개발 플랫폼으로서의 윈도우 2003

시스템 관리자는 개발 프로젝트의 시작에서부터 종료까지 개발팀의 일원으로 혹은 조력자로써 함께 생각하고 작업해야 한다. 또한 가까운 미래에는 일정 수준의 개발 능력을 가져야만 진정한 시스템 관리자의 역할을 수행할 수 있을것이다. 물론 관리자와 개발자의 영역은 서로 다르지만, 프로젝트에 있어서 개발자와 관리자간의 의사 소통에 장애가 생긴다면 개발 프로젝트, 혹은 시스템 운용에 크고 작은 많은 문제들이 발생할 것이다. 개발자와 관리자의 긴밀한 협조는 프로젝트 성공의 주요 요건으로 작용한다.

윈도우 서버 2003은 닷넷과 통합되어 있고, 기존의 개발 관련 지원 기능도 대폭 향상되었다. 이 문서는 최신 개발 환경인 닷넷 프레임워크, 새롭게 지원되는 기능인 UDDI 서비스, 향상된 기능을 제공하는 구성요소 서비스 등에 대한 이해를 목표로 한다.

닷넷 프레임워크

마이크로소프트의 닷넷 프레임워크는 인터넷 기반의 분산 컴퓨팅 환경하에서 가장 효율적인 개발을 가능케 해주는 최신의 개발 환경이다. 닷넷 프레임워크는 기존 분산 컴퓨팅의 많은 문제점들을 해결할 수 있으며, 현재에도 많은 기능들이 추가되고 있다.

닷넷 프레임워크의 소개

닷넷 프레임워크는 인터넷 기반의 분산 환경에서의 응용 프로그램 개발을 효율적으로 하기 위한 목적으로 개발되었다. 닷넷 프레임워크의 가장 큰 특징은 안전한 코드 실행과 기존의 스크립트 환경에서의 성능 문제를 해결해 준다는 것이다.

닷넷 프레임워크의 구성 요소들 중, 가장 중요한 두 가지의 구성요소는 공용 언어 런타임 (Common Language Runtime) 과 닷넷 프레임워크 클래스 라이브러리이다. 공용 언어 런타임은 메모리 관리, 스레드 실행, 코드 안전성 검사 등의 여러 가지 기능을 수행하며, 공용 언어 런타임에서 실행되는 코드를 '관리되는 코드 (Managed Code)' 라고 부른다. 닷넷 프레임워크 클래스 라이브러리는 재 사용 가능한 타입들의 집합이며, 공용 언어 런타임과 밀접하게 관련되어 있다.. 클래스 라이브러리는 개체 지향적으로 구현되어 있으며, 기능의 확장이 용이하다. 클래스 라이브러리를 사용하면 일반적인 용도의 응용프로그램을 쉽고 빠르게 개발할 수 있다.

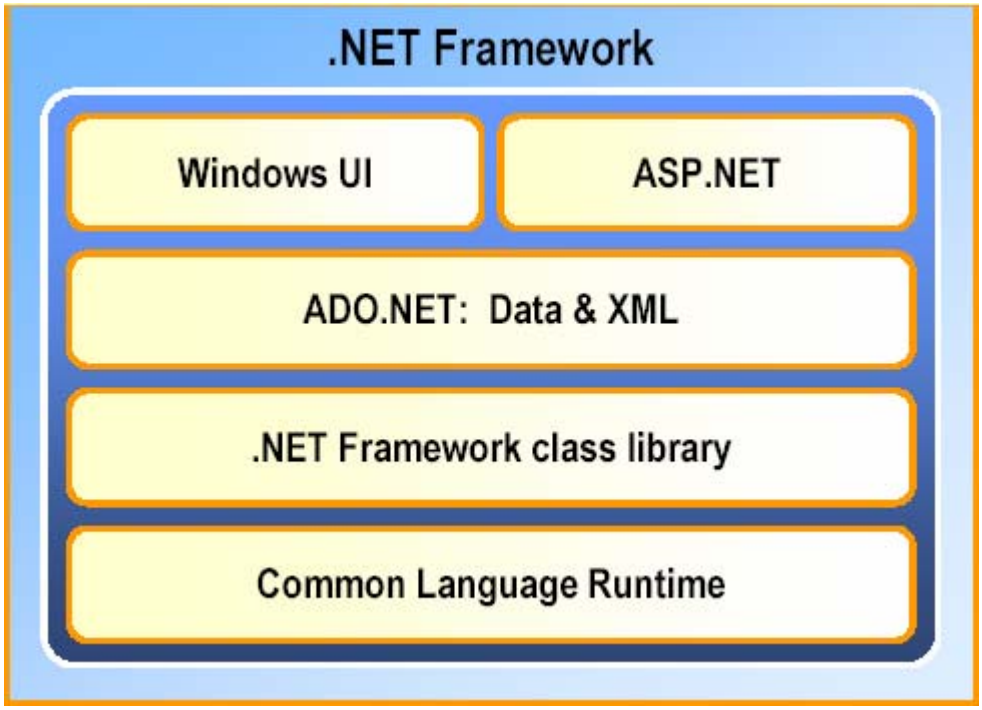

그림 1. 닷넷 프레임워크

공용 언어 런타임

런타임 자체는 닷넷 프레임워크에서 새롭게 등장한 요소는 아니다. 이전의 많은 개발 언어들 (비주얼 베이직 6.0, 비주얼 C++, Jscript 등) 이 각각 자신의 런타임을 사용했다. 공용 언어 런타임을 기존의 런타임들과 구분 지어주는 가장 큰 특징은 공용 언어 런타임이 닷넷을 지원하는 모든 언어들을 위한 통합된 런타임이라는 사실이다.

공용언어 런타임을 사용함으로써 개발 프로세스가 단순해지고, 개발된 응용 프로그램이 실행될 때 메모리의 할당, 프로세스와 스레드 관리, 보안 정책의 사용, 의존성 관리 등의 작업이 자동화 되기 때문에 개발자는 필요한 업무 기능의 구현에만 신경 쓰면 된다. 또한 개발 시에 는 코드의 정확성과 타입 안전성을 검사하는 역할을 하므로 많은 부분에서 개발자의 수고를 덜어준다.

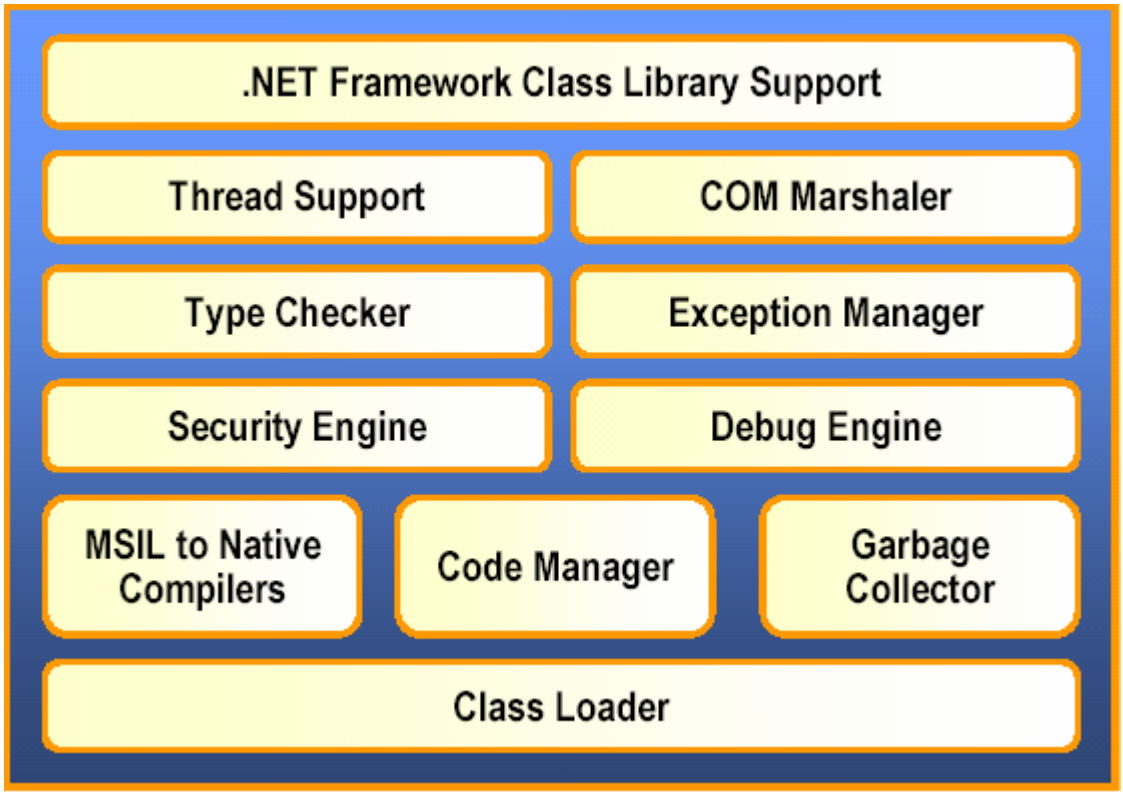

그림 2. 공용 언어 런타임

닷넷 프레임워크 클래스 라이브러리

닷넷 프레임워크 클래스 라이브러리는 재사용 가능한 클래스, 혹은 타입들의 집합이다. 이 라이브러리는 몇 가지 네임스페이스로 구성되어 계층 구조를 이루며, 모든 클래스는 System 네임스페이스로 부터 시작한다. 네임스페이스의 디자인은 매우 직관적이다. 예를 들면 System.IO 네임스페이스는 입출력과 관련된 클래스들을, System.Net 네임스페이스는 TCP/IP와 소켓에 관련된 클래스들을 포함한다.

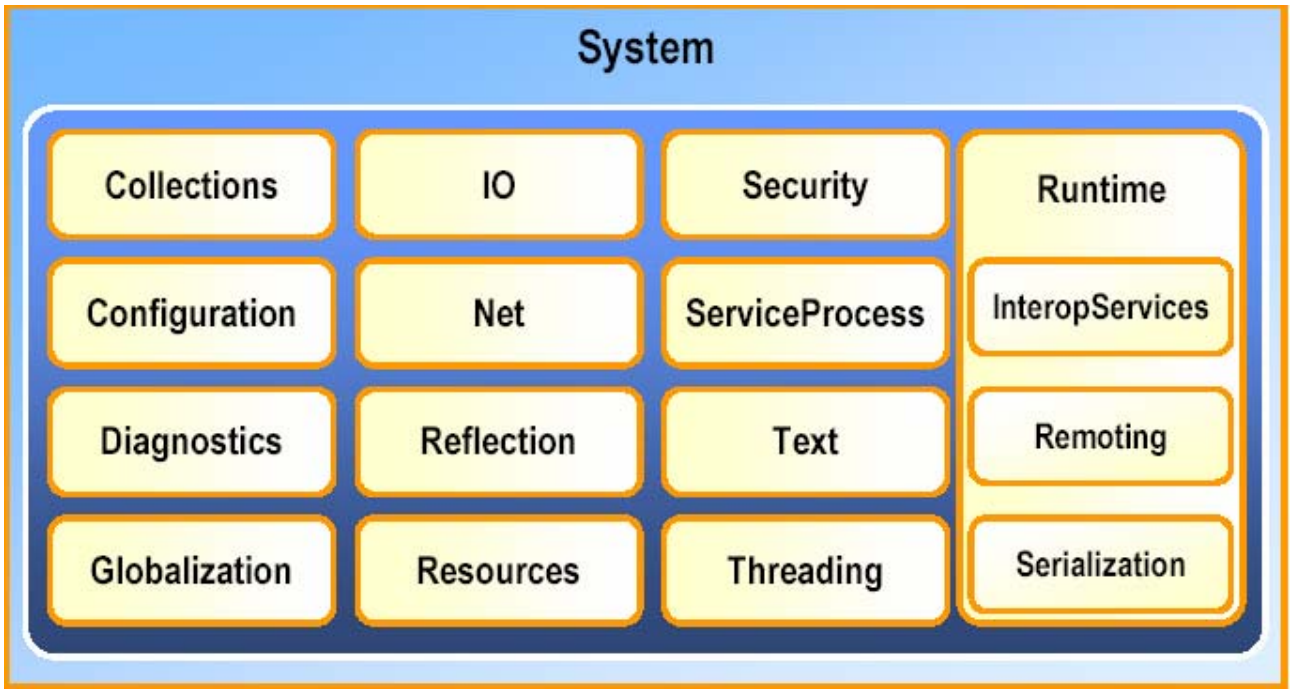

그림 3. 닷넷 프레임워크 클래스 라이브러리

ASP.NET 과 XML 웹 서비스

ASP.NET은 엔터프라이즈급의 웹 응용 프로그램을 위한 통합 개발 플랫폼이다. ASP.NET은 새로운 프로그래밍 모델을 제공하며, 안전하고 확장성 있는 구조를 가지고 있다.
또한 ASP.NET은 고용 언어 런타임상에서 작동하며, 웹 프로그래밍을 위한 강력한 프레임워크로 기능한다. ASP.NET의 웹 폼 (Web Forms) 은 동적인 사용자 인터페이스를 쉽게 만들 수 있도록 해 주며, ASP.NET을 이용해 개발된 XML 웹 서비스는 웹 기반의 분산 응용 프로그램의 구성 요소가 된다.

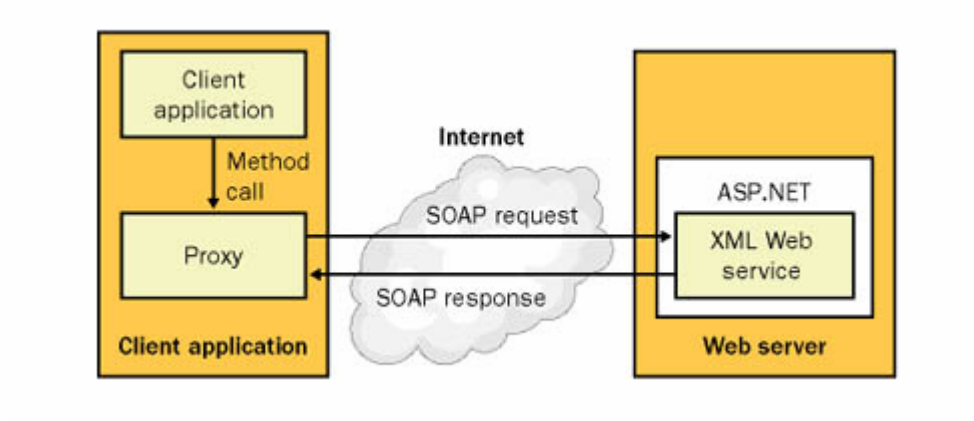

그림 4. XML 웹 서비스

현재의 개발 환경에서 가장 크게 이슈화 되고 있는 부분은 아마도 응용 프로그램 통합일 것이다. 대부분의 기업들은 다양한 플랫폼과 개발 기술을 이용해서 업무 환경을 구축하고 있다. 이런 환경에서 사용되는 다양한 응용 프로그램들은 기본적으로 호환성이 없으며, 이 문제를 해결하기 위해 미들웨어를 도입하기도 하지만, 아직까지는 이기종간의 비호환성을 해결하는 것은 어려운 문제이다.

XML 웹 서비스는 이러한 환경에서 다양한 응용 프로그램을 통합시키기 위한 이상적인 대안이다. XML 웹 서비스는 이기종 운영환경, 서로 다른 개발 언어 문제를 HTTP, XML, SOAP 등의 인터넷 표준 기술을 사용하여 해결하고 있다.

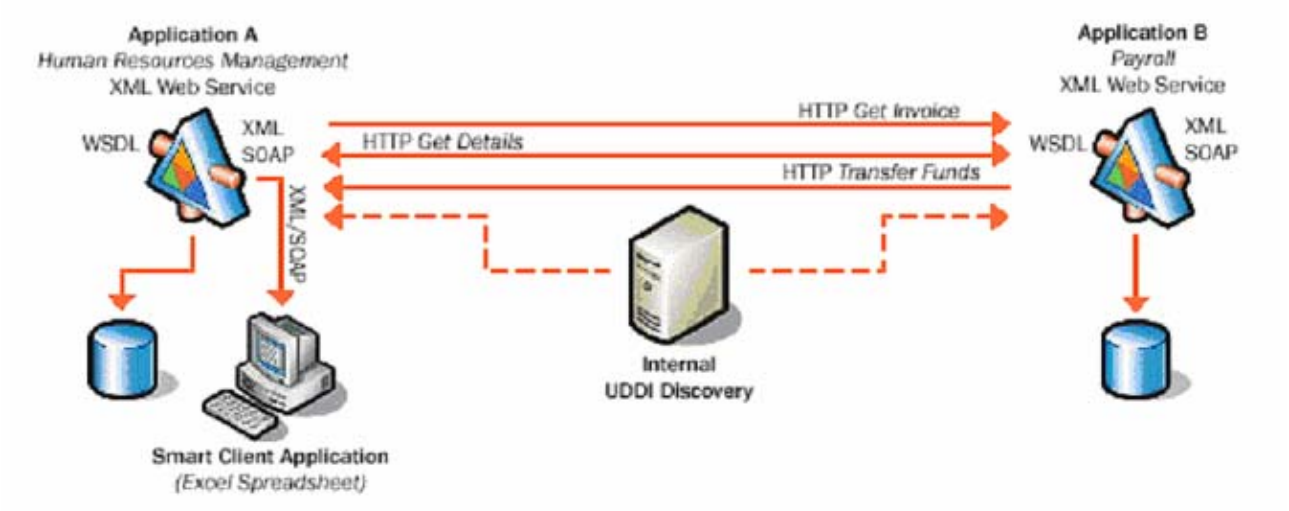

그림 5. XML 웹 서비스를 통한 응용프로그램 통합

닷넷 프레임워크 1.1 구성

윈도우 서버 2003에는 닷넷 프레임워크 1.1 버전이 포함되어 있으며, 보안 정책과 설정을 위한 구성 도구를 제공한다.

닷넷 프레임워크의 보안정책

닷넷 프레임워크에서의 보안은 코드 그룹, 권한 집합, 정책 어셈블리의 세가지 요소의 관계로 정의된다.

공용언어 런타임에서 실행되는 코드는 어셈블리라는 단위로 배포되고, 로드 시 런타임은 코드 작성자의 디지털 서명 및 코드의 출처 같은 증명 정보를 검사한다. 공용 언어 런타임의 보안 관리자는 보안 정책을 기반으로 어셈블리를 코드 그룹에 매핑한다.

윈도우 운영체제는 권한이 없는 사용자가 운영체제에 액세스하는 것을 막아준다. 하지만 운영체제에 내장된 보안 기능은 권한 있는 사용자가 악의적인 프로그램 코드를 다운로드하거나 실행하는 것을 막지 못한다. 이런 구조는 고의가 아닌 실수로 인해 시스템을 손상 시킬 수 있다. 그런 한 예로, 악성코드가 숨겨져 있는 웹사이트에 접속하거나 웜이나 바이러스에 감염된 이메일을 열어보는 경우를 들 수 있다.

닷넷 프레임워크의 코드 액세스 보안은 악의적이거나 결함 있는 코드의 실행으로부터 시스템을 보호할 수 있다. 코드 액세스 보안은 코드의 출처나 기타 증명 정보에 기반하여 신뢰 수준을 결정한다. 예를 들면 내부 인트라넷에서 다운로드한 코드는 높은 수준의 신뢰도를 갖고, 인터넷 상의 알 수 없는 사이트로부터 다운로드한 코드는 낮은 수준의 신뢰성을 가질 것이다.

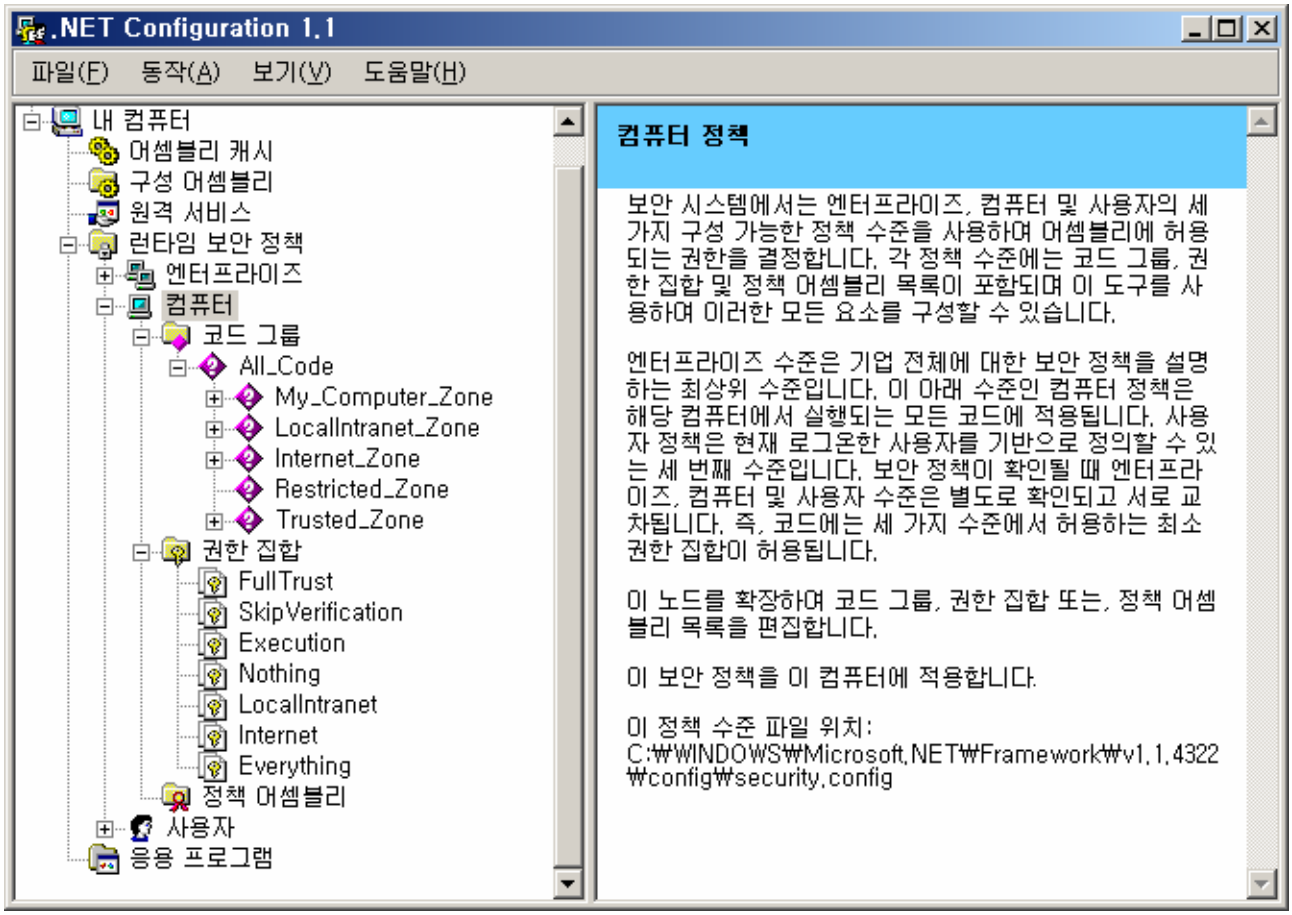

그림 6. 코드 액세스 보안 정책

닷넷 프레임워크 구성파일

닷넷 프레임워크는 응용프로그램이 실행되는 방법을 개발자와 관리자가 제어할 수 있도록 해준다. 관리자는 구성 파일을 사용하여 응용 프로그램이 액세스할 수 있는 리소스, 응용 프로그램이 사용하는 어셈블리 버전, 응용 프로그램의 위치 등을 제어할 수 있으며, 구성 파일은 XML 형식으로 개별 컴퓨터나 네트워크에 연결된 여러 컴퓨터에 있는 응용 프로그램을 관리할 수 있도록 해준다.

닷넷 프레임워크의 구성 파일에는 여러 종류가 있으며, 각 파일이 영향을 미치는 범위는 그 종류에 따라 다르다. 먼저 컴퓨터의 구성과 응용 프로그램의 구성 파일은 바인딩 리디렉션과 코드 위치 등의 정보를 제공하고, 보안 구성 파일은 보안 정책을 설정하는 데 사용된다.

컴퓨터 구성, 응용 프로그램 구성 파일

컴퓨터 구성 파일에는 컴퓨터 전체에 적용되는 어셈블리 바인딩, 기본 제공 원격 채널 및 ASP.NET 에 대한 구성 설정이 들어 있다. 이 파일은 Machine.config 이란 이름을 가지며, %CLR install path%₩Config 의 경로에 위치한다. 또한 응용 프로그램 구성 파일은 특정 응용 프로그램에 대한 설정을 담고 있으며, ASP.NET 응용프로그램인 경우 Web.Config 이라는 이름을 갖고, 구성 파일은 응용 프로그램의 URL 경로에 위치한다.

마지막으로 윈도우 기반의 응용 프로그램의 구성 파일은 응용 프로그램과 같은 폴터에 위치하며, 응용 프로그램의 이름에 .config 이라는 확장자가 붙는 구조로 명명된다. 예를 들어, myApp.exe 라는 응용 프로그램의 구성 파일은 myApp.exe.config 이 된다.

보안정책 구성 파일

보안 정책은 엔터프라이즈 정책, 컴퓨터 정책, 사용자 정책의 세가 지 수준으로 구분되며, 수준에 따라 각각이 구성 파일을 가지고 있다. 먼저 엔터프라이즈 정책은 기업 전체에 적용되는 정책을 말하며, 해당 컴퓨터에만 적용되는 정책과 로그온한 사용자를 기반으로 적용되는 사용자 정책이 있다.

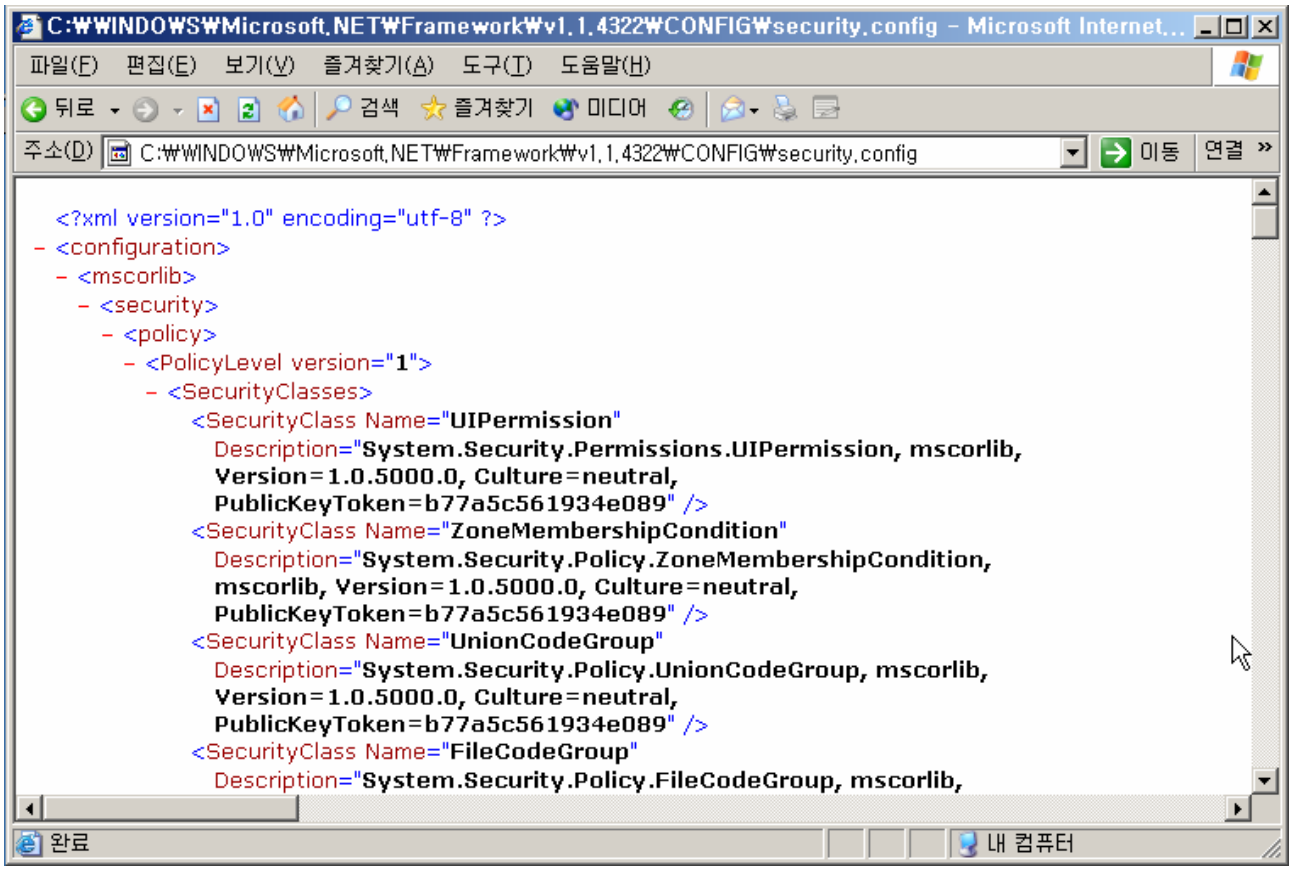

그림 7. 보안 정책 파일

구성 요소 서비스

구성요소 서비스를 한마디로 정의한다면, 마이크로 소프트의 미들웨어 (middleware) 라고 할 수 있다. 일반적으로 미들웨어는 한 기업 안에 설치된 다양한 하드웨어, 네트워크 프로토콜, 응용 프로그램, 운영 체제들의 차이를 메워주는 소프트웨어로 생각할 수 있다. 이 미들웨어의 기능에 트랜잭션 모니터링 기능이 부가된 것이 윈도우 2003의 구성요소 서비스이다.

구성요소 서비스 소개

구성요소 서비스는 COM+ 응용프로그램 수준에서 보안, 트랜잭션 등의 속성을 설정할 수 있도록 해주며, 이런 장치는 개발 프로젝트에서 일일이 보안과 트랜잭션 관리 기능을 구현하지 않고도 완성된 응용 프로그램을 개발 할 수 있도록 도와주는 역할을 한다.

COM+ 응용프로그램은 구성 요소 서비스의 관리와 보안의 기본 단위이며, 다음 그림과 같이 각 기능을 구현하는 여러 개의 컴포넌트로 구성된다.

그림 1. COM+ 구성요소

대부분의 구성 요소 서비스 관리 작업은 배포된 응용 프로그램 및 구성 요소에 대해 고성능과 보안을

보장하고 설정하기 위한 것이다. 수행하는 설정 작업은 응용 프로그램의 종류와 응용프로그램에서 사용하는 서비스에 따라 달라지겠지만, 구성 요소 서비스 관리 콘솔을 통하여 일관된 방식으로 수행할 수 있다. 구성 요소 서비스 관리 도구를 사용하여 COM 응용프로그램이나 COM+ 응용프로그램을 관리 할 수 있다.

COM 응용프로그램은 함께 동작하도록 개발 된 COM 구성 요소의 그룹을 뜻하는 용어이다. COM 응용프로그램의 예로는 마이크로소프트 엑셀 같은 프로그램을 들 수 있다. COM+ 응용프로그램은 대기열, 역할기반 보안등의 COM+ 서비스를 사용하기 위해 개발되고 구성된 COM구성 요소들의 그룹이다. COM+ 응용 프로그램이 가지는 몇 가지 특징은 COM+ 응용 프로그램이 구성요소 코드로 쓰여졌다는 것과 구성 요소 서비스 관리 도구를 통해 정의된다는 것이다.

또한 COM+ 응용 프로그램은 COM+ 서버 응용 프로그램과 COM+ 라이브러리 응용 프로그램의 두 가지 종류로 나뉘며, 응용 프로그램에 따라 관리 방식이 달라진다. 서버 응용프로그램은 자체 프로세스 공간에서 실행되고, 라이브러리 응용 프로그램은 다른 호스트의 프로세스에서 실행되도록 개발된다.

응용 프로그램 풀링 관리

응용프로그램 풀링 서비스는 단일 쓰레드 프로세스를 확장할 수 있도록 해준다. 이런 기능은 쓰레딩 모델이 정의되지 않은 레거시 코드가 단일 쓰레드 아파트먼트에서 실행되어야 할 때 매우 유용하다.

응용프로그램 풀링은 활성화를 처리할 수 있는 다른 프로세스를 제공하기 때문에 한 프로세스에서 오류가 발생할 경우 복수하는 데 도움이 된다. 예를 들면, 실행 프로세스가 3개 있고, 한 프로세스에서 오류가 발생했을 경우 나머지 프로세스는 영향을 받지 않기 때문에 작업 손실을 최소화 할 수 있다.

응용프로그램 풀링 구성

응용프로그램 풀링을 사용하면 지정된 값만큼의 Dllhost.exe 프로세스를 사용하여 응용프로그램 활성화를 처리하게 된다. 만약 7을 지정했다면, 처음 7개의 프로세스는 새로운 Dllhost.exe 프로세스로 시작하고, 그 이후는 라운드 로빈 방식을 사용하여 처리된다. 다음의 단계를 따라서 응용프로그램 풀링을 구성할 수 있다.

1. 구성요소 서비스 관리 도구의 콘솔 트리에서 구성할 COM+ 응용 프로그램을 선택한다. 컨텍스트 메뉴에서 [속성]을 선택한다.

2. [풀링 및 재생] 탭의 [응용프로그램 풀링]항목의 풀 크기를 원하는 값으로 설정한다. [풀 크기]가 1로 설정되어 있으면, 풀링 서비스는 사용되지 않는다.

응용 프로그램 재생관리

사용중인 응용프로그램에 문제가 있다는 사실을 발견한 경우에는 어떻게 대처할 것인가? 혹은 새로 개발된 응용프로그램에 아무런 문제가 없다고 장담할 수 있겠는가?

응용프로그램 재생은 알려진 문제의 수정과 예상치 못한 문제에 대한 예방책을 제공하여 응용프로그램의 전반적인 안정성을 높인다. 메모리 누수, 프로세스 오류 등의 문제로 실행 시간이 길어지면서 응용프로그램의 성능이 떨어지는 경우가 종종 있다. 그런 경우에 응용 프로그램 재생은 오류가 발생한 프로세스를 자동으로 종료시키고, 다시 시작하는 방법으로 프로세스를 초기화하고 메모리를 재 할당한다.

응용프로그램 재생 구성

응용 프로그램 재생은 응용 프로그램과 관련 있는 Dllhost.exe 프로세스의 복사본을 만들어서 실행시키는 방식으로 수행된다. Dllhost.exe 프로세스의 복사본은 앞으로 있을 모든 요청을 처리하고, 이전의 Dllhost.exe 프로세스는 개체에 대한 외부 참조가 종료되는 것을 감지하거나 만료 시간에 도달할 경우에 종료된다. 이러한 방식을 사용하여 응용 프로그램 재생은 응용프로그램 서비스가 중단되지 않도록 해준다. 다음의 순서에 따라서 응용프로그램 재생을 설정할 수 있다.

1. 구성요소 서비스 관리 도구의 콘솔 트리에서 구성할 COM+ 응용 프로그램을 선택한다. 컨텍스트 메뉴에서 [속성]을 선택한다.
2. 사용할 조건에 따라 [풀링 및 재생]탭에서 [최대 실행 시간 제한], [메모리 제한], [만료시간 제한], [호출 제한 및 활성화 제한]에 값을 입력한다. [최대 실행 시간 제한] 또는 [메모리 제한], [활성화 제한] 항목의 값을 0으로 설정하면 응용프로그램 재생이 중지된다.
   - 최대 실행 시간은 0~30,240 (21일) 사이의 값을 가지며, 기본 설정은 0 이다.
   - 메모리 제한은 0~1,048,576KB 사이의 값을 가지며, 기본 설정은 0이다.
   - 만료 시간 제한은 1~1440분 (24시간) 사이의 값을 가지며, 기본 설정은 15다.
   - 호출 제한은 0~1,048,576회 사이의 값을 가지며, 기본 설정은 0이다.
   - 활성화 제한은 0~1,048,576 사이의 값을 가지며, 기본 설정은 0이다.

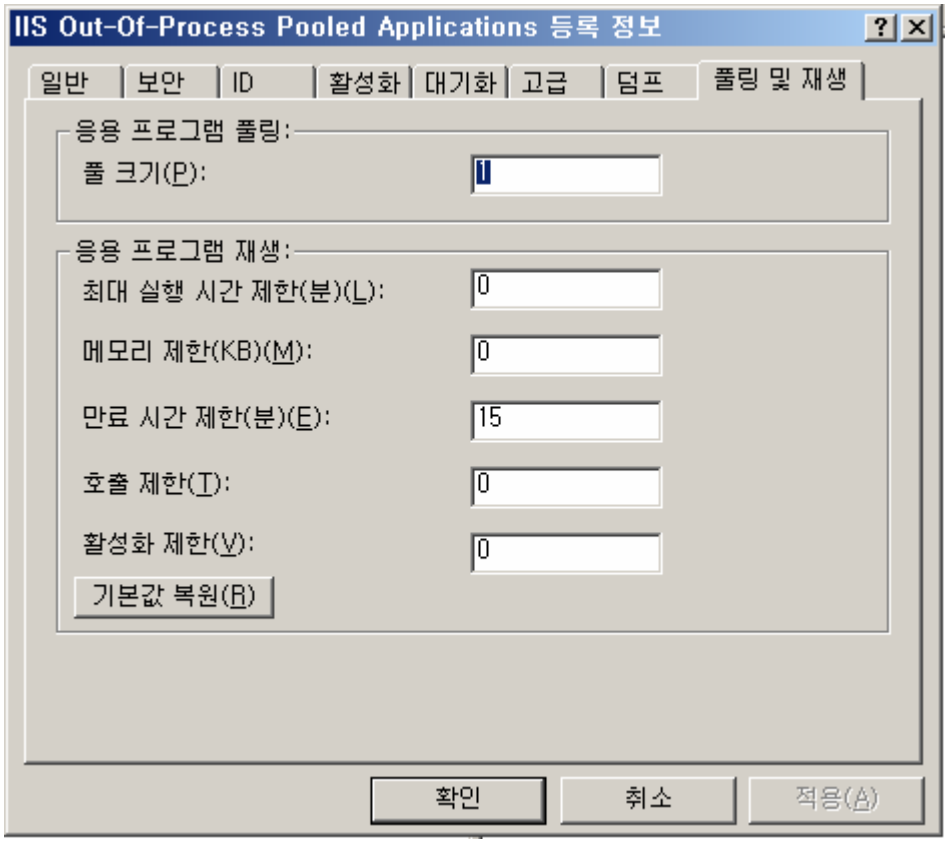

그림 2. 응용프로그램 풀링 및 재생

응용프로그램 보안 관리

구성요소 서비스는 COM+ 응용 프로그램, 데이터, 리소스 등이 실수나 고의적으로 손상되는 것을 방지할 수 있는 보안 모델을 제공한다.

역할 기반 보안

역할 기반 보안을 사용하면 COM+ 응용프로그램이나 COM+ 파티션에 대한 접근 제어를 강화할 수 있다. 역할이란 응용프로그램 리소스에 대한 접근 권한을 결정하기 위해 응용프로그램이나 COM+ 파티션에 정의된 사용자 그룹이다. 역할은 초기에 개발자에 의해서 응용프로그램에 연결되며, 그 후에는 시스템 관리자가 실제 사용자들로 할당한다.

역할을 할당하는 효율적인 방법은 윈도우 보안 그룹을 사용하는 것이다. 먼저 사용자 계정을 적절한 보안 그룹에 할당한 다음 해당 그룹을 적절한 역할에 연결하는 식이다. 따라서 COM+ 응용프로그램에서

역할 기반 보안을 사용하려면 해당 응용프로그램에 대한 역할 기반 인증 검사를 먼저 설정해야 한다. 이렇게 하면 해당 응용프로그램을 액세스 중인 사용자가 속한 역할을 확인한 다음 해당 응용프로그램에서 작업을 수행할 수 있는 권한을 부여한다. 다음의 순서에 따라서 역할기반 보안 설정을 할 수 있다.

1. 인증 검사를 사용할 COM+ 응용프로그램을 선택하고, 컨텍스트 메뉴를 사용하여 [속성]을 선택한다.
2. [응용프로그램 속성] 대화 상자에서 [보안]탭을 선택한다.
3. [인증]에서 [이 응용 프로그램에 대한 액세스 검사 수행]확인란을 선택한다.

위의 작업이 수행된 상태에서 실제 사용자 계정이나 그룹을 역할에 할당하여야 한다.

1. [구성요소 서비스]관리 도구의 콘솔 트리에서 사용자 계정 또는 그룹을 추가할 역할이 포함된 COM+ 응용 프로그램을 찾는다. 응용 프로그램 역할이 나타날 때까지 콘솔 트리를 확장한다.
2. 사용자 계정 또는 그룹을 추가할 역할을 선택한다.
3. [역할의 사용자 폴더]에서 컨텍스트 메뉴를 이용하여 [새로 만들기]를 선택한 다음 [사용자]를 선택한다.
4. [사용자 선택] 또는 [그룹 선택]창의 아래 창에 추가할 사용자 또는 그룹의 정식 이름을 입력하고 작업을 마친다.

이제 액세스 검사를 위한 보안 수준 설정을 해보자.

1. [구성요소 서비스]관리 도구의 콘솔 트리에서 보안 수준을 설정할 COM+ 응용 프로그램을 선택한다. 컨텍스트 메뉴를 이용하여 [속성]을 선택한다.
2. [응용프로그램 속성]대화 상자에서 [보안]탭을 클릭한다.
3. [보안 수준]에서 다음 중 하나를 선택한다.
   - 프로세스 수준에서만 액세스 검사를 수행: 역할에서 응용 프로그램에 할당된 사용자가 프로세스 보안 설명자 (Security Descriptor) 에 추가됨을 나타낸다.
   - 프로세스 및 구성 요소 수준에서 액세스 검사 수행: 프로세스 수준 보안 설명자 검사와 전체 역할 기반 보안 검사가 수행됨을 나타낸다.

이후에 응용 프로그램을 다시 시작하면 설정된 수준에 맞춰 보안 검사가 수행되고, 역할에 할당된 사용자에게만 액세스 권한이 주어진다. 다음의 단계를 수행하면 소프트웨어 제한 정책을 구성할 수 있다.

1. 제한 정책을 설정할 COM+ 응용 프로그램을 선택한 후, 컨텍스트 메뉴를 이용하여 [속성]을

선택한다.

2. [응용프로그램 속성]대화 상자에서 [보안]탭을 클릭한다.
3. [소프트웨어 제한 정책]에서 [소프트웨어 제한 정책 사용]확인란을 선택하여 신뢰 수준을 설정한다.
4. [제한 수준]에서 적합한 수준을 선택한다.
   - 허용 안 함: 응용프로그램에서 사용자의 모든 권한을 사용할 수 없다.
   - 제한 안 됨: 응용프로그램이 모든 사용자 권한을 사용할 수 있다.

COM+ 인증 설정

인증이란 클라이언트와 서버간에 자격증명을 사용하여 서로의 ID를 확인하고, 상호간 데이터를 검사하며, 암호화하는 프로세스이다. COM+에서 인증 설정과 효과는 서버응용프로그램인지 라이브러리 응용프로그램인지에 따라서 달라진다.

서버 응용프로그램에 대한 인증 수준을 설정하면 클라이언트가 해당 응용프로그램을 호출할 때 수행되는 인증 수준이 결정된다. 일반적으로 높은 인증 수준은 성능의 저하를 가져올 수 있지만, 높은 수준의 보안과 데이터 무결성을 제공한다.

1. 인증을 설정할 COM+ 서버 응용프로그램을 선택하고, 컨텍스트 메뉴를 이용하여 [속성]을 선택한다.
2. [응용프로그램 속성]대화상자에서 [보안]탭을 선택한다.
3. [호출 인증 수준]대화 상자에서 적합한 수준을 선택한다.
   - 없음: 인증이 이루어지지 않는다.
   - 연결: 연결이 이루어질 때만 자격 증명을 인증한다.
   - 호출: 모든 호출의 시작 지점에서 자격 증명을 인증한다.
   - 패킷: 자격 증명을 인증하고 호출 데이터가 모두 수신되었는지를 검사한다. (이 설정이 COM+ 서버 응용프로그램의 기본 값이다.)
   - 패킷 무결성: 자격 증명을 인증하고 전송 중에 호출 데이터가 변조되지 않았는지를 검사한다.
   - 패킷 개인 정보: 자격 증명을 인증하고, 데이터 및 보낸 사람의 ID와 서명을 비롯하여 해당 패킷을 암호화한다.

COM+ 라이브러리 응용 프로그램은 다른 프로세스에 의해 호스트 되기 때문에 그 보안 요구 사항이 서버 응용프로그램의 보안 요구 사항과 다를 수 있다.

COM+의 가장 수준 설정

가장이란 서버가 클라이언트를 대신하여 호출하고, 호출 시 서버ID와 자격증명 대신 클라이언트의 ID와 자격 증명을 제시하는 프로세스이다. 이는 클라이언트가 서버에게 자신의 ID를 사용하도록 명시적으로 허가한 경우에만 가능하다.

응용 프로그램에 대한 가장 수준을 설정하면 응용 프로그램이 다른 응용프로그램을 호출 할 때 자신의 ID를 사용하도록 권한 수준이 결정된다. 가장 수준은 COM+ 서버 응용 프로그램에 대해서만 설정할 수 있다. 라이브러리 응용 프로그램은 호스트 프로세스의 ID로 실행되며, 해당 프로세스에 지정된 가장 수준을 사용한다.

1. 가장을 설정할 COM+서버 응용 프로그램을 선택한 다음 컨텍스트 메뉴를 이용하여 [속성]을 선택한다.
2. 응용 프로그램 속성 대화 상자에서 [보안] 탭을 선택한다.
3. [가장 수준]에서 적절한 수준을 선택한다.
   - 익명: 클라이언트는 서버에 대해 익명으로 처리된다. 서버가 클라이언트를 가장할 수 있지만 가장 토큰에는 클라이언트에 대한 정보가 전혀 포함되지 않는다.
   - ID: 서버가 클라이언트 ID를 받아 클라이언트를 가장하여 액세스 검사 수준을 결정할 수 있다.
   - 가장: 이 설정은 COM+서버 응용 프로그램의 기본 설정이다. 서버가 자체 기능을 수행하면서 클라이언트를 가장할 수 있지만 약간의 제한이 있다. (클라이언트가 위치한 컴퓨터의 리소스를 서버가 액세스할 수 있다. 서버와 클라이언트가 같은 컴퓨터에 있을 때는 서버가 클라이언트로서 네트워크 리소스를 액세스할 수 있다. 서버가 클라이언트와 다른 컴퓨터에 있을 때는 서버가 위치한 컴퓨터의 리소스만 액세스할 수 있다.)
   - 대리인: 서버는 클라이언트와 같은 컴퓨터에 있는지 여부에 관계 없이 자체 기능을 수행하면서 클라이언트를 가장할 수 있다. 가장하는 동안, 클라이언트 자격 증명을 여러 대의 컴퓨터에 전달할 수 있다.

COM+ 파티션 관리

COM+ 파티션은 하나의 응용 프로그램이 다른 버전의 응용 프로그램과 격리된 환경에서 실행될 수 있도록 해 주는 논리적 컨테이너이다. 이런 개념은 하나의 컴퓨터 안에 자리잡은 가상 서버의 개념으로 생각할 수도 있다. 디스크 드라이브의 파티션을 생각해보자, 파티션을 사용하면 하나의 물리적 디스크 드라이브를 여러 개의 논리적 드라이브로 나누고, 각각에 별도의 설정을 적용해서 사용할 수 있다. COM+ 파티션은 이런 파티션 개념을 응용 프로그램 수준에 적용시킨 것으로 생각할 수 있다. 즉 COM+ 파티션을 사용하면 물리적인 서버 하나를 논리적으로 여러 개로 구성하여, 각각에 응용 프로그램을 설치하고, 개별적인 구성으로 사용할 수 있다.

운영 환경에서는 동일한 응용프로그램의 설정을 서로 다르게 설정하여 동시에 서비스하여야 하는 경우가 종종 발생한다. 예를 들면, 응용 프로그램을 교육용, 테스트용, 개발용으로 각각 구성하고, 동시에 실행시켜야 하는 경우를 생각할 수 있다.

여러 형태로 구성된 응용프로그램을 동시에 운영하려면, 각 구성 개수만큼의 컴퓨터가 필요해지고, 이는 비용과 관리상의 부담을 증가시킨다. COM+ 파티션을 사용하면 이런 문제를 효율적으로 해결할 수 있다. COM+ 파티션은 단일 컴퓨터에서 서로 다르게 구성된 응용프로그램을 동시에 설치하고, 사용할 수 있도록 해준다.

COM+ 파티션을 사용하면 응용프로그램의 배포와 관리에 있어서 많은 이점을 누릴 수 있다. COM+ 파티션을 활용할 수 있는 환경은 다양하다. 다음의 ASP (Application Service Provider) 시나리오는 COM+ 파티션 사용의 이점을 직관적으로 보여준다.

<예> 병원 관리용 소프트웨어를 서비스 하는 경우
어는 소프트웨어 개발 회사에서 병원 관리용 COM+ 응용 프로그램을 개발하여 서비스한다. 응용 프로그램의 클라이언트는 웹 브라우저를 통하여 웹 서비스와 연결되고, 환자의 진료 기록은 SQL 데이터 베이스에 저장된다. 서비스 공급자는 A, B, C의 세 병원 고객이 있다. 이때 각 고객은 클라이언트 측의 응용프로그램을 로컬에서 실행하고, 서버 측의 COM+ 응용프로그램은 서비스 공급자의 내부 웹 서버에서 실행된다. 각 병원은 각각의 환경에 따른 응용프로그램 설정을 요구할 것이고, 자체 데이터 저장 공간이 필요하다.

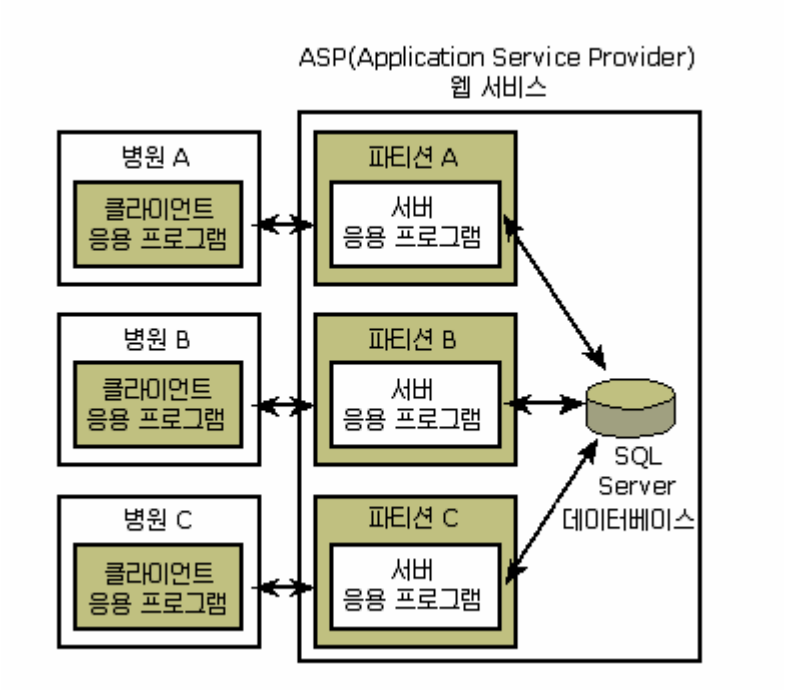

그림 3. COM+ 파티션 활용 시나리오

위의 환경에서 COM+ 파티션은 매우 유용하게 사용될 수 있다.

COM+ 파티션 사용

구성 요소 서비스 관리 도구를 사용해서 시스템에서 COM+ 파티션을 사용하도록 설정 할 수 있다.

1. 구성 요소 서비스 콘솔 트리에서 컴퓨터 폴더를 열고 컴퓨터를 선택한다.
2. 해당 컴퓨터 이름을 선택한 다음 컨텍스트 메뉴를 이용하여 속성 메뉴 항목을 선택한다.
3. 옵션 탭에서 파티션 사용 확인란을 선택한다.

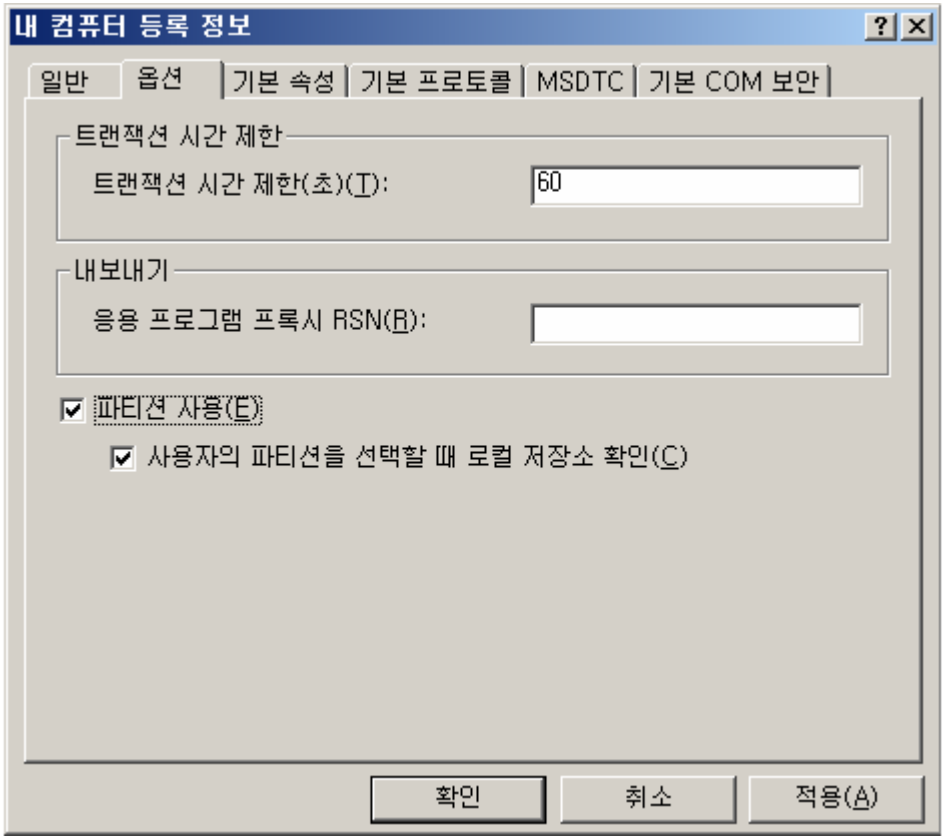

그림 4. COM+ 파티션 사용

도메인 수준의 COM+ 파티션 만들기

COM+ 파티션은 도메인과 로컬 수준에서 생성될 수 있다. 도메인 수준의 COM+ 파티션은 도메인 환경에서 COM+ 응용프로그램 관리를 위한 기본적인 수단이 된다. 예를 들어, 조직 내의 각 업무 부서를 액티브 디렉터리의 OU로 매핑하고, 각 OU를 도메인 수준의 COM+ 파티션에 할당하는 방식은 액티브 디렉터리 관리의 연장선상에서 아주 자연스러운 발상이다.

도메인 수준의 COM+ 파티션을 만들려면 [Active Directory 사용자 및 컴퓨터] 관리 도구를 사용하여 액티브 디렉터리 내에 파티션과 해당 파티션에 대한 파티션 집합을 만들어야 한다. 이 후에 [구성요소 서비스] 관리 도구를 사용하여 COM+ 응용 프로그램 서버에 액티브 디렉터리의 파티션과 동일한 파티션을 만들어야 한다. 액티브 디렉터리의 COM+ 파티션과 동일한 GUID를 갖는 COM+ 파티션을 응용프로그램 서버에 만들면 해당 파티션이 링크되고, 단일 도메인 차원의 COM+ 파티션이 된다.

다음의 단계를 수행하여 Active Directory에 COM+ 파티션 만들 수 있다.

1. Active Directory 사용자 및 컴퓨터 관리 도구를 사용하여 보기 메뉴에서 고급 기능을 선택한다.
2. 도메인의 system 폴더에서 ComPartitions 폴더를 마우스 오른쪽 단추로 클릭한 다음 새 파티션을 선택한다.
3. 새 파티션의 일반 속성 페이지에서 새 파티션에 대한 이름과 설명을 입력하고 확인을 클릭한다.

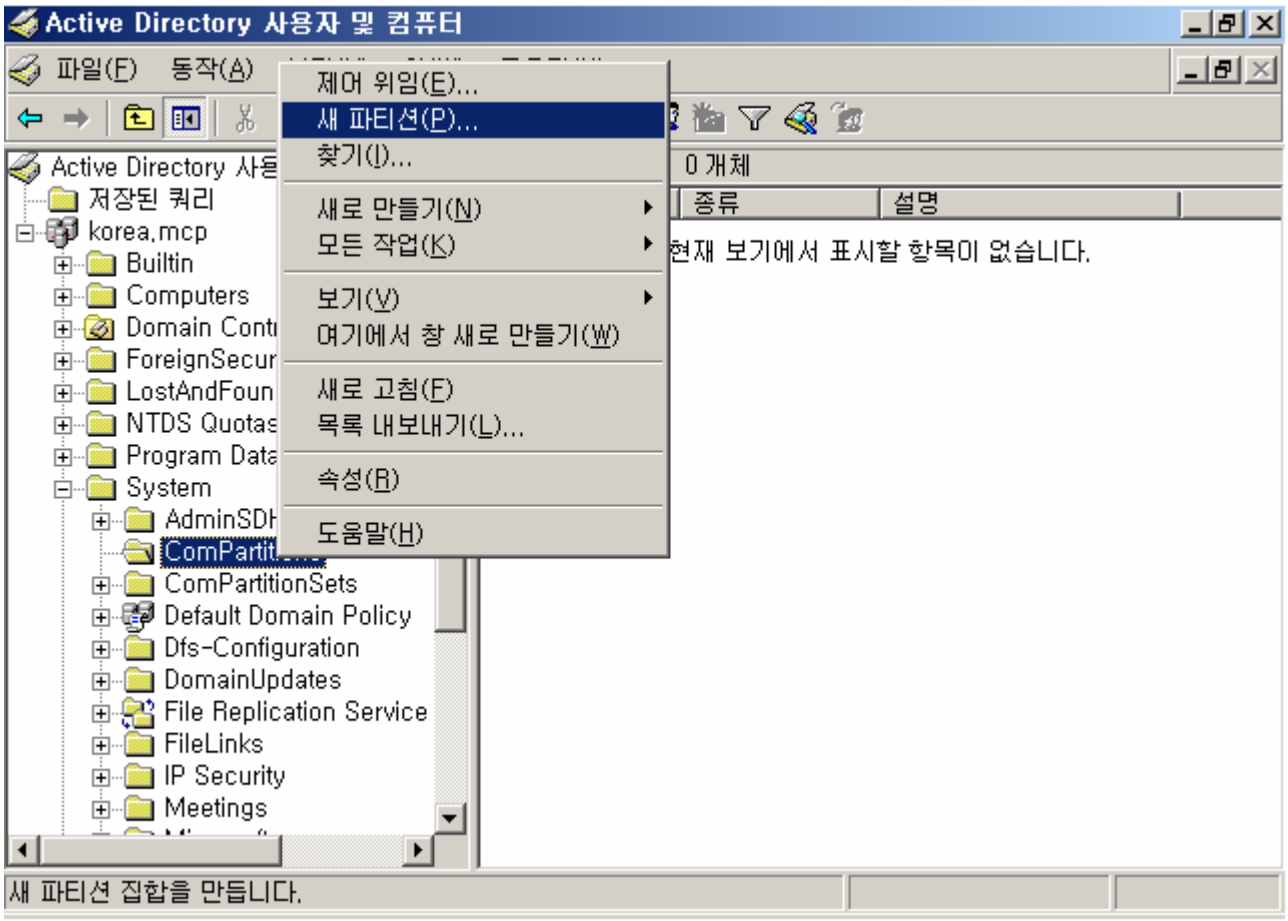

그림 5. COM+ 파티션 생성

Active Directory에 파티션 집합 만들려면 다음의 단계를 수행하라.

1. 도메인의 system 폴더에서 ComPartitinSets 폴더를 마우스 오른쪽 단추로 클릭한 다음 새 파티션 집합을 선택한다.
2. 새 집합의 일반 속성 페이지에서 새 파티션 집합에 대한 이름과 설명을 입력하고 다음을 클릭한다.

3. 새 집합의 파티션 구성 페이지에서 집합에 포함 시키려는 파티션을 클릭한 다음 추가를 클릭하고 다음을 클릭한다.
4. 새 집합의 사용자 구성 페이지에서 사용자 추가 및 조직 구성 단위 추가를 클릭하여 파티션 집합에 매핑될 사용자와 조직 구성 단위를 추가한 다음 마침을 클릭한다.

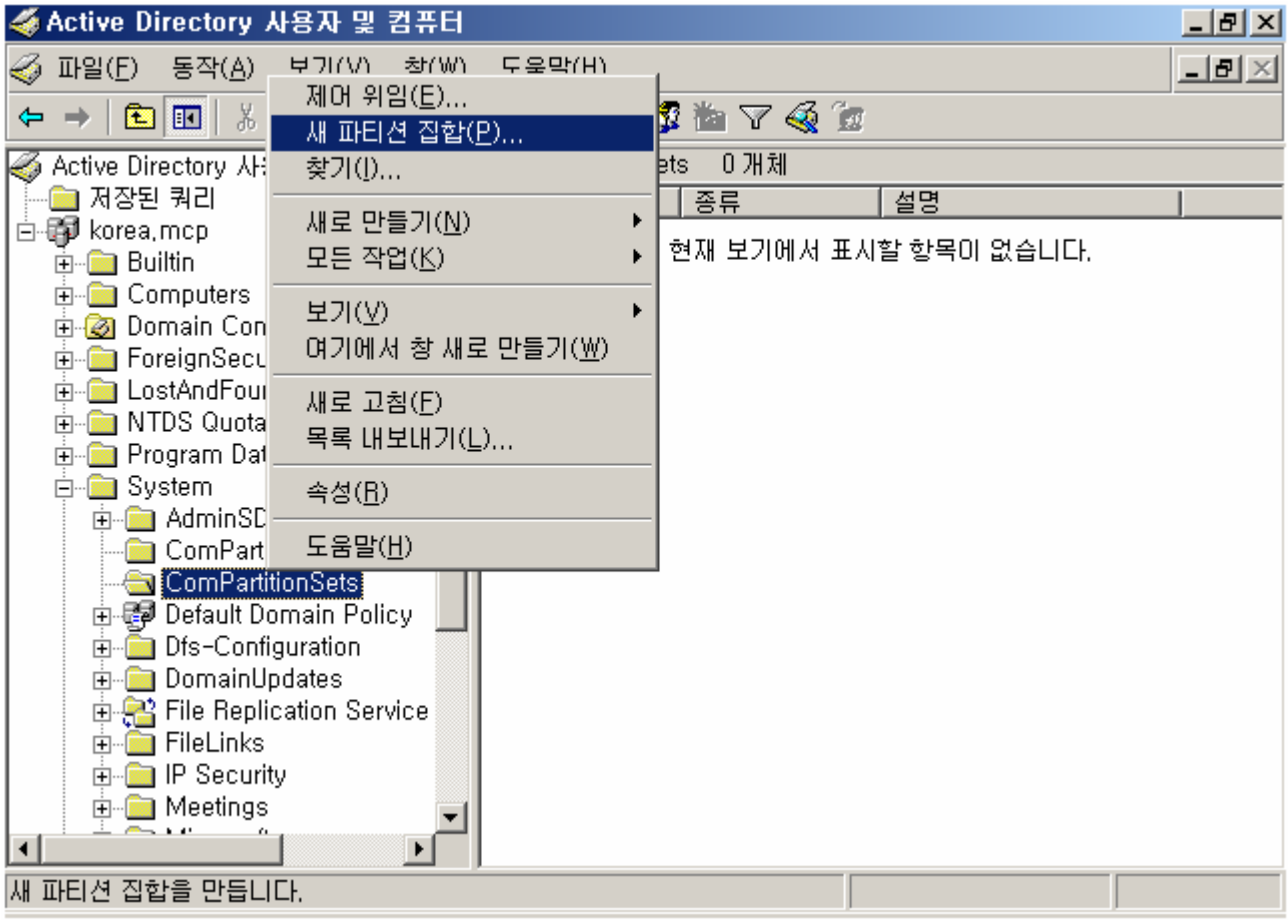

그림 6. COM+ 파티션 집합 생성

도메인 사용자나 조직 구성 단위를 파티션 집합에 매핑하려면 다음의 단계를 수행하라.

1. Active Directory 사용자 및 컴퓨터 관리 도구에서 파티션 집합에 매핑할 사용자 이름이나 조직 구성 단위를 마우스 오른쪽 단추로 클릭한 다음 속성을 클릭한다.
2. COM+파티션 탭의 파티션 집합 목록 상자에서 사용자나 조직 구성 단위를 매핑할 파티션 집합을 선택한다.

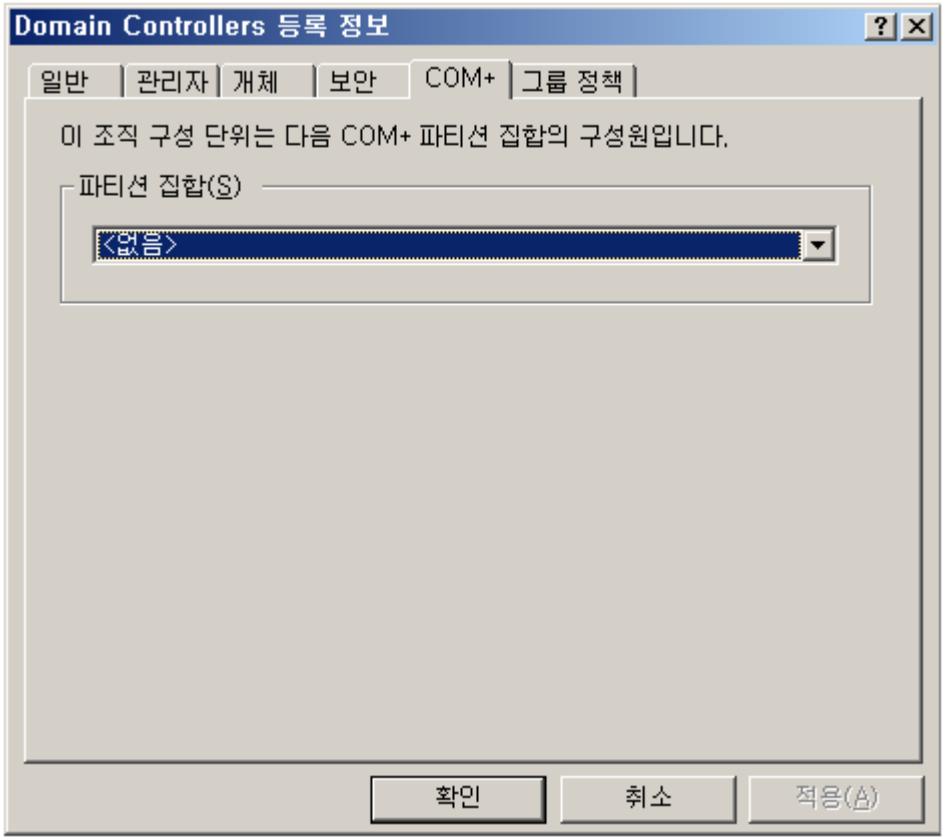

그림 7. 파티션 집합 매핑

다음은 응용 프로그램 서버에 파티션을 만들어 Active Directory의 파티션에 링크하기 위한 단계이다.

1. 응용 프로그램 서버에 COM+ 파티션을 사용하도록 설정 했는지 확인한다.
2. 구성 요소 서비스 콘솔 트리에서 파티션을 만들 컴퓨터의 이름을 찾는다.
3. COM+파티션 폴더를 마우스 오른쪽 단추로 클릭하고 새로 만들기를 선택한 다음 하위 메뉴에서 파티션을 클릭한다.
4. COM+파티션 설치 마법사에서 다음을 클릭한다.
5. 빈 파티션 만들기를 클릭한다.
6. 디렉터리 찾아보기 단추를 클릭하고 링크하려는 COM+파티션을 선택한다.
7. 다음을 클릭해서 작업을 마친다.

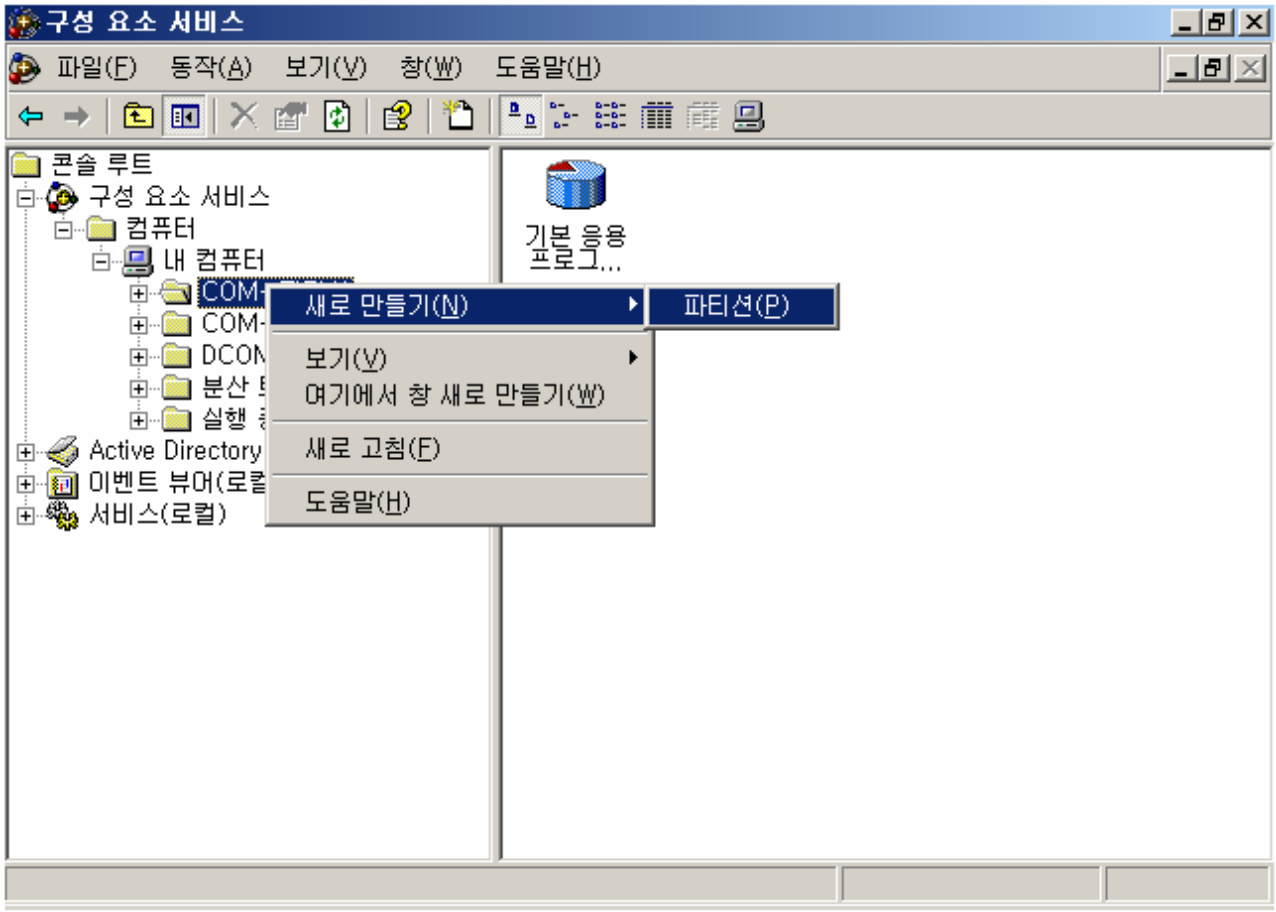

그림 8. 파티션 만들기

새로운 COM+ 파티션 사용자에게 기본 COM+ 파티션 할당

사용자가 COM+ 응용 프로그램을 액세스 할 때 COM+ 서비스는 해당 응용 프로그램에 대한 사용자의 기본 파티션을 검색한다. 다음의 단계에 따라 사용자의 기본 파티션을 설정할 수 있다.

1. 구성 요소 서비스 관리 도구의 콘솔 트리에서 사용자가 있는 컴퓨터의 이름을 찾는다.
2. COM+ 파티션 사용자 폴더를 마우스 오른쪽 단추로 클릭하고 새로 만들기를 선택한 다음 사용자를 선택한다. 기본 파티션 할당 대화 상자가 표시된다.
3. 추가를 클릭하면 사용자 선택 대화 상자가 표시된다.
4. 사용자 이름을 입력하고 작업을 마친다.

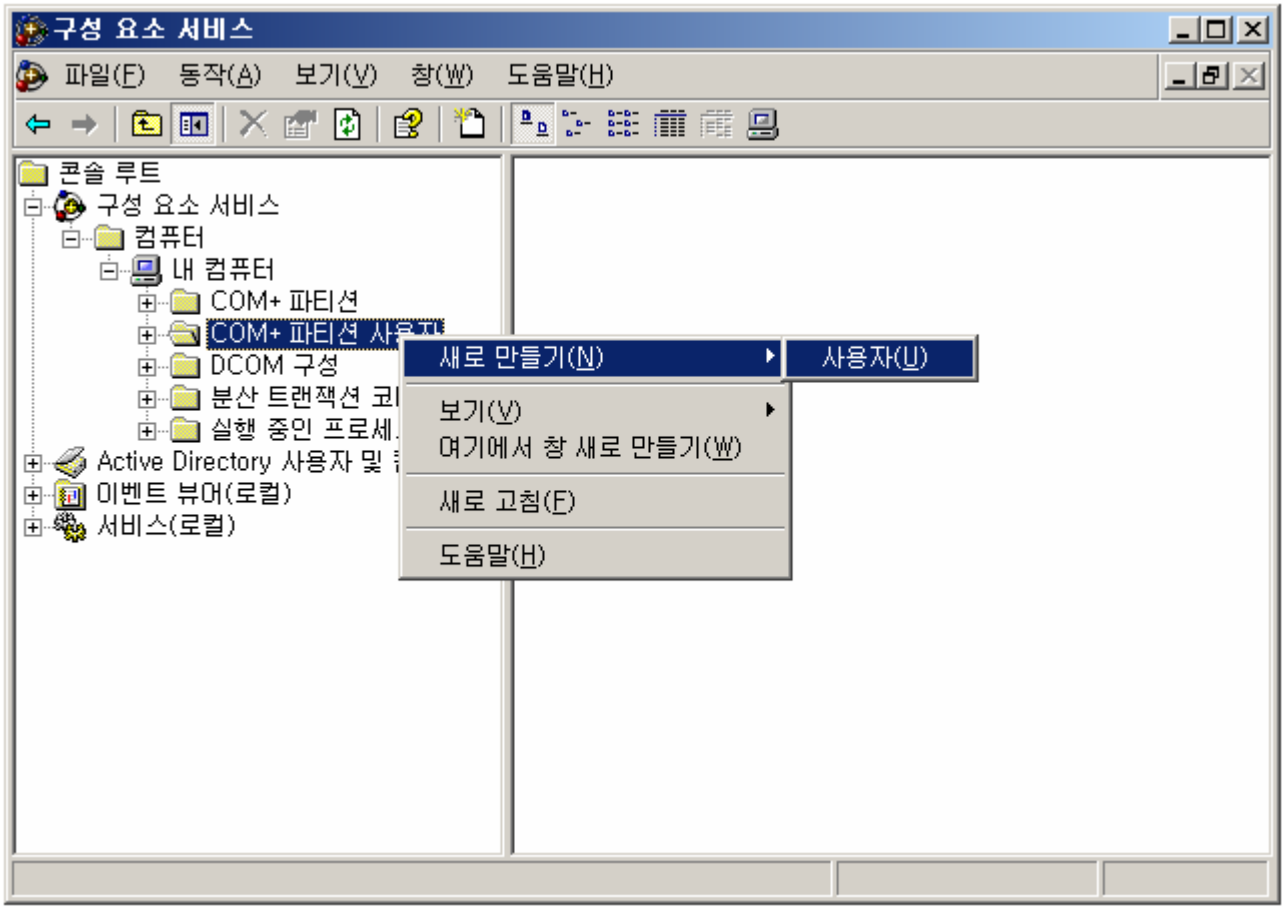

그림 9. COM+ 파티션 할당

로컬 COM+ 파티션 만들기

도메인이 아닌 환경에서 COM+ 파티션을 사용하게 될 경우도 있을 것이다. 도메인 수준의 파티션 대신 로컬 파티션을 사용하기로 결정했다면, 다음의 순서를 따라 설정 작업을 진행할 수 있다. 도메인의 다른 사용자는 사용할 수 없는 로컬 COM+ 파티션을 만들어 보자.

1. 구성 요소 서비스 콘솔 트리에서 파티션을 만들 컴퓨터의 이름을 찾는다.
2. COM+파티션 폴더를 마우스 오른쪽 단추로 클릭하고 새로 만들기를 선택한 다음 하위 메뉴에서 파티션을 클릭한다..
3. COM+ 파티션 설치 마법사에서 다음을 클릭한다.
4. 빈 파티션 만들기를 클릭하고 새 파티션의 이름을 입력한다.
5. 새 파티션에 특정 GUID를 지정하려면 파티션 필드에서 기존 파티션 ID를 삭제하고 새 파티션 ID를 입력한다.
6. 다음을 클릭해서 작업을 마친다.

활성자 역할에 사용자 할당

특정 COM+ 파티션 내에서 COM+ 구성 요소를 활성화 할 수 있는 권한을 사용자에게 부여하려면 사용자를 해당 파티션의 활성자 역할에 할당하여야 한다.

1. 구성 요소 서비스 관리 도구의 콘솔 트리에서 COM+ 파티션이 설치된 컴퓨터의 이름을 찾는다.
2. COM+파티션 폴더를 확장한다.
3. 새 활성자를 추가할 파티션을 확장한다.
4. 역할 폴더, 활성자의 순서로 폴더를 확장한다.
5. 사용자 폴더를 마우스 오른쪽 단추로 클릭하고 새로 만들기를 선택한 다음 사용자를 선택한다.
6. 활성자 역할에 추가할 사용자의 계정을 입력하고 작업을 마친다.

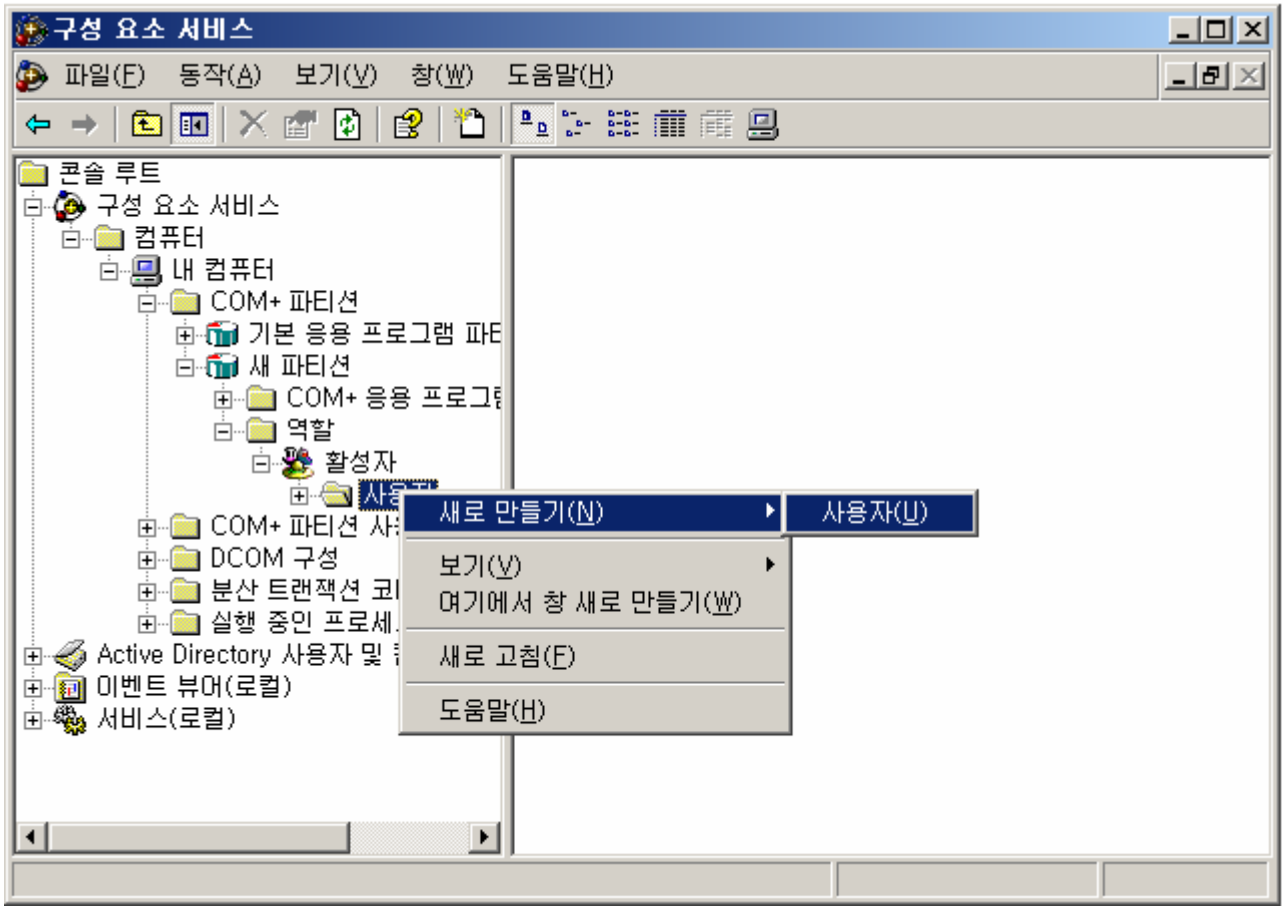

그림 10.활성자 역할에 사용자 할당

COM+ 파티션 관리 할당

모든 COM+파티션의 관리를 한 사용자에게 위임하려면 COM+ System Application 내에서 사용자를 Administrator 역할에 할당하여야 한다.

1. 구성 요소 서비스 관리 도구의 콘솔 트리에서 COM+ 응용 프로그램이 설치된 컴퓨터의 이름을 찾는다.
2. COM+파티션 폴더가 표시될 때까지 콘솔 트리의 보기를 확장한다.
3. 기본 응용 프로그램 파티션 아이콘을 확장한다.
4. COM+ 응용 프로그램, System Application 순서로 폴더를 확장한다.
5. 역할, 관리자의 순서로 폴더를 확장한다.
6. 사용자 폴더를 마우스 오른쪽 단추로 클릭하고 새로 만들기를 선택한 다음 사용자를 선택한다.
7. 관리자 역할에 추가할 사용자의 계정을 입력하고 작업을 마친다.

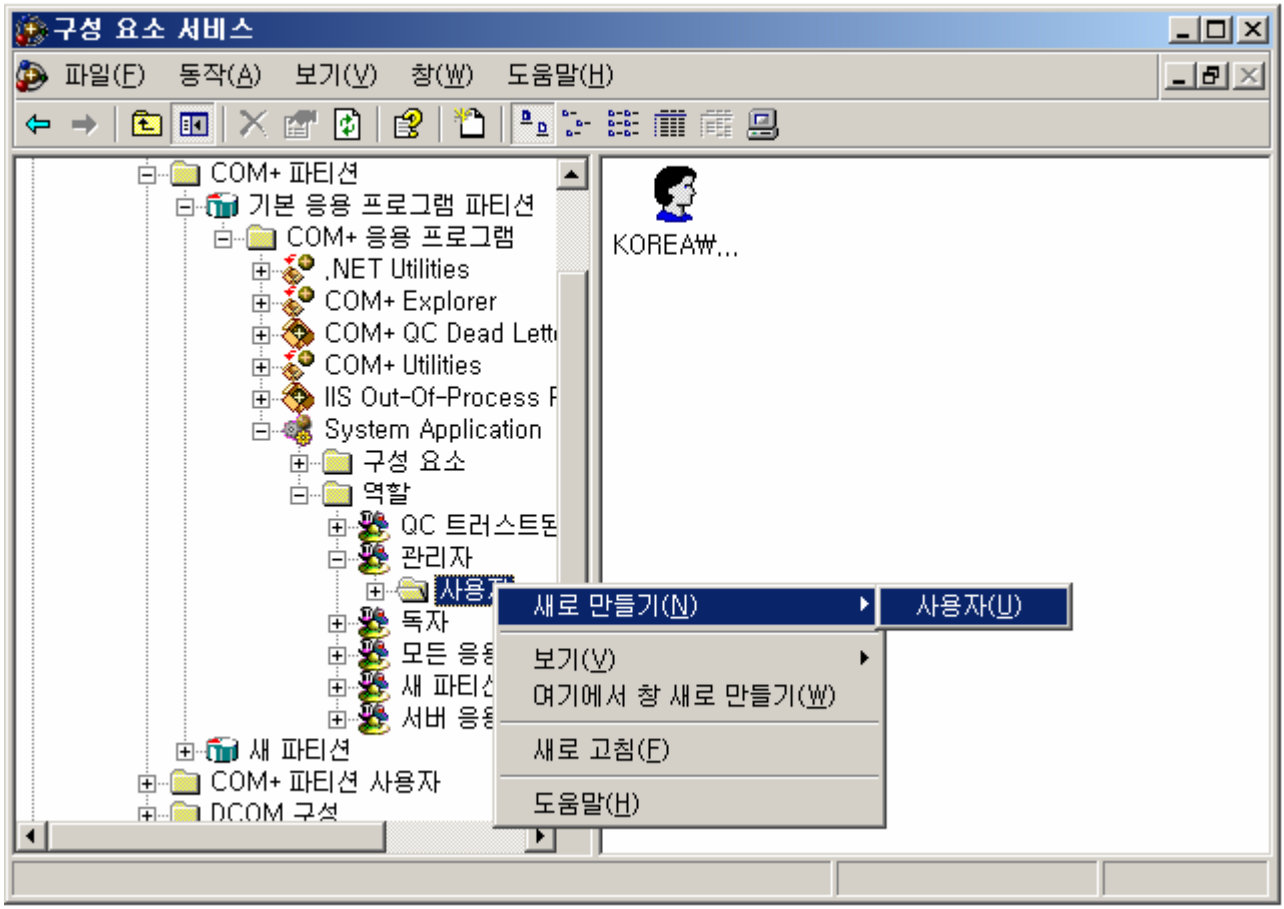

그림 11. COM+ 파티션 관리 할당

COM+ 파티션 내보내기

COM+파티션 내보내기 마법사를 사용하면 기존 파티션의 구성과 내용을 추출하고 패키징하여 다른 컴퓨터로 이동시킬 수 있다.

1. 구성 요소 서비스 관리 도구의 콘솔 트리에서 내보내려는 COM+ 파티션을 포함하는 컴퓨터의 이름을 찾는다.
2. COM+파티션 폴더를 확장한다.
3. 내보낼 COM+파티션을 마우스 오른쪽 단추로 클릭하고 내보내기를 선택한다. COM+파티션 내보내기 마법사가 표시된다.
4. 마법사를 사용해서 작업을 완료한다.

COM+파티션 내보내기 마법사에서는 Windows Installer 패키지와 캐비닛 파일을 만든다. 이동 하려는 컴퓨터에 두 개의 파일을 복사하고 패키지를 설치할 수 있다.

COM+ SOAP 서비스 관리

[구성요소 서비스]관리 도구에서 [SOAP]사용 확인란을 선택하는 간단한 설정으로 COM+ 응용프로그램을 웹 서비스로 노출 시킬 수 있다. 또한 COM+를 사용하여 배포했는지에 관계 없이 XML 웹 서비스를 투명하게 COM 구성요소로 사용할 수 있다. 이런 기능을 이용하여 기존의 COM+ 응용프로그램 로직을 쉽게 XML 웹 서비스로 제공할 수 있고, 또한 XML웹 서비스를 새로운 COM+ 응용프로그램으로 통합 할 수 있다.

COM+ SOAP 서비스 개요

HTTP의 사용은 네트워크를 통해 원격 서버에 있는 데이터를 웹 브라우저로 쉽게 액세스 할 수 있게 함으로써 컴퓨터 사용에 일대 전환을 가져왔다. 브라우저가 HTTP를 이용하는 방식은 형식화된 텍스트 데이터를 가져오는 수준에 머물렀으나, SOAP을 사용하면 클라이언트 응용프로그램이 XML 웹 서비스의 메서드를 원격 호출할 수 있게 된다.

클라이언트 응용프로그램에서 원격 서버에 구현된 메서드를 호출할 수 있도록 하는 기능은 매우 유용하

다. 이런 방식을 사용하면 개발자가 해당 기능을 사용하는 모든 응용프로그램을 재 배포하지 않고도 서버 측의 구현을 업그레이드 함으로써 응응 프로그램의 성능을 향상 시킬 수 있다.

웹 페이지와 마찬가지로 XML 웹 서비스는 IIS 와 같은 웹 서버에서 HTTP를 사용하여 액세스될 수 있다. 하지만 이런 종류의 HTTP 패킷에는 HTML로 인코드 된 웹 페이지 대신 SOAP으로 인코드 된 서버 측 메서드를 호출하는 입력 및 출력 매개변수가 포함된다.

윈도우 서버 2003의 구성요소 서비스는 모든 COM+ 응용프로그램에 구성된 COM 구성요소의 기본 인터페이스에 메서드를 노출시켜서 XML 웹 서비스로 사용할 수 있는 기능을 제공한다. 이런 기능을 염두에 두고서 구성요소를 작성하려면 노출시킬 메서드가 구성요소의 기본 인터페이스여야 한다는 것과 구성 요소를 서버의 COM+ 카탈로그에 등록해야 한다는 것 정도만 주의하면 된다.

COM+를 사용하여 XML웹 서비스를 만들면 COM+나 심지어 윈도우를 사용하지 않는 클라이언트라 할지라도 권한만 있다면 웹 서비스를 이용할 수 있다. XML웹 서비스로 노출된 구성요소에 대한 액세스는 IIS 와 COM+ 응용프로그램 모두의 액세스 제어 방식을 병행 적용할 수 있다.

COM+ SOAP 서비스 보안 고려 사항

닷넷 프레임워크 1.1. 을 사용할 경우, COM+ SOAP 서비스를 사용하여 만든 XML 웹 서비스는 해당되는 모든 COM+ 응용프로그램의 역할 기반 설정을 사용한다. 이들 설정을 사용하면 클라이언트와 서버 사이의 통신을 암호화 할 수 있고, 클라이언트 ID의 인증서 기반 인증을 요구할 수 있다.

COM+ 응용 프로그램에 대해 패킷 인증 및 역할 기반 보안 사용을 중지하면, 시스템에 대해 네트워크 액세스 권한을 가진 모든 클라이언트가 해당 XML 웹 서비스에 익명으로 액세스 하도록 할 수 있다.

COM+ 응용프로그램에 대해 패킷 인증과 역할 기반 보안을 사용하명, 권한을 가진 COM+ 클라이언트만이 해당 XML 웹 서비스를 액세스할 수 있다. 보안 모드에서는 익명 액세스가 허용되지 않으며, WSDL 파일이 게시되지 않는다. XML 웹 서비스를 보안 모드에서 만든 경우는 보안되는 않는 모드로 변경할 수 없다.

기본적으로 암호화 되지 않은 SOAP 통신은 HTTP 포트(80)에서 수신되며, 암호화 된 SOAP 통신은 HTTPS 포트 (443)에서 수신된다. 클라이언트에서 XML 웹 서비스를 성공적으로 액세스하려면 TCP SYN 패킷이 적절한 서버 포트에 전달되도록 클라이언트와 서버 사이의 방화벽을 구성해야 한다.

XML 웹 서비스 보안

닷넷 프레임워크 1.1 은 COM+ 응용프로그램의 속성 페이지 중 [보안] 탭의 설정을 따른다.

SOAP 을 사용하는 응용 프로그램 내보내기

COM+ SOAP 서비스를 사용하여 응용프로그램을 XML 웹 서비스로 노출하면 권한이 있는 모든 클라이언트가 해당 서비스를 사용할 수 있지만, COM+ 클라이언트에서 SOAP을 사용하는 COM+ 서버 응용프로그램을 사용하는 것이 훨씬 간결하다. 그렇게 하려면 프록시 모드에서 SOAP을 사용하는 응용프로그램으로 내보낸 다음 이를 해당 XML 웹 서비스를 액세스할 클라이언트로 가져오면 된다.

SOAP 을 사용하는 응용프로그램 가져오기

SOAP을 사용하는 응용 프로그램에서 서버에서 프록시 모드로 내보내면, 이 응용프로그램을 가져오는 클라이언트는 그에 포함된 구성요소의 메서드를 액세스 할 수 있게 된다. 이 방법을 사용하여 가져온 응용 프로그램의 구성요소를 클라이언트에서 사용할 경우 해당 구성 요소는 클라이언트에서 실행되지 않고, 해당 응용프로그램을 내보낸 서버가 제공하는 XML 웹 서비스 형태로 원격으로 액세스하게 된다. 따라서 네트워크를 통해 SOAP을 사용하는 COM+응용프로그램의 기능을 쉽게 사용할 수 있다.

대기중인 구성요소 관리

대기중인 COM+ 구성요소 기능을 사용하면 클라이언트는 메시지 대기열을 통해서 COM+ 서버 응용 프로그램의 서비스를 요청할 수 있다. 클라이언트 응용프로그램의 요청은 아웃바운드 대기열에서 기록되고, 메시지 대기열을 통해 서버 응용프로그램 대기열로 전송되며, 서버 응용프로그램이 요청을 처리하기 전까지 해당 요청은 서버 응용 프로그램의 대기열에 저장된다.

대기중인 구성요소 서비스를 이용하면 비 동기 클라이언트-서버 통신, 트랜잭션 클라이언트-서버 통신, 인증된 클라이언트-서버 통신을 구현할 수 있다.

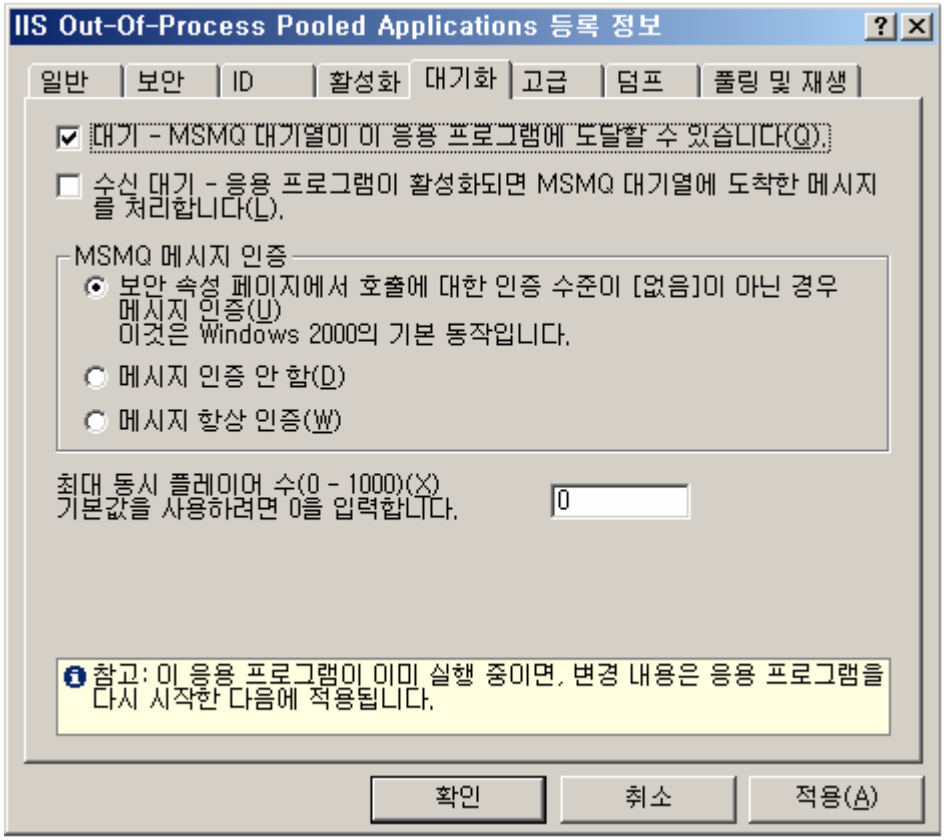

그림 12. 대기중인 구성요소 사용

메시지 큐 (Message Queuing)

정보를 저장하는 방식에는 여러 가지 방법이 있다. 이제는 너무 구식처럼 보이는 텍스트 기반의 저장 방식부터, 고도로 전문화된 DBMS, 혹은 근래에 들어서 주목 받는 XML 기술 까지 각각의 방식은 나름대로의 영역을 확보하고 사용되고 있다. SQL 서버의 경우는 여러 개의 테이블을 갖는 데이터베이스에 정보를 나누어 저장한다. 메시지 큐는 정보를 각 작업 별로 분류하고 개개의 분류마다 별도의 큐를 사용하여 저장한다. 예를 들면 전자상거래 솔루션이나 데이터마이닝 솔루션에 필요한 정보를 제공할 목적으로 전용의 큐를 생성하고 사용할 수 있다.

그림 1. 메시지 큐를 사용한 메시지 전송

메시지 큐의 종류

메시지 큐잉이 동작하기 위해서는 몇 가지 종류의 큐가 필요하다.

퍼블릭 큐 (Public Queues)

퍼블릭 큐는 액티브 디렉토리에 퍼블리싱 되고, 도메인 안에서 복제가 된다. 퍼블릭 큐를 사용하면, 해당 큐가 존재하는 컴퓨터의 이름을 모르는 경우에도 아무 문제 없이 정보를 검색하고 살펴볼 수 있다. 또한 그러한 이유로 한 컴퓨터에 있던 큐를 다른 컴퓨터로 옮기더라도, 그 큐를 사용하던 클라이언트는 아무런 영향도 받지 않는다.

프라이비트 큐 (Private Queues)

프라이비트 큐는 워크그룹 환경에서 사용될 수 있다. 네트워크 환경에서 전용의 메시지 큐잉 서버가 없는 경우, 프라이비트 큐는 인증기능을 제공하지 못한다. 또한 프라이비트 큐를 사용할 때는 해당 큐가 존재하는 컴퓨터의 이름을 알고 있어야 한다. 프라이비트 큐는 개발 환경이나, 소규모의 용도에 적합하다.

아웃고잉 큐 (Outgoing Queues)

아웃고잉 큐는 로컬에서 외부로 보내질 메시지가 전송되기 전에 보관되는 장소로 쓰이는 큐이다. 만약 컴퓨터가 어프라인 상태로 오래 유지된다면, 메시지는 아웃고잉 큐에서 보관되다가 네트워크 연결이 이루어지면 자동적으로 전송될 수 있다. 또한 아웃고잉 큐에 보관된 메시지는 수동으로 만들어지거나 지워지지 않는다.

커넥터 큐 (Connector Queues)

커넥터 큐는 메시지 큐 서버내에 존재한다. 커넥터 큐는 다른 외부 큐들에 대한 프락시 역할을 하며, 다른 메시징 시스템에 의해서 관리된다. 예를 들면, HIS 2000 (Host Integration Server) 에 포함되어 있는 MSMQ-MQSeries 브릿지를 사용하면, 윈도우 응용프로그램이 IBM MQ eries와 상호 작동하도록 구성할 수 있다. 비슷한 방식으로 MQC (Message Queuing Connector)를 사용하여 유닉스나 AS/400 같은 비 윈도우 기반의 기종들과의 연결을 설정할 수 있다. 커넥터 큐는 외부의 다른 이기종에서 제공되는 큐를 윈도우 기반의 큐처럼 이용할 수 잇게 하는 기능을 한다. 커넥터 큐는 서로 다른 메시지 큐 제품간의 상호 동작을 가능하게 해 준다.

저널 큐 (Journal Queues)

저널 큐는 이미 보내졌거나, 처리된 큐의 복사본을 저장한다. 저널 큐에는 소스 (Source)과 타겟 (Target)의 두 가지 타입이 있다. 메시지를 전송할 때 전송자는 메시지의 복사본을 현재의 컴퓨터에 저장하도록 할 수 있다. 소스 저널링은 각 메시지를 기반으로 작동하고, 타겟 저널링은 각 큐를 기반으로 동작한다. 만약 목적지 큐에 저널링이 활성화 되어 있다면, 목적지 큐는 전송 받는 모든 메시지의 복사본을 별도의 분리된 저널 큐에 저장한다.

데드-레터 큐 (Dead-Letter Queues)

데드-레터 큐는 성공적으로 전송되지 못하고 만료되어버린 메시지를 저장한다. 어떤 경우에는 목적 큐에 도달하거나, 전달된 후에 메시지가 만료되어 버리는 경우도 있다. 앞의 경우가 발생하면 그러한 메시지들은 데드-레터 큐에 보관된다. 다음과 같은 경우에는 메시지가 제대로 전송되지 못하는 경우가 발생할 수 있다.

- 큐의 용량이 다 차버린 경우, 큐를 찾을 수 없는 경우, 전송자가 목적지 큐를 알지 못하는 경우
- 비-트랙잭션 메시지가 트랜잭션 큐로 보내진 경우.
- 메시지에 설정된 최대 홉(hop) 수나, 시간을 초과한 경우.

관리 큐 (Administration Queues)

관리 큐는 승인 메시지를 저장하기 사용된다. 메시지 큐가 다른 큐들을 관리하고, 조사 목적으로 내부적으로 사용하는 큐들과는 다른 목적을 가진다.

리포트 큐 (Report Queues)

리포트 큐는 메시지 라우팅 트래킹이 활성화 된 경우, 보낸 메시지를 추적하기 위하여 사용된다. 하나의 컴퓨터에는 한 개의 리포트 큐만 존재할 수 있다.

메시지 큐 (Message Queue) 서비스

메시지 큐잉은 메시지를 분배하고 전송 받기 위한 메커니즘으로 윈도우 서비스를 사용한다.

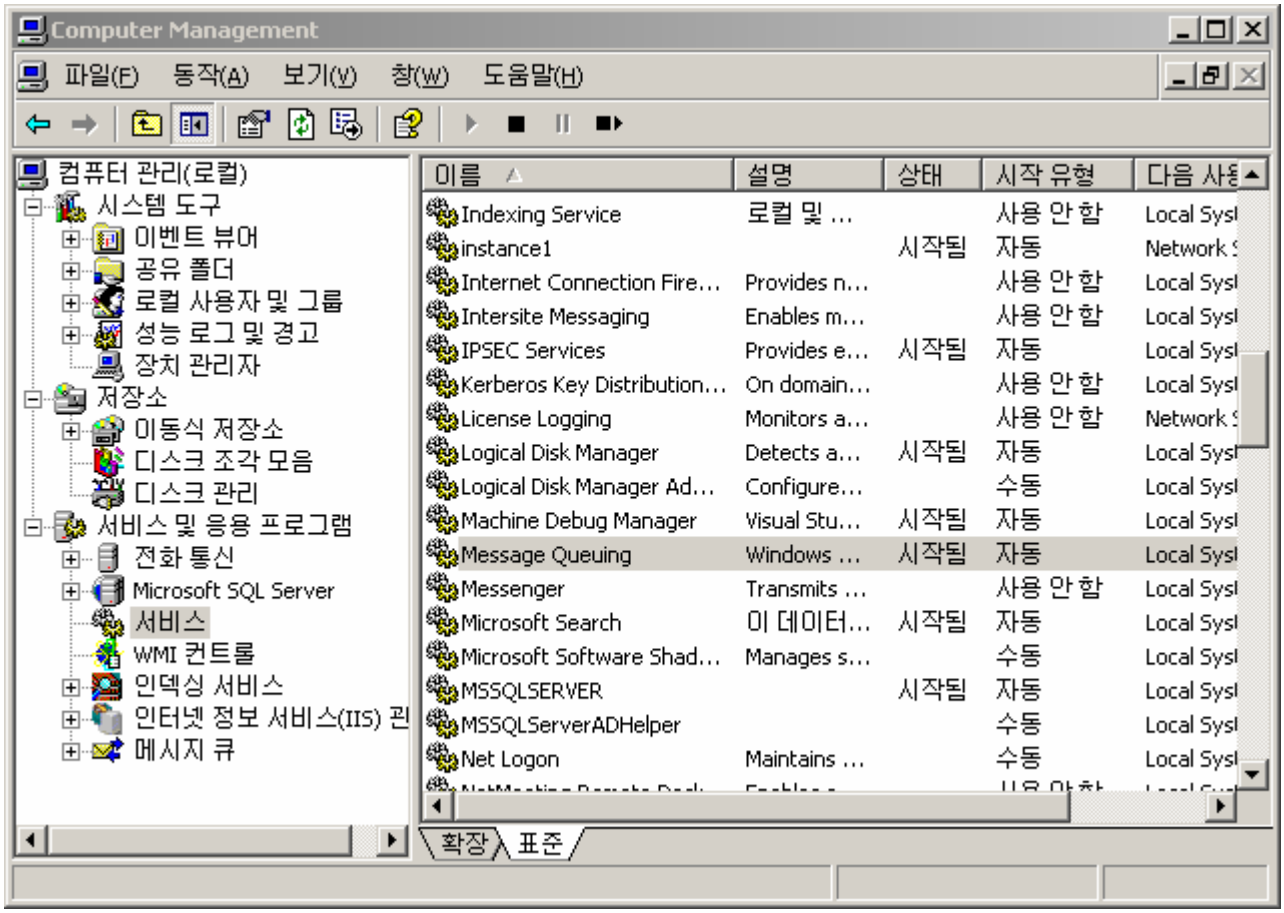

그림 2. 메시지 큐잉 서비스

메시지 큐 설정

메시지 큐는 프로그램적으로 생성하고 제거할 수 있지만, 대부분의 경우에는 컴퓨터 관리 콘솔을 사용하여 생성하고 설정하는 것이 간편하다.
메시지 큐는 컴퓨터 관리 콘솔에서 간단히 생성하고, 필요한 속성을 설정할 수 있다. 주요한 속성 중에 '트랜잭션' 속성이 있다. 트랜잭션 큐는 메시지가 전송된 순서대로 도착함을 보장해준다. 하지만 관계형 데이터베이스에서처럼 '원자성'이 보장되는 형태로 작동하는 것은 아니다.

큐가 만들어진 다음 등록정보 창을 이용하여 필요한 설정을 할 수 있다.

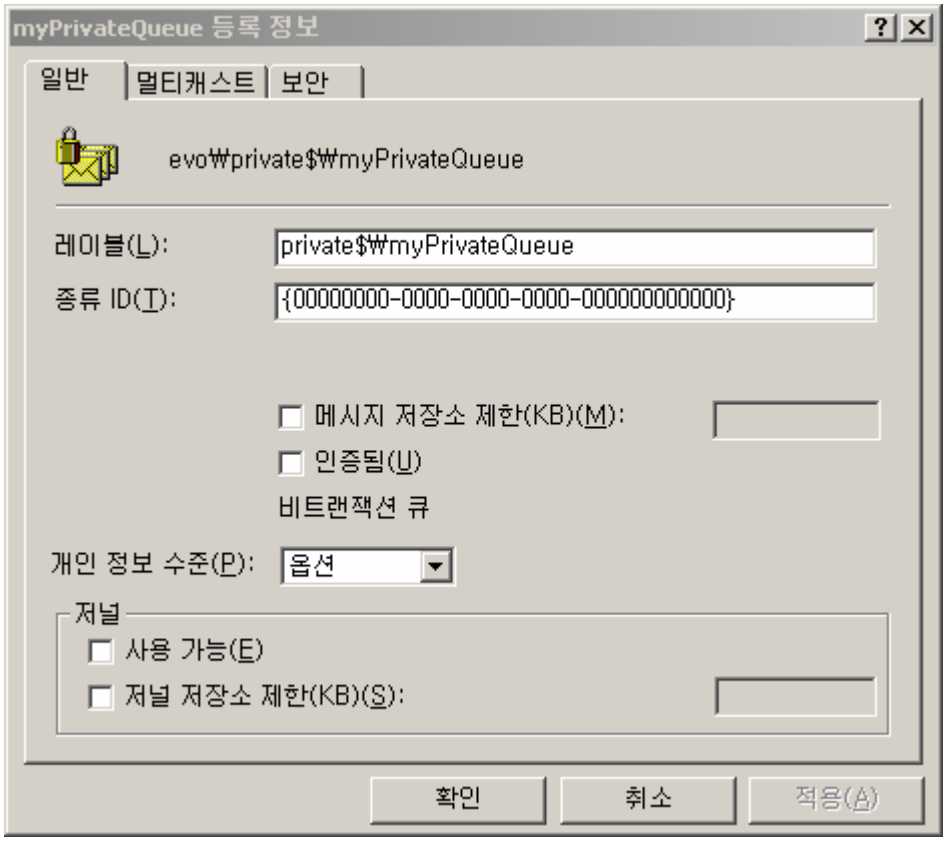

그림 3. 메시지 큐 등록정보

등록정보 창을 이용하여 다음과 같은 항목들을 설정할 수 있다.

- 레이블 (Label): 쉽게 검색할 수 있도록 큐를 구별해주는 이름
- 종류 ID (Type ID): 이 옵션은 공유 GUID를 사용해서 여러 개의 큐를 그룹핑 할 수 있게 해 준다. GUID는 guidgen.exe 나 uudigen.exe 같은 유틸리티를 사용해서 만들 수 있다.

- 메시지 저장소 제한 (Maximum Size): 최대 용량을 지정해서 메시지로 디스크가 꽉차는 일을 예방하여야 한다. 윈도우 2000까지는 최대 2GB의 시스템적 제한이 있었으나, XP 이후로는 최대 용량 제한은 없다.
- 개인 정보 수준 (Privacy Level): 메시지가 암호화 될지를 결정한다. 메시지를 암호화 하려면 액티브 디렉토리가 필요하다.
- 저널 (Journal); 저널 옵션을 사용하면, 현재 큐를 위한 저널 큐가 만들어지고 전송 받은 모든 메시지의 복사본이 저장된다. 저널 큐의 용량 제한을 둘 수 있으며, 디버깅시에 매우 유용하다.

메시지 큐: 메시지 분석

닷넷 환경에서 메시지 큐 다루기

닷넷의 System.Messaging 네임스페이스에는 메시지 큐를 다루기 위한 각종 타입들이 포함되어 있다. 그 중 가장 주요한 몇가지 타입을 다음의 표로 정리했다.

Class	설명
BinaryMessageFormatter	이진 포맷을 사용하여, 닷넷 객체를 직렬화, 역직렬화 한다.
Message	메시지 큐잉의 메시지와 그의 속성을 캡슐화 한다.
MesageQueue	메시지 큐잉 서버의 메시지 큐에 접근성을 제공한다.
MessageQueueException	내부적인 메시지 큐잉 관련 에러를 나탸낸다.
MessageQueueInstaller	메시지 큐를 인스톨할 수 있게 해준다.
XmlMessageformantter	XML 포맷을 이용하여 닷넷 객체를 직렬화, 역직렬화한다.

[표 1.] Queue관련 주요 클래스

UDDI 서비스

UDDI (Universal Description Discovery and Integration) 는 XML 웹 서비스에 대한 정보를 게시하고 찾기 위한 표준 규약이다. UDDI 서비스는 각 조직과 그 조직이 제공하는 서비스 그리고 기술적 세부 사항들을 설명하는 프레임워크를 제공한다.

UDDI 서비스 소개

앞에서는 비교적 새롭게 등장한 닷넷 프레임워크, ASP.NET, XML 웹서비스 등을 살펴보았다. 분명히 XML 웹 서비스의 사용은 이전보다 진보된 방식으로 분산 컴퓨팅을 구현할 수 있도록 해준다. 하지만 응용 프로그램의 구성 요소 역할을 하는 웹 서비스들을 적절히 조직화하고 관리할 수 잇는 수단이 제공되지 않는다면, 그 운영 효율성은 상당부분 저하될 것이다. UDDI 서비스는 표준에 기초한 분류 기법을 제공하여 웹 서비스의 게시와 검색을 위한 기본 토대를 제공한다.

XML 웹 서비스를 이용하는 가상의 개발 환경을 하나 생각해보자. '갑'이라는 개발자는 웹 서비스의 개발을 담당하고, '을'이라는 개발자는 그 웹 서비스를 이용하는 클라이언트 모듈을 담당하고 있다. '을'이 '갑'의 웹 서비스를 이용하려면 '을'은 해당 웹 서비스의 위치, 이름, 매개변수 등의 정보를 미리 알아야 한다. 가장 초보적인 단계의 방식을 생각한다면 직접 대화를 통해 세부 정보를 알려주는 이메일 같은 수단도 가능할 것이다. 하지만 프로그래밍 리소스의 규모가 커질수록 이런 전달 방식은 효율성 면에서 한계에 도달하게 될 것이다.

위의 문제점을 극복할 수 있는 대안으로 UDDI 서비스를 생각해 볼 수 있다. 우리가 네트워크 환경에서 디렉터리 서비스를 이용하여 네트워크 리소스를 조직화 하고 편하게 사용하는 것처럼 UDDI 서비스는 여러 프로그래밍 리소스를 조직화하여 관리할 수 있도록 해주며, 사용자에게 검색 서비스 기능을 제공한다. 정리하면 UDDI 서비스는 각종 웹 서비스를 설명하고, 검색할 수 있는 방법을 제공한다.

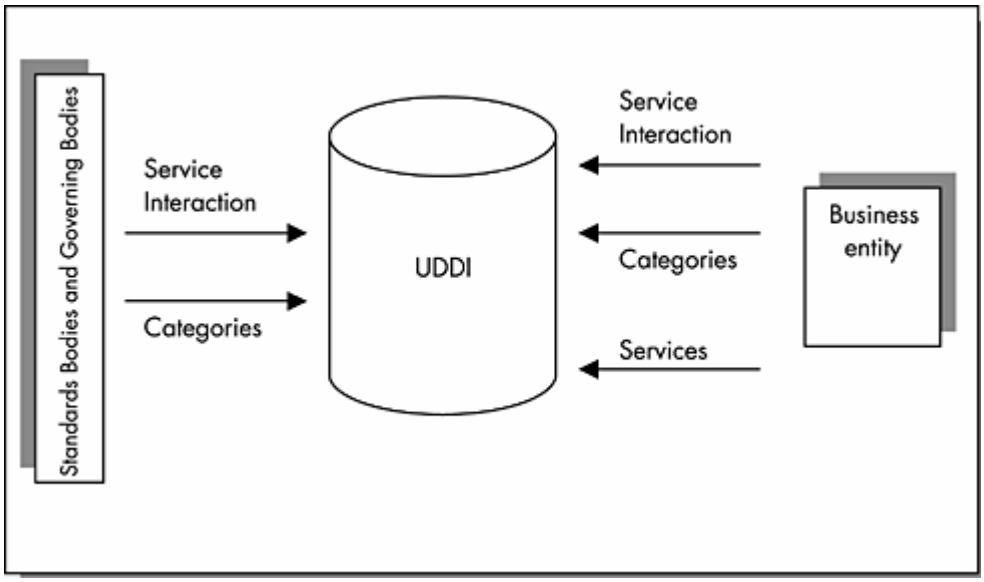

그림 1. UDDI 서비스 개념

UDDI 서비스 데이터 모델링

UDDI 서비스는 공급자, 서비스, 바인딩, tModel의 핵심 요소들을 사용하여 서비스를 체계적으로 분류하고, 데이터베이스화 한다. 네 가지의 핵심 요소는 서비스를 정리하는 수단이 될 뿐만 아니라 사용자가 사용하게 될 검색조건이 되기도 한다. 위의 요소를 사용하여 데이터베이스가 구축되었다면 사용자는 간단히 서비스 공급자의 이름이나 서비스 이름 등을 사용하여 자신이 원하는 서비스를 검색할 수 있을 것이다. 다음은 UDDI 서비스의 네 가지 핵심 요소들의 간단한 설명이다.

1) 공급자

공급자 (Providers)는 계층구조에서 가장 상위 수준의 요소이며, 서비스를 제공하는 실제의 단체나 개념적인 그룹으로 생각할 수 있다. 예를 들면, 공급자는 서비스를 제공하는 조직이나 부서, 혹은 그 서비스를 제공하는 사람이나 서비스가 호스팅되고 있는 컴퓨터의 이름이 될 수 있다. 또한 공급자는 연락처 (Contact)가 연결될 수 있는 유일한 요소이며, 공급자끼리 연관 관계를 맺을 수 있다. 공급자는 논리적으로 서비스를 포함한다.

2) 서비스

서비스 (Service)는 UDDI 서비스의 사용자들에게 제공될 서비스에 대한 액세스 지점을 제공하고, 그 기능을 서술하는 역할을 한다. 서비스는 논리적으로 바인딩을 포함한다.

3) 바인딩

바인딩 (Binding)은 웹 서비스의 액세스 지점과 인터페이스 정보의 액세스 지점, 즉 해당 정보로 접근할 수 있는 URL을 포함한다.

4) tModel

tModel은 매우 융통성 있는 구조를 가지고 있으며, 일반적으로는 웹 서비스의 WDSL (Web Service Description Language) 문서를 나타낸다.

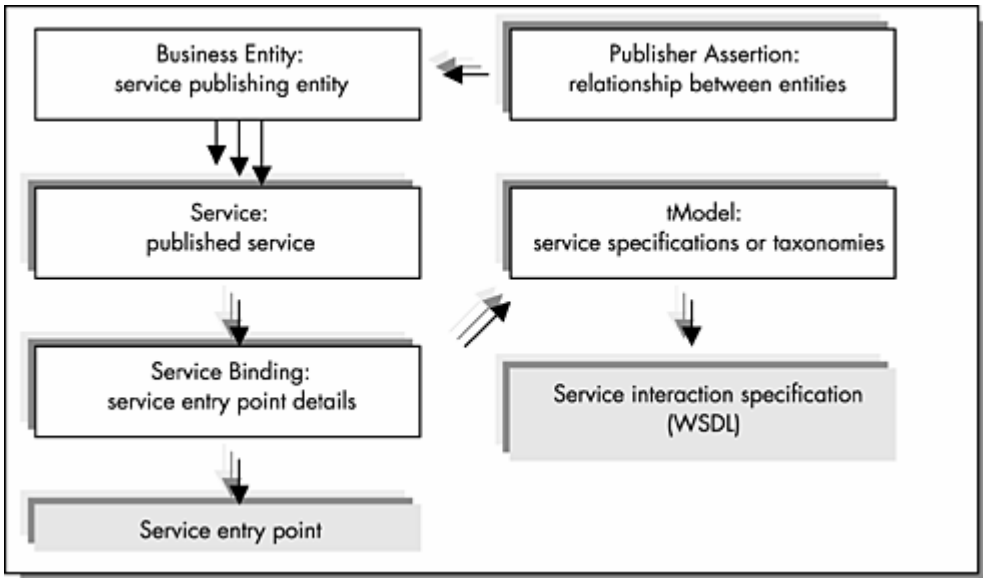

그림 2. UDDI 핵심 요소의 상관 관계

UDDI 서비스에서 범주 만들고 사용하기

UDDI 서비스에 게시되는 서비스들의 양이 늘어날수록 서비스나 공급자 이름을 사용하는 검색 방법은 효율성이 떨어진다. 만약 사용 가능한 몇 천, 몇 만개의 웹 서비스가 있는 상황이라면 자신이 필요로 하는 서비스의 이름이나 공급자의 이름으로 검색하는 방법보다는 서비스를 성격 별로 분류하고, 범주화 시키는 방법이 훨씬 유리해진다. 예를 들어, 우리가 인터넷 검색 엔진을 사용해서 이메일 계정을 제공하는 업체를 찾는 경우를 생각해보자. 실제 이메일 서비스를 제공하는 업체의 이름을 아는 경우라면, 해당 업체의 이름을 검색어로 사용하여 빠르게 원하는 정보를 찾을 수 있을 것이다. 하지만 특정 서비스 업체의 이름을 알지 못한다면 '컴퓨터와 인터넷 -> 통신, 네트워크 -> 전자우편 (E-Mail) -> Email 계정 서비스' 같은 분류를 따라 검색을 진행 할 것이다.

위와 같은 경우를 생각할 때 두 가지의 주요 분류 기법은, 서비스 수준에서 범주를 적용하고 하위요소에 상속시키는 방법이 첫 번째이고 tModel에 범주를 적용시키고 바인딩의 인스턴스 정보를 이용하는 두 번째의 방법이 있다.

UDDI 서비스 설치

이제 직접 UDDI 서비스를 설치하고, 기본적인 설정을 해보자. UDDI 서비스는 관리 콘솔, 데이터베이스 구성요소, 그리고 웹 서버 구성 요소의 세가지 하위 구성 요소로 구성되어 있다. 다음은 각 구성요소의 간단한 설명이다.

- 웹 서버 구성요소: ASP.NET을 이용하여 비즈니스 로직과 메시지 처리 부분을 담당하며, IIS 6.0을 이용한다.
- 데이터베이스 구성요소: UDDI 서비스의 정보를 저장하는 역할을 한다. SQL 서버 2000이나 MSDE가 사용될 수 있다.
- 관리 콘솔: MMC 스냅인으로 제공되며, 여러 개의 UDDI 서비스 사이트를 관리 할 때 사용된다.

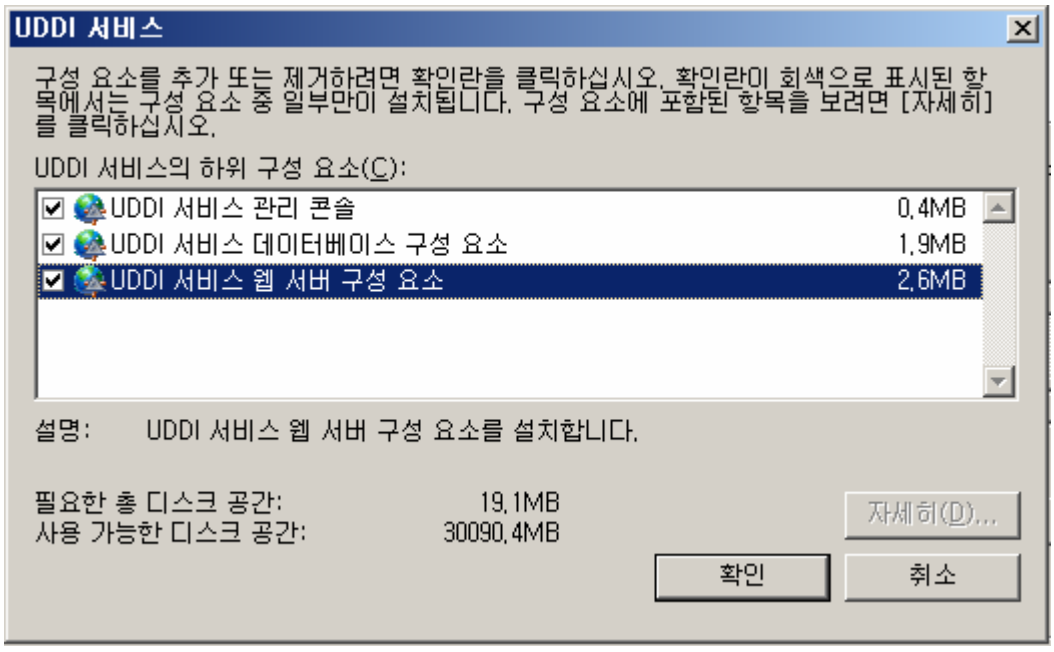

그림 3. UDDI 서비스 하위 구성 요소

우선 구성요소 마법사를 사용하여 UDDI 서비스를 선택하고 다음 화면으로 진행한다. 다음 화면에서는 UDDI 서비스 사이트와의 통신에 SSL을 사용할 것인지를 결정한다. (이 선택은 설치 후 변경할 수 있다.) 테스트 환경에서 인증서 서비스를 사용할 수 있다면 SSL을 사용하도록 설정하고 진행하자.

데이터베이스의 저장 위치와 서비스 계정 설정은 기본 설정을 이용한다. 다음 단계는 UDDI 서비스 사이트의 이름을 지정하는 화면이다. 적당한 이름을 입력한다.

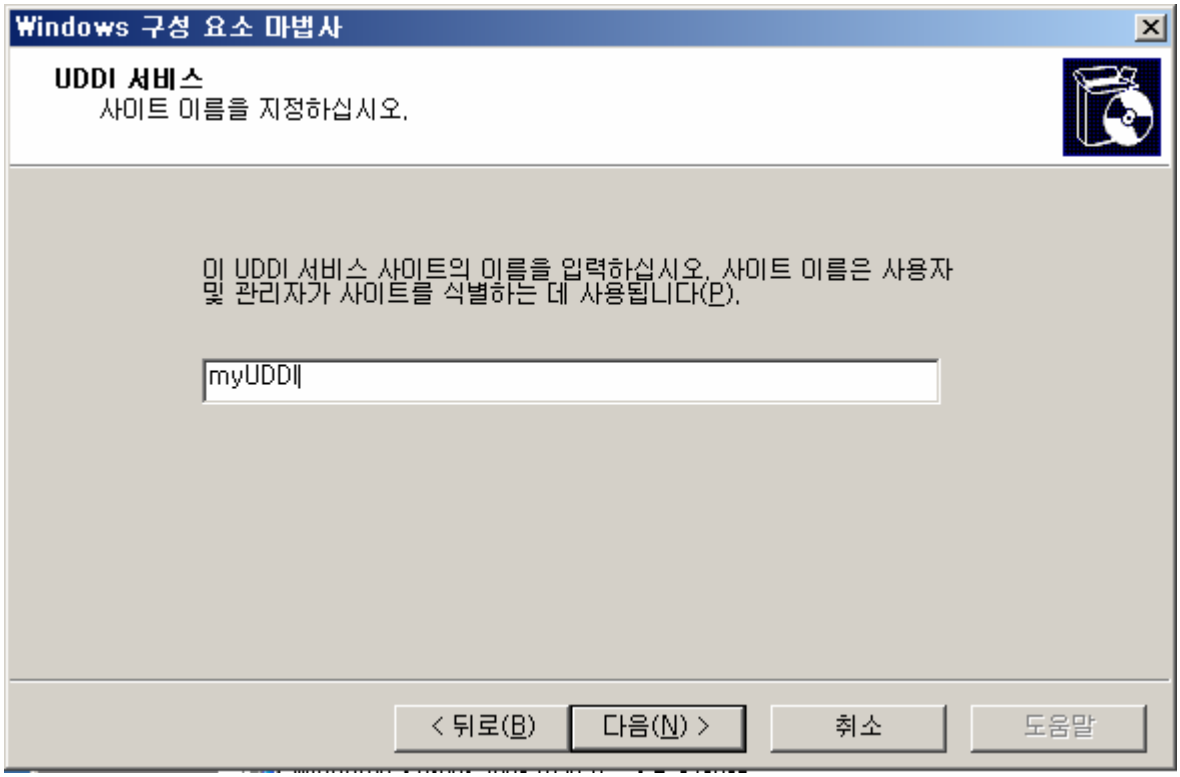

그림 4. 서비스 사이트 이름 지정

나머지의 서비스 설치를 진행한 후, 설정 작업을 소개하는 웹 페이지를 볼 수 있다.

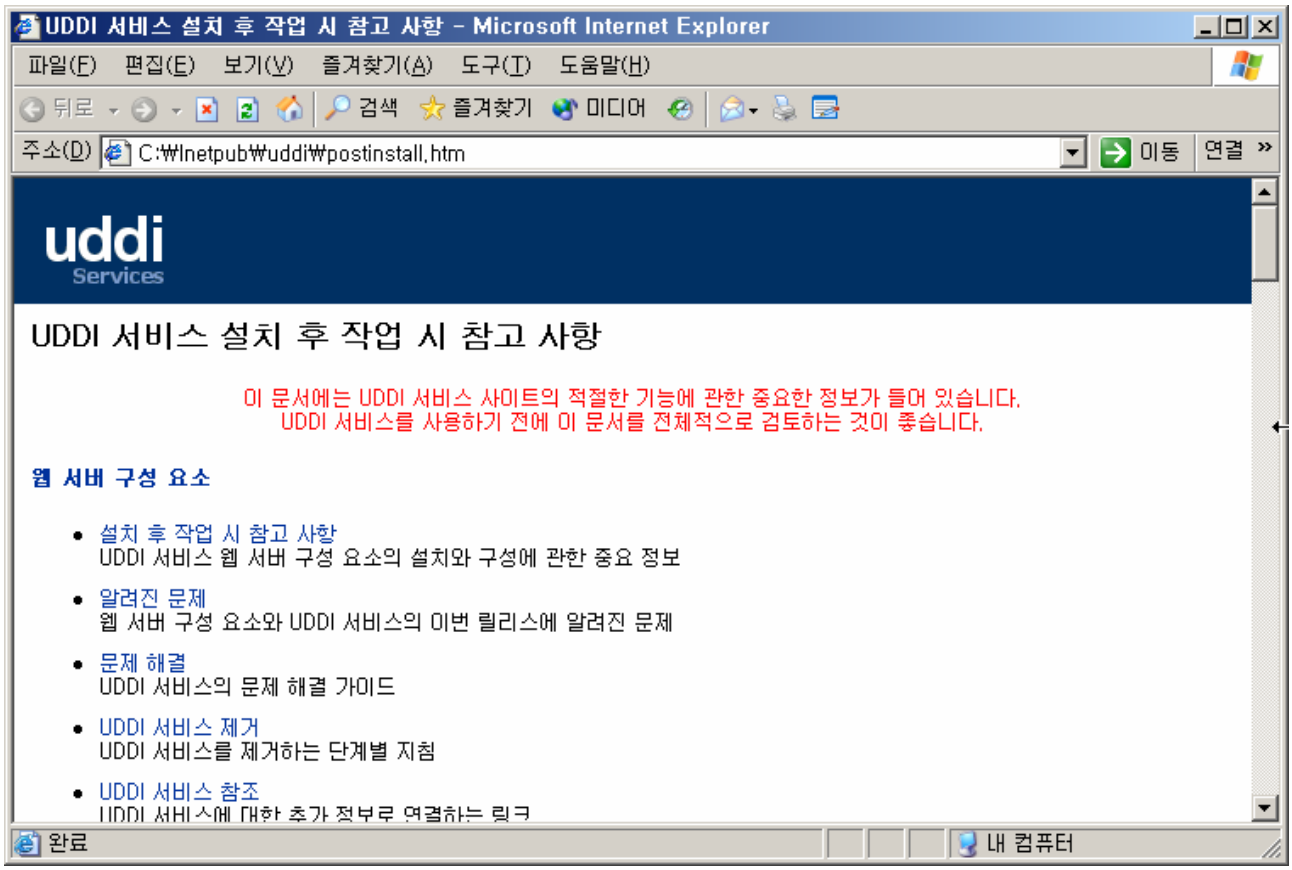

그림 5. 설치 완료 화면

UDDI 서비스 설정

역할 할당

UDDI 서비스는 다음과 같은 4개의 역할을 포함하고 있고, 각각의 역할에 따라 서비스를 사용할 수 있는 권한이 달라진다.

- 사용자 그룹은 UDDI 서비스를 검색할 수 있는 권한만 가진다.
- 게시자 그룹은 UDDI 서비스를 검색하고, 게시할 수 있는 권한을 가진다.
- 코디네이터 그룹은 UDDI 게시자 그룹의 권한과 UDDI 서비스에 저장되어 있는 정보를 보고, 수정할 수 있는 권한을 가진다.
- 관리자 그룹은 코디네이터 그룹의 권한과 UDDI 서비스 자체의 관리 권한을 가진다. 이런 관리 작업에는 보안설정, 데이터 백업 등의 작업이 포함된다.

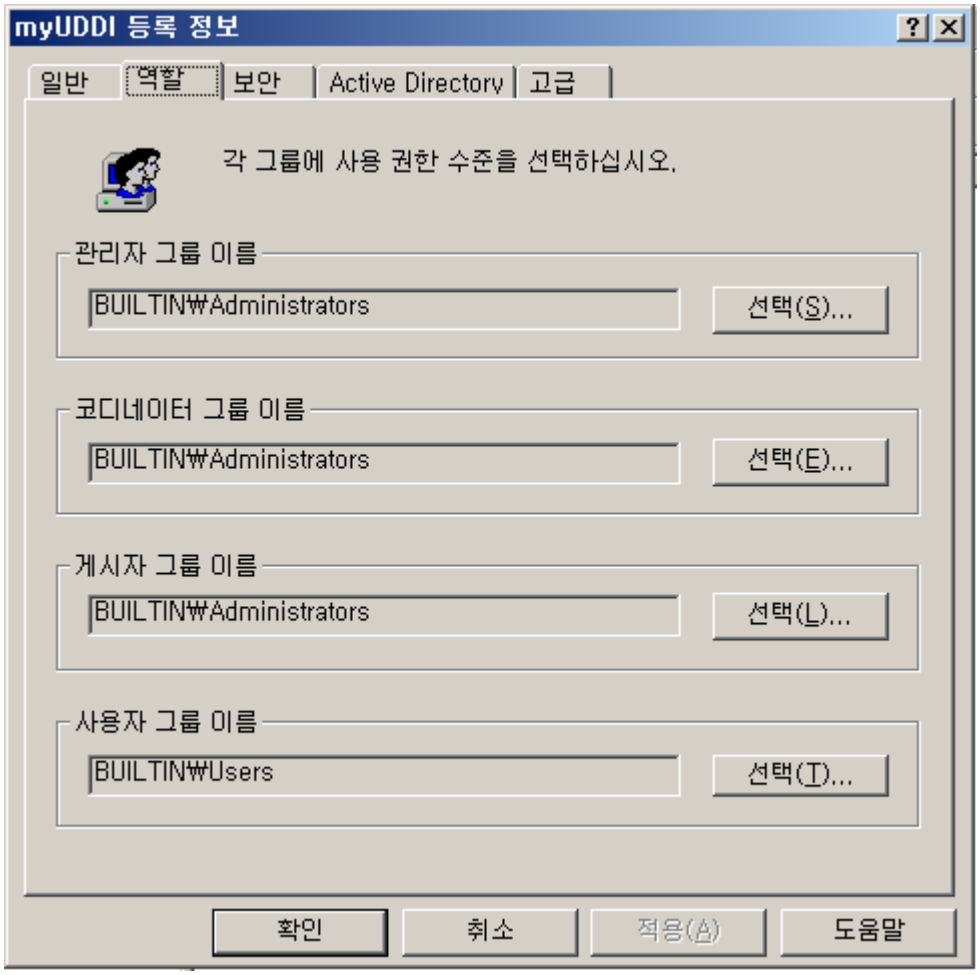

그림 6. UDDI 등록정보-역할

보안설정

UDDI 서비스는 인증방법, SSL의 사용, 암호화 설정을 통해 보안을 강화 할 수 있는 옵션을 제공한다. 여타 서비스와 마찬가지로 가능한 한 높은 보안 수준을 유지하여 운영 환경의 신뢰성과 안정성을 높여야 한다.

본질적으로 UDDI서비스는 프로그램 코드를 통해서 사용하기 위한 서비스라는 것을 기억할 필요가 있다. XML 웹 서비스를 사용하는 응용프로그램은 코딩 시에 정해진 웹 서비스를 사용하도록 작성될 수도 있고, 실행 시에 필요한 웹 서비스를 동적으로 검색하고 연결하여 사용하도록 작성될 수도 있다. 이런 이유로 UDDI 서비스는 다음과 같은 세 가지의 인터페이스를 제공한다.

1. 웹 사용자 인터페이스

   웹 브라우저를 사용하여 UDDI 서비스를 이용할 수 있도록 하며, 사용자의 인증 여부에 따라 각기 다른 URL을 제공한다. 윈도우 통합 인증이 사용될 때는 http(s)://<servername>/uddi의 URL을 사용하고, 인증이 필요 없는 읽기만 가능한 접속은 http(s)://<servername>/uddipublic의 URL을 사용한다.

그림 7. 웹 사용자 인터페이스

2. 비주얼 스튜디오 닷넷을 위한 인터페이스

   개발 환경 내에서 UDDI 서비스의 검색, 통합 기능을 사용하도록 제공되는 인터페이스이다. 이 인터페이스 역시 윈도우 통합 인증이 사용되는 경우 http(s)://<servername>/uddi/addwebrefernce 를, 인증이 필요 없는 접속인 경우 http(s)://<servername>/uddipublic/addwebrefernce의 URL을 사용하도록 나누어져 있다.

3. UDDI API를 이용한 인터페이스

   UDDI API 1.0과 2.0을 지원하며, 개발도구나 응용프로그램에서 직접 UDDI 서비스에 액세스 할 수 있는 기능을 제공한다. 이 방식은 인증과 비인증 연결, 게시할 때와 요청할 때를 기준으로

다음의 4개의 URL을 제공한다.

- 통합인증을 사용하고, 게시할 때: http(s)://<servername>/uddi/publish.asmx
- 통합인증을 사용하고, 요청할 때: http(s)://<servername>/uddi/inquire.asmx
- 비 인증, 게시할 때: http(s)://<servername>/uddipublic/publish.asmx
- 비 인증, 요청할 때: http(s)://<servername>/uddipublic /inquire.asmx

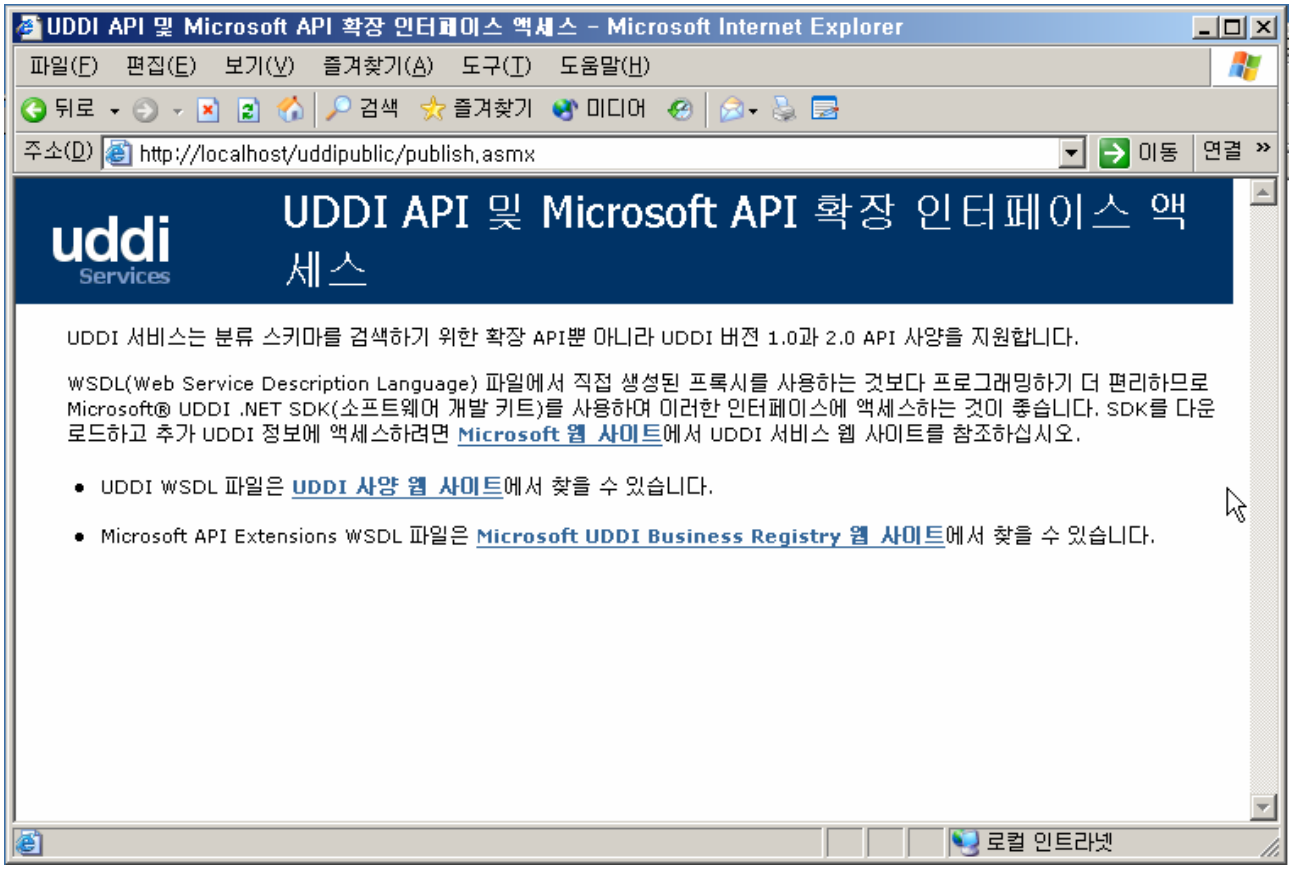

그림 8. UDDI API 액세스

위와 같이 UDDI 서비스는 사용되는 인터페이스에 따라 접속 URL과 게시 기능, 인증 정보 전달 방법 등이 다르게 적용된다.

UDDI 서비스는 윈도우 통합 게시자 인증, UDDI 게시자 인증의 두 가지 인증 방식을 지원하고, 두 가지를 같이 사용할 수 있도록 해준다. 가장 보안성이 좋은 설정은 윈도우 통합 게시자 인증을 사용하고, 동시에 읽기 액세스에 대한 인증을 사용하도록 설정하는 것이다. UDDI 게시자 인증은 윈도우 플랫폼이 아닌 클라이언트가 UDI 서비스에 액세스 하여야 하는 경우에만 한정하여 사용하도록 한다. 또한 사용자

가 인터넷을 통하여 서비스에 액세스하는 경우라면 SSL의 사용을 고려하여야 한다.

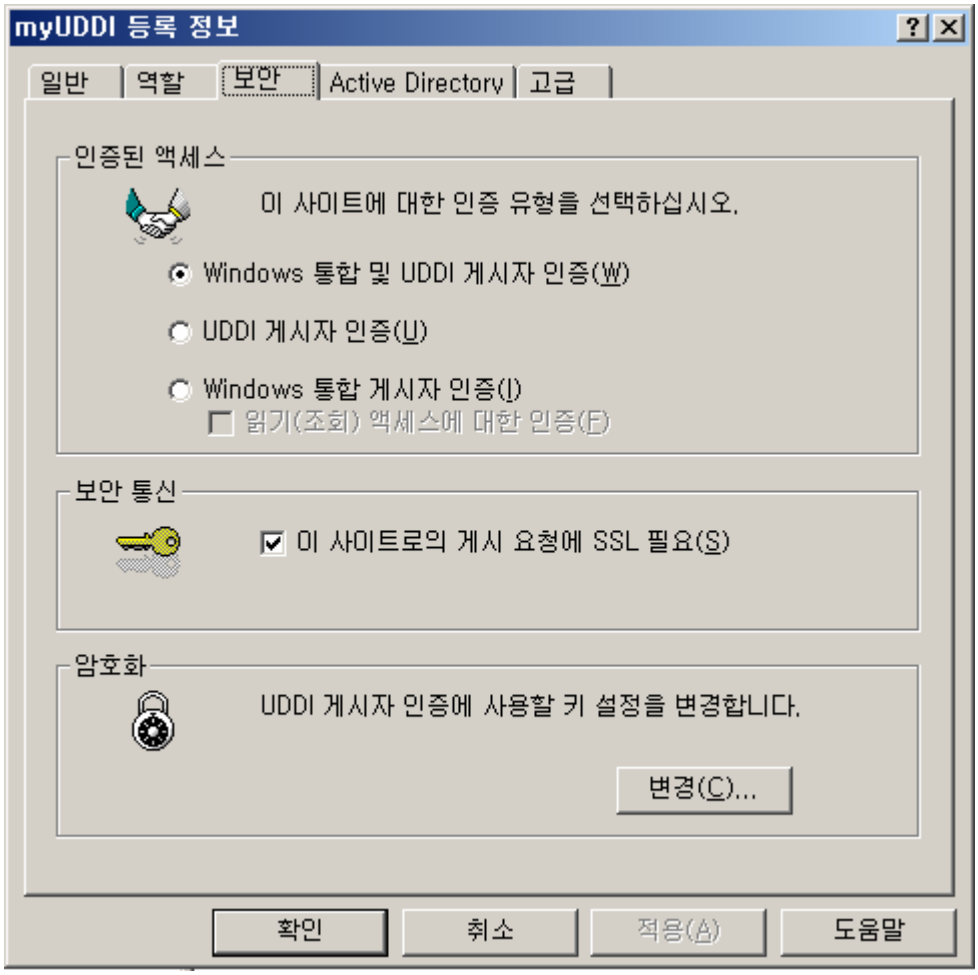

그림 9. UDDI 등록 정보-보안

웹 서비스 게시

일반적인 관점에서 생각하면 서비스 게시와 검색 같은 활동들은 주로 개발자의 몫이다. 하지만 간단한 수준의 게시 작업을 직접 수행하여 보면 UDDI 서비스의 동작을 보다 쉽게 이해할 수 있다.

UDDI 서비스 관리

UDDI 서비스 관리 작업은 관련 데이터베이스의 백업 및 복원 그리고 로그 설정 등의 작업을 포함한다.

데이터베이스 백업과 복원

데이터베이스를 설치할 때, SQL 200 서버를 데이터 저장소로 사용했다면 백업과 복원은 SQL 2000 서버의 클라이언트 도구를 사용하여 수행한다. MSDE를 사용한 경우에는 설치 경로의 data 디렉터리에 설치된 두 가지의 유틸리티를 사용할 수 있다. 기본 경로를 이용하여 설치하였다면 “C:₩inetpub₩uddi₩data” 의 경로에서 uddi.database.backup.cmd 와 uddi.database.restore.cmd 라는 두 명령행 유틸리티를 발견할 수 있을 것이다.

로그설정

UDDI 서비스는 이벤트 로그나 텍스트 파일 형식의 로그 혹은 두 가지 모두를 수행하도록 설정될 수 있다. 자세한 로깅 옵션을 사용할수록 성능에 영향을 미치므로, 디버깅 용도를 제외하고는 낮은 수준의 로깅을 수행하도록 한다.

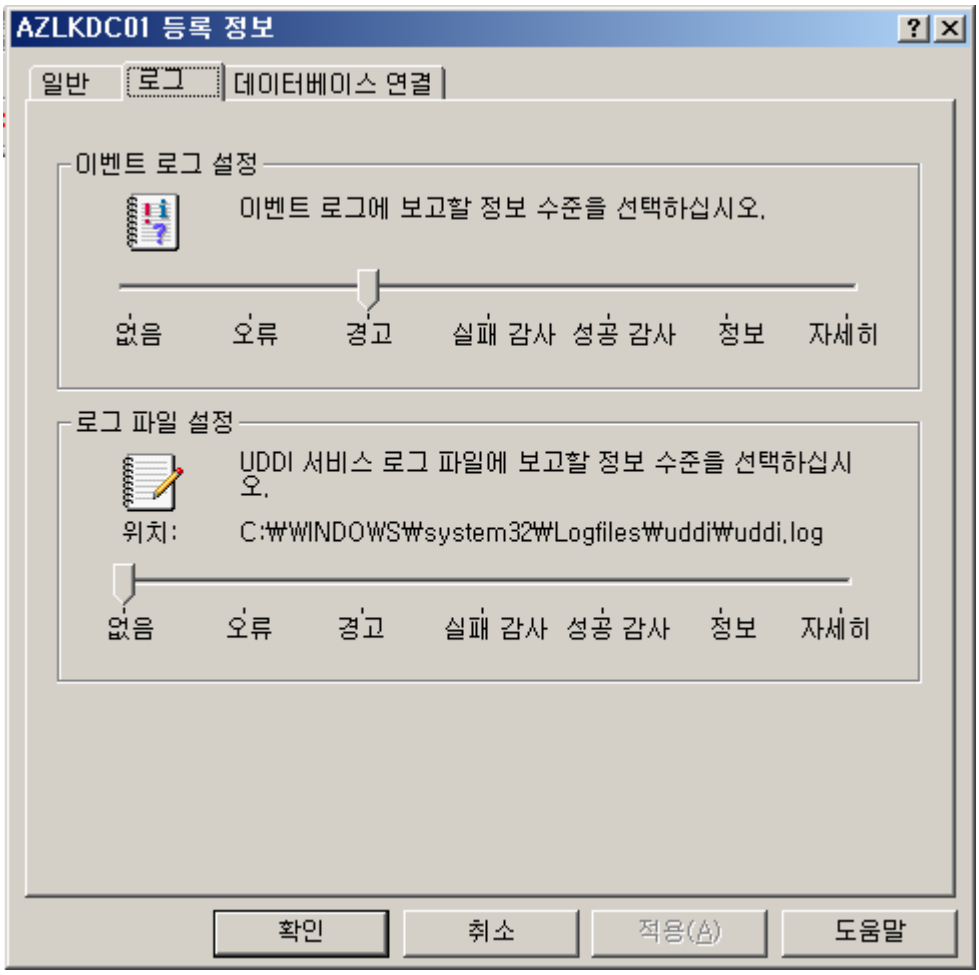

그림 10. UDDI 로그설정

WMI (Windows Management Instrumentation)

새로운 업무용 프로그램을 전사적으로 배포하기 위해서 클라이언트 컴퓨터들의 시스템 사양을 조사해야 하는 경우를 생각해 보자. 시스템 관리자의 현재 임무는 네트워크상의 모든 클라이언트에 장착된 메모리 양을 알아내고, 메모리가 모자라는 경우 업그레이드를 위한 보고서를 작성하는 일이다. 클라이언트 컴퓨터의 수가 많지 않다면 별 문제가 되지 않겠지만, 중 규모 이상의 네트워크에서 이런 일은 꽤 수고스러운 작업이 된다.

물론 원격으로 시스템 정보를 살펴볼 수도 있지만, 실제 물리적인 정보 (가령 메인보드의 메모리 소켓은 몇 개며, 몇 개가 비어 있는지, 사용된 메모리의 용량 정보 등)를 일일이 조사하는 것은 매우 시간 소모적인 작업이다.

WMI 관련 기술을 사용하면 이런 지루한 작업들을 매우 단순한 일로 만들 수 있다. 필요한 준비물은 스크립트언어에 대한 일정수준의 지식과 “메모장” 만 있으면 된다.

WBEM과 WMI

WBEM (Web-Based Enterprise Management) 은 기업 환경의 관리 작업을 통합할 목적으로 만들어진 관리 기술과 인터넷 표준 기술의 집합이다. WMI는 WBEM 을 윈도우 환경에 구현한 것으로, 관리 대상에 대한 일반적인 인터페이스를 제공한다.

WMI의 목표
WMI는 윈도우 환경에서 일관되고 표준화된 기반 구조를 제공하는 것에 목적이 잇다. WMI를 이용하면 서로 다른 인터페이스, 프로토콜과 데이터 유형에 신경 쓰지 않고도 관리 작업을 수행할 수 있다. 관리자는 특정 작업을 위하여 DMI, SNMP, WDM 같은 여러 가지의 복잡한 기술에 주의하지 않아도 된다.

WMI를 이용한 일반적인 관리 작업
WMI를 이용하는 관리 작업은 다음과 같은 몇 가지의 범주로 정리할 수 있다.

- WMI는 시스템의 상태 정보를 얻을 목적으로 사용할 수 있다. 이런 정보들은 하드웨어와 소프트웨어에 대한 정보, 즉 운영체제, 디스크 드라이브, 프로세서, 프린터, BIOS에 대한 정보 등이 될 수 있다.

- WMI는 시스템 모니터링 목적으로 사용할 수 있다. 원격 시스템의 이벤트 로그에 기록되는 정보를 읽어오거나 성능 모니터의 실시간 정보를 얻는데 WMI를 사용할 수 있다.
- 레지스트리나 이벤트 관련 관리 작업에 WMI를 활용할 수 있다.

WMI 아키텍처

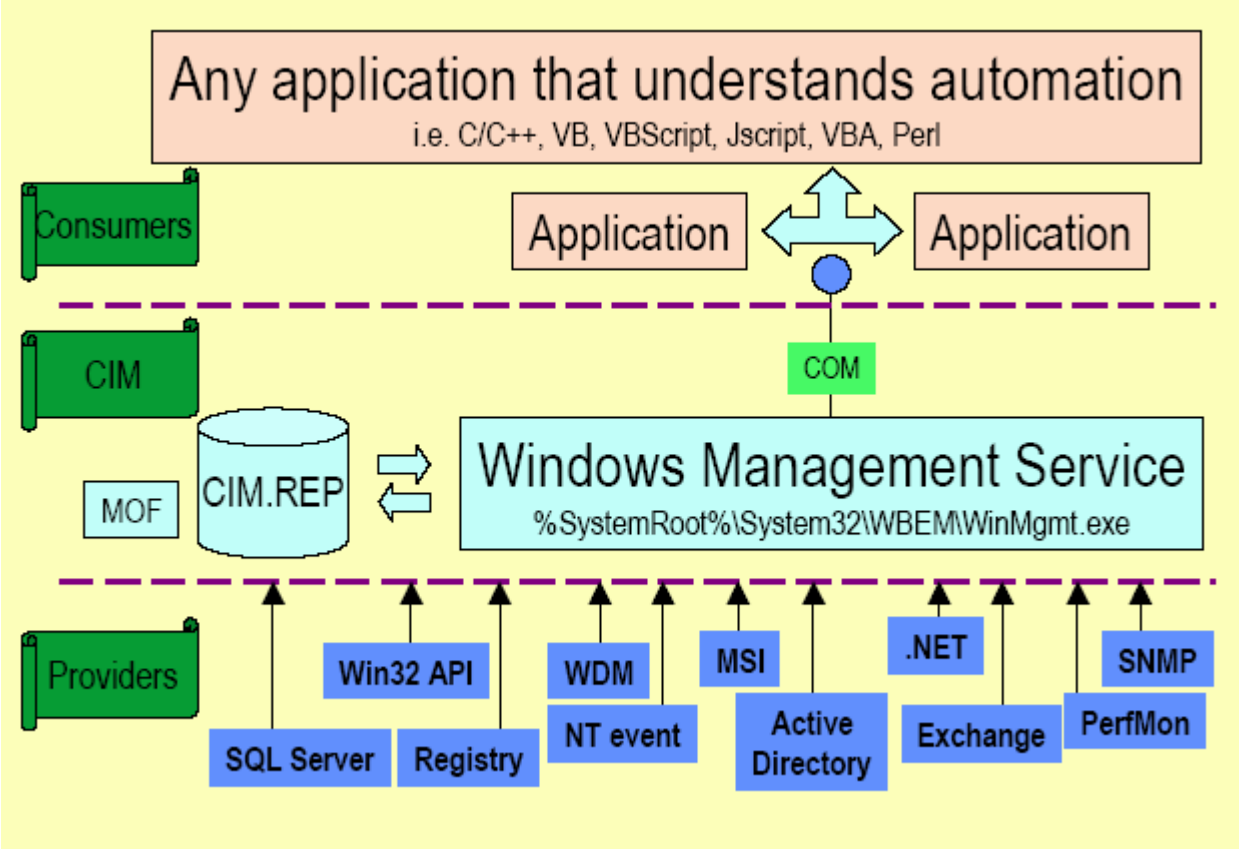

그림 1. WMI 아키텍처

WMI는 CIM (Common Information Model) 을 사용하여 운영체제 내의 관리 대상 개체를 표현한다. 논리적이거나 물리적인 모든 구성 요소들 (소프트웨어, 하드웨어 장치 등) 은 관리 대상 객체가 된다. WMI는 소비자 CIM, 제공자의 3 개의 계층으로 구상된다.

WMI의 소비자는 WMI를 통해 제공되는 정보를 사용하는 응용 프로그램이다. 마이크로소프트의 SMS (System Management Server), MOM (Microsoft Operation Manager) 혹은 사용자가 작성한 스크립트 등의 모든 소비자는 COM 객체를 통해 노출되는 인터페이스를 사용하여 WMI 정보에 접근한다.

CIM (Common Information Model) 은 WMI의 핵심요소이다. CIM은 네트워크, 엔터프라이즈 환경에서의 관리 정보들을 기술하는 모델이다. “%SystemRoot%₩System32₩WBEM₩” 폴더의 WMI 테스터 (wbemtest.exe)를 이용하여 모델링 된 클래스 정의를 살펴 볼 수 있다.

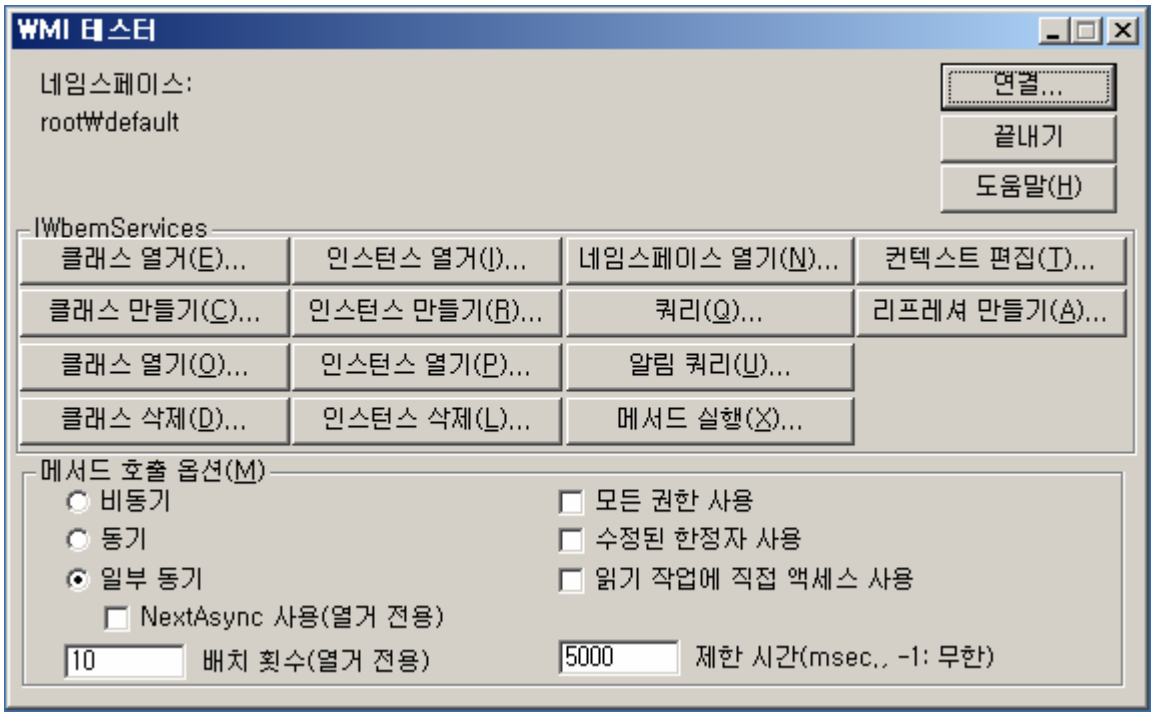

그림 2. WMI 테스터 화면

MOF (Management Object Format) 파일은 CIM 상의 클래스, 개체와 이름공간을 정의하는 목적으로 사용된다. WMI 테스터와 같은 경로에 있는 MOF 컴파일러 (mofcomp.exe)를 이용하면 표준 텍스트 포맷인 MOF 파일을 바이너리 데이터로 컴파일 할 수 있다.

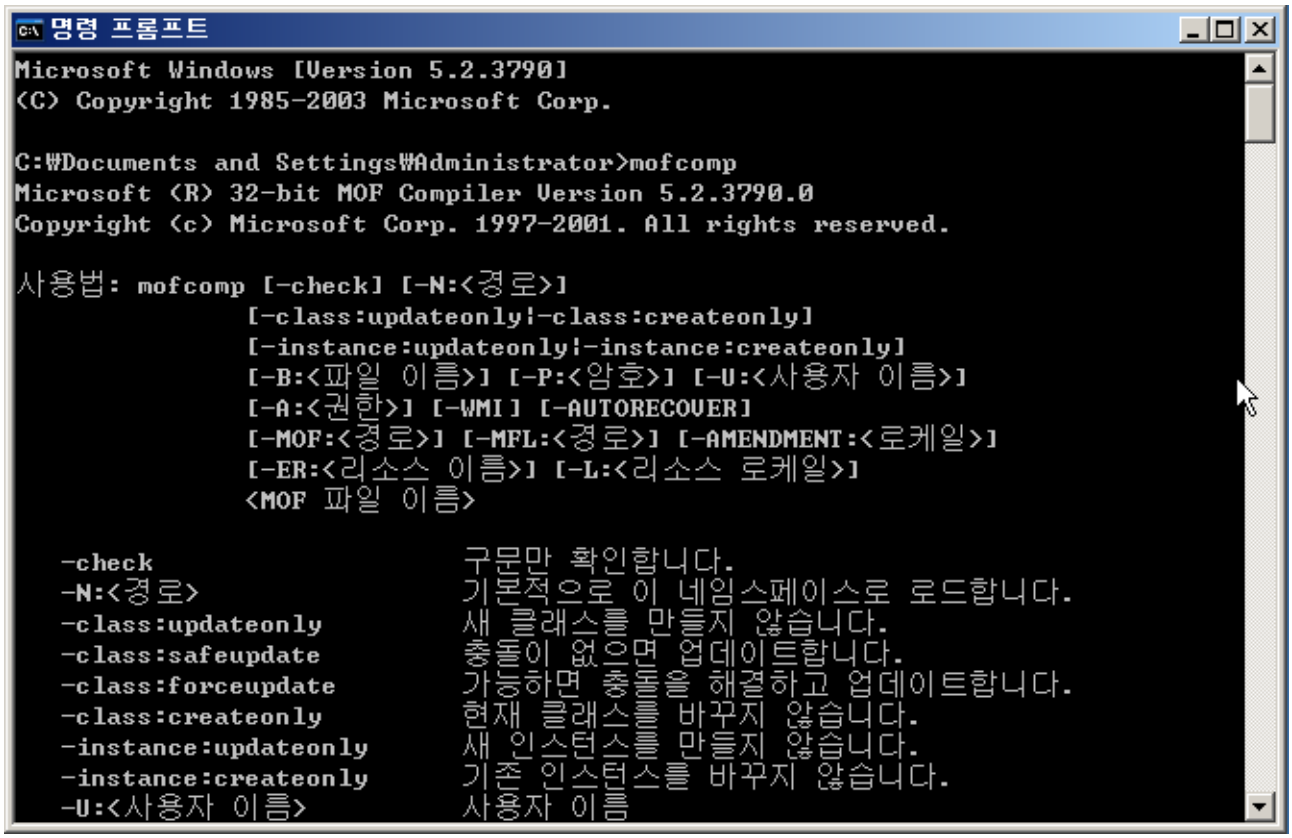

그림 3. mof 컴파일러 사용법

WMI 제공자는 관리 대상 개체들과 직접적으로 통신하는 소프트웨어이다. 제공자들은 대상 개체와 각각의 API를 이용하여 통신하며, 대상 개체의 종류만큼의 개별적인 제공자가 존재한다. (ActiveDirectory, PerMon, WDM 등)

WMI 스크립트 활용

이제 이전의 시나리오로 돌아가서 보고서를 작성해야 하는 관리자가 되어보자.

```
strIPSubnet = “192.168.0.”
On Error Resume Next
For strIPNode = 1 To 254
strComputer = strIPSubnet & strIPNode
WScript.Echo “****************”
WScript.Echo “Computer: “ & strComputer
WScript.Echo “****************”
```

```
Set objWMIService = GetObject("winmgmt:₩₩" & strComputer & "₩root₩cimv2")
Set colItems = objWMIService.ExecQuery("Select * from Win32_PhysicalMemory", , 48)
For each objItem in colItems
    WScript.Echo "Bank Label: " & objItem.BankLabel
    WScript.Echo "Capaciry: " & objItem.Capacity
    WScript.Echo "Data Width: " & objItem.DataWidth
    WScript.Echo " Description: " & objItem.Description
    WScript.Echo "Device Locator: " & objItem.DeviceLocator
    WScript.Echo "Form Factor: " & objItem.FormFactor
    WScript.Echo "Hot Swappable: " & objItem.HotSwappable
    WScript.Echo "Manufactorer: " & objItem.Manufacturer
    WScript.Echo "Memory Type: " & objItem.MemoryType
    WScript.Echo "Name: " & objItem.Name
    WScript.Echo "Part Number: " & objItem.PartNumber
    WScript.Echo "Position In Row: " & objItem.PositionInRow
    WScript.Echo "Spied: " & objItem.Speed
    WScript.Echo "Tag: " & objItem.Tag
    WScript.Echo "Type Detail: " & objItem.TypeDetail
  Next
  Next
```

위의 스크립트는 192.168.0.0 서브넷의 모든 컴퓨터들의 메모리에 대한 상세 정보를 출력하는 내용으로, VBScript 언어로 작성되었다. 메모장을 이용해서 위의 내용을 입력하고, 저장한 다음, 서브넷 부분을 알맞게 수정한다면 자신의 네트워크 환경에서 위의 정보가 출력되는 것을 볼 수 있다.

앞서 "메모장과 일정수준의 스크립트 언어지식" 이라는 조건만을 이야기 했지만, WMI의 개체 모델을 살펴보는 것은 그리 간단한 일이 아니다. 그리고 필요가 발생할 때 마다 해당 내용들을 찾아서 스크립트를 작성해야 한다면 작업 효율성의 측면에서 꽤 부담스러운 일이 된다.

마이크로소프트의 테크넷의 스트립트 센터 (http://www.microsoft.com/technet/scriptcenter)를 방문하면 거의 모든 경우에 해당되는 방대한 양의 스크립트들을 접할 수 있다.

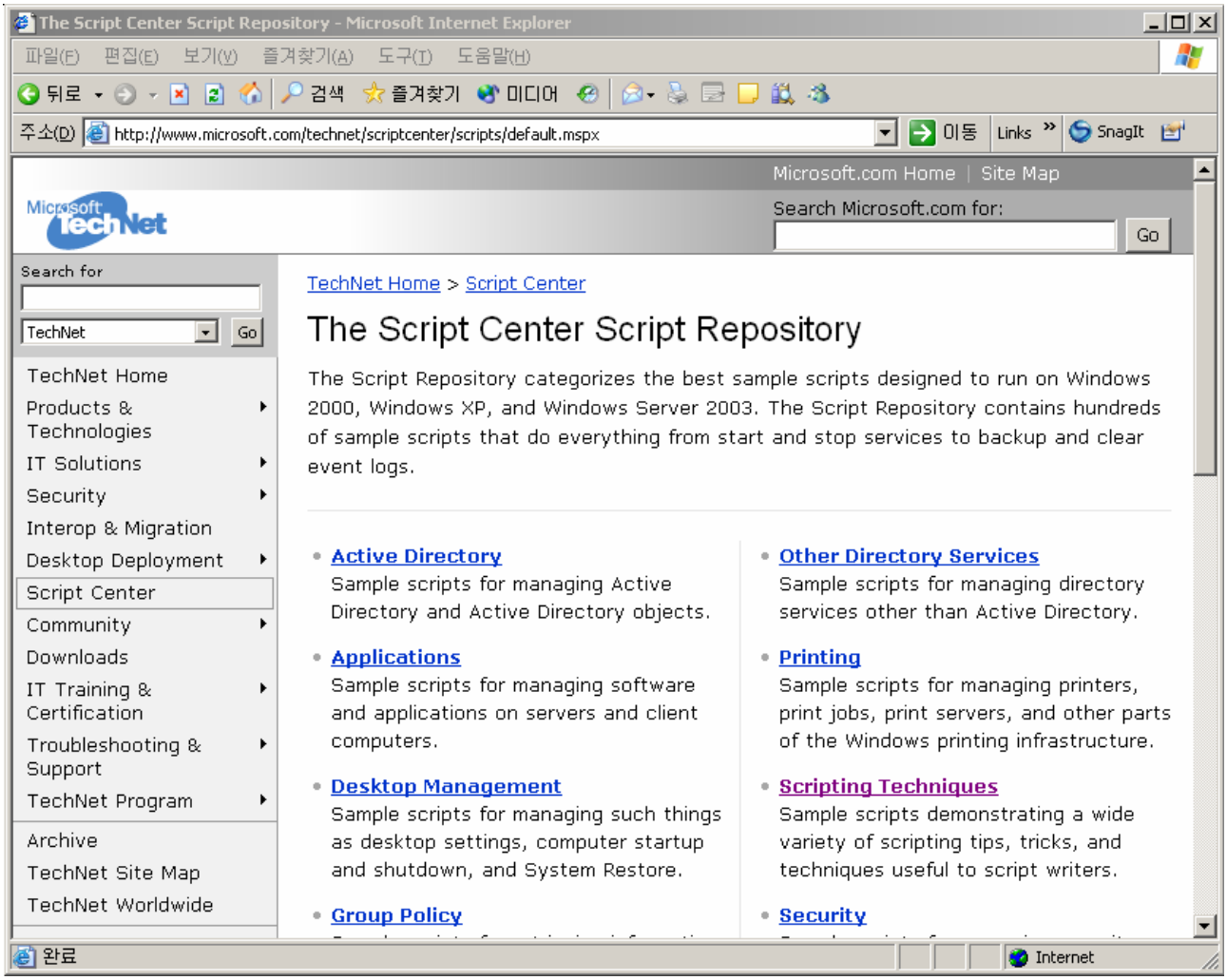

그림 4. 테크넷 스트립트 센터

스크립트 센터의 샘플 스크립트 수준을 넘어서 보다 고수준의 스크립트를 작서앟고자 한다면, WMI SDK를 설치하고 포함된 문서를 참조하여야 한다. WMI SDK에는 CIM Studio등의 유용한 도구들을 비롯하여 WMI와 관련된 완벽한 레퍼런스 문서를 제공한다.

WMIC

WMI 의 기능을 이용할 수 있는 준비된 프로그램이 있다면 사용자 정의 스크립트를 작성해야 하는 부담을 상당부분 덜 수 있을 것이다. 윈도우 2003은 WMIC (Windows Management Instrumentation Command-line)을 제공하여 스크립트를 작성하지 않고도 WMI를 이용한 관리 작업을 수행할 수 있도록 해준다.

WMIC 개요

WMIC는 WMI 기능을 이용하기 위한 간단한 명령 행 인터페이스를 제공한다. WMIC는 스크립트나 관리용 응용프로그램과 상호 동작하며, 그 기능이 확장될 수 있다.

WMIC 를 이용한 관리작업

WMIC를 사용하기 위한 기반 지식은 WMI를 이용하는 스크립트와 같다. 직접 WMIC를 사용해 보면서 그 기능을 살펴보자.

WMIC는 다른 많은 윈도우 2003의 명령어처럼 대화형 모드 (Interactive Mode) 와 비 대화형 모드 (Non-Interactive Mode)의 두 가지 작동 모드를 가진다. 명령 프롬프트에서 “wmic”를 입력하고, 상호작용 모드에서 “/?”를 입력하면 도움말을 볼 수 있다.

WMIC 별칭

많은 수의 WMIC 별칭이 준비되어 있다. WMIC의 도움말을 참조하면 각 별칭에 대한 정보를 얻을 수 있다. 몇 가지의 별칭을 보면, BIOS, CDROM, OS, PROCESS 등이 직관적인 이름을 가지고 있다. CPU에 대한 사용 예를 보려면 WMIC 상에서 “CPU /?”를 입력한다.

WMIC 동사, 스위치, 명령

별칭 다음에 동사를 입력하는 방식으로 WMIC의 동사를 사용할 수 있다. 다음 표는 WMIC의 동사, 스위치, 명령과 그 사용 예이다.

[표 1] WMIC 동사

동사	사용 예
ASSOC	OS ASSOC
CALL	SERVICE WEHRE CAPTION=’TELNET’ CALL STARTSERVICE
CREATE	ENVIRONMENT CREATE NAME=”TEMP”, VARIABLEVALUE=”NEW”
DELETE	PROCESS WHERE NAME=”CALC.exe” DELETE
GET	PROCESS GET NAME
LIST	PROCESS LIST BRIEF
SET	ENVIRONMENT SET NAME=”TEMP”, VARIABLENAME=”NEW”

[표 2] WMIC 스위치

스위치	사용 예
/NAMESPACE	/NAMESPACE:₩₩root
/ROLE	/ ROLE:₩₩root
/NODE	..
/IMPLEVEL	/IMPLEVEL:Anonymous
/AUTHLEVEL	/AUTHLEVEL:Pkt
/LOCALE	/LOCALE:MS_411
/PROVILEGES	/PRIVILEGES:ENABLE
/TRACE	/TRACE:ON
/RECORD	/RECORD:MyOutput.xml
/INTERACTIVE	/INTERACTIVE:ON
/FAILFAST	/FAILFAST:ON
/USER	/USER:Admin
/PASSWORD	/PASSWORD:password
/OUTPUT	/OUTPUT:CLIPBOARD
/APPEND	/APPEND:CLIPBOARD
/AGGREGATE	/AGGREGATE:ON

[표 3] WMIC 명령

명령	사용 예
CLASS	WMIC /OUTPUT:C:₩ClassOutput.htm CLASS Win32_SoundDevice
PATH	WMIC /OUTPUT:C:₩PathOUtput.txt PATH Win32_SoundDevice GET /VALUE
CONTEXT	WMIC CONTEXT
QUIT	WMIC QUIT
EXIT	WMIC EXIT