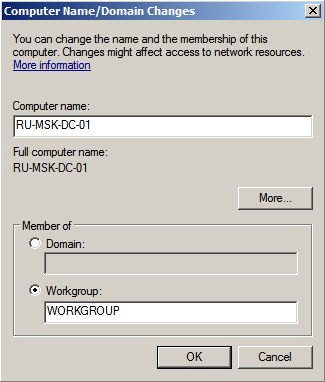
Перед началом установки роли Active Directory Domain Services необходимо присвоить серверу корректное имя, а затем указать статический IP-адрес в настройках сетевого подключения.

В окне «System Properties» на вкладке «Computer Name» нажимаем на кнопку «Change».

[](http://www.vmkh.net/wp-content/uploads/ustanovka-active-directory-domain-services-na-windows-server-2008-r2-3.jpg)

Настоятельно рекомендую заранее продумать, как будут называться сервера в вашей организации.

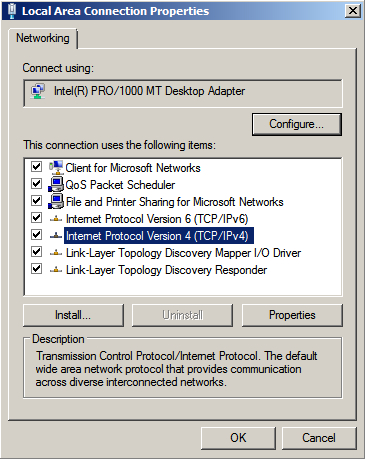
Далее указываем новое имя сервера в поле «Computer Name» и нажимаем на кнопку «OK».

[](http://www.vmkh.net/wp-content/uploads/ustanovka-active-directory-domain-services-na-windows-server-2008-r2-4.jpg)

Система предупредит о том, что для применения новых настроек необходимо перезагрузить сервер.

Нажимаем кнопку «OK». Restart.

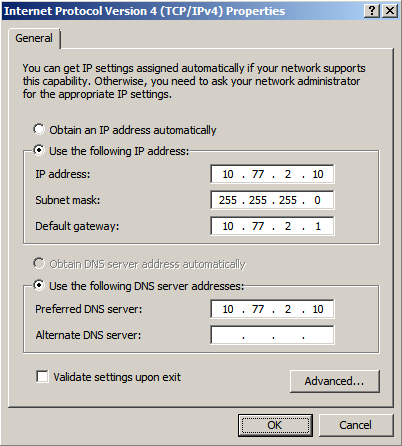
Далее открываем окно «Local Area Connection Properties». Выбираем «Internet Protocol Version 4» и нажимаем на кнопку «Properties».

[](http://www.vmkh.net/wp-content/uploads/ustanovka-active-directory-domain-services-na-windows-server-2008-r2-12.jpg)

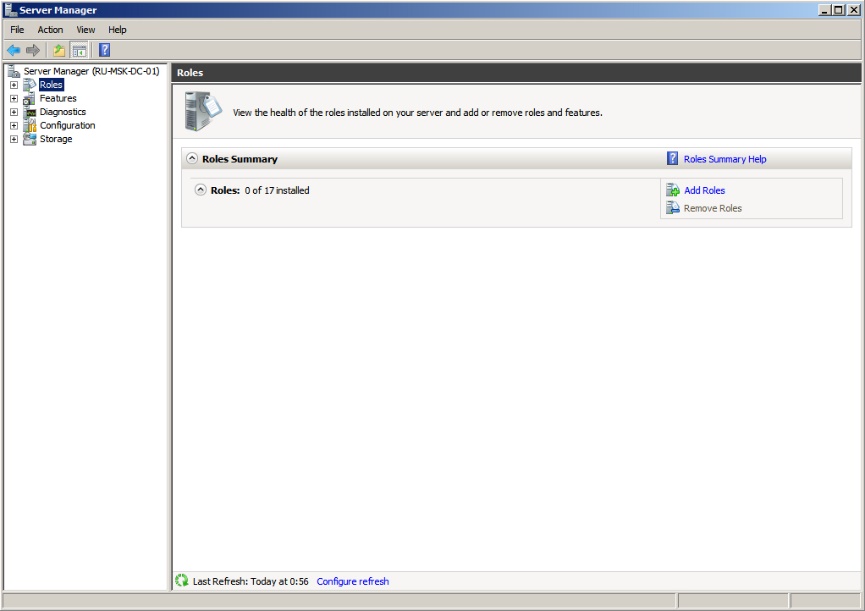
Далее выбираем пункт «Use the following IP address» и указываем свободный IP-адрес, маску подсети и шлюз. Обратите внимание, вы должны заранее понимать, как устроена ваша сеть и знать какие IP-адреса свободны.

В поле «Preferred DNS server» указываем IP-адрес этого сервера, так как на вашем сервере будет присутствовать роль DNS Server, которая устанавливается вместе с ролью «Active Directory Domain Services».

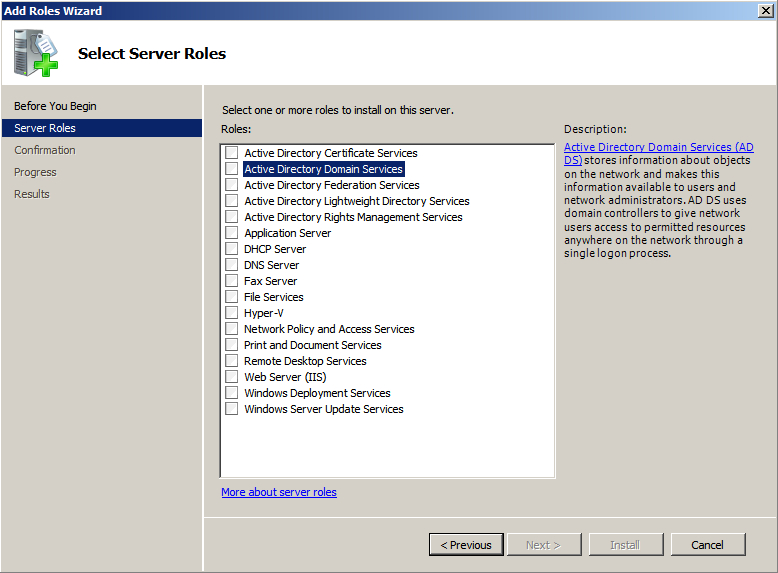
Нажимаем на кнопку «OK».

[](http://www.vmkh.net/wp-content/uploads/ustanovka-active-directory-domain-services-na-windows-server-2008-r2-13.jpg)

Открываем «Server Manager» и в пункте «Roles» выбираем «Add Roles».

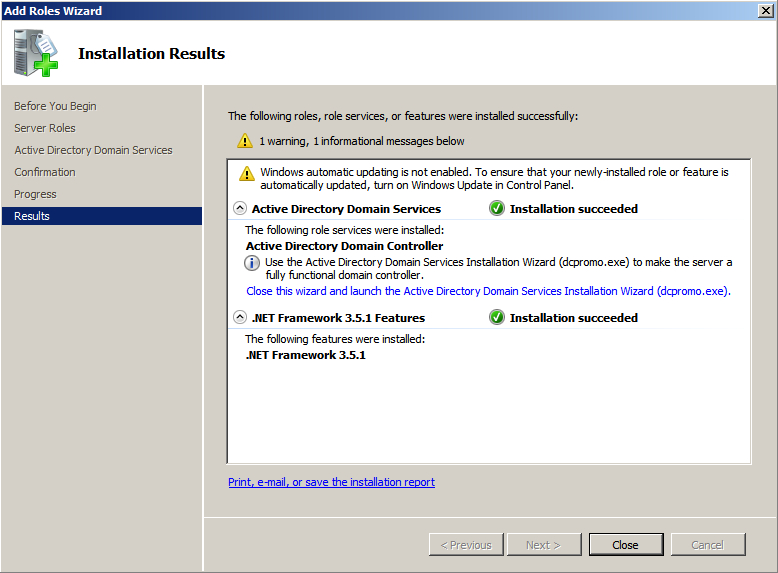
[](http://www.vmkh.net/wp-content/uploads/ustanovka-active-directory-domain-services-na-windows-server-2008-r2-15.jpg)

Выбираем роль «Active Directory Domain Services».

[](http://www.vmkh.net/wp-content/uploads/ustanovka-active-directory-domain-services-na-windows-server-2008-r2-17.jpg)

Далее жмем «Next» и устанавливаем роль.

Теперь нажимаем на кнопку «Close this wizard and launch the Active Directory Domain Services Installation Wizard (dcpromo.exe)», для того чтобы повысить роль вашего сервера до уровня контроллера домена.

[](http://www.vmkh.net/wp-content/uploads/ustanovka-active-directory-domain-services-na-windows-server-2008-r2-23.jpg)

Настоятельно рекомендую заранее продумать какое доменное имя вы будете использовать при добавлении нового леса.

Нажимаем на кнопку «Next».

[](http://www.vmkh.net/wp-content/uploads/ustanovka-active-directory-domain-services-na-windows-server-2008-r2-24.jpg)

Дальее следуем указаниям мастера установки.

## Схема службы каталогов

### 2.2.1.. Описание схемы службы каталогов

Схема (schema) Active Directory представляет собой набор определений всех типов объектов каталога и связанных с ними атрибутов. Схема определяет способ хранения и конфигурирования в AD всех пользователей, компьютеров и других данных объек­тов, стандартизуя всю структуру Active Directory. Схема, защищенная с помощью спи­сков разграничительного контроля доступа (Discretionary Access Control List — DACL), управляет возможными атрибутами каждого объекта из Active Directory. По существу, схема представляет собой базовое определение самого каталога и является основой функциональности среды домена. Делегировать управление схемой следует осторож­но и лишь избранной группе администраторов, поскольку любые изменения схемы влияют на всю среду AD.

**Объекты схемы**

Объекты структуры Active Directory — User (пользователь), Printer (принтер), Computer (компьютер) и Site (сайт) — определены в схеме как объекты. Каждый объ­ект имеет список атрибутов, которые определяют его и могут быть использованы для поиска этого объекта. Например, объект User для работника с именем Вера Сердючка будет иметь атрибут FirstName (имя) Вера и атрибут LastName (фамилия) Сердючка. Кроме этих, можно назначить и другие атрибуты: название подразделения, адрес электронной почты и целый ряд других. Пользователи, выполняя поиск информации в Active Directory, могут делать запросы на основе этой информации, например, искать всех пользователей, работающих в отделе сбыта. Представление о том, сколько атрибутов имеется в Active Directory, дает тот факт, что сразу после инсталляции каж­дому объекту можно присвоить более 1000 атрибутов.

**Расширение схемы**

Одним из основных достоинств структуры Active Directory является возможность непосредственной модификации и расширения схемы для включения пользователь­ских атрибутов. Обычное расширение атрибутов происходит во время инсталляции последней версии Microsoft Exchange, которая расширяет схему, увеличивая ее размер в два раза. Модернизация Active Directory Windows 2000 до Active Directory Windows Server 2003 также расширяет схему, включая в нее атрибуты, специфические для Win­dows Server 2003.

Схема Active Directory расположена в контейнере с характерным именем (distinguished name) «CN=Schema,CN=Configuration,DC=<ForestRoot>»

**Модификация схемы с помощью интерфейса службы Active Directory**

Есть интересный метод преставления всех подробностей схемы Active Directory — с помощью утилиты интерфейса службы Active Directory (Active Directory Service Interfaces — ADSI). Эта утилита предназначена для упрощения доступа к Active Directory, и с ее помощью можно просмотреть любой внешний совместимый каталог LDAP. Ути­лита ADSI, позволяет просматривать, удалять и модифицировать атрибуты схемы. При модифицировании схемы необходимо соблюдать особую осто­рожность, поскольку проблемы, возникающие в схеме, исправлять очень трудно.

Для модификации схемы необходимо выполнение следующих условий:

* 1. Пользователь должен входить в группу «**Администраторы схемы**».
  2. Необходимо получить идентификатор для нового объекта (**object identifier – OID**). OID – это уникальный номер создаваемого объекта, который выделяет его в Active Directory. Для получения номера можно воспользоваться скриптом от Microsoft.

Каждый класс AD и каждый атрибут должны быть уникальны, поэтому каждому определению класса и атрибута нужно назначать уникальный идентификатор OID. Идентификатор представляет собой строку из десятичных чисел, разделенных точками - так же, как в IP-адресах. OID объекта classSchema помещается в его атрибуте governsID, а идентификатор OID объекта attributeSchema - в атрибуте attributeID.

Эти идентификаторы назначаются центром регистрации имен (Name Registration Authority) Международной организации по стандартизации (International Organization for Standardizatin, ISO).

Используемая в идентификаторах OID цифровая система обозначений имеет древовидную структуру. За отдельные участки дерева именования отвечают конкретные инстанции с четко разграниченными полномочиями, которые могут выделять "подветви" (arcs) другим инстанциям более низкого уровня, а те, в свою очередь, выделять "подветви подветвей" инстанциям еще более низкого уровня.

Кстати, не путайте идентификаторы OID с глобальными уникальными идентификаторами типа GUID (globally unique ID). OID содержит уникальное обозначение класса, а GUID - уникальное обозначение экземпляра объекта. Например, OID класса user - это 1.2.840.113556.1.5.9; каждый объект User, создаваемый из этого класса, будет иметь уникальный идентификатор GID, но являться потомком одного и того же класса (т.е. класса, OID которого равен 1.2.840.113556.1.5.9).

### 2.2.2. Классы

Каждый объект AD в контексте именования домена или в контексте именования конфигурации представляет собой пример, или экземпляр объектного класса (object class). Так, объект User представляет собой экземпляр класса user; а объект Computer, соответственно, экземпляр класса computer. Класс описывает объект AD и ассоциированные с ним свойства, или атрибуты, необходимые для формирования экземпляра объекта данного класса (т.е. обязательные атрибуты), либо те, которые могут присутствовать в экземпляре класса (т.е. факультативные атрибуты). Контекст именования схемы содержит определения всех классов и атрибутов, имеющихся в других контекстах именования AD. Но дело осложняется тем, что служба AD предусматривает хранение этих определений, которые в данном случае именуются объектами.

Таким образом, следует постоянно иметь в виду различие между объектом, который является экземпляром класса (это может быть, скажем, экземпляр объекта контекста именования конфигурации или контекста именования домена), и объектом, который является определением класса или атрибута (это может быть объект схемы в контексте именования схемы). Объекты схемы делятся на две категории: объекты classSchema определяют классы, а объекты attributeSchema - атрибуты. Каждый атрибут имеет лишь одно определение, но может использоваться с несколькими различными классами.

Представьте себе, что вы создаете объект User с именем James Bond. Этот объект является экземпляром класса user, и контекст именования схемы определяет его с помощью особого объекта classSchema, именуемого user. Служба AD сохраняет регистрационное имя объекта James Bond (пусть это будет имя BondJ), в атрибуте, именуемом SAMAccountName, определяемом контекстом именования схемы через особый объект attributeSchema, называемый SAMAccountName. Объект user категории classSchema определяет связь между объектом SAMAccountName и классом user. Вернее, объект SAMAccoutName представляет собой имя объекта attributeSchema, отображаемое при использовании протокола Lightweight Directory Access Protocol, или LDAP. Фактическое же имя объекта хранится в его атрибуте common name (cn) именуемом SAM-Account-Name.

#### Определение объектов: объекты категории classSchema

Объекты контекста именования схемы, относящиеся к категории classSchema, содержат определения классов. В этих определениях заключены следующие сведения:

правила, определяющие, где можно создавать объекты AD (например, в организационной единице (organizational unit, или OU), взаимосвязи, или производные классы, между объектами classchema (например, родительский класс данного класса, именуемый суперклассом (superclass)), отношения между атрибутами и объектами категории classSchema, в частности, обязательные и факультативные атрибуты класса.

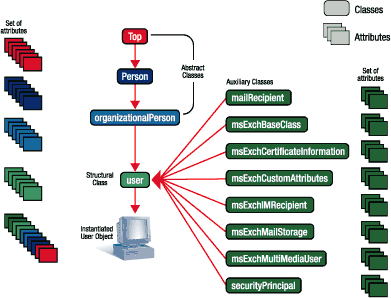
Каждый объект classSchema относится к одной из трех категорий классов: абстрактная (abstract), вспомогательная (auxiliary) и структурная (structural).

**Абстрактные классы.** Объекты classSchema (иначе говоря, классы), относящиеся к абстрактной категории, выступают в качестве формальных параметров-заполнителей; эти классы нельзя использовать для создания экземпляра объекта в другом контексте именования AD. В службе AD абстрактные классы используются главным образом как родительские классы других классов, или суперклассы, так что администратор может создавать иерархическую структуру из объектных классов и определять систему наследования атрибутов между классами (в атрибуте subClasOf каждого класса хранится имя его суперкласса). Хотя наследование и не является прерогативой абстрактных классов, это именно то качество, из-за которого данные классы используются. Значение атрибута objectClassCategory абстрактных объектов classSchema равно 2.

**Вспомогательный класс.** Вспомогательные объекты classSchema подобны абстрактным классам в том, что они тоже не используются для создания экземпляров объектов AD. Вспомогательные классы напоминают файлы include и заголовочные файлы; с помощью этой категории классов можно включать набор из нескольких атрибутов в определение другого класса. Атрибуты systemAuxiliaryClass и auxiliaryClass являются атрибутами, которые способны содержать несколько значений (multivalue attributes). Иными словами, одному объекту classSchema можно поставить в соответствие множество вспомогательных классов. Атрибут с одним значением (single-value attribute) может содержать только одно значение, тогда как атрибут со множеством значений может содержать несколько значений.

При установке пакета Exchange 2000 в лесу AD программа установки Exchange 2000 Setup дополняет схему AD множеством новых классов. Некоторые из них относятся к категории вспомогательных классов и определяют атрибуты, которые используются только в Exchange 2000. Принадлежащие объекту classSchema атрибуты systemAuxiliaryClass или auxiliaryClass определяют объекты любого применимого вспомогательного класса. Значение атрибута objectClassCategory вспомогательных классов равно 3.

**Структурный класс.** Это единственный тип класса, с помощью которого можно создавать экземпляры объектов в других контекстах именования службы AD. Определения классов, внутри которых могут располагаться созданные экземпляры класса, определяемого объектами classSchema, содержатся в атрибутах systemPossSuperiors и possSuperiors. Так, принадлежащий классу user атрибут systemPossSuperiors определяет класс organizationalUnit в качестве возможного контейнера для экземпляров класса user, поэтому объект User можно создавать в экземпляре класса organizationalUnit. Эти определения известны как правила реализации (instantiation rules). Значение атрибута objectClassCategory структурных объектов classSchema равно 3.



На Рисунке  показано, как взаимодействуют три описанных типа классов. Вверху расположен абстрактный объект категории classSchema, который является суперклассом по отношению к абстрактному объекту Person категории classSchema. Person, в свою очередь, выступает в роли суперкласса по отношению к абстрактному объекту organizationaPerson, принадлежащему категории classSchema. Последний объект является суперклассом класса user (т.е. объекта user категории clssSchema). Наконец, объект user категории classSchema представляет собой структурный объект класса, определяющий объекты User (иными словами, экземпляры класса user).

Класс user наследует атрибуты своих суперклассов (см. Рисунок), равно как и различных вспомогательных классов. Каждый объект имеет набор обязательных атрибутов и набор факультативных атрибутов. Атрибуты systemMustContain и mustContain объекта категории classScema определяют его обязательные атрибуты, тогда как в атрибутах systemMayContain и mayContain объекта категории classSchema записаны факультативные атрибуты данного объекта. Оснастка ADSI Edit не дает возможности увидеть определенное значение для атрибутов systemMustContain и mustContain класса user. Это не означает, что у класса user нет обязательных атрибутов. Класс user наследует обязательные атрибуты своих суперклассов (скажем, класса Person) и вспомогательных классов (например, класса mailRecipient).

Ниже перечислены ключевые атрибуты класса «classSchema».

|  |  |
| --- | --- |
| **lDAPDisplayName** | **Описание** |
| **cn** | У каждого объекта есть атрибут именования, из которого формируется относительное характерное имя (RDN). У класса classSchema это атрибут «cn» (Common-Name). |
| **lDAPDisplayName** | Имя, используемое LDAP клиентами, чтобы обращаться к классу. Это имя должно быть уникальным. |
| **schemaIDGUID** | Один из уникальных идентификаторов класса. Сервер сам генерирует значение этого атрибута, если он не был указан при создании класса или атрибута. |
| **adminDisplayName** | Имя, отображаемое в средствах администрирования. |
| **governsID** | Уникальный идентификатор класса. |
| **rDnAttId** | Определяет атрибут именования. Рекомендуется использовать «cn» в качестве значения этого атрибута. |
| **mustContain**,**systemMustContain** | Обязательные атрибуты. Могут быть заданы только при создании класса. |
| **mayContain**,**systemMayContain** | Необязательные атрибуты (возможные). |
| **possSuperiors**,**systemPossSuperiors** | Атрибуты, которые определяют какие классы могут быть законными родителями экземпляров данного класса. |
| **objectClassCategory** | Номер, указывающий тип класса:  Структурный – можно создавать экземпляры.  Абстрактный – класс предоставляет базовое определение для дочерних структурных классов.  Вспомогательный – может расширять другие типы классов. |
| **subClassOf** | Атрибут определяет родительский класс. |
| **auxiliaryClass**,**systemAuxiliaryClass** | Вспомогательные классы. |
| **defaultObjectCategory** | Устанавливается характерное имя суперкласса. |
| **defaultHidingValue** | TRUE, чтобы скрыть ненужный класс в средствах администрирования. |
| **systemFlags** | Числовое значение, которое содержит флаг для дополнительных свойств класса. |
| **systemOnly** | Логическое значение. При TRUE класс может изменять только система. |
| **defaultSecurityDescriptor** | Задает дескриптор безопасности по умолчанию для новых объектов данного класса |
| **isDefunct** | True – класс активный, false – класс деактивирован. |
| **description** | Описание класса. |
| **objectClass** | Определяет экземпляром какого класса является данный класс. |

### 2.2.3. Атрибуты

Объекты attributeSchema контекста именования схемы определяют атрибуты, предусмотренные в службе AD. Так же, как и объекты категории classSchema, все объекты типа attributeSchema имеют наборы атрибутов, в которых определены их характеристики.

Каждый атрибут схемы AD имеет заранее заданный синтаксис, определяющий тип информации, который может храниться в данном атрибуте. Например, телефонные номера хранятся в атрибутах иначе, нежели двоичные значения или строки. Синтаксис атрибута atributeSyntax объекта attributeSchema определяется с помощью идентификатора объектов Object Identifier (OID), а для определения синтаксиса атрибута oMSyntax того же объекта применяется обозначение объектной модели X/Open (X/Open Object Model). Атрибут объекта oMObjectClass тоже описывает его синтаксис, правда, в другой форме.

Значение атрибута oMSyntax всегда равно 127 (это значение свидетельствует о том, что синтаксис указывает на тип данных «объект»). Необходимо следить за тем, чтобы атрибут oMObjectClass объекта отображал корректное значение для избранного синтаксиса.

Ниже перечислены ключевые атрибуты класса «attributeSchema».

|  |  |
| --- | --- |
| **lDAPDisplayName** | **Description** |
| **cn** | Атрибут контекста именования. |
| [**lDAPDisplayName**](https://msdn.microsoft.com/en-us/library/windows/desktop/ms676828(v=vs.85).aspx) | Имя, используемое LDAP клиентами, чтобы обращаться к классу. Это имя должно быть уникальным. |
| [**schemaIDGUID**](https://msdn.microsoft.com/en-us/library/windows/desktop/ms679645(v=vs.85).aspx) | Один из уникальных идентификаторов класса. Сервер сам генерирует значение этого атрибута, если он не был указан при создании класса или атрибута. |
| [**adminDisplayName**](https://msdn.microsoft.com/en-us/library/windows/desktop/ms675214(v=vs.85).aspx) | Имя, отображаемое в средствах администрирования. |
| [**attributeID**](https://msdn.microsoft.com/en-us/library/windows/desktop/ms675234(v=vs.85).aspx) | Уникальный идентификатор атрибута. |
| [**attributeSecurityGUID**](https://msdn.microsoft.com/en-us/library/windows/desktop/ms675235(v=vs.85).aspx) | Атрибут для управления доступом. |
| [**attributeSyntax**](https://msdn.microsoft.com/en-us/library/windows/desktop/ms675236(v=vs.85).aspx) | Синтаксис атрибута. |
| [**oMSyntax**](https://msdn.microsoft.com/en-us/library/windows/desktop/ms679072(v=vs.85).aspx) | XDS представление синтаксиса атрибута. |
| [**oMObjectClass**](https://msdn.microsoft.com/en-us/library/windows/desktop/ms679071(v=vs.85).aspx) | Если oMSyntax равен 127, используется этот атрибут для определения объекта по умолчанию. |
| [**rangeLower**](https://msdn.microsoft.com/en-us/library/windows/desktop/ms679437(v=vs.85).aspx),[**rangeUpper**](https://msdn.microsoft.com/en-us/library/windows/desktop/ms679438(v=vs.85).aspx) | Определяют границы значения атрибута. |
| [**isSingleValued**](https://msdn.microsoft.com/en-us/library/windows/desktop/ms676811(v=vs.85).aspx) | Одно значение или несколько у атрибута. |
| [**searchFlags**](https://msdn.microsoft.com/en-us/library/windows/desktop/ms679765(v=vs.85).aspx) | Определяют поисковую информацию атрибута. |
| [**isMemberOfPartialAttributeSet**](https://msdn.microsoft.com/en-us/library/windows/desktop/ms676807(v=vs.85).aspx) | Определяет находится ли атрибут в глобальном каталоге. |
| [**linkID**](https://msdn.microsoft.com/en-us/library/windows/desktop/ms676831(v=vs.85).aspx) | Числовое значение, определяющее является ли атрибут связанным. Четное число – переднее связывание, нечетное – заднее. |
| [**systemFlags**](https://msdn.microsoft.com/en-us/library/windows/desktop/ms680022(v=vs.85).aspx) | Определяет дополнительные параметры атрибута. |
| [**systemOnly**](https://msdn.microsoft.com/en-us/library/windows/desktop/ms680025(v=vs.85).aspx) | Может изменяться только системой. |
| [**mAPIID**](https://msdn.microsoft.com/en-us/library/windows/desktop/ms676860(v=vs.85).aspx) | Числовое значение для клиентов MAPI. |
| [**isDefunct**](https://msdn.microsoft.com/en-us/library/windows/desktop/ms676800(v=vs.85).aspx) | Определяет активен или неактивен атрибут. |
| [**description**](https://msdn.microsoft.com/en-us/library/windows/desktop/ms675492(v=vs.85).aspx) | Описание. |
| [**objectClass**](https://msdn.microsoft.com/en-us/library/windows/desktop/ms679012(v=vs.85).aspx) | Определяет экземпляром какого класса является данный атрибут. |

В Active Directory определено всего 23 типа синтаксиса атрибутов. Ниже перечислены основные (значения SyntaxID и OM ID необходимы для модифицирования схемы, а именно добавления новых атрибутов):

* Boolean: Syntax ID - 2.5.5.8, OM ID - 1
* Enumeration: Syntax ID - 2.5.5.9, OM ID - 10
* Integer: Syntax ID - 2.5.5.9, OM ID - 2
* Interval: Syntax ID - 2.5.5.16, OM ID - 65
* Object(Access-Point): Syntax ID - 2.5.5.14, OM ID - 127
* Object(DN-Binary): Syntax ID - 2.5.5.7, OM ID - 127
* Object(DN-String): Syntax ID - 2.5.5.14, OM ID - 127
* Object(DS-DN): Syntax ID - 2.5.5.1, OM ID - 127
* Object(OR-Name): Syntax ID - 2.5.5.7, OM ID - 127
* String(Case Sensitive) : Syntax ID - 2.5.5.3, OM ID - 27
* String(Generalized-Time) : Syntax ID - 2.5.5.11, OM ID - 24
* String(IA5) : Syntax ID - 2.5.5.5, OM ID - 22
* String(NT-Sec-Desc) : Syntax ID - 2.5.5.15, OM ID - 66
* String(Numeric) : Syntax ID - 2.5.5.6 18
* String(Object-Identifier) : Syntax ID - 2.5.5.2, OM ID - 6
* String(Octet) : Syntax ID - 2.5.5.10, OM ID - 4
* String(Printable) : Syntax ID - 2.5.5.5, OM ID - 19
* String(Sid) : Syntax ID - 2.5.5.17, OM ID - 4
* String(Teletex) : Syntax ID - 2.5.5.4, OM ID - 20
* String(Unicode) : Syntax ID - 2.5.5.12, OM ID - 64
* String(UTC-Time) : Syntax ID - 2.5.5.11, OM ID - 23

## Обзор методов управления схемой Active Directory

* + 1. LDAP

**LDAP** (англ. *Lightweight Directory Access Protocol* — «облегчённый протокол доступа к каталогам») — протокол прикладного уровня для доступа к службе каталоговX.500, разработанный IETF как облегчённый вариант разработанного ITU-T протокола DAP. LDAP — относительно простой протокол, использующий TCP/IP и позволяющий производить операции аутентификации (*bind*), поиска (*search*) и сравнения (*compare*), а также операции добавления, изменения или удаления записей. Обычно LDAP-сервер принимает входящие соединения на порт 389 по протоколам TCP или UDP. Для LDAP-сеансов, инкапсулированных в SSL, обычно используется порт 636.

Всякая запись в каталоге LDAP состоит из одного или нескольких *атрибутов* и обладает *уникальным именем* (DN — англ. *Distinguished Name*). Уникальное имя может выглядеть, например, следующим образом: «cn=Иван Петров, ou=Сотрудники, dc=example, dc=com». Уникальное имя состоит из одного или нескольких*относительных уникальных имен* (RDN — англ. *Relative Distinguished Name*), разделённых запятой. Относительное уникальное имя имеет вид ИмяАтрибута=значение. На одном уровне каталога не может существовать двух записей с одинаковыми относительными уникальными именами. В силу такой структуры уникального имени записи в каталоге LDAP можно легко представить в виде дерева.

Запись может состоять только из тех атрибутов, которые определены в описании класса записи (*object class*), которые, в свою очередь, объединены в схемы (*schema*). В схеме определено, какие атрибуты являются для данного класса обязательными, а какие — необязательными. Также схема определяет тип и правила сравнения атрибутов. Каждый атрибут записи может хранить несколько значений.

Возможности LDAP описаны в 4 моделях:

* Информационная модель определят, каким образом информация или данные представлены в системе LDAP.
* Модель именования определяет, каким образом данные организованы.
* Функциональная модель определяет, что можно сделать (чтение, поиск, запись и т.д.).
* Модель безопасности определяет то, как информация защищена.

Стандартизация поведения LDAP с помощью этих моделей делает LDAP предпочтительным протоколом доступа к службам каталогов.

В протоколе LDAP определены следующие операции для работы с Каталогом:

* Операции подключения/отключения
  + Подключение (*bind*) — позволяет ассоциировать клиента с определённым объектом Каталога (фактическим или виртуальным) для осуществления контроля доступа для всех прочих операций чтения/записи. Для того, чтобы работать с Каталогом, клиент обязан пройти аутентификацию как объект, отличительное имя (Distinguished Name) которого находится в пространстве имён, описываемом Каталогом. В запросе операции *bind* клиент может не указывать отличительное имя, в таком случае будет осуществлено подключение под специальным псевдонимом *anonymous* (обычно это что-то наподобие гостевой учетной записи с минимальными правами)
  + Отключение (*unbind*) — позволяет клиенту в рамках сеанса соединения с LDAP-сервером переключиться на аутентификацию с новым отличительным именем. Команда unbind возможна только после аутентификации на сервере с использованием bind, в противном случае вызов unbind возвращает ошибку
* Поиск (*search*) — чтение данных из Каталога. Операция сложная, на вход принимает множество параметров, среди которых основными являются:
  + База поиска (*baseDN*) — ветка DIT, от которой начинается поиск данных
  + Глубина поиска (*scope*) — может иметь значения (в порядке увеличения охватываемой области): base, one, sub
    - base — поиск непосредственно в узле — базе поиска
    - one — поиск по всем узлам, являющимся прямыми потомками базового в иерархии, то есть лежащим на один уровень ниже него
    - sub — поиск по всей области, нижележащей относительно базы поиска (baseDN)
  + Фильтр поиска (*searchFilter*) — это выражение, определяющее критерии отбора объектов каталога, попадающих в область поиска, задаваемую параметром scope. Выражение фильтра поиска записывается в польской (префиксной) нотации, состоящей из логических (булевых) операторов и операндов, в свою очередь являющихся внутренними операторами сопоставления значений атрибутов LDAP (в левой части) с выражениями (в правой части) с использованием знака равенства.

Логические операторы представлены стандартным «набором»: **&** (логическое «И»), **|** (логическое «ИЛИ») и **!** (логическое «НЕ»).

Пример фильтра поиска:

(&(!(entryDN:dnSubtreeMatch:=dc=Piter,dc=Russia,ou=People,dc=example,dc=com))(objectClass=sambaSamAccount)  
(|(sn=Lazar\*)(uid=Nakhims\*)))

* Операции модификации — позволяют изменять данные в Каталоге, при этом в понятие модификации входит как добавление, удаление и перемещение записей целиком, так и редактирование записей на уровне их атрибутов. Подтипы модификации:
  + Добавление (*add*) — добавление новой записи
  + Удаление (*delete*) — удаление записи
  + Модификация RDN (*modrdn*) — перемещение/копирование записи
  + Модификация записи (*modify*) — позволяет редактировать запись на уровне её атрибутов,
    - добавляя новый атрибут или новое значение многозначного атрибута (add)
    - удаляя атрибут со всеми его значениями (delete)
    - заменяя одно значение атрибута на другое (replace)
    - а также увеличивая (уменьшая) значение атрибута в рамках атомарной операции (increment)
* Операция сравнения (*compare*) — позволяет для определённого отличительного имени сравнить выбранный атрибут с заданным значением

В стандарте LDAP определена специальная операция, позволяющая клиентам получать информацию о поддерживаемых сервером версиях протокола и возможностях LDAP-сервера. Эта команда является надстройкой (расширением) для операции *search* и выполняется при следующем сочетании параметров последней:

* BIND анонимный
* База поиска *baseDN* указана как **""** (пустая строка)
* Глубина поиска *scope* указана как **base**
* Фильтр поиска: **(objectClass=\*)**
* Перечень запрашиваемых атрибутов: либо явное перечисление, либо «**+**».

Active Directory поддерживает LDAP v2 и LDAP v3.

Ниже перечислены особенности 3 версии.

* Можно указать элементы управления (как на сервере, так и на клиенте), которые расширяют функциональные возможности операции LDAP.
* Можно сделать запрос сервер для выполнения расширенных операций (помимо стандартных операций LDAP).
* Повышенная безопасность при использовании Simple Authentication Security Layer (SASL)
* Возможность модифицирования схемы.
* И клиентская, и серверная часть поддерживают формат UTF-8. Можно отправлять и принимать данные на любом языке. Active Directory посылает все ответы в этом формате.
* Операционные атрибуты, которые служба каталогов использует для собственных нужд.
* Сервер может направить клиента на другие сервера.

LDAP v3 совместим с LDAP v2. Есть только одно условие: клиент LDAP v2 должен иметь возможность подключиться к серверу.

* + 1. ADSI

**ADSI** (от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Active Directory Service Interfaces* — «интерфейсы службы Active Directory») — [интерфейс программирования приложений](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9), разработанный компанией[Microsoft](https://ru.wikipedia.org/wiki/Microsoft) и предназначенный для доступа к различным [службам каталогов](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B1%D0%B0_%D0%BA%D0%B0%D1%82%D0%B0%D0%BB%D0%BE%D0%B3%D0%BE%D0%B2), в первую очередь к [Active Directory](https://ru.wikipedia.org/wiki/Active_Directory). ADSI позволяет создавать, изменять и удалять объекты в каталогах, выполнять поиск и т. д.

Интерфейсы Microsoft Active Directory Service Interfaces (ADSI) представляют собой модель службы каталогов на основе модели компонентных объектов, позволяющую клиентским приложениям, совместимым с ADSI, получать доступ к широкому кругу разных протоколов каталогов, включая службу каталогов Windows®, LDAP и NDS, используя всего один стандартный набор интерфейсов. ADSI ограждает клиентское приложение от реализации и вмешательства в работу базового хранилища данных или протокола.

Приложение, называемое поставщиком ADSI, делает себя доступным для клиентских приложений ADSI. Данные, предоставляемые поставщиком, организованы в специальное пространство имен, определенное поставщиком. Вдобавок к реализации интерфейсов, определенных ADSI, поставшик может реализовывать схему ADSI. Схема используется для предоставления метаданных о структуре пространства имен и объектов, предоставляемых поставщиком ADSI.

Дополнительные сведения о ADSI содержатся в Active Directory Service Interfaces Version 2.0 SDK и Active Directory Schema SDK в пакете SDK операционной системы.

Имена объектов ADSI называются строками связывания (Binding String) или строками ADsPath, которые состоят из двух частей. Первая часть имени определяет, к какой именно службе каталогов (или провайдеру ADSI) мы обращаемся:

|  |  |
| --- | --- |
| **Обращение** | **Описание** |
| LDAP:// | Для службы каталогов, созданной на основе протокола LDAP (Lightweight Directory Access Protocol, упрощённый протокол для доступа к каталогу), в том числе для Active Directory в Windows 2000/2003. |
| WinNT:// | Для службы каталогов в сети Windows NT 4.0 или на локальной рабочей станции Windows XP/2000. |
| NDS:// | Для службы каталогов NetWare NDS (Novell Directory Service). |
| NWCOMPAT:// | Для службы каталогов NetWare Bindery. |

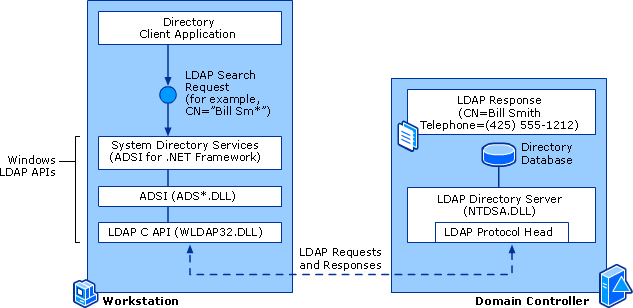
Вторая часть строки ADsPath определяет расположение объекта в конкретном каталоге (для каждого провайдера ADSI - по-своему).

ADSI может выступать в роли провайдера OLE DB, что позволяет с помощью ADO (ActiveX Data Object) выполнять "естественные" запросы к пространству имён службы каталога (провайдер "ADsDSOObject"). Использовать ADSI в качестве провайдера OLE DB можно в запросах к пространствам имён LDAP и NDS.

AD is Microsoft's implementation that's equivalent to LDAP. It's one among the many new aspects that come with Win2K. In an attempt to make access to the directory a simpler task, as well as less daunting to those who are used to COM objects and Microsoft’s Visual Studio, Microsoft introduced a technology known as Active Directory Services Interface (ADSI). Application developers and system administrators alike can use ADSI, which lets them use anything that supports OLE automation (e.g., COM+, OLE DB, or ODBC) to access directory services.

* + 1. Определение способа управления схемой службы каталогов

Ниже приведена иллюстрация различий между LDAP API и ADSI. Как мы видим, ADSI является оболочкой.



On computers running a Windows operating system, directory client applications can use one of three different LDAP APIs. As shown in the following figure, all LDAP requests from a directory client application ultimately go through the native LDAP C API, which runs as Wldap32.dll.

При написании программы мы будем использовать LDAP API. Все остальные технологии работают именно с этим API и являются более тяжеловесными, так как они были разработаны, чтобы упростить процесс программного взаимодействия со службами каталогов.

Программирование с Windows API, куда относится и LDAP, можно назвать низкоуровневым программированием. И работа на этом уровне позволит взглянуть поглубже на то, как работает сеть.

## Обзор существующих решений для управления схемой службы каталогов

### **CSVDE**

Средство CSVDE позволяет импортировать новые объекты в Active Directory, используя исходный CSV-файл; оно также дает возможность экспортировать существующие объекты в файл CSV. CSVDE нельзя использовать для изменения существующих объектов; при использовании этого средства в режиме импорта можно лишь создавать новые объекты.

Экспорт списка существующих объектов с помощью CSVDE довольно прост. Ниже показано, как экспортировать объекты Active Directory в файл под названием ad.csv:

**csvde –f ad.csv**

Параметр –f указывает, что за ним следует имя выходного файла. Но следует понимать, что, в зависимости от среды, этот базовый синтаксис может привести к выводу огромного и неудобного файла. Чтобы ограничить средство экспортом лишь объектов внутри определенного структурного подразделения (OU), команду можно изменить следующим образом:

**csvde –f UsersOU.csv –d ou=Users,dc=contoso,dc=com**

Предположим далее, что мне необходимо экспортировать лишь объекты пользователя в мой файл CSV. В таком случае можно добавить параметр –r, позволяющий указать фильтр протокола LDAP для данного поиска, который ограничит число экспортируемых атрибутов (заметьте, что все нижеследующее является одной строкой):

**csvde –f UsersOnly.csv –d ou=Users,dc=contoso,dc=com –r  
“(&(objectcategory=person)(objectclass=user))” –l  
DN,objectClass,description**

Параметр –i позволяет импортировать объекты в Active Directory из исходного файла CSV. Однако создание объектов пользователя с помощью CSVDE имеет один важный недостаток: с помощью этого средства нельзя устанавливать пароли пользователей, поэтому я бы не стал использовать CSVDE для создания объектов пользователей.

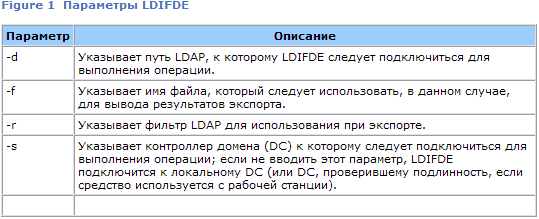
### **LDIFDE**

Active Directory предоставляет второе встроенное средство для пакетных операций пользователей, именуемое LDIFDE и обладающее более широкими и гибкими возможностями, чем CSVDE. Помимо создания новых объектов, LDIFDE позволяет модифицировать и удалять существующие объекты и даже расширять схему Active Directory. Платой за гибкость LDIFDE является то, что необходимый входной файл (файл LDIF) с расширением .ldf использует более сложный формат, чем простой файл CSV. (Немного поработав, можно также настраивать пароли пользователей, но об этом чуть позже.)

Начнем с простого примера — экспорта пользователей в структурном подразделении в файл LDF (отметьте, что все нижеследующее является одной строкой ):

**ldifde -f users.ldf -s DC1.contoso.com -d “ou=UsersOU,dc=contoso,dc=com”  
–r “(&(objectcategory=person)(objectclass=user))”**

Как и в случае большинства средств командной строки, полное описание параметров LDIFDE можно получить, запустив команду LDIFDE /? . На Рис. 1 показаны те, что я использовала здесь. (Заметьте, что параметры для команд CSVDE и LDIFDE одинаковы.)



По-настоящему возможности LDIFDE раскрываются при создании объектов и управлении ими. Однако перед этим необходимо создать входной файл. Нижеследующий код создает две новых учетных записи пользователя — afuller и rking; для создания входного файла введите текст в блокноте (или другом редакторе открытого текста) и сохраните его как NewUsers.ldf:

dn: CN=afuller, OU=UsersOU, DC=contoso, DC=com  
**changetype: add**  
**cn: afuller**  
**objectClass: user**  
**samAccountName: afuller**

**dn: CN=rking, OU=UsersOU, DC=contoso, DC=com**  
**changetype: add**  
**cn: rking**  
**objectClass: user**  
**samAccountName: rking**

После того как создание файла завершено, запустите следующую команду:

**ldifde –i –f NewUsers.ldf –s DC1.contoso.com**

Единственный новый параметр здесь — это -i, который, как несложно догадаться, указывает, что выполняется операция импорта, а не экспорта.

При модификации или удалении существующих объектов синтаксис команды LDIFDE не меняется; вместо этого изменяется содержимое файла LDF. Для изменения поля описания учетных записей пользователей создайте текстовый файл, именуемый ModifyUsers.ldf, такой как показано на Рис. 2.

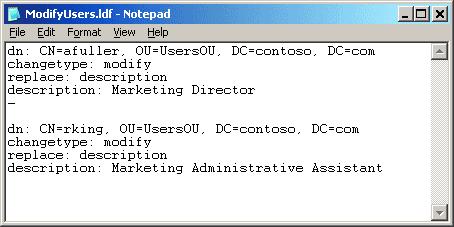


Рис. 2 Файл LDF ModifyUsers

Изменения импортируются путем запуска того же синтаксиса команды LDIFDE, что и раньше, с указанием нового файла LDF после параметры -f. Формат LDF для удаления объектов еще проще; для удаления пользователей, с которыми вы работали, создайте файл, именуемый DeleteUsers.ldf, и введите следующее:

**dn: CN=afuller OU=UsersOU, DC=contoso, DC=com**  
**changetype: delete**

**dn: CN=rking, OU=UsersOU, DC=contoso, DC=com**  
**changetype: delete**

Отметьте, что, в отличие от CSVDE, LDIFDE может настраивать пароли пользователей. Однако перед настройкой атрибута unicodePWD для учетной записи пользователя необходимо настроить шифрование SSL/TLS на контроллерах домена.

Вдобавок, LDIFDE может создавать и модифицировать любые объекты Active Directory, а не только учетные записи пользователей. Например, нижеследующий файл LDF создаст новое расширение схемы, именуемое EmployeeID-example, в схеме леса contoso.com:

**dn: cn=EmployeeID-example,cn=Schema,**  
**cn=Configuration,dc=contoso,dc=com  
changetype: add  
adminDisplayName: EmployeeID-Example  
attributeID: 1.2.3.4.5.6.6.6.7  
attributeSyntax: 2.5.5.6  
cn: Employee-ID  
instanceType: 4  
isSingleValued: True  
lDAPDisplayName: employeeID-example**

Поскольку в файлах LDIFDE используется стандартный отраслевой формат файла LDAP, приложения от сторонних производителей, которым необходимо модифицировать схему Active Directory, часто поставляют файлы LDF, с помощью которых можно изучить и одобрить изменения, прежде чем применять их к производственной среде.

**Оснастка «Схема Active Directory»**

Оснастка консоли управления (MMC) "Схема Active Directory" используется для просмотра объектов схемы доменных служб Active Directory (AD DS) и управления ими. Требования к установке и использованию оснастки "Схема Active Directory" более строгие по сравнению с другими средствами администрирования. Для внесения изменений в оснастку "Схема Active Directory" необходимо быть членом группы безопасности "Администраторы схемы" в корневом домене леса Active Directory. По умолчанию в группу "Администраторы схемы" входит только учетная запись администратора для корневого домена леса.

Оснастка "Схема Active Directory" служит для просмотра определений классов и объектов, обеспечения административного доступа к схеме и позволяет изменять классы и атрибуты схемы.

Оснастка "Схема Active Directory" также может быть использована для просмотра объектов схемы служб Active Directory облегченного доступа к каталогам (AD LDS) и управления ими.

2.4.2. Программа LDAPAdmin

Ldap Admin is a free Windows LDAP client and administration tool for LDAP directory management. This application lets you browse, search, modify, create and delete objects on LDAP server. It also supports more complex operations such as directory copy and move between remote servers and extends the common edit functions to support specific object types (such as groups and accounts).

