**Active Directory**

**개요**

이 글에서는 Active Directory의 중요한 개념 및 특징을 비롯하여 기술적인 면을 개괄적으로 살펴보고, Active Directory 아키텍처를 소개하며 Microsoft® Windows NT® 운영 체제 버전 4.0에서 Microsoft Windows® 2000으로 마이그레이션하는 방법에 대한 정보, 그리고 자주 질문하는 사항(FAQ)에 대해 설명합니다. Active Directory는 Microsoft BackOffice® 패밀리가 통합 구축되는 플랫폼인 Windows 2000 분산 시스템의 기초입니다.

**머리말**

여기서는 Microsoft® Windows® 2000 Server 운영 체제에서 제공하는 새로운 디렉터리 서비스인 Active Directory의 기술적인 특징을 소개합니다. 중요한 Active Directory 개념, 아키텍처 요소 및 기능을 자세하게 소개합니다. "중요 개념" 단원에서는 Active Directory 자체를 살펴 보기 전에 먼저 이해해야 하는 용어들에 대해서 설명합니다. "아키텍처"와 "Active Directory 특징" 단원에서는 Active Directory가 수행하는 기능, Windows에 제공하는 특징과 이것이 구현되는 방법을 자세하게 설명합니다. "마이그레이션"에서는 Windows NT 4.0에서 Windows 2000으로 마이그레이션하는 도메인 모델과 디렉터리 구조를 다루고, 마지막 단원인 "자주 질문하는 사항(FAQ)"에서는 Active Directory에 대해 실제 문의하는 질문에 대해 답변하고 Active Directory가 작동하는 방식을 설명합니다.

*디렉터리*란 관심 있는 개체에 대한 정보를 저장하는 데 사용하는 정보 소스입니다. 전화 번호 디렉터리에는 전화 가입자들에 대한 정보를 저장하고 파일 시스템에는 파일에 대한 정보를 저장합니다.

분산 컴퓨팅 시스템이나 인터넷 같은 공용 컴퓨터 네트워크에는 프린터, 팩스 서버, 응용 프로그램, 데이터베이스, 사용자 같이 흥미로운 개체들이 많이 있습니다. 사용자들은 이 개체들을 찾아 사용하기를 원하고 관리자는 이 개체들을 사용하는 방법을 관리하기를 원합니다.

이 글에서 *디렉터리*와 *디렉터리 서비스*는 공용 네트워크 또는 개인 네트워크에서 볼 수 있는 디렉터리를 말합니다. *디렉터리 서비스*는 *디렉터리*와는 다릅니다. 디렉터리 서비스는 디렉터리 정보 소스이면서 *동시에* 사용자들에게 이 정보를 사용할 수 있고 이용할 수 있게 해 주는 서비스입니다.

디렉터리 서비스는 확장 컴퓨터 시스템의 가장 중요한 구성 요소 중 하나입니다. 사용자들과 관리자들은 관심 있는 개체의 정확한 이름을 모르는 경우가 많습니다. 그러나 그 개체의 *특성*은 하나 이상 알 수도 있으며 디렉터리를 *쿼리*하여 알고 있는 특성과 일치하는 개체 목록을 얻을 수 있습니다. 예를 들어 "26 빌딩에 있는 모든 이중 프린터" 같은 특성을 이용합니다. 디렉터리 서비스는 사용자들이 개체의 특성 중 하나를 이용하여 그 개체를 찾을 수 있도록 합니다.

디렉터리 서비스가 하는 일은 다음과 같습니다.

* 침입자로부터 정보를 안전하게 보관하기 위해 관리자가 정의하는 *보안*을 강화시킵니다.
* 네트워크의 많은 컴퓨터 간에 디렉터리를 *분산*합니다.
* 디렉터리를 더 많은 사용자가 사용할 수 있고 오류를 일으키지 않도록 *복제*합니다.
* 많은 수의 개체를 저장할 수 있도록 여러 개의 저장소로 디렉터리를 *분할*합니다.

디렉터리 서비스는 관리 도구이자 일반 사용자 도구이기도 합니다. 네트워크에 개체의 수가 점점 더 많아짐에 따라 디렉터리 서비스가 점점 더 필수적인 사항이 되고 있습니다. 디렉터리 서비스는 대규모 분산 시스템이 연결되는 허브 같은 것입니다.

Active Directory는 Windows 2000 Server에 포함되어 있는 디렉터리 서비스로, 이전의 Windows 기반 디렉터리 서비스 기능을 확장하고 전체적으로 새로운 기능을 추가합니다. Active Directory는 *안전하고 분산되어 있으며*,*분할되고 복제됩니다*. 또한 몇 백개 정도의 개체만 가지는 단일 서버에서부터 수 천 개의 서버와 수 백만 개의 개체를 가지는 규모에 이르기까지 어떤 환경에서도 잘 작동하도록 만들어졌습니다. Active Directory에 추가된 여러 새 기능은 관리자나 일반 사용자 모두로 하여금 많은 양의 정보를 쉽게 탐색하고 쉽게 관리할 수 있도록 함으로써 시간 절감의 효과를 가져다 줍니다.

**중요 개념**

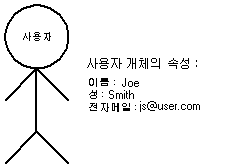
Active Directory를 설명하는 데 사용되는 개념 중에는 새로운 개념도 있고 이전에 사용하던 개념도 있습니다. 불행하게도 한동안 많이 사용되었던 일부 용어들은 구체적인 한 가지 사항이 아니라 여러 의미로 사용되고 있습니다. 따라서 계속하기 전에 먼저 다음 개념과 용어들이 Active Directory와 관련된 맥락에서 어떻게 정의되고 있는가를 이해하는 것이 중요합니다.

Active Directory의 *범위*는 매우 포괄적입니다. 여기에는 모든 단일 개체(프린터, 파일, 사용자)와 모든 서버, 단일 영역 네트워크에 있는 모든 도메인이 포함될 수 있습니다. 또한 여러 개의 네트워크들이 결합되어 있는 넓은 영역이 포함될 수 있습니다. 다음 용어 중에서 일부는 단일 네트워크 이상의 넓은 영역에 적용됩니다. 따라서 Active Directory는 하나의 컴퓨터에서 단일 네트워크, 많은 컴퓨터 네트워크가 결합된 규모에 이르기까지 여러 규모로 확장될 수 있다는 사실을 염두에 두는 것이 중요합니다.

Active Directory는 다른 디렉터리 서비스와 마찬가지로 우선 *이름 공간*입니다. 전화 번호 디렉터리 역시 이름 공간입니다. 주어진 이름을 확인할 수 있는, 경계가 있는 영역은 모두 하나의 이름 공간입니다. 이름 확인이란 이름을 이름이 나타내는 개체 또는 정보로 변환하는 과정입니다. 전화 번호 디렉터리는 전화 번호로 전화 가입자를 확인할 수 있는 이름 공간을 형성합니다. Windows 파일 시스템은 파일의 이름으로 그 파일을 확인할 수 있는 이름 공간을 형성합니다.

Active Directory는 디렉터리에 있는 개체의 이름으로 그 개체를 확인할 수 있는 이름 공간을 형성합니다.

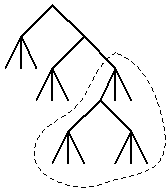
*개체*는 사용자, 프린터, 응용 프로그램 같은 구체적인 무엇인가를 나타내는 구분되고 명명된 *특성* 집합입니다. 특성은 디렉터리 개체가 식별할 수 있는 주제를 설명하는 데이터를 가지고 있습니다. 한 사용자의 특성에는 성, 이름, 전자 메일 주소 등이 포함될 수 있습니다.



**그림 1 사용자 개체와 그 특성**

*컨테이너*는 특성을 가지며 Active Directory 이름 공간의 일부이라는 점에서 개체와 비슷합니다. 그러나 개체와는 달리 구체적인 무엇인가를 나타내지는 않습니다. 말 그대로 개체 그룹과 다른 컨테이너를 담을 수 있는 컨테이너입니다.

*트리*는 이 글에서 개체와 컨테이너의 계층 구조를 설명하는 데 사용됩니다. 트리의 끝점은 보통 개체입니다. 트리의 노드(트리가 가지를 뻗는 점)는 컨테이너입니다. 트리는 개체들이 서로 어떻게 연결되어 있는지 그리고 한 개체에서 다른 개체로 경로가 어떻게 연결되어 있는지를 보여 줍니다. 간단한 디렉터리 하나는 하나의 컨테이너이고 컴퓨터 네트워크나 도메인도 하나의 컨테이너라고 할 수 있습니다. 트리에서 끊어지지 않은 모든 경로가 인접한 하위 트리가 됩니다. 이 경로에는 해당 경로에 있는 컨테이너의 모든 구성원을 포함됩니다.



**그림 2 파일 디렉터리의 인접한 하위 트리**

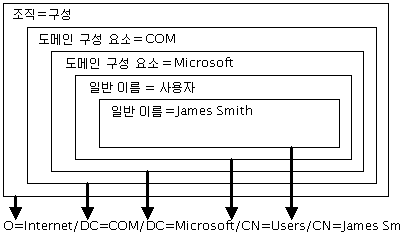
*이름*은 Active Directory에 있는 모든 개체를 식별할 때 사용합니다. 이름에는 두 가지 종류가 있습니다.

**고유 이름(DN)**

Active Directory의 모든 개체는 *고유 이름*(DN)을 가집니다. 고유 이름으로 개체를 보유하는 도메인을 식별하고 그 개체가 닿아 있는 컨테이너 계층 구조의 전체 경로를 확인합니다. 전형적인 DN 형식은 다음과 같습니다.

**/O=Internet/DC=COM/DC=Microsoft/CN=Users/CN=James Smith**

이 DN은 Microsoft.com 도메인의 "James Smith"라는 사용자 개체를 식별해 줍니다.



**그림 3. 고유 이름의 그래픽 표현**

**RDN**

개체의 *상대 고유 이름*(RDN)은 개체의 특성인 이름의 일부입니다. 이전 예에서 보면 "James Smith"` 사용자 개체의 RDN은 **CN=James Smith**이고, 부모 개체의 RDN은 **CN=Users**입니다.

Active Directory는 하나 이상의 *명명 컨텍스트* 또는 *파티션*으로 구성되어 있으며, 디렉터리의 인접한 하위 트리는 하나의 명명 컨텍스트입니다. 명명 컨텍스트는 또한 복제 단위입니다.

Active Directory에서 단일 서버는 최소한 세 개의 명명 컨텍스트를 항상 보유합니다.

* 스키마
* 구성(복제 토폴로지 및 관련 메타데이터)
* 하나 이상의 사용자 명명 컨텍스트(디렉터리에 실제 개체를 가지고 있는 하위 트리)

*도메인*은 Windows NT 또는 Windows 2000 컴퓨터 네트워크의 단일 보안 경계입니다. (도메인에 대한 자세한 내용은 Windows 설명서를 참고하십시오.) Active Directory는 하나 이상의 도메인으로 구성되어 있습니다. 독립형 워크스테이션에서는 그 컴퓨터가 바로 도메인입니다. 도메인은 둘 이상의 물리적인 위치로 그 범위가 확장될 수 있습니다. 모든 도메인은 자체 보안 정책을 가지고 있으며 다른 도메인과의 보안 관계를 가집니다. 여러 도메인이 트러스트 관계에 의해 연결되어 있고 일반 스키마, 구성, 글로벌 카탈로그를 공유할 때 *도메인 트리*를 가지게 됩니다. 여러 도메인 트리는 하나의 *포리스트*로 연결할 수 있습니다.

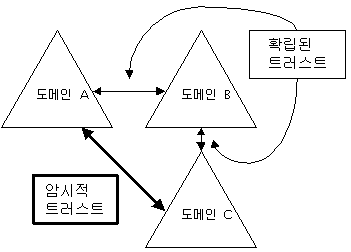
도메인 트리(*트리*)는 일반 스키마, 구성을 공유하여 인접한 이름 공간을 형성하는 여러 개의 도메인으로 구성되며, 트리에 있는 도메인들은 트러스트 관계에 의해 서로 연결되어 있습니다. Active Directory는 바로 이런 트리들의 집합니다.

트리는 두 가지 관점에서 볼 수 있습니다. 한 가지는 도메인 간의 트러스트 관계를 보는 것이고, 다른 관점은 도메인 트리의 이름 공간을 보는 것입니다.

**트러스트 관계 보기**

개별 도메인과 이 도메인들이 서로를 트러스트하는 방법을 기초로 하여 도메인 트리의 그림을 그려볼 수 있습니다.

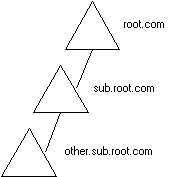
Windows 2000은 Kerberos 보안 프로토콜을 기초로 하여 도메인 간의 트러스트 관계를 구축합니다. Kerberos 트러스트는 전이적이면서도 계층적입니다. 도메인 A가 도메인 B를 트러스트하고 도메인 B는 도메인 C를 트러스트한다면 도메인 A는 자동으로 도메인 C를 트러스트합니다.



**그림 4 트러스트 관계로 바라본 도메인 트리**

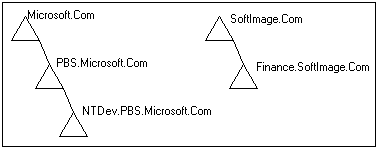
**이름 공간 보기**

또한 이름 공간을 기초로 하여 도메인 트리의 그림을 그려볼 수도 있습니다. 도메인 트리의 이름 공간 경로를 따라서 개체의 고유 이름을 결정할 수 있습니다. 이 관점은 개체를 논리적인 계층 구조로 같이 묶어 그룹화할 때 유용합니다. 인접한 이름 공간의 주된 장점은 이름 공간의 루트에서부터 깊은 검색을 하면 전체 계층 구조를 검색하게 된다는 것입니다.



**그림 5 도메인 트리를 이름 공간으로 표시**

*포리스트*는 인접한 이름 공간을 형성하지 **않는** 트리의 집합입니다. 포리스트의 모든 트리는 일반 스키마, 구성, 글로벌 카탈로그를 공유합니다. 주어진 포리스트의 모든 트리는 전이적이고 계층적인 Kerberos 트러스트 관계를 통해 서로를 트러스트합니다. 그러나 포리스트는 트리와는 달리 구분되는 뚜렷한 이름이 필요 없습니다. 한 포리스트는 상호 참조하는 개체와 구성원 트리에 알려진 Kerberos 트러스트 관계들의 집합으로 존재합니다. 포리스트에 있는 트리들은 Kerbores 트러스트를 위해 계층 구조를 형성하고 트러스트의 루트에 있는 트리 이름은 그 포리스트를 참조하는 데 사용할 수 있습니다.



**그림 6 한 포리스트 안에 있는 여러 개의 트리**

*사이트*는 Active Directory 서버를 포함하고 있는 네트워크에 있는 하나의 위치입니다. 사이트는 잘 연결된 하나 이상의 TCP/IP 서브넷으로 정의됩니다. "잘 연결된다"는 말은 네트워크 연결이 매우 안정되어 있고 빠르다(예: LAN 속도가 초 당 천만비트 이상)는 것을 의미합니다. 서브넷의 집합으로 사이트를 정의하면 관리자가 Active Directory 액세스와 복제 토폴로지를 쉽고 빠르게 구성할 수 있어서 물리적인 네트워크의 장점을 이용할 수 있습니다. 사용자가 로그온하면 Active Directory 클라이언트는 사용자와 동일한 사이트에서 Active Directory 서버를 찾습니다. 동일한 사이트에 있는 시스템들은 네트워크 측면에서 볼 때 서로 가까이 있기 때문에 이들 시스템 간의 통신은 안전하고 빠르며 효율적입니다. 사용자의 워크스테이션은 이미 자신이 사용하는 TCP/IP 서브넷을 알고 있고 서브넷은 Active Directory 사이트로 직접 변환되기 때문에 로그온 시 로컬 사이트를 쉽게 알 수 있습니다.

**아키텍처**

여기서는 Active Directory의 주요 아키텍처 구성 요소 중 일부만 소개합니다.

Active Directory 데이터 모델은 X.500 데이터 모델에서 파생된 것입니다. 디렉터리는 특성으로 설명되는 다양한 종류의 실체를 나타내는 개체들을 보유합니다. 디렉터리에 저장할 수 있는 개체의 범용성은 스키마에서 정의됩니다. 각 개체 클래스에 대해 스키마는 클래스의 인스턴스가 가져야 하는 특성, 가질 수 있는 추가 특성, 그리고 현재 개체 클래스의 상위가 될 수 있는 개체 클래스를 정의합니다.

Active Directory 스키마는 디렉터리에 저장되어 있는 개체 클래스 인스턴스의 집합으로 구현됩니다. 이것은 스키마를 가지는 많은 디렉터리와는 다르지만 시작 시에 읽을 수 있도록 텍스트 파일로 저장합니다. 디렉터리에 스키마를 저장해 두면 많은 이점이 있습니다. 예를 들어, 사용자 응용 프로그램은 스키마를 읽어서 사용 가능한 개체와 속성을 알 수 있습니다.

Active Directory 스키마는 동적으로 업데이트될 수 있습니다. 즉, 응용 프로그램은 새 특성과 클래스로 스키마를 확장할 수 있고 이 확장을 즉시 사용할 수도 있습니다. 스키마 개체를 작성하거나 디렉터리에 저장된 스키마 개체를 수정하여 스키마를 업데이트할 수 있습니다. Active Directory에 있는 다른 개체처럼, 스키마 개체는 액세스 제어 목록(ACL)에 의해 보호되고 따라서 승인된 사용자만 스키마를 변경할 수 있습니다.

디렉터리는 Windows 2000 trusted Coputing Base의 일부이며 Windows 2000 보안 구조에 완전히 통합되어 있습니다. ACL은 Active Directory에 있는 모든 개체를 보호합니다. Windows 2000에서는 액세스 유효성 검사 루틴이 ACL을 사용하여 Active Directory의 개체나 특성에 대한 액세스 시도를 검사합니다.

승인된 사용자는 Active Directory에서 관리 작업을 수행합니다. 사용자는 더 높은 권한에 의해 지정된 개체 집합과 디렉터리의 일부 식별된 하위 트리 상에 있는 개체 클래스에 대해 지정된 작업 집합을 수행할 권한을 허가 받습니다. 이를 *위임된관리*라고 합니다. 위임된 관리를 통해 누가 무엇을 할 수 있는지에 대해 매우 세밀하게 제어할 수 있으며 높은 권한을 주지 않고도 권한을 위임할 수 있습니다.

DSA(Directory System Agent)는 디렉터리의 물리적인 저장소를 관리하는 프로세스입니다. 클라이언트는 지원되는 인터페이스 중 하나를 사용하여 DSA에 연결한 다음 디렉터리 개체와 그 특성을 검색하고, 읽고, 쓸 수 있습니다. DSA는 디렉터리 데이터의 물리적인 저장소 형식으로부터 클라이언트를 분리시켜 줍니다.

**Active Directory 특징**

여기서는 Active Directory의 주요 기능과 구성 요소 중 일부에 대해서 설명합니다.

Active Directory는 도메인 이름 시스템(DNS)와 긴밀하게 통합되어 있습니다. DNS는 컴퓨터 및 서비스 이름을 TCP/IP 주소로 확인하기 위해 인터넷 상에서 사용하는 분산된 이름 공간입니다. 인트라넷을 사용하는 대부분의 기업은 이름 확인 서비스로 DNS를 사용합니다. Active Directory는 DNS를 *위치 서비스*로 사용합니다.

Windows 2000 도메인 이름이 DNS 도메인 이름입니다. 예를 들어, "Microsoft.com"은 유효한 DNS 도메인 이름이자 Windows 2000 도메인의 이름일 수도 있습니다. 긴밀한 DNS 통합이란 Active Directory가 인터넷과 인트라넷 환경에 자연스럽게 조화된다는 의미입니다. 클라이언트는 디렉터리 서버를 쉽고 빠르게 찾습니다. 기업은 Active Directory 서버를 직접 인터넷에 연결하여 고객 및 협력업체들과 안전하게 통신하고 전자 상거래를 할 수 있습니다.

**위치 서비스**

Active Directory 서버는 클라이언트가 도메인 이름만 알아도 서버를 찾을 수 있도록 서버 주소를 게시합니다. Active Directory 서버는 DNS에 있는 서비스 리소스 레코드(SRV RR)를 통해 게시됩니다. SRV RR은 서비스 이름을 그 서비스를 제공하는 서버 주소에 매핑할 때 사용하는 DNS 레코드입니다. SRV RR 이름 형식은 다음과 같습니다.

<service>.<protocol>.<domain>

Active Directory 서버는 TCP 프로토콜을 통해 LDAP 서비스를 제공하므로 게시된 이름은 다음과 같은 형식으로 사용합니다.

ldap.tcp.<domain>

따라서 "Microsoft.com"의 SRV RR은 "ldap.tcp.microsoft.com"입니다. SRV RR의 추가 정보에는 서버의 중요도와 그 비중을 나타내서 클라이언트가 자신에 맞는 가장 좋은 서버를 선택할 수 있도록 합니다.

Active Directory 서버는 설치될 때 *동적 DNS*를 통해 그 자신을 게시합니다. (여기에 대해서는 다음에 설명합니다.) TCP/IP 주소는 시간이 지나면서 변하게 되므로 서버는 주기적으로 등록 상태를 점검하여 등록이 제대로 되어 있는지 검사하고 필요하면 업데이트합니다.

**동적 DNS**

동적 DNS는 DNS 표준1에 최근에 추가되었습니다. 동적 DNS는 DNS 서버를 새 값 또는 변경된 값으로 동적으로 업데이트할 때 사용할 프로토콜을 정의합니다. 동적 DNS 이전에는 관리자들이 DNS 서버가 저장한 레코드를 수동으로 구성해야 했습니다.

하나의 개체는 정확하게 하나의 이름, *고유 이름*(DN)을 가집니다. DN은 개체를 고유하게 식별하며 클라이언트가 디렉터리에서 그 개체를 검색할 수 있도록 그 개체에 대한 충분한 정보를 가지고 있습니다. 개체의 DN은 길어서 기억하기 힘들 수도 있습니다. 더군다나 개체의 DN은 변경될 수도 있습니다. 개체의 DN은 개체의 RDN, 그 상위 이름으로 이루어져 있으므로 개체의 이름이나 상위의 이름을 바꾸면 DN도 같이 변경됩니다.

DN은 기억하기도 복잡하고 변경되기 때문에 개체를 검색할 때 사용할 다른 수단을 가지고 있는 것이 좋습니다. Active Directory는 특성별 쿼리를 지원하므로 정확한 DN을 모르거나, DN이 변경되더라도 개체를 찾을 수 있습니다. 쿼리로 개체를 찾는 과정을 단순하게 만들기 위해 Active Directory 스키마는 두 개의 유용한 속성을 정의합니다2.

* **전역 고유 개체 식별자(GUID)**- 개체마다 고유한 128비트 번호입니다. GUID는 개체가 작성될 때 지정됩니다. 개체를 이동하거나 이름을 바꾸어도 GUID는 변하지 않습니다. 응용 프로그램은 개체의 GUID를 저장하여 현재 DN에 상관 없이 해당 개체를 검색할 수 있도록 해 줍니다.
* **사용자 이름** - 각 보안 이름(사용자 및 그룹)은 "친숙한" 이름인 *사용자 이름*(UPN)을 가집니다. UPN은 DN보다 길이도 짧아 기억하기도 더 쉽습니다. 사용자 이름은 사용자를 나타내는 "기호나 약어" 이름과 사용자 개체가 있는 도메인 트리의 DNS 이름으로 구성됩니다. 예를 들어, microsoft.com 트리의 James Smith 사용자의 UPN은 JamesS@Microsoft.com입니다.

**이름의 고유성**

고유 이름(DN)은 말 그대로 고유해야 합니다. Active Directory에서는 같은 부모 아래에 같은 RDN을 가지는 개체가 하나 이상 있을 수 없습니다. DN은 RDN으로 구성되므로 역시 고유합니다. GUID는 정의상으로 고유하고 고유성이 GUID를 생성하도록 보장하는 알고리즘입니다. 고유성은 다른 속성에 대해서는 강제되지 않습니다.

Active Directory에 대한 액세스는 *유선프로토콜*을 통해 이루어집니다. 유선 프로토콜은 클라이언트와 서버 간에 주고 받는 메시지와 상호 작용의 형식을 정의합니다. 다양한 API를 통하여 개발자들은 이 프로토콜에 액세스할 수 있습니다.

**프로토콜 지원**

다음과 같은 프로토콜이 지원됩니다.

* **LDAP**- Active Directory 핵심 프로토콜은 LDAP(Lightweight Directory Access Protocol)입니다. LDAP 버전 2와 버전 33이 지원됩니다.
* **MAPI-RPC**- Active Directory는 MAPI 인터페이스를 지원하는 원격 프로시저 호출(RPC)을 지원합니다.
* **X.500**- Active Directory 정보 모델은 X.500 정보 모델에서 나온 것입니다. X.500은 Active Directory가 구현하지 않는 여러 유선 프로토콜을 정의합니다. 다음과 같은 프로토콜이 여기에 해당됩니다.

DAP - Directory Access Protocol

DSP -Directory System Protocol

DISP - Directory Information Shadowing Protocol

DOP - Directory Operational Binding Management Protocol

Active Directory가 이 프로토콜을 구현하지 않는 이유는 다음과 같습니다.

* + 이 프로토콜은 별로 중요하지 않고 구현되는 경우도 적습니다.
  + 이 프로토콜은 OSI 네트워킹에 종속적입니다. OSI는 TCP/IP의 대안으로 사용되는 프로토콜이지만 널리 구현되어 있지는 않습니다. TCP/IP 네트워크에서 OSI를 전송하면 TCP/IP를 직접 사용하는 것에 비해 효율성에서 떨어집니다.
  + LDAP는 DAP와 DSP가 제공하는 가장 중요한 기능들을 제공하며 TCP/IP에서 OSI를 "감싸는" 오버헤드 없이 TCP/IP를 통해 작동하도록 만들었습니다.
  + 1993년과 1997년 DISP 및 DOP 사양에는 모호한 부분이 상당히 많습니다. 이에 따라 구현했을 때는 상호 운용성이 보장되지 않고 따라서 이 프로토콜의 중요도는 아주 낮아졌습니다.

**응용 프로그래밍 인터페이스(API)**

지원되는 API는 다음과 같습니다.

* ADSI - Active Directory 서비스 인터페이스(ADSI)는 간단하고 강력한 개체 지향 인터페이스를 Active Directory4에 제공합니다. 개발자는 Java, Visual Basic® 프로그래밍 시스템, C, C++ 등을 비롯하여 각종 프로그래밍 언어를 사용할 수 있습니다. ADSI는 스크립팅이 가능하므로 시스템 관리자들이 쉽게 사용할 수 있습니다. ADSI는 또한 사용자들이 볼 수 없도록 LDAP 통신의 자세한 정보를 숨깁니다.
* LDAP API - RFC 1823에 정의된 LDAP C API는 C 프로그래머가 사용할 수 있는 하위 수준 인터페이스입니다.
* MAPI - Active Directory는 역 호환성을 위해 MAPI를 지원합니다. 새 응용 프로그램은 ADSI 또는 LDAP C API를 사용해야 합니다.

Active Directory에서는 *가상 컨테이너*를 통해 다른 디렉터리를 제공합니다. 가상 컨테이너는 Active Directory에 저장되어 있는 지식 정보를 통해 구현됩니다. 지식 정보는 Active Directory에서 외부 디렉터리가 나타나야 하는 위치, 외부 DS에서 검색 작업을 시작하기 위한 외부 디렉터리의 복사본을 보유하는 서버의 DNS 이름과 고유 이름(DN)이 있는 위치를 기술합니다.

Active Directory는 많은 파티션과 명명 컨텍스트로 이루어질 수 있습니다. 개체의 *고유 이름*(DN)에는 그 개체가 있는 파티션의 복제본의 위치를 찾을 수 있는 충분한 정보가 포함되어 있습니다. 그러나 사용자나 응용 프로그램이 대상 개체의 DN을 모르거나 개체가 있는 파티션을 모르는 경우가 많습니다. *글로벌 카탈로그*(GC)는 사용자나 응용 프로그램이 대상 개체의 속성을 하나 이상 알면 Active Directory 도메인 트리에서 개체를 찾을 수 있게 해 줍니다.

글로벌 카탈로그에는 디렉터리의 모든 사용자 명명 컨텍스트의 부분 복제본이 있습니다. 여기에는 또한 스키마 및 구성 명명 컨텍스트가 들어 있습니다. GC에는 Active Directory에 있는 모든 개체의 복제본 뿐 아니라 그 특성들도 있다는 의미입니다. GC에 있는 특성은 사용자의 성, 이름, 로그인 이름 같이 검색 작업에 자주 사용하는 특성과 개체의 전체 복제본의 위치를 아는 데 필요한 특성들입니다. GC는 사용자가 관심 있는 개체가 있는 도메인 이름이나, 기업 내에서 인접한 확장 이름 공간이 없어도 그 개체를 빨리 찾을 수 있게 해 줍니다.

Active Directory 복제 시스템은 자동으로 글로벌 카탈로그를 만들고 복제 토폴로지를 생성합니다. 글로벌 카탈로그에 복제된 속성들에는 마이크로소프트가 정의한 기본 집합이 포함됩니다. 관리자는 설치에 필요한 추가 속성들을 지정할 수 있습니다.

여기까지 Active Directory의 보안 기능에 대해 간단히 살펴 보았습니다. Windows 2000 보안 모델에 대한 자세한 정보는 "Secure Networking Using Microsoft Windows 2000 Distributed Security" 백서를 참고하십시오.

**개체 보호**

Active Directory의 모든 개체는 액세스 제어 목록(ACL)으로 보호됩니다. ACL은 개체를 볼 수 있는 사용자와 각 사용자가 해당 개체에 대해 수행할 수 있는 작업을 결정합니다. 개체를 볼 권한이 없는 사용자에게는 그 개체 자체가 존재하지 않게 됩니다.

ACL은 보호하는 개체와 같이 저장된 액세스 제어 항목(ACE) 목록입니다. Windows 2000에서 ACL은 *보안 설명자*라고 하는 이진값으로 저장됩니다. 각 ACE에는 보안 식별자(SID)가 있어서 ACE가 적용되는 *이름* (사용자 또는 그룹)을 식별하고 ACE에 의해 부여되거나 거부된 정보에 대한 액세스 유형을 식별합니다.

디렉터리 개체에 대한 ACL에는 개체 전체에 적용되는 ACE와 개체의 개별 특성에 적용되는 ACE가 포함되어 있습니다. ACE를 통해 관리자는 개체를 볼 수 있는 사용자 뿐 아니라 볼 수 있는 속성까지도 제어할 수 있습니다. 예를 들어, 모든 사용자는 다른 모든 사용자에 대해 전자 메일과 전화 번호 특성을 읽을 권한을 받을 수 있지만 사용자의 보안 속성은 특별한 보안 관리자 그룹의 구성원 외 다른 사용자는 볼 수 없도록 거부될 수 있습니다. 개별 사용자는 자신의 사용자 개체에 대해서 전화 번호와 메일 주소와 같은 개인 특성에 쓸 권한을 받을 수 있습니다.

**위임**

*위임*은 Active Directory의 가장 중요한 보안 기능 중 하나입니다. 위임을 사용해서 더 높은 관리 권한이 컨테이너나 하위 트리에 대한 특정 관리 권한을 개인이나 그룹에 부여할 수 있습니다. 이렇게 함으로써 사용자 인구가 많은 부분을 통째로 관리하는 "도메인 관리자"에 대한 필요는 줄어듭니다.

ACE는 컨테이너에 있는 개체에 대한 특정 관리 권한을 사용자나 그룹에 부여할 수 있습니다. 컨테이너의 ACL에 있는 ACE를 통해 특정 개체 클래스에 대해 특정 작업을 할 수 있는 권한이 부여됩니다. 예를 들어, "James Smith" 사용자를 "회계부"라는 조직 구성 단위의 관리자로 만들려면 "회계부"의 ACL에 다음과 같은 ACE를 추가합니다5.

"James Smith";Grant ;Create, Modify, Delete;Object-Class User  
"James Smith";Grant ;Create, Modify, Delete;Object-Class Group  
"James Smith";Grant ;Write;Object-Class User; Attribute Password

이제 James Smith는 회계부에 새 사용자와 그룹을 만들 수 있고 기존 사용자에 대한 암호를 설정할 수 있지만 다른 개체 클래스를 만들 수는 없습니다. ACE에서 다른 컨테이너에 대한 액세스 권한을 주지 않는 한 다른 컨테이너에 있는 사용자에는 영향을 미칠 수 없습니다.

**상속**

*상속*은 컨테이너로부터 주어진 ACE를 해당 컨테이너의 모든 자식 항목에 전파하는 것입니다. 상속을 위임과 같이 결합하여 한 번에 디렉터리의 전체 하위 트리에 관리 권한을 부여할 수 있습니다.

Active Directory에서는 *멀티마스터 복제* 기능을 제공합니다. 멀티마스터 복제란 해당 파티션의 모든 복제본이 쓰기 가능하다는 의미입니다. 이 기능을 사용하여 그 파티션의 어느 복제본에든 업데이트를 적용할 수 있습니다. Active Directory 복제 시스템은 지정된 복제본에서 다른 복제본으로 변경 내용을 전파합니다. 복제는 자동으로 진행되고 투명합니다.

**업데이트 전파**

일부 디렉터리 서비스는 변경 내용을 감지하고 전파하는 데 타임스탬프를 사용합니다. 이런 시스템에서는 모든 디렉터리의 클럭을 동기화하는 것이 매우 중요합니다. 네트워크에서는 시간 동기화를 유지하기가 어렵습니다. 뛰어난 네트워크 시간 동기화 기능을 가지고 있다고 하더라도 해당 디렉터리 서버의 시간이 잘못 설정될 가능성도 있습니다. 이렇게 되면 업데이트를 놓치게 됩니다.

Windows 2000에서는 분산 시간 동기화를 제공하지만, Active Directory 복제 시스템에서는 시간에 따라 업데이트 전파를 수행하지 않습니다. 대신 Active Directory 복제 시스템은 *업데이트 시퀀스 번호*(USN)를 사용합니다. USN은 각 Active Directory 서버가 유지 관리하는 64비트 번호입니다. 서버가 Active Directory에 어떤 속성을 쓸 때 USN이 올라가고 방금 쓴 속성과 같이 저장됩니다. 이 작업은 자동으로 이루어지는데, USN의 증가와 저장 및 속성값의 쓰기가 하나의 단위로 성공하거나 실패합니다.

각 Active Directory 서버는 복제 파트너로부터 받은 USN 테이블을 유지 관리합니다. 파트너로부터 받은 가장 높은 USN이 이 테이블에 저장됩니다. 해당 파트너가 복제가 필요한 디렉터리 서버를 알려 주면 서버는 가장 최신에 받은 USN 값보다 더 높은 USN으로 모든 변경 내용을 요청합니다. 이 간단한 방법은 타임스탬프의 정확도에 의존하지 않습니다.

테이블에 저장된 USN은 받은 각 업데이트에 대해 하나의 단위로 업데이트되기 때문에 오류가 발생해도 간단하게 복구할 수 있습니다. 복제를 다시 시작하려면 서버는 테이블에 있는 가장 최근 값보다 더 높은 USN을 사용하여 모든 변경 내용을 파트너에게 요청합니다. 테이블은 변경 내용이 적용될 때마다 하나의 단위로 업데이트되기 때문에 복제 주기가 중간에 중단되더라도 중단된 위치를 정확하게 파악하므로 업데이트를 놓치거나 중복하지 않습니다.

**충돌 감지 및 속성 버전 번호**

Active Directory같은 멀티마스터 복제 시스템에서는 동일한 속성이 서로 다른 두 개 또는 그 이상의 복제본으로 업데이트될 가능성이 있습니다. 첫 번째 복제본에서 변경 내용이 완전히 전파되기도 전에 두 번째(또는 세 번째, 네 번째...) 복제본에서 속성이 변경되면 *복제충돌*이 일어납니다. 이런 충돌은 *속성 버전 번호*를 사용하여 감지됩니다. 서버별 값인 USN과는 달리 속성 버전 번호는 Active Directory 개체의 속성별 번호입니다. 속성이 처음 Active Directory 개체에 쓰여질 때 속성 버전 번호가 초기화됩니다.

*원래쓰기*가 일어날 때 속성 버전 번호가 증가합니다.*.*원래 쓰기는 변경 작업을 시작하는 시스템에서 속성에 대한 쓰기입니다. 복제로 인한 속성 쓰기는 원래 쓰기가 아니므로 속성 버전 번호가 증가하지 않습니다. 예를 들어 사용자가 암호를 업데이트할 때 원래 쓰기가 발생하면서 암호 속성 버전 번호가 증가합니다. 변경된 암호를 다른 서버에 복제하여 쓸 때는 속성 버전 번호가 변하지 않습니다.

복제에 의해 변경 내용을 받을 때 속성 버전 번호가 로컬에 저장된 버전 번호와 동일하고 받은 값과 저장된 값이 다르게 되면 이 때 충돌이 감지됩니다. 이런 현상이 일어나면 받는 시스템이 최신 타임스탬프를 가지는 업데이트를 적용합니다. 이런 경우는 복제에 시간이 사용되는 상황에서만 발생합니다.

받은 속성 버전 번호가 로컬에 저장된 버전 번호보다 낮으면 이 업데이트는 이미 오래된 것으로 간주되어 무시됩니다. 받은 속성 버전 번호가 로컬에 저장된 버전 번호보다 높으면 업데이트를 받아들입니다.

**전파 완충**

Active Directory 복제 시스템에서는 복제 토폴로지에서 루프를 허용합니다. 이를 통해 관리자는 성능과 가용성을 높이기 위해 서버들 사이에 여러 경로를 가지는 복제 토폴로지를 구성할 수 있습니다. Active Directory 복제 시스템은 변경 내용이 끝없이 전파되지 않도록 하고 이미 최신 상태인 복제본에 변경 내용이 중복해서 전달되지 않도록 *전파 완충* 기능을 제공합니다.

Active Directory 복제 시스템은 *최신 벡터*를 채용하여 전파를 완충합니다. 최신 벡터란 각 서버가 보유하는 서버-USN 쌍 목록입니다. 최신 벡터는 서버로부터 받은 원래 쓰기의 가장 높은 USN을 서버-USN 쌍으로 표시합니다. 해당 사이트에 있는 최신 벡터에는 그 사이트에 있는 모든 서버가 열거됩니다.

복제 주기가 시작되면 요청하는 서버가 자신이 가지고있는 최신 벡터를 보내는 서버에게 보냅니다. 보내는 서버는 최신 벡터를 사용하여 요청하는 서버에 보낸 변경 내용을 필터링합니다. 주어진 원래 쓰기의 높은 USN이 특정 업데이트에 대한 원래 쓰기 USN보다 높거나 같은 경우에는 보내는 서버가 변경 내용을 보낼 필요가 없고 요청하는 서버는 이미 원래 쓰기와 관련하여 최신 상태를 유지하고 있는 것입니다.

Windows 2000 도메인 트리는 Windows 2000 도메인의 계층 구조이며 각 도메인은 하나의 Active Directory 파티션입니다. 트리의 모양과 트리 구성원의 관계는 도메인의 DNS 이름으로 결정됩니다. 트리의 도메인은 서로 인접한 이름 공간을 형성해야 합니다. 즉 **a.myco.com**은 **myco.com**의 자식이고 **b.myco.com**은 a**.myco.com**의 자식입니다.

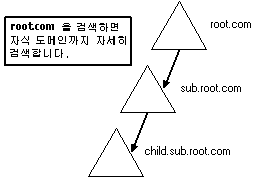
**양방향 전이 트러스트**

Windows 2000 도메인 트리에 도메인이 추가될 때는 이 추가된 도메인과 트리의 모든 자식 도메인 간에 자동으로 양방향 전이 트러스트 관계가 구축됩니다. 트러스트는 전이적이고 양방향이기 때문에 트리 구성원 간에 트러스트 관계를 추가할 필요가 없습니다. 트러스트 계층 구조는 **구성** 컨테이너에 디렉터리 메타데이터의 일부로 저장됩니다.

도메인이 트리에 연결될 때 디렉터리에 이 필수 개체들이 만들어집니다.

**이름 공간**

Windows 2000 도메인 트리의 도메인들은 인접한 이름 공간을 형성해야 합니다. 기본적으로 각 도메인의 바로 밑에 있는 자식 도메인은 인접하므로 이미 이름 공간의 일부가 됩니다. 이것은 자식 도메인의 모든 개체의 고유 이름은 그 부모 도메인의 이름을 접두사로 가진다는 것을 의미합니다. 이 트리들을 하나의 포리스트로 결합할 수 있습니다.



**그림 7 자식 도메인의 이름이 부모 도메인 이름 바로 아래 종속되므로 부모 도메인과 자식 도메인은 서로 인접한 이름 공간을 형성합니다.**

예를 들어 부모 도메인, O=Internet/DC=COM/DC=Microsoft, 그리고 자식 도메인, O=Internet/DC=COM/DC=Microsoft/DC=PBS는 인접한 이름 공간을 형성합니다. 부모 도메인의 루트 개체, O=Internet/DC=COM/DC=Microsoft가 자식의 루트 개체, O=Internet/DC=COM/DC=Microsoft/DC=PBS**의 루트 개체 바로 위 부모입니다.**부모와 자식이 명명 트리를 형성하기 때문에 부모에서 시작되는 깊은 검색은 자동으로 자식 도메인까지도 검색하게 됩니다.

**한 개의 도메인 트리를 형성하는 시기**

Windows 2000 도메인 트리는 기업 수준의 Active Directory입니다. 해당 기업의 모든 Windows 2000 도메인들은 기업 도메인 트리에 속해야 합니다. 도메인에 대해 분리된 DNS 이름을 지원해야 하는 기업의 경우에는 포리스트를 형성해야 합니다.

**도메인 트리 또는 포리스트를 형성하는 방법**

Windows 2000 도메인은 설치 과정 동안 하나의 도메인 트리로 결합됩니다. 새 Windows 2000 서버를 설치하거나 Windows NT 이전 버전에서 업그레이드하는 동안 관리자는 다음 옵션을 선택하게 됩니다.

* 새 포리스트에 첫째 트리 만들기
* 기존 포리스트에 새 트리 만들기
* 기존 도메인의 새 복제본 만들기
* 자식 도메인 설치

도메인 트리를 결합하려면 **자식 도메인 설치**를 선택한 다음 새 자식 도메인의 부모 도메인을 확인합니다. Windows로 나중에 업데이트하면 더 큰 단일 트리로 기존 트리를 연결할 수 있는 기능이 추가됩니다.

기존 트리 내에서는 도메인을 자유롭게 이동하여 전체 트리 모양을 변경할 수 있습니다. 좋은 트리를 계획하는 것은 중요하지만 좋다는 것은 주관적인 개념이며 각 조직에서 원하는 바에 따라 달라집니다. 필요할 때 트리를 다시 구성할 수 있다면 정확한 디자인의 중요성은 그만큼 줄어듭니다.

사이트는 시스템 간에 연결성이 좋다고 간주되는 네트워크의 영역입니다. Windows 2000은 하나 또는 그 이상의 IP 서브넷으로 한 사이트를 정의합니다. 이것은 같은 서브넷 주소를 가지는 컴퓨터는 LAN이나 Frame Relay, ATM 같은 다른 고대역6 환경인 동일한 네트워크 세그먼트에 연결된다는 가정을 기초로 한 것입니다.

Windows 2000은 사이트 정보를 사용하여 사용자에게 가장 가까운 Active Directory 서버의 위치를 찾습니다. 사용자 워크스테이션이 네트워크에 연결할 때 DHCP 서버로부터 TCP/IP 주소를 받습니다. 이 TCP/IP 주소는 워크스테이션이 연결되어 있는 서브넷을 식별합니다. 동적으로 IP 주소가 구성된 워크스테이션은 또한 동적으로 구성된 서브넷 정보를 가집니다. 그렇지 않은 경우에는 Windows 2000 도메인 컨트롤러(DC) 로케이터가 워크스테이션에 알려진 서브넷 정보를 바탕으로 사용자와 동일한 서브넷에 있는 Active Directory 서버를 찾으려고 합니다.

**사이트 및 복제**

Windows 2000 복제 시스템은 그 사이트의 Active Directory 서버 간 복제를 위한 링 토폴로지를 자동으로 생성합니다. 한 사이트 내에서는 원격 프로시저 호출(RPC)을 통해 디렉터리 복제가 이루어집니다. 사이트 간 복제에서는 RPC나 메시징 중에서 하나를 사용하도록 구성할 수 있습니다. Windows 2000에서는 표준 기능으로 간단한 SMTP 메시징을 제공합니다. Microsoft Exchange를 사용할 수 있는 경우에는 Exchange가 지원하는 많은 메인 전송 중 하나를 사용하여(SMRP, X 400 등) Exchange를 통해 사이트 간 디렉터리 복제를 수행할 수 있습니다.

**사이트 계획**

가장 작은 사이트는 단일 IP 서브넷으로 구성됩니다. Windows 2000에서는 같은 사이트에 있는 모든 시스템은 공용 고대역폭 네트워크를 공유한다고 가정합니다. 이런 가정 하에서 좋은 사이트 디자인이란 그 사이트에 할당된 모든 서브넷이 그런 네트워크를 공유하는 것입니다. WAN, 다중 라우터 또는 더 느린 링크로 분리되어 있는 네트워크의 영역은 별도의 사이트여야 합니다.

Active Directory 스키마는 디렉터리에 저장할 수 있는 모든 개체 클래스 집합과 특성들을 정의합니다. 각 개체 클래스에 대해 스키마는 클래스의 유효한 부모를 지정함으로써 디렉터리 트리에서 개체 클래스를 만들 수 있는 위치를 정의합니다. 한 클래스의 내용은 클래스가 가져야 하는 또는 가질 수 있는 특성 목록에 의해 정의됩니다.

**스키마 확장 시기**

사용자와 응용 프로그램은 필요에 맞는 기존 개체 클래스가 없을 때 스키마를 확장합니다. 스키마 확장은 간단한 프로세스입니다.

**특성 추가**

언제든지 스키마에 새 특성을 추가할 수 있습니다. 특성 정의는 이름, 고유한 개체 식별자(OID), 특성이 보유할 수 있는 데이터 종류를 정의하는 구문, 선택적 범위 제한으로 구성됩니다. 문자열의 경우 값을 제한하여 문자열의 최대 길이와 최소 길이를 설정합니다. 정수의 경우에는 값을 제한하여 정수의 최대값과 최소값을 설정합니다.

Active Directory에서 쿼리의 성능은 쿼리를 최적화할 때 사용할 수 있는 인덱스의 가용성과 직접 관련이 있습니다. 해당 쿼리를 만족하는 인덱스가 없으면 LDAP 서버가 쿼리를 만족시키기 위해 전체 파티션을 읽어야 합니다. 특성을 정의할 때 그 특성에 대한 인덱스를 만들 수도 있습니다. **attributeSchema**개체의 **searchFlags** 특성을 1로 설정하여 해당 특성의 인덱스를 만들 수도 있으며 특성을 인덱스로 만들었음을 정의해야 합니다.

* 특성은 쿼리에 자주 사용됩니다. 인덱스를 추가하면 항목이 삽입될 때 반드시 인덱스를 유지 관리해야 하기 때문에 저장소를 소비하며 삽입 성능에 영향을 미칩니다. 따라서 자주 사용하지 않을 인덱스는 추가하지 말아야 합니다.
* 특성 값은 고유해야 합니다. 부울값은 참 또는 거짓, 이 두 가지 값만 가지므로 부울값 특성을 인덱스로 만들면 안됩니다. 직원 번호와 이름은 고유하므로 인덱스로 설정하기에 좋습니다.
* 특성은 관심 있는 개체에서 발생합니다. 특성을 인덱스로 만들면 그 특성을 인스턴스화한 개체도 쉽게 찾을 수 있습니다. 인덱스를 추가하기 전에 먼저 인덱스로 만드는 특성이 검색하려는 개체가 반드시 가져야 하는 필수 특성인지, 또는 가질 수도 있는 특성인지를 확인하십시오.

**새 개체 추가**

언제든지 스키마에 새 개체를 추가할 수 있습니다. 개체 정의는 이름, 개체 식별자(OID), 포함할 수 있는 특성 및 포함해야 하는 특성 목록, 개체의 부모 항목이 될 수 있는 클래스 목록, 개체가 파생된 클래스, 그리고 개체에 적용되는 보조 클래스 목록으로 구성됩니다.

**스키마 확장 방법**

스키마는 디렉터리에 저장할 수 있는 사항을 제어하고 이미 저장되어 있는 사항을 설명하기 때문에 스키마에 대한 쓰기 권한은 기본적으로 관리자로 제한됩니다. Windows 2000에서는 Microsoft Management Console (MMC)에 대한 스키마 관리 스냅인이 제공됩니다. 스키마를 확장하기 위해 적절한 권한이 있는 사용자가 새 특성과 클래스를 만들 수 있습니다. 그런 다음 새 클래스나 기존 클래스에 새 특성을 추가할 수 있습니다. 각각의 새로운 특성이나 클래스에는 OID가 필요합니다.

**개체 식별자(OID)**

개체 식별자는 디렉터리 서비스에 있는 개체 클래스나 특성을 뚜렷하게 식별하는 번호입니다7 . OID는 *발급 기관*에서 발급하며 계층 구조를 형성합니다. OID는 점으로 구분되는 십진수 문자열로 표시합니다(예: "1.2.3.4"). 기업과 개인은 발급 기관으로부터 루트 OID를 획득하고 이를 사용하여 추가 OID를 할당합니다. 예를 들어 마이크로소프트가 발급 받은 루트 OID는 1.2.840.113556입니다. 마이크로소프트는 이 루트로부터 뻗어나오는 가지를 내부적으로 관리합니다. 이 가지들 중 하나는 Active Directory 클래스를 위한 OID를 할당하는 데 사용하고 다른 하나는 Active Directory 특성용으로 사용하며 또 다른 가지는 다른 용도로 사용하는 방식으로 OID를 관리합니다.

전 세계의 많은 나라들은 기업에 OID를 발급하는 일을 맡고 있는 *국가 등록 기관(*NRA)을 가지고 있습니다. 미국의 NRA는 ANSI(American National Standards Institute)입니다. NRA는 루트 OID를 발급합니다. 기업은 OID에 사용할 이름을 등록할 수도 있습니다. 루트 OID와 등록 이름을 사용하려면 요금을 내야 합니다. 자세한 내용은 해당 국가의 NRA에 문의하십시오8 .

게시란 사용 가능하게 하려는 정보를 직접 포함하거나, 그 정보에 대한 참조를 포함하는 개체를 만드는 작업입니다. 예를 들어, 사용자 개체에는 전화 번호, 전자 메일 주소같이 사용자에 대한 유용한 정보가 있고 볼륨 개체에는 공유 파일 시스템 볼륨에 대한 참조가 있습니다.

**게시 시기**

Active Directory에 게시하는 정보는 유용하거나 사용자 중 많은 사람이 관심을 갖고 있고 많은 사람이 사용하는 정보이어야 합니다.

Active Directory에 게시하는 정보는 두 가지 특징을 지닙니다.

* 비교적 정적이고 자주 변경되지 않습니다. 전화 번호와 전자 메일 주소는 비교적 정적인 정보로 게시하기에 적합합니다. 반면 현재 선택된 사용자의 전자 메일 메시지는 휘발성이 높은 정보의 예입니다.
* 뚜렷한 특성 집합으로 나타낼 수 있고 구조화되어 있습니다. 사용자의 사무실 주소는 게시하기에 적합한 구조화된 정보의 예입니다. 반면 사용자의 목소리를 담은 오디오 클립은 파일 시스템에 더 적합한 구조화되지 않은 정보의 예입니다

응용 프로그램에서 사용하는 작업 정보는 Active Directory에 게시하기에 가장 적합한 정보입니다. 이 정보에는 그 응용 프로그램의 모든 인스턴스에 적용되는 글로벌 구성 정보가 포함되어 있습니다. 예를 들어, 관계형 데이터베이스 제품은 데이터베이스 서버에 대한 기본 구성 정보를 Active Directory에 개체로 저장할 수 있습니다. 제품을 새로 설치할 때 이 개체에서 기본 정보를 수집하여 설치 과정을 단순화하고 기업에서 제품 설치의 일관성을 향상시킬 수 있습니다.

응용 프로그램은 또한 그 *연결 지점*을 디렉터리에 게시할 수 있습니다. 연결 지점은 클라이언트/서버가 만나는 데 사용됩니다. Active Directory는 *Service Administration Point*개체를 사용하여 통합 서비스 관리를 위한 아키텍처를 정의하고 RPC, Winsock, COM 응용 프로그램에 대한 표준 연결 지점을 제공합니다. 연결 지점을 게시하는 데 RPC나 Winsock 인터페이스를 사용하지 않는 응용 프로그램은 Active Directory에 *Service Connection Point*개체를 명시적으로 게시할 수 있습니다.

응용 프로그램 데이터 역시 응용 프로그램별 개체를 사용하여 디렉터리에 게시할 수 있습니다. 응용 프로그램별 데이터는 위에서 논의한 기준에 맞아야 합니다. 즉, 여러 사용자에게 흥이 있으며 비교적 자주 변경되지 않으며 조직화되어 있는 데이터여야 합니다.

**게시 방법**

정보를 게시하는 방법은 응용 프로그램이나 서비스에 따라 달라집니다.

* RPC - RPC 응용 프로그램은 API 중에서 RpcNs\* 패밀리를 사용하여 연결 지점을 디렉터리에 게시하고 연결 지점을 게시한 서비스들의 연결 지점을 쿼리합니다.
* Windows 소켓 응용 프로그램은 Winsock 2.0에서 사용 가능한 API 중에서 Registartion and Resolution 패밀리를 사용하여 연결 지점을 게시하고 연결 지점을 게시한 서비스의 연결 지점을 쿼리합니다.
* DCOM - DCOM 서비스는 Active Directory에 있는 DCOM Class Store를 사용하여 연결 지점을 게시합니다.

Windows 2000에서는 새로운 그룹 특징을 도입합니다.

* Exchange의 다음 주요 릴리스를 설치하면 그룹이 메일 그룹으로 간주됩니다.
* 그룹에는 비 보안 구성원이 포함될 수 있습니다. (이것은 그룹이 보안 목록과 메일 그룹 두 가지 모두에 사용될 때 중요합니다.)
* 그룹을 보안용으로 사용하지 않을 수 있습니다. (이것은 그룹을 메일 그룹으로만 사용할 때 중요합니다.)
* 그룹을 중첩시킬 수 있습니다.
* 새로운 그룹인 *유니버설* 그룹이 도입되었습니다.

유니버설 그룹은 가장 간단한 형태의 그룹입니다. 유니버설 그룹은 포리스트의 어느 위치에서나 ACL로 나타날 수 있고 다른 유니버설 그룹, 글로벌 그룹, 그리고 포리스트의 어느 위치에 있는 사용자든지 포함할 수 있습니다. 최소 설치는 유니버설 그룹만을 사용하며 글로벌 그룹이나 로컬 그룹은 관심을 두지 않습니다.

*글로벌*그룹은 포리스트 어느 위치에서나 ACL로 나타날 수 있습니다. 글로벌 그룹에는 그 도메인의 사용자와 다른 글로벌 그룹을 포함할 수 있습니다.

도메인 로컬 그룹은 그 도메인에서만 ACL로 사용할 수 있습니다. 도메인 로컬 그룹에는 포리스트에 있는 모든 도메인의 사용자와 글로벌 그룹을 포함할 수 있고 그 도메인의 유니버설 그룹과 다른 도메인 로컬 그룹을 포함할 수 있습니다.

이 세 가지 종류의 그룹은 그룹 구성원을 변경함으로써 야기되는 글로벌 카탈로그의 복제 소통량을 줄이면서도 풍부하고 신축적인 액세스 제어 환경을 제공합니다. 유니버설 그룹은 GC에 나타나지만 포리스트에 있는 도메인의 글로벌 그룹을 우선적으로 포함합니다. 일단 글로벌 그룹이 형성되면 유니버설 그룹의 구성원은 자주 바뀝니다. 글로벌 그룹은 GC에 나타나지만 구성원은 나타나지 않습니다. 글로벌 그룹의 구성원이 변경되면 이 그룹이 정의된 도메인의 외부에는 복제되지 않습니다. 도메인 로컬 그룹은 이 그룹이 정의된 도메인에서만 유효하며 GC에는 전혀 나타나지 않습니다.

**마이그레이션**

여기서는 Windows NT 이전 버전에서 마이그레이션하는 과정을 간단하게 살펴 보겠습니다. 마이그레이션에 대해 자세하게 설명한 글은 <http://www.microsoft.com/ntserver/>에서 볼 수 있습니다.

Windows NT 3.51 및 4.0 시스템에서 Windows 2000으로 직접 업그레이드할 수 있습니다. Windows NT 3.1과 3.5 시스템은 Windows 2000으로 업그레이드하기 전에 먼저 Windows NT 3.51이나 4.0으로 업그레이드해야 합니다. *Windows 2000하드웨어 호환 목록*에 있는 시스템에서는 Windows 2000으로 업그레이드할 수 있습니다.

도메인 컨트롤러는 디렉터리의 복사본을 보관합니다. Windows NT 3.51와 4.0에는 두 가지 종류의 도메인 컨트롤러, 주 도메인 컨트롤러(PDC)와 백업 도메인 컨트롤러(BDC)가 있습니다. 주 도메인 컨트롤러는 읽기/쓰기 복사본을 보유하는 반면 백업 도메인 컨트롤러는 읽기 전용 복사본을 보유합니다.

Windows 2000에서는 해당 도메인의 모든 도메인 컨트롤러가 쓰기 가능한 디렉터리 사본을 보유합니다. 주 도메인 컨트롤러와 백업 도메인 컨트롤러 간에 구분이 없고 모든 도메인 컨트롤러가 동등합니다.

Windows NT 3.51 또는 4.0 도메인에서 Windows 2000으로 마이그레이션하려면 먼저 도메인의 주 도메인 컨트롤러를 Windows 2000으로 업그레이드해야 합니다. 이렇게 하면 도메인 디렉터리의 사용자와 그룹을 자동으로 Active Directory로 로드합니다. 업그레이드 과정에서 이 도메인의 다음 중 하나로 지정합니다.

* 새 포리스트의 새 트리의 루트
* 기존 포리스트의 새 도메인 트리의 루트
* 기존 도메인 트리의 자식

이 때는 도메인이 *혼합 도메인*입니다. Windows 2000 관리 도구를 사용하여 도메인을 관리할 수 있고 디렉터리에 조직 구성 단위의 계층 구조를 만들어 관리 권한을 위임할 수 있습니다. 백업 도메인 컨트롤러, 구성원 서버, 클라이언트는 변경되지 않고 PDC가 이제 Active Directory 서버라는 사실도 알지 못합니다.

백업 도메인 컨트롤러를 마이그레이션하려면 각 도메인 컨트롤러를 Windows 2000으로 업그레이드합니다. 이렇게 하면 백업 도메인 컨트롤러를 자동으로 Active Directory 복제본으로 설치하고 복제 토폴로지에 삽입합니다. 모든 도메인 컨트롤러를 Windows 2000으로 업그레이드하면 도메인은 더 이상 혼합 도메인이 아니며 중첩 그룹이 지원됩니다.

구성원 서버를 마이그레이션하려면 Windows 2000으로 업그레이드합니다. 로컬 사용자와 로컬 그룹은 구성원 서버의 레지스트리에 저장되어 있으며 Active Directory로 옮겨지지 않습니다. 구성원 서버를 Windows 2000으로 업그레이드하면 서버는 Kerberos 보안에 참가하고, Active Directory 인식 응용 프로그램에 대한 지원을 추가하며 Microsoft Management Console과 Active Directory 인식 셸과 공용 대화 상자도 추가할 수 있습니다.

Windows NT Workstation 기반 클라이언트를 마이그레이션하려면 Windows 2000 Professional로 업그레이드합니다. Active Directory를 인식하는 데 필요한 소프트웨어 구성 요소를 추가로 설치하여 Windows 95 기반 클라이언트를 마이그레이션할 수도 있습니다. 클라이언트를 업그레이드하면 서버는 Kerberos 보안에 참가하고 Active Directory 인식 응용 프로그램에 대한 지원을 추가하며 Active Directory 인식 셸과 공용 대화 상자를 추가할 수 있습니다.

하위 PDC (3.51 또는 4.0)가 Windows 2000으로 업그레이드하면 사용자들도 Active Directory로 마이그레이션되고 도메인 루트의 "Users"라는 컨테이너에 배치됩니다.

하위 PDC (3.51 또는 4.0)을 Windows 2000으로 업그레이드하면 시스템 계정도 Active Directory로 마이그레이션되고 도메인 루트의 "Computers"라는 컨테이너에 배치됩니다.

하위 PDC (3.51 또는 4.0)을 Windows 2000으로 업그레이드하면 그룹이 Active Directory로 마이그레이션되고 도메인 루트의 "Users" 컨테이너에 배치됩니다. 기본 제공 그룹은 "Built-in"이라는 특별 컨테이너에 배치됩니다.

**자주 질문하는 사항(FAQ)**

워크스테이션은 처음으로 연결한 Active Directory 서버에 자신의 서브넷을 제공함으로써 사이트를 찾습니다. 워크스테이션은 자신의 서브넷 마스크를 IP 주소에 적용하여 서브넷을 결정합니다. 서브넷 마스크와 IPO 주소는 DHCP에 의해 지정되거나 정적으로 구성될 수 있습니다. 처음으로 연결된 서버는 제공된 서브넷을 사용하여 워크스테이션이 위치하고 있는 사이트에 대한 Site 개체를 찾습니다. 현재 서버가 해당 사이트에 없으면 서버는 사용하기에 더 나은 서버를 워크스테이션에게 알려 줍니다.

워크스테이션은 DNS를 쿼리하여 디렉터리 서버를 찾습니다. 해당 도메인의 디렉터리 서버는 SRV 리소스 레코드를 다음과 같은 형식의 DNS 이름으로 게시합니다.

LDAP.TCP.<domain name>

따라서, Microsoft.com에 로그인하는 워크스테이션은 LDAP.TCP.Microsoft.com에 대한 SRV 레코드용 DN을 쿼리합니다. 서버는 목록에서 선택하여 연결합니다. 연결된 서버는 워크스테이션이 제공하는 서브넷 정보를 사용하여 이전 질문에서 설명한 가장 적합한 서버를 결정합니다.

사용자는 다양한 형식의 다양한 이름으로 Windows 2000 Professional에 로그온할 수 있습니다. 여기에는 Win32® 응용 프로그래밍 인터페이스인 **DsCrackNames**가 지원하는 이름 형식도 포함됩니다. 이 인터페이스는 필요할 때 이름 형식을 변환하는 데 사용됩니다.

* 도메인 NETBIOS 이름과 SAM-Account-Name- 이것은 Windows NT 4.0 스타일의 로그온 이름입니다. 도메인 NETBIOS 이름은 도메인이 마이그레이션 이전에 가지고 있던 이름입니다. SAM-Account-Name은 사용자가 마이그레이션 이전에 가지고 있던 이름입니다.
* 사용자 이름 -이 이름은 <friendly name>@<dotted-dns-domain-name>형식을 사용합니다. 이름이 고유하지 않으면 로그온 시 **알 수 없는 사용자 오류**로 로그온이 실패합니다.

액세스 제어 목록은 마이그레이션에 의해 직접적인 영향을 받지는 않습니다. Windows NT 3.51과 Windows NT 4.0의 모든 도메인이 제대로 마이그레이션되면 ACL 측면에서는 변경되는 것이 아무 것도 없습니다.

서버를 리소스 도메인에서 마이그레이션된 계정 도메인의 한 조직 구성 단위로 옮기고 *그 리소스 도메인을 삭제하는 경우*, 삭제된 도메인을 참조하는 ACE가 들어 있는 ACL을 편집해야 합니다. 이것은 Windows 2000 마이그레이션의 기능은 아닙니다. Windows NT 버전 도메인을 삭제하면 그 도메인에서 발급한 보안 식별자는 유효하지 않게 됩니다.

ACL 리소스를 다시 적용하는 데 드는 수고를 줄이기 위해서 Windows NT 3.51이나 4.0을 사용하는 리소스 도메인을 가지고 있고 이 도메인을 OU로 대체한 다음 Windows 2000에서 이를 삭제하려고 하는 경우에는, 그룹을 리소스 도메인에서ACL로 이동해서는 안됩니다. 이렇게 하면 구성원 서버의 로컬 그룹에는 영향을 미치지 않으며 도메인 컨트롤러의 그룹에만 영향을 줍니다.

마이크로소프트는 ACL을 리소스들에게 다시 적용하도록 도와 주는 도구를 Windows 2000의 일부로 제공할 것입니다.

도메인에 그룹을 만들면 그 그룹은 도메인에서 발급하는 보안 식별자(SID)를 가지게 됩니다. 그 SID를 ACL로 넣으면 ACL은 그 토큰에서 해당 SID를 가지는 사용자에 대한 액세스 권한을 부여합니다. 사용자는 SID를 발급한 도메인에 로그온하여 그 토큰에서 SID를 가져옵니다. 이것이 네트워크 로그온이고 투명하게 일어납니다.

ACL을 편집할 때 **LookupAccountName** API는 SID와 같이 호출됩니다. SID를 발급한 도메인을 삭제하면 ACL의 사용자 및 그룹 목록에 "알 수 없는 사용자"라고 나타나는 것을 보게 됩니다. Windows NT 3.51와 Windows NT 4.0에서 도메인이나 트러스트 링크를 삭제하면 이런 현상이 일어납니다.

여기에 대해서는 두 부분으로 나누어 답할 수 있습니다.

* **1부: 구성원 서버의 로컬 그룹** - 구성원 서버의 로컬 그룹은 여전히 로컬로 남습니다. 그들은 구성원 서버의 SAM에만 존재하고 Active Directory로 마이그레이션되지 않습니다. 구성원 서버의 로컬 그룹에 대한 일반적인 사용 예는 계정 도메인으로부터 글로벌 그룹을 보유하는 것입니다. 서버 관리자는 서버 리소스에 대한 ACL에 로컬 그룹을 넣고 로컬 그룹에 글로벌 그룹을 추가합니다. 이것은 Windows 2000에서도 바뀌지 않습니다. 다른 점은 글로벌 그룹이 정상 그룹이 되어 Active Directory에 게시된다는 것입니다.
* **2부: PDC와 BDC의 로컬 그룹** - 백업 도메인 컨트롤러는 읽기 전용 SAM을 가지고 있습니다. BDC에 로컬 그룹을 만들 때 작성 동작은 PDC에서 원격으로 이루어져서 모든 BDC로 복제됩니다. 의미상으로는 이 로컬 그룹들은 구성원 서버의 로컬 그룹과 동일하지만 실제로는 PDC와 모든 BDC에 존재합니다. Windows 2000으로 업그레이드할 때 업그레이드 과정 중에 Active Directory에 각각에 대한 도메인 로컬 그룹 개체가 만들어 집니다.

글로벌 카탈로그 검색은 여러 가지 방법 중 하나로 시작됩니다.

* Null DN(이름 공간의 루트)에 루트되어 있는 하위 트리 또는 일수준 LDAP 검색 - 이 검색 방법은 글로벌 카탈로그에 대한 참조를 생성합니다.
* GC 복제본에 있는 GC 포트를 직접 참조.
* GC ADSI 공급자(GC://)에 대한 명시적인 참조.

아닙니다. Microsoft DNS를 사용하면 상당한 이점이 있지만 RFC 호환 DNS 서버도 잘 작동합니다. 동적 DNS를 권장합니다. 왜냐하면 동적 DNS 서버와 같이 사용하면 Active Directory 서버는 자동으로 DNS에 필수적인 레코드를 등록할 수 있기 때문에 동적 DNS를 사용하는 것이 좋습니다. 정적 DNS 서버 역시 잘 작동하지만 각 Active Directory 서버에 대한 DNS 등록은 수동으로 해야 합니다.

Windows 2000에 포함된 DNS 서버는 동적 DNS를 RFC 및 BIND를 준수하여 구현한 것입니다. 이 서버는 Windows 2000용으로 구현한 것이지만 공용 도메인 BIND 구현의 포트가 아닙니다. Microsoft DNS 서버는 Active Directory에서 믿을만한 DNS 영역을 저장합니다. DNS 데이터는 영역 전송이 아니라 Active Directory 복제에 의해 Microsoft DNS 서버들 간에 복제됩니다. Microsoft DNS 서버는 다른 DNS 서버와의 상호 운용성을 위해 표준 DNS 영역 전송을 지원합니다.

DHCP 서버는 Windows 2000에서도 변하지 않습니다. DHCP 클라이언트는 DNS를 인식하고 동적 DNS 서비스를 사용하여 DNS에서 직접 DHCP에 의해 발급되는 주소를 등록합니다. DHCP 클라이언트는 DHCP가 WINS 서버를 확인하면 WINS를 사용하여 등록을 계속합니다.

WINS는 Windows 2000에서 변하지 않습니다. Windows 2000 클라이언트(와 Active Directory 업그레이드가 설치된 Windows 95 클라이언트)는 더 이상 NETBIOS 이름 공간을 사용할 필요가 없습니다. WINS는 하위 클라이언트가 서버를 찾거나 서버가 하위 클라이언트를 찾을 때 여전히 필요합니다. 기업에 더 이상 하위 클라이언트가 없으면 WINS 서버를 꺼도 됩니다.