Borland®

델파이를 이용한 인터넷 프로그래밍

Marco Cantù
(http://www.marcocantu.com)

목차

인터넷 프로그래밍에의 도전	1
델파이는 어디에 적합한가?	2
핵심적인 TCP/IP 지원	3
클라이언트측 프로토콜 지원	4
서버측 프로토콜 지원	4
클라이언트측 웹 지원	5
서버측 웹 개발	6
HTML 페이지 생성	6
WebBroker 프레임워크	7
마이크로소프트의 윈도우 DNA 구조 지원	9
InternetExpress: XML, ECMAScript, DHTML	10
써드파티 웹 서버 확장	13
인터넷에서 활동중인 델파이	13
결론: 델파이의 장점	14

볼랜드 델파이는 마이크로소프트 윈도우 플랫폼에서의 독립 애플리케이션 및 클라이언트 서버 애플리케이션 개발을 위한 훌륭한 환경으로 알려져 있습니다. 델파이의 장점은 완전한 OOP 지원에서 비주얼한 개발에 이르기까지 많으며, 강력함과 쉬운 사용법이 독특하게 결합되어 있습니다.. 그러나 이제 개발의 새로운 분야는 인터넷 프로그래밍입니다. 이러한 새로운 상황에서 델파이는 무엇을 제공해야 할까요? 델파이를 가지고 인터넷 애플리케이션을 구축하려면 어떤 기능을 활용할 수 있을까요? 이것이 여기서 다루고자 하는 내용입니다. 우리는 다음과 같은 델파이의 용도를 살펴볼 것입니다:

- direct 소켓 및 TCP/IP 프로그래밍
- 클라이언트 또는 서버 측에서 가장 일반적인 인터넷 프로토콜을 구현하는 써드파티 컴포넌트를 이용
- WebBroker 및 Internet Express 구조로 서버측에서 HTML 페이지 생성
- MTS, COM, ASP 및 ActiveX를 포함한 마이크로소프트의 핵심 기술을 이용한 작업.

인터넷 프로그래밍에의 도전

인터넷 프로그래밍은 전통적인 개발자 환경과 그것을 사용하는 프로그래머들에게 새로운 도전을 제시합니다. 표준 프로토콜의 구현, 다양한 플랫폼 사용 (마이크로소프트 윈도우는 대부분의 클라이언트 컴퓨터에서 사용되고 있지만 서버는 일부에만 사용됨) 및 일부 RDBMS 시스템의 라이센싱 체계에 관련된 문제들이 있습니다.

Delphi

white paper

white paper

그러나 대부분의 문제들은 HTTP 개발과 관련되어 있습니다. 기존의 윈도우 애플리케이션을 웹 브라우저에서 실행되는 애플리케이션으로 변환하는 작업은 생각보다 복잡합니다. 웹은 현재 상태를 유지하지 않으므로, 유저 인터페이스 개발이 어려운 문제가 될 수 있습니다. 또한 가장 널리 사용되는 브라우저간의 비호환성도 고려해야 합니다. 인터넷 프로그래밍의 영역(특히 HTTP 영역)을 겨냥한 새로운 플랫폼이 등장하였습니다. 이러한 환경은 서버측 개발이 더 많아지고, 때로 웹 페이지내의 데이터베이스 정보도 포함합니다. 일반적인 해법은 HTML 페이지를 특수한 "스크립팅" 태그와 함께 작성하는 것인데, 이 태그는 서버에 있는 엔진에 의하여 확장됩니다. 작은 프로젝트에 대해서는 흥미롭지만, 이러한 기술의 대부분은 스크립팅 파워가 제한되어 있기 때문에 사용자가 HTML 코드와 스크립팅 코드 및 GUI 스크립팅 코드와 데이터베이스 지향 코드를 혼합해야 합니다. 대형 애플리케이션에 있어서는 프로그램의 서로 다른 영역들이 구별되지 않는 것은 전혀 좋은 구조가 아닙니다.

더욱이, 마이크로소프트의 DNA는 새로운 마이크로소프트 .NET 구조로 교체될 예정이고, 이 새로운 이름은 이전의 구조에 정말로 심각한 제약이 있었다는 것을 암시하는 것 같습니다. .NET은 분명히 더 "개방적"일 것이고, SOAP(Simple Object Access Protocol) 호출 프로토콜에 대한 지원의 강화를 포함한 XML의 중요성을 강조합니다. .NET의 또 다른 핵심 요소는 COM이 점차 사라질 것이 명백하다는 것입니다 (마이크로소프트의 예전 접근 방식에 투자해 왔던 사람들에게는 좋은 소식이 아니죠).

.NET이 도래하여도 HTML 스크립팅을 위해 ASP에 기반을 두고 데이터베이스 조작을 위해 COM/MTS/COM+에 기반을 두고 있는 마이크로소프트의 DNA 구조는 전망이 더 좋기는 하지만, 윈도우 플랫폼과 마이크로소프트 자신의 IIS 웹 서버에 제한되어 주로 인터넷 익스플로러 브라우저에서만 작동하는 경향이 있습니다. DNA의 현재 상태는 방화벽과 친하지 않은 DCOM, 복잡한 환경설정과 관리, 마이크로소프트 기술과의 결합, 포함된 데이터베이스 및 제한된 범위성 등의 몇 가지 제약이 있습니다. 또한, 부분적으로 상태를 유지하고, 부분적으로는 상태를 유지하지 않는 많은 레이어로 구성된 전체 구조는 아직 인터넷의 도전에 대응하기에는 여전히 제한되어 있는 것으로 보입니다.

델파이는 어디에 적합한가?

현재와 같이 다소 복잡한 상황이 진행되면 델파이와 같은 "전통적인" 개발 플랫폼은 어떤 분야에 적합할까요? 인터넷 시대에 델파이의 목표는 약간의 성능, 유연성, 사용의 용이성 및 데이터베이스 지원입니다.

- 델파이의 강력함은 많은 스크립트 기반의 기술들과는 달리, 완전히 컴파일된 코드와 완전히 객체 지향적인 구조에서 나옵니다. (이것은 이후 버전에서 이루어진 것이 아니라, 1.0 버전부터 언어와 라이브러리의 기반 구조였습니다.) 델파이로 네이티브 컴파일된 애플리케이션은 기본적으로 단일 실행 파일로 만들어지기 때문에 배포가 간단합니다. (추가 런타임 라이브러리나 사용자 컴포넌트 파일이 없음) 사실, 단일 EXE를 여러 패키지로 나누는 것은 델파이가 제공하는 유용한 옵션으로, 프로그래머들은 가장 좋은 배포 솔루션을 선택할 수 있습니다.
- 델파이의 유연성은 HTTP에 제한되지 않고 대부분의 인터넷 프로토콜에 개방적인 지원에서 나옵니다. 많은 써드파티 원시 컴포넌트 개발자들이 그랬던 것처럼 새로운 프로토콜을 지원하는 저수준의 코드를 작성할 수 있는 개발 환경을 제공합니다.
- 델파이의 사용상의 용이성은 컴포넌트 기반의 환경에서 나옵니다. 메일 클라이언트를 개발하기 위해서는 몇 가지 컴포넌트를 추가하고 몇가지 속성을 수정한 후 최소한의 코드를 작성하면 됩니다. 델파이와 써드파티 컴포넌트에 있는 예제 중 몇가지는 기본적으로 완전한 기능을 가진 이메일 프로그램입니다.

  델파이 5의 InternetExpress 기술에 있어서 사용상의 용이성은 데이터셋 기반의 HTML 페이지를 비주얼하게 개발하는 데까지 확장되었습니다.
- 데이터베이스와 클라이언트/서버 지원은 오래전부터 델파이의 가장 강력한 기능 가운데 하나였으며, 클라이언트/서버 구조는 대부분의 웹 애플리케이션과 인터넷 사이트의 핵심적 요소로 남아 있습니다.

실제로, 유저 인터페이스와 비즈니스 로직(보통 데이터 모듈을 이용)을 분리하여 델파이 애플리케이션을 구축하면, 기존의 비즈니스 로직 코드에 새로운 유저 인터페이스를 바로 접목할 수 있게 됩니다.
이것은 특히 비즈니스 로직과 유저 인터페이스를 서로 다른 애플리케이션으로 분리하여 클라이언트와 애플리케이션 서버 컴퓨터에서 실행되는 멀티티어 MIDAS 애플리케이션에 대해서는 더욱 그렇습니다.
InternetExpress기술을 이용하면, 웹서버 애플리케이션을 구축하기 위해서는 단순히 새로운 유저 인터페이스를 구축하는 것만으로 클라이언트쪽 브라우저에서 사용할 수 있게 됩니다.

기존의 델파이 코드를 활용할 수 있다는 점과 동일한 핵심 애플리케이션에 대해 윈도우와 브라우저 기반의 두가지 유저 인터페이스의 구축이 가능하다는 점이 델파이를 인터넷 개발 플랫폼으로 선택하는 두 가지 중요한 이유입니다. 그러나 이것이 모두는 아닙니다. 델파이에 포함된 기술들은 다른 인터넷 개발 영역들도 마찬가지로 지원하기 때문입니다.

마지막으로, 볼랜드는 카일릭스를 통해 "리눅스용 델파이"를 제공하여 사용자의 서버 애플리케이션이 마이크로소프트 윈도우나 리눅스 OS에서 모두 작동되도록 지원합니다. 델파이는 특정 OS에 구속되지 않으므로, 사용자의 웹 서버 애플리케이션이 인터넷 서버로 가장 많이 사용되는 두 가지 OS에서 작동되도록 해줍니다.

핵심적인 TCP/IP 지원

모든 인터넷/인트라넷 애플리케이션을 위한 공통적인 요소는 TCP/IP 소켓을 통한 통신입니다. 대개의 경우 통신은 통신 프로토콜이라고 불리는 규칙을 따릅니다. 예를 들어, SMTP와 POP3는 각각 메일 메시지를 주고 받기 위한 아주 단순한 프로토콜로, 인터넷 표준으로 정의되어 있습니다.

델파이를 사용하면 다음과 같은 작업을 할 수 있습니다.

- 컴포넌트 팔레트의 Internet 페이지에 있는 TServerSocket과 TClientSocket 컴포넌트를 이용하여 클라이언트 및 서버측의 독자적인 프로토콜을 구현할 수 있습니다. 이 방법은 분산 애플리케이션에 대해서는 간편하지만, 같은 개발자가 작성하지 않은 다른 프로그램과는 연동할 수 없는 폐쇄된 시스템이 됩니다(상황에 따라 장점이 될 수도 있고 단점이 될 수도 있습니다). 물론 그런 새로운 프로토콜을 다른 사람들과 함께 하기 위해 공개하지 않는 경우를 말하는 것입니다.

그림 1: 컴포넌트 팔레트의 Internet 페이지: 소켓과 HTML producer 컴포넌트들이 포함되어 있다

- 클라이언트측이나 서버측에서 기존의 프로토콜을 구현할 수 있습니다. 이것은 위에서 언급한 일반적인 소켓 컴포넌트로 구현할 수도 있지만, 보통은 특정 프로토콜에 전문화된 써드파티의 델파이 컴포넌트를 사용하며, 이 가운데 몇가지는 델파이 IDE에 이미 설치되어 있습니다.
- 웹의 핵심인 HTTP 프로토콜과 HTML 파일 포맷을 지원합니다. 이들의 역할은 매우 중요하기 때문에 다른 프로토콜들과는 별도로 다루겠습니다.

델파이에서의 TCP/IP와 소켓 프로그래밍에 대한 지원은 직접 API(윈도우 플랫폼의 경우 Winsock)를 사용하는 것만큼 강력하지만, 훨씬 더 간단합니다. 사실 소켓 컴포넌트는 복잡하고 세부적인 기술로부터 프로그래머를 보호하지만, 윈도우 핸들과 저수준 API를 노출하므로 사용자정의 저수준 호출을 가능하게 해줍니다.

단순한 C 윈도우 API 호출로 간단한 소켓 프로그램을 작성하는 작업은 수백 줄의 코드가 필요한 반면, 델파이 소켓 컴포넌트를 사용하면 복잡한 작업의 경우에도 단 몇 줄의 코드로 충분합니다. 이것이 컴포넌트를 기반으로 하는 개발의 가장 잘 알려진 장점입니다. 또한, 프로그램을 위한 간단한 유저 인터페이스를 구축하는 것은 델파이에서는 종종 사소한 작업이기 쉽습니다. 다른 개발 환경에서는 C와 같은 저수준의 언어로 소켓을 프로그래밍한 후 다른 비주얼 툴로 유저 인터페이스를 작성하여 두 가지를 통합하여야 하므로 여러 가지 언어에 대한 지식이 필요하게 됩니다.

클라이언트측 프로토콜 지원

델파이는 클라이언트측 인터넷 애플리케이션의 개발을 위한 즉시 사용할 수 있는 컴포넌트들을 제공합니다. 여러가지 선택 가능한 원시 VCL 컴포넌트들이 준비되어 있으며, 모두 비슷한 철학에 바탕하고 있습니다.

- NetMaster 컴포넌트는 델파이 환경에 이미 설치되어 있으며(컴포넌트 팔레트의 FastNet 페이지), 가장 일반적인 인터넷 프로토콜들(POP3, SMTP, NNTP, FTP, HTTP 등)에 대한 클라이언트측 지원을 포함하고 있습니다.
- Indy 오픈 소스 컴포넌트("Internet Direct", 이전에는 WinShoes라고 불렸으며 현재는 Jedi Project에 통합됨)는 델파이 5의 컴패니언 CD나 웹 사이트에서 구할 수 있습니다(http://www.nevrona.com/indy). 델파이의 6 이상 버전이나 카일릭스(리눅스용 델파이 IDE)에는 Indy가 기본으로 설치되어 있습니다.
- 무료 ICS 컴포넌트("Internet Component Suite", http://overbyte.delphicenter.com/eng/products/ics.html에서 전체 소스코드를 포함해서 다운로드 가능)는 Francois Piette에 의해 설계 및 관리되고 있으며, 또 하나의 매우 인기있는 델파이 컴포넌트로서 대부분의 인터넷 프로토콜을 지원합니다.
- 상업적으로 판매되고 있는 것으로 IP*Works 컴포넌트(http://www.dev-soft.com/products/ipworks.aspx)와 Turbo Power의 Internet Professional (http://www.turbopower.com/products/IPRO/) 등.

이들 컴포넌트들 중 일부는 자신만의 WinSock 래퍼에 직접 매핑되고, 나머지는 클라이언트 FTP와 HTTP 프로토콜을 지원하는 마이크로소프트 시스템 DLL인 WinInet 라이브러리를 이용합니다. 사용할 컴포넌트 셋이 무엇이든 작업하기는 매우 간단합니다. 기존의 애플리케이션에 메일 기능을 추가하고자 하는 경우, 폼(또는 데이터 모듈)에 두 가지 컴포넌트를 놓고 속성을 설정(연결할 메일 서버 설정 등)한 후에 몇 줄의 코드를 작성하면 됩니다.

예를 들어, NetMaster의 컴포넌트로 이메일을 보내려면 다음과 같은 간단한 코드를 사용할 수 있습니다.

```
// component properties

object Mail: TNMSMTP
  Host = 'mail.server.com' // your web service
  Port = 25
  PostMessage.FromAddress =
    'marco@marcocantu.com'
end

// code to send the above email message
Mail.PostMessage.ToAddress.Add(
  'davidi@borland.com');
PostMessage.Subject := 'borland Community Site';
PostMessage.Body.Add(
  'Hi David, I wanted to ask you...');
Mail.Connect;
Mail.SendMail;
Mail.Disconnect;
```

요약하면, 인터넷 클라이언트 애플리케이션을 지원하기 위해 델파이를 사용할 때의 장점은 다음과 같습니다.

- 다양한 컴포넌트들 중에서 선택 가능 (몇가지는 완전히 무료이고 오픈 소스임)
- 기존의 애플리케이션과 쉽게 통합됨
- 델파이의 비주얼하고 객체지향적 구조로 새로운 특정 유저 인터페이스의 개발이 쉬움

서버측 프로토콜 지원

기존 애플리케이션에 웹 프로토콜을 지원하거나 특정 프로토콜을 위한 클라이언트 프로그램을 작성하는 경우 외에, 기업 환경에서는 종종 독자적인 인터넷 서버 애플리케이션을 개발할 필요가 있습니다. 물론, 이미 존재하는 여러 서버들은 새로운 기능을 추가하여 사용될 수 있으나, 때로는 기존의 프로그램이 지원하지 않는 것을 제공할 필요가 생깁니다.

이러한 경우, 너무 복잡하지 않다면 독자적인 서버를 작성하는 것을 생각할 수 있습니다. 델파이와 서버측 컴포넌트 셋을 이용하면 클라이언트 프로그램에 비해 약간의 수고만 더하면 사용자 정의 서버를 구축할 수 있고, 전문가 품질의 인터넷 서버 프로그램의 성능을 (때로는 더 나은 성능을) 얻을 수 있습니다.

서버측 컴포넌트는 웹 서버 컴포넌트를 제공했던 Jaadu(http://www.jaadu.com)에 의하여 시작되었으며, 위에서 언급한 Indy 컴포넌트 셋에 포함되어 있습니다. 또한, DXSock (http://www.dxsock.com)이라고 불리는 훌륭하게 최적화된 네이티브 델파이 컴포넌트 셋이 있는데, 이것은

구체적으로 인터넷 서버 프로그램의 개발을 목표로 하고 있습니다. 이러한 컴포넌트 셋의 일부 데모는 실제로 완벽한 기능의 HTTP, 메일 및 뉴스 서버입니다.

클라이언트측 웹 지원

많은 인터넷 프로토콜들이 중요하고 이메일이 가장 많이 사용되는 인터넷 서비스 가운데 하나이지만, 웹(HTTP 프로토콜)이 대부분의 인터넷 개발의 원동력이라는 것은 부인할 수 없습니다. 델파이에서의 웹 지원은 특별히 더 풍부합니다. 여기에서는 우선 클라이언트측에서 사용할 수 있는 기능들을 살펴본 후에 (기존 브라우저와 통합하기 위해), 서버측으로 이동하여 델파이로 할 수 있는 웹 서버 개발에 충분한 시간을 할애할 것입니다.

대부분의 컴포넌트 셋에 포함된 HTTP 컴포넌트는 애플리케이션내에서 사용자정의 브라우저를 생성할 수 있도록 해줍니다. 웹 사이트에서 HTML 파일을 받거나 질의를 전송할 수 있습니다. 이 때 HTTP 서버에서 받은 HTML의 내용을 기존의 브라우저로 보내거나 사용자의 애플리케이션에 별도의 HTML 프로세서나 HTML 뷰어를 통합할 수 있습니다. 적용 가능한 방법은 다음과 같습니다.

- ShellExecute API 함수를 호출하여 외부 브라우저(마이크로소프트 인터넷 익스플로러 혹은 넷스케이프 네비게이터)로 HTML 파일을 보내거나 델파이에서 WebBrowser 컴포넌트로 지원하는 인터넷 익스플로러 ActiveX 컨트롤을 통합하여 HTML 파일을 읽어들임.

```
ShellExecute (Handle, 'open',
'c:\tmp\test.htm', '', '',
sw_ShowNormal);
```

- 특정한 정보를 추출하기 위해 HTML을 사용자 정의 방법으로 처리. 이 방법은 HTML을 사용자에게 보여주지 않고 처리하여 특정한 정보를 추출하고자 할 때 유용합니다.
- 네이티브 델파이 컴포넌트를 사용하여 HTML을 사용자의 프로그램 안에 보여줌. (따라서 사용자에게 인터넷 익스플로러가 설치되어 있을 필요가 없음) 이러한 컴포넌트는 써드파티로부터 입수할 수 있으며, 가장 유명한 HTML 뷰어 컴포넌트는 David Baldwin이 제공하는 것입니다 (웹 사이트 http://www.pbear.com 참고).

브라우저를 애플리케이션 내에 통합하는 작업과 반대로, 애플리케이션을 브라우저 내에 통합할 수도 있습니다. 이것은 인터넷 익스플로러와 ActiveX 기술을 사용하면 조금 더 쉽습니다. 델파이는 ActiveForm 기술을 이하여 ActiveX 기술을 완벽하게 지원합니다. ActiveForm은 단순한 델파이 폼처럼 비주얼한 방법으로 만들지만, ActiveForm은 인터넷 익스플로러의 HTML 페이지 안에 포함될 수 있습니다. ActiveForm에 메인 폼을 포함시키면 기존의 프로그램을 웹으로 옮길 수도 있습니다.

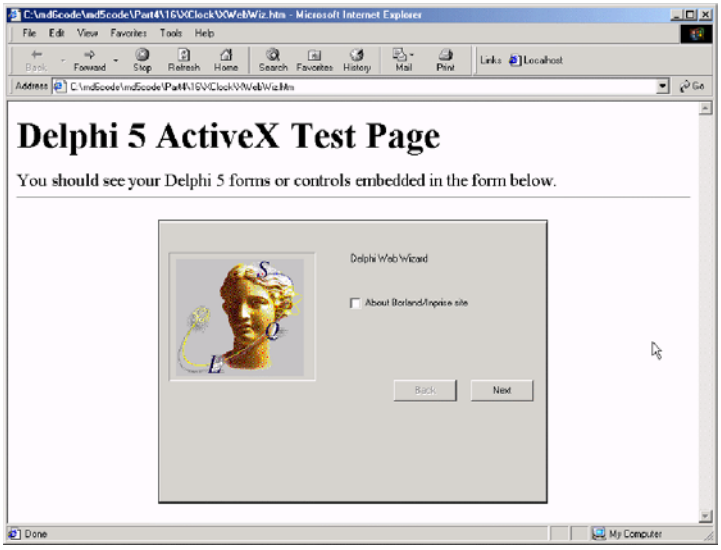

그림 2: 델파이의 ActiveForm으로 개발한 위저드의 예 (PageControl 컴포넌트를 이용). 버튼을 이용하여 브라우저의 현재 페이지 내에서 폼의 다른 페이지로 이동할 수 있음

이 마이크로소프트에 한정된 기술(다른 브라우저는 ActiveX를 지원하지 않음)은 사용자가 브라우저를 특정 페이지로 이동함으로써 필요한 프로그램을 다운로드 받을 수 있게 해주므로 인트라넷 안에서의 간단한 델파이 애플리케이션의 배포가 간편합니다. 그러나, ActiveX 기술은 너무 많은 사람들이 다른 브라우저 혹은 OS를 사용하거나, 웹에서 다운로드한 프로그램이 자동으로 실행되어 피해를 입는 것을 우려하여 브라우저에서 이 기능을 비활성화 시키기 때문에 인터넷에는 적절하지

않습니다.

- 이 분야에 있어서 델파이의 장점은 인터넷의 잇점을 활용하기 위해 기존 프로그램을 수정하여 웹 브라우저와 매끄럽게 연동할 수 있게 할 수 있다는 것입니다. 또한, ActiveX 기술을 이용하지 않기로 선택한다면 어떤 특정 브라우저나 플랫폼에도 구속되지도 않게 됩니다.

서버측 웹 개발

웹 서버에 관련된 모든 개발은 인터넷 개발에서 단연 가장 중요한 분야이며, 많은 대체 솔루션을 사용할 수 있는 분야이기도 합니다. 델파이는 버전 3부터 개발자들에게 강력한 서버측 웹 개발을 제공하였으며, 델파이 5는 웹 서버 확장을 개발하기 위한 성숙된 환경입니다. 델파이에는 서버측 개발을 지원하기 위한 여러 기술들이 포함되어 있으므로, 여기에서 다음과 같은 주제들을 다루기로 합니다.

- HTML Producer 컴포넌트
- WebBroker 기술: CGI, WinCGI 및 ISAPI/NSAPI 서버측 확장의 개발에 이용
- Internet Express 기술(델파이 5에서 도입): 데이터베이스 기반 서버측 애플리케이션 개발에 이용. XML 및 ECMAScript (이전의 JavaScript) 등의 표준 기술을 이용

HTML 페이지 생성

델파이에는 동적인 HTML 페이지를 생성하기 위한 여러 가지 컴포넌트가 포함되어 있습니다. Producer 컴포넌트에는 페이지 지향적인 것과 테이블 지향적인 것의 두 가지 세트가 있습니다. PageProducer와 같은 페이지 지향적인 HTML Producer 컴포넌트를 사용하는 경우에는 Producer 컴포넌트에 사용자 정의 태그(# 문자로 표시됨)가 있는 HTML 파일을 넘겨줍니다. 그러면 이 컴포넌트의 OnTag 이벤트에서 이런 사용자 정의 태그를 특정한 HTML 코드로 치환시킬 수 있습니다.

다음은 OnTag 이벤트의 예제 코드입니다.

```
Procedure TFormProd.PageProducer1HTMLTag(Sender:
  TObject; Tag: TTag; const TagString: String;
  TagParams: TStrings; var ReplaceText: String);
var
  nDays: Integer;
begin
  if TagString = 'date' then
    ReplaceText := DateToStr (Now)
  else if TagString = 'expiration' then
  begin
    nDays := StrToIntDef
    (TagParams.Values['days'], 0);
    if nDays <> 0 then
      ReplaceText := DateToStr (Now + nDays)
    else
      ReplaceText := '<I>{expiration
        tag error}</I>';
  end;
end;
```

이 코드는 단순 태그인 날짜를 현재 날짜로 치환시키며, 파라미터를 가진 태그인 expiration은 페이지에 있는 정보가 유효한 날의 수를 가리키는 파라미터 하나를 가집니다. 이 사용자 정의 태그에 대한 HTML은 다음과 같습니다.

```
Prices valid until <b><#expiration days=21></b>
```

그 결과는 그림 3에서처럼, "The prices in this catalog are valid until **12/24/2000**"의 형태로 나타날 것입니다.

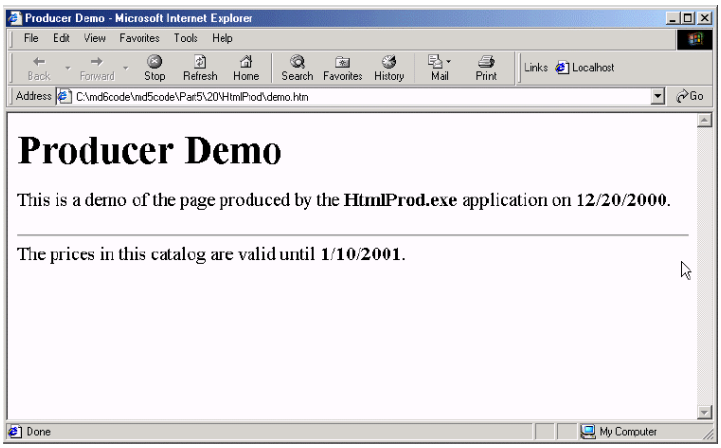

그림 3: PageProducer 컴포넌트로부터 생성된 HTML 파일

이러한 접근 방법의 장점은 사용자가 선호하는 HTML 편집기를 사용하여 페이지를 작성한 후 단순히 사용자 정의 태그를 추가하기만 하면 된다는 것입니다. 또한, 동일한 OnTag 이벤트 처리기를 다수의 Producer 컴포넌트가 공유할 수 있으므로 동일한 프로그램에서 같은 태그 확장을 위해 여러 번 코딩하지 않아도 됩니다.

이 그룹의 두 번째 컴포넌트인 DataSetPageProducer는 태그 이름을 데이터셋의 현재 레코드의 필드값으로 자동으로 바꿀 수 있습니다.

또 다른 그룹에는 HTML 테이블 지향적인 컴포넌트가 포함되어 있습니다. 이것은 데이터셋(테이블, 또는 쿼리나 스토어드프로시저의 결과)을 자동으로 HTML 테이블로 변환하는 것입니다. 기본적인 변환 방법이 있긴 하지만, 표와 각 컬럼, 심지어는 특정 셀에 대해서 사용자 정의 태그와 스타일을 추가할 수 있습니다. 사용자 정의 과정은 윈도우 애플리케이션 안에서 비주얼한 DBGrid에 적용하는 것과 유사합니다. 예를 들어, 아래의 코드는 두 번째 컬럼의 셀 가운데 8자리가 넘는 값을 가지는 모든 셀을 빨간색(그림 4 참고)으로 바꿉니다.

```
procedure
TFormProd.DataSetTableProducer1FormatCell(
  Sender: TObject;
  CellRow, CellColumn: Integer;
  var BgColor: THTMLBgColor;
  var Align: THTMLAlign;
  var VAlign: THTMLVAlign;
  var CustomAttrs, CellData: String);
begin
  if (CellColumn = 1) and (Length (CellData) > 8)
  then
    BgColor := 'red';
end;
```

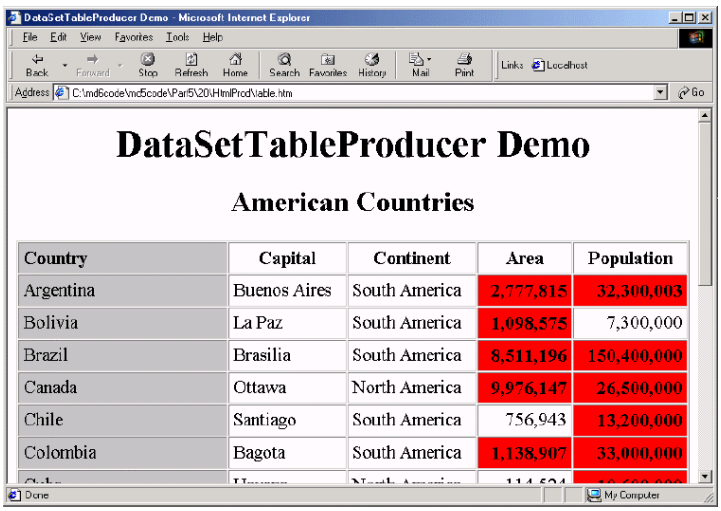

그림 4: 특정 셀에 대하여 사용자 정의 색을 표시한 DataSetTableProducer의 결과

두 번째 컴포넌트인 QueryTableProducer는 특별히 HTML 탐색 폼으로부터의 입력에 기초한 매개변수를 가진 쿼리를 구축하기 위하여 만들어진 것입니다. 폼에 입력된 매개변수들은 컴포넌트에 의하여 자동으로 SQL 쿼리의 매개변수로 변환되고, 그 결과 데이터셋은 HTML 테이블로 포맷됩니다. 이 모든 복잡한 작업을 사용자 정의 코드 없이 설정할 수 있습니다!

장점

- 선호하는 편집기로 기본 HTML 코드를 작성하여 단순히 사용자 정의 코드를 추가하기만 하면 됩니다.
- 스크립팅 코드와 HTML 코드가 섞이지 않고 완전히 분리되게 됩니다. HTML은 단순히 생성될 코드에 대한 장소 유지 역할만 하게 됩니다.
- 스크립트는 완전한 성능을 내는 컴파일된 코드로 대체됩니다.
- 이 기술로 데이터베이스 데이터에 쉽게 액세스할 수 있게 되며, 복잡한 쿼리 결과를 사용자 정의 코드 없이 HTML 테이블로 표시할 수 있습니다.

참고

HTML Producer 컴포넌트는 정적인 HTML 파일을 생성하기 위해서도 이용될 수 있는데, 다시 말하면 정적인 파일을 서버가 아닌 프로그램에서 동적으로 생성하여 웹서버에 위치시킬 수 있다는 뜻입니다.

또 한가지 주목할 점은, 델파이 프로그램은 HTML 파일 이외에 TJPEGImage 컴포넌트를 사용하여 JPEG 파일을 생성할 수 있습니다. 이러한 파일들도 서버에 위치시키거나 서버 확장으로부터 동적으로 생성될 수 있습니다. 복잡한 비즈니스 그래프를 생성할 수도 있는데, 네이티브 TeeChart 컴포넌트를 이용하면 됩니다.

WebBroker 프레임워크

웹 서버 확장(웹 서버와 완벽하게 통합되는 사용자 정의 애플리케이션)의 개발은 다음을 포함한 다양한 기술에 기반합니다.

- CGI (Common Gateway Interface, UNIX 머신에서 주로 사용됨)
- WinCGI (동일한 기술의 윈도우 변종)

- ISAPI(Internet Server API; 마이크로소프트의 IIS에 맞추어진 라이브러리) 및 NSAPI (넷스케이프 웹서버에서 지원하는 비슷한 API)
- Apache 모듈 (동일한 아이디어이지만 오픈 소스 Apache 웹 서버를 위한 것) – 델파이 6 이후 버전과 카일릭스에서 지원

이들 대부분의 기술에 있어서의 문제는, 모두 HTTP 프로토콜을 기반으로 하더라도 같은 정보를 받아서 웹 서버에 활용하는 방법은 크게 다르다는 것입니다. 이런 이유로, 볼랜드는 VCL에 WebBroker라는 작은 객체 지향의 프레임워크를 내장하여 이러한 차이를 제거하도록 하였습니다. 개발자는 모든 코드를 몇 가지 포괄적인 베이스 클래스를 기초로 작성한 후에, 델파이를 이용하여 CGI나 ISAPI 등의 특정 기술을 구현하도록 합니다. 이것은 다른 프로젝트 소스 코드와 몇 줄의 코드만 추가하면 아무리 복잡한 프로그램이라도 하나의 기술에서 다른 기술로 옮길 수 있다는 것을 의미합니다.

- 이같은 브리지 아이디어는 델파이6와 카일릭스의 등장 이후로 윈도우 기반 서버 외에 리눅스 기반 서버도 포함하게 되었으므로, WebBroker 기술은 다양한 OS 플랫폼의 다양한 웹서버들 사이를 오갈 수 있습니다.

WebBroker는 여러 기술 사이를 연결할 뿐 아니라, 많은 핵심 루틴과 기능을 제공하여 서버측 개발을 간단하게 합니다. 예를 들어, 복잡한 문자열을 분석할 필요 없이 다음과 같은 문장으로 쿼리 안에서 특정한 값을 요구할 수 있습니다.

```
stringName := Request.QueryString['Name'];
```

이것은 단지 간단한 하나의 예에 불과하며, WebBroker 구조에는 개발 속도를 높여주기 위해 시간을 절약할 수 있는 수많은 방법이 있습니다.

또한, WebBroker 구조는 일반적으로 HTML Producer 컴포넌트들과 함께 사용된다는 것도 알아두십시오. SQL 서버에서 쿼리를 실행하고, HTML 테이블을 이용하여 포맷한 후에, 서버측 애플리케이션으로부터 그 결과를 받아오는 프로그램을 개발하는 데에는 20번 미만의 마우스 클릭만으로 충분하며 코드 작성도 거의 필요하지 않습니다!

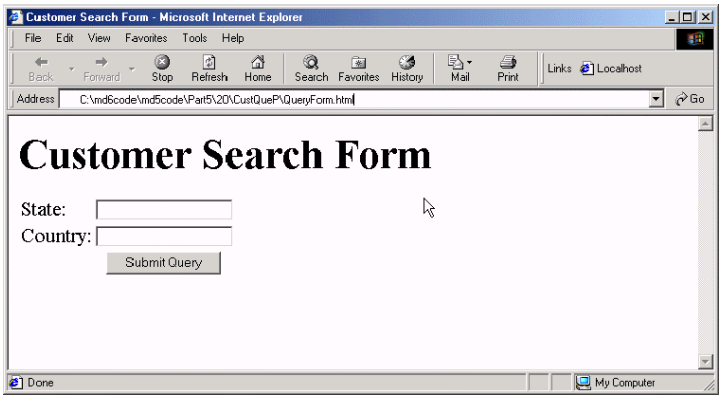

그림 5: 이와 같은 쿼리 폼은 QueryTableProducer 컴포넌트를 이용하면 코드를 거의 작성할 필요 없이 SQL 쿼리의 매개변수에 직접 연결될 수 있습니다!

예를 들어, 테이블이 있는 다음과 같은 HTML 파일이 있을 경우(그림 5 참조), 스크립트(CustQuery.exe)와 연결하여 요청을 처리할 수 있습니다. 이것은 2개의 입력 필드가 있는 테이블을 가진 HTML 코드입니다.

```
<html><head>
    <title>Customer Search
Form</title>
</head>
<body>
<h1>Customer Search Form</h1>
<form
action="/scripts/CustQuery.exe/search"
method="POST">
<table>
<tr><td>State:</td>
    <td><input type="text"
name="State"></td></tr>
<tr><td>Country:</td>
    <td><input type="text"
name="Country"></td></tr>
    <tr><td></td>
<td><center><input
type="Submit"></center></td></tr>
</form>

</body>
</html>
```

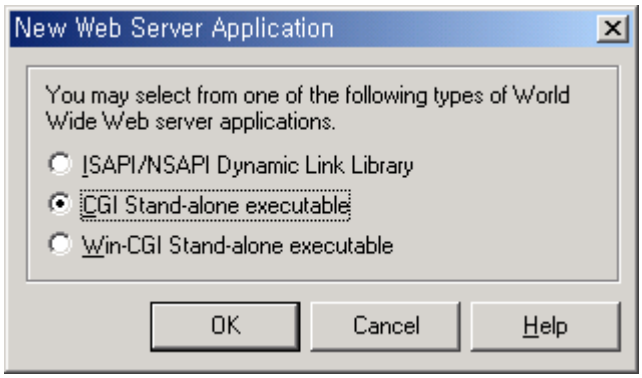

그림 6: 델파이의 웹 서버 애플리케이션 위저드에서 웹 서버 기술을 선택

이제 웹 서버 애플리케이션 마법사(그림 6)에서 CGI (혹은 기타 원하는 기술)를 선택하여 델파이 WebBroker 애플리케이션을 생성할 수 있습니다. WebModule에서 오브젝트 트리뷰(그림 7의 위)의 Actions 항목을 오른쪽 마우스로 클릭하거나 액션리스트 에디터 (그림 7의 아래)에서 추가 버튼을 이용하여 액션을 추가할 수 있습니다. WebModule을 더블 클릭하면 액션리스트 에디터가 열립니다.

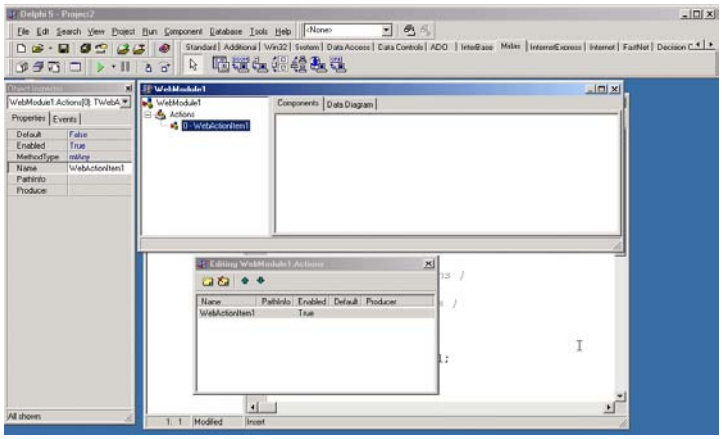

그림 7: WebModule의 액션들이 오브젝트 트리뷰(위)와 액션리스트 에디터(아래)에 표시됩니다. 속성은 오브젝트 인스펙터(왼쪽)에서 설정합니다.

액션의 PathInfo 속성을 "/search" 값으로 설정합니다. 이 컴포넌트는 Producer 속성을 이용하여 데이터모듈에 추가한 QueryTableProducer 컴포넌트에 연결할 수 있습니다. 또, 이 컴포넌트는 Query 속성을 통하여 Query 컴포넌트에 연결됩니다.

Query 컴포넌트는 액션이 호출될 때 실행되어 웹 요청의 QueryStrings 또는 ContentStrings 매개변수를 Params로 보냅니다. 이것은 HTML input box에 입력된 값들을 델파이가 추출하여 같은 이름의 쿼리 매개변수로 복사한다는 뜻입니다. 따라서, 입력 필드와 같은 이름을 가지는 매개변수에 대하여 쿼리를 다음과 같이 사용할 수 있습니다. (위의 HTML 코드를 다시 참고하십시오).

```
SELECT Company, State, Country
FROM CUSTOMER.DB
WHERE
    State = :State OR Country = :Country
```

이것이 전부입니다! 델파이 코드도 없이 데이터베이스 검색을 위한 웹페이지를 구현했습니다. HTML 테이블 출력을 사용자 정의하고, 스타일 시트를 첨부하고, 사용자 정의 프로세스를 위하여 추가 코드를 추가하면 몇 시간 안에 이 프로그램의 전문가적인 버전을 구축할 수 있습니다.

이상을 요약하면 다음과 같습니다.

- WebBroker는 CGI, Win-CGI, ISAPI/NSAPI 및 Apache Module 등 다양한 기술을 위한 단일 소스 코드 솔루션을 제공합니다.
- HTML producer 컴포넌트와 함께 사용하면 서버측 애플리케이션은 HTML 페이지를 쉽게 생성할 수 있으며, 특히 데이터베이스 데이터를 표시할 수 있습니다.
- 카일릭스를 이용하면 WebBroker 코드를 쉽게 리눅스에 이식할 수 있습니다.

마이크로소프트의 윈도우 DNA 구조 지원

WebBroker 프레임워크와 나중에 설명할 Internet Express 기술에 관한 것 이외에, 델파이는 전체적인 윈도우 DNA 구조(후에 .NET으로 대체되겠지만 당분간 계속 사용될 것임)에 대하여 완전하고도 고품질의 지원을 제공합니다. 델파이는 마이크로소프트 자체 비주얼 개발툴의 지원 이전부터 ActiveX와 MTS 기술을 지원하는 첫번째 비주얼한 개발 환경이었습니다.

이것뿐 아니라 델파이의 간단하고도 완벽한 COM 지원은 아직도 업계에서 최고입니다. 이러한 고품질의 저수준 COM 지원을 바탕으로 델파이는 윈도우 셸 프로그래밍, 오토메이션, Active Document, ActiveX (이미 설명한 웹 지향의 ActiveForm 기술 포함), MTS 및 COM+ 등과 같은 대부분의 COM 관련 기술을 지원합니다.

델파이를 사용하면 비즈니스 룰과 데이터베이스 통합 코드를 정의하는 MTS 객체를 작성하고, HTML과 유저 인터페이스 요소를 생성하기 위한 ASP에서 사용가능한 COM 객체의 층을 제공하며, 모든 것을 ASP 스크립트 (마이크로소프트의 Active Server Page 기술)로 감쌀 수 있습니다. 이것은 마이크로소프트의 제안을 완벽한 고품질 지원을 통하여 전적으로 수용하는 것에 해당합니다.

이러한 과거의 전력으로 볼 때, 우리는 앞으로 출시될 델파이가 COM+ (물론 약간의 노력을 기울이면 지금도 델파이 5로 COM+ 애플리케이션을 작성할 수 있기는 합니다) 및 새로운 마이크로소프트 표준을 완전히 지원할 것으로 기대할 수 있습니다.

이상을 요약하면 다음과 같습니다.

- 윈도우 플랫폼에 대한 델파이의 지원은 완벽하며, 전체 COM 구조 및 윈도우 DNA 모델에 대한 지원을 포함합니다.
- 델파이는 저수준의 COM 코드를 작성할 수 있기 때문에 개발 환경에서 볼랜드의 지원을 기다릴 필요 없이 새로운 표준을 목표로 할 수 있습니다. 다른 비주얼 개발환경에 대해서는 이와 같은 말을 할 수 없을 것입니다.
- 델파이를 사용하면 마이크로소프트 기술을 완전하게 지원하면서도 마이크로소프트의 툴을 사용함으로써 제한되는 것을 피할 수 있습니다. 약간의 공학적 노력을 기울이면 델파이 클래스를 구축하여 마이크로소프트의 DNA 기술(예를 들어 ISAPI 서버와 COM 객체에서 델파이 코드를 사용하는 것)에 적합하게 기능을 노출시킬 수 있고, 동일한 핵심 코드에 다른 래퍼를 제공함으로써 다른 웹 서버나 리눅스와 같은 다른 플랫폼으로 이식할 수도 있습니다.

InternetExpress: XML, ECMAScript

DHTML

델파이 5는 가장 최근의 공개 표준에 기반을 둔 전혀 새로운 기술을 제공함으로써 인터넷 개발 영역에서 기존의 델파이 기능을 더욱 확장하였습니다. InternetExpress는 두 가지 핵심적인 컴포넌트에 기반을 두고 있습니다.

- XMLBroker 컴포넌트는 기존의 데이터셋을 (MIDAS 데이터 스트림 포맷을 사용하여) XML 데이터로 변환할 수 있습니다. 쿼리 결과와 테이블 전체를 변환할 수 있으며 델파이 데이터셋 컴포넌트(BDE나 마이크로소프트의 ADO에 기반을 둔 컴포넌트 또는 네이티브 InterBase 컴포넌트)를 이용하여 XMLBroker에 데이터를 제공하고 웹 페이지에 출력할 수 있습니다.
- MidasPageProducer 컴포넌트는 XMLBroker가 제공하는 데이터에 기초한 HTML 폼을 위하여 고안된 비주얼 컴포넌트입니다. 이러한 페이지들이 일단 브라우저에서 사용되면, 사용자가 데이터베이스 데이터를 볼 수 있을 뿐만 아니라, 데이터베이스에 대한 편집, 삭제 및 데이터 삽입을 완벽하게 지원할 수 있습니다.

유저 인터페이스 구성은 일반 윈도우 유저 인터페이스와 비슷하지만, 이것은 플러그인이나 사용자 정의의 확장 없이도 여러 브라우저에서 실행되는 네이티브 웹 애플리케이션입니다. 이러한 공개성은 Internet Express 기술이 다음과 같은 공개 표준에 기반을 두고 있기 때문입니다.

- XML 및 XML DOM
- ECMAScript (JavaScript 기술의 공식 명칭)는 대부분의 웹 브라우저가 지원하는 유일한 스크립팅 언어이며, 유저 인터페이스를 사용자 정의하고, 클라이언트측에서 간단한 입력과 편집 규칙을 적용하며, 유저 인터페이스가 XML 데이터와 상호 작용하도록 사용할 수 있습니다.
- 동적 HTML과 CSS(Cascading Style Sheets)는 브라우저 안에서 현대적인 유저 인터페이스를 개발하도록 하고, 그래픽 요소가 HTML이나 데이터 규칙에 구속되지 않도록 합니다. 사실 반대로 다양한 요소들(HTML 코드, 비즈니스 규칙, SQL 서버 액세스)은 잘 분리되어

있습니다.

위에서 언급한 바와 같이 InternetExpress 구조는 볼랜드의 MIDAS 3(델파이 6 이후 버전에서는 DataSnap) 기술에 기반을 두고 있습니다. 완전히 확장된 상태에서 그 구조는 다음과 같은 4개의 레이어(그림 8 참조)로 구성되어 있습니다.

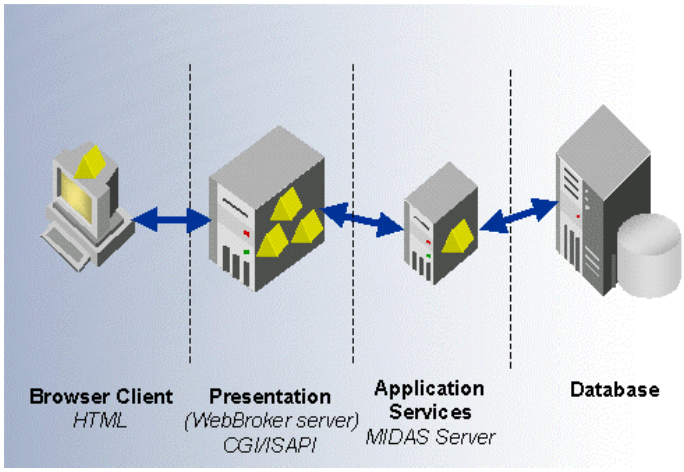

그림 8: 멀티티어 Internet Express 애플리케이션의 여러 레이어

- SQL 서버: 별도의 컴퓨터에서 실행 (BDE, ADO 또는 네이티브 컴포넌트가 지원하는 서버)
- MIDAS 애플리케이션 서버: SQL 서버에 연결하여 특정 비즈니스 룰을 적용하고 클라이언트에게 데이터를 제공
- WebBroker 웹 서버 확장: 웹 서버와 결합되어 MIDAS 서버가 수신한 데이터를 XML로 변환하고 적절한 HTML 유저 인터페이스를 제공
- 웹 브라우저: 인터넷 익스플로러 또는 넷스케이프, 또는 표준에 적합한 다른 것으로 운영 시스템에서 작동하는 것

간단한 프로젝트의 경우 이 그림은 MIDAS 서버와 XML 제작 컴포넌트를 하나의 애플리케이션으로 합쳐서 간략하게 표시될 수 있습니다. 이러한 전체 구조의 장점은 비즈니스 로직의 적절한 추출, 전송 독립성 (TCP/IP, HTTP, DCOM, MTS 및 CORBA보다 상위에서 작동할 수 있음) 및 (SQL 서버 연결 공유를 위한) 리소스 풀링 등을 포함한 MIDAS 구조의 혜택을 얻을 수 있다는 것입니다.

실습을 위해서 간단한 데이터베이스 테이블을 편집하기 위한 간단한 Internet Express 앞단을 구축하는 단계를 살펴보기로 합니다. 마법사를 이용하여 새로운 웹 서버 애플리케이션을 생성하는 것에서 시작합니다. 웹 모듈에서 액션(그림 7과 관련 예제의 설명 참조)을 추가하고, Default 속성을 True로 설정하여 이 액션을 기본 액션으로 표시합니다. 이제 다음과 같은 컴포넌트가 필요합니다.

- 테이블이나 쿼리 컴포넌트 등의 실제 데이터셋: BDE, ADO 또는 기타 데이터베이스 액세스 기술을 사용하여 데이터베이스에 연결된 데이터셋. 가장 간단한 방법은 컴포넌트 팔레트의 Data Access 탭의 BDE Table을 사용하고 DatabaseName과 TableName 속성을 설정하는 것입니다. DatabaseName 속성에는 샘플 DBDEMOS 알리아스를 사용할 수 있으며, 속성의 드롭다운 메뉴에서 사용 가능한 아무 테이블이나 선택할 수 있습니다.
- 데이터셋 프로바이더 컴포넌트(컴포넌트 팔레트의 MIDAS 탭): DataSet 속성을 사용하여 데이터셋(여기에서는 Table1)에 연결
- XML 브로커 컴포넌트(컴포넌트 팔레트의 InternetExpress 탭): ProviderName 속성을 사용하여 ("단일" 티어 접근에서) 프로바이더 컴포넌트에 직접 연결
- Midas Page Producer 컴포넌트(컴포넌트 팔레트의 InternetExpress 탭): 액션의 Producer 속성을 설정함으로써 웹 모듈의 기본 액션에 연결

이 시점에서 HTML 폼을 준비하기 위하여 Midas Page Producer를 더블 클릭으로 열고 특수 에디터를 사용할 수 있습니다. 예를 들어, 데이터 폼을 추가하고 거기에 (XML broker에 연결된) 데이터 그리드와 (XMLComponent 속성을 사용하여 그리드에 연결된) 내비게이터를 추가할 수 있습니다.

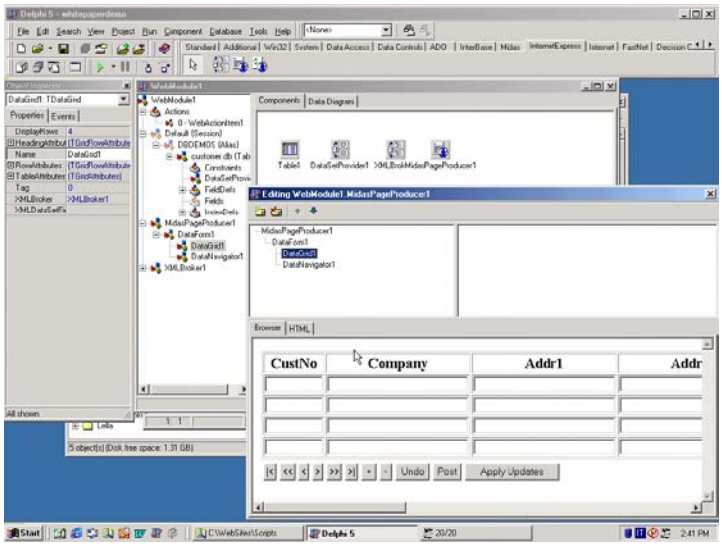

그림 9: InternetExpress 애플리케이션의 WebModule 안에 있는 편집기 MidasPageProducer 컴포넌트

이러한 단계를 거쳐서 복잡한 데이터 모듈을 갖게 됩니다. 주요 옵션을 요약하기 위하여 비주얼한 개발 활동의 결과를 반영하는 내부 델파이 파일인 DFM 파일의 텍스트 버전을 보이도록 하겠습니다. 여러분은 이 코드를 타이핑할 일이 절대로 없으며, 이것은 단지 오브젝트 인스펙터에서의 비주얼한 설정에 대한 요약으로서, 참고나 문서화의 목적상 매우 유용합니다. 마우스 오른쪽 버튼을 클릭하여 "View As Text" 명령을 선택하면 자신의 WebModule을 볼 수 있습니다. 여기에서는 DFM 파일의 텍스트 정의에서 주요 부분만 추출하였기 때문에 실제는 이것보다 코드가 훨씬 많습니다.

```
object WebModule1: TWebModule1
  Actions = <
    item
      Default = True
      Name = 'WebActionItem1'
      PathInfo = '/MidasPageProducer1'
      Producer = MidasPageProducer1
    end>
  object Table1: TTable
    Active = True
    DatabaseName = 'DBDEMOS'
    TableName = 'country.db'
  end
  object DataSetProvider1: TDataSetProvider
    DataSet = Table1
  end
  object XMLBroker1: TXMLBroker
    ProviderName = 'DataSetProvider1'
    WebDispatch.PathInfo = 'XMLBroker1'
  end
  object MidasPageProducer1:
    TMidasPageProducer
    HTMLDoc.Strings = (...)
    IncludePathURL = '/include/'
    object DataForm1: TDataForm
      object DataGrid1: TDataGrid
        XMLBroker = XMLBroker1
      end
      object DataNavigator1:
        TDataNavigator
        XMLComponent = DataGrid1
      end
    end
  end
end
```

Midas Page Producer의 IncludePathUrl 속성을 브라우저가 필요로 하는 JavaScript 파일들이 들어 있는 디렉토리를 표시하는 URL에 설정해야 합니다. 그렇지 않으면 브라우저가 오류 메시지를 표시하고 아무 데이터도 표시하지 않습니다.

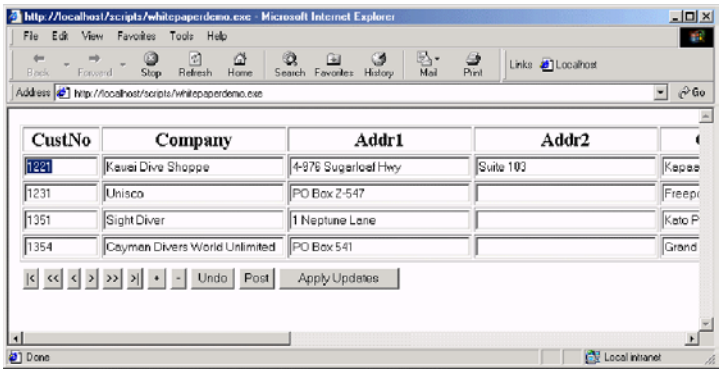

그림 10: 브라우저(여기에서는 마니크로소프트 인터넷 익스플로러) 안에 구축한 InternetExpress 애플리케이션

이것이 전부입니다! 이제 완전한 애플리케이션을 갖게 되었으며, 사용자는 어떠한 브라우저 안에서나 데이터를 볼 수 있을 뿐 아니라 데이터를 편집하고 다시 서버로 보내어 데이터베이스에 넣을 수 있습니다. 그림 9의 예를 참조하십시오. 또한, 코드 작업의 필요 없이 기본 옵션만을 선택했지만, 약간의 연구(참고 서적 가운데 하나는 필자의 저서인 "Mastering Delphi 5")와 노력을 하면 현대적인 브라우저 기반의 UI와 함께 복잡한 프로그램을 구축할 수 있습니다.

이상을 요약하면 다음과 같습니다.

- 델파이 InternetExpress 구조를 이용하면 브라우저

기반의 유저 인터페이스를 가지고 MIDAS 멀티티어 애플리케이션을 확장할 수 있습니다. HTTP, XML 및 ECMAScript에 의존하는 구조는 다른 비슷한 성능의 솔루션들과는 달리 어떠한 브라우저나 OS에도 구속되지 않습니다.

- InternetExpress로 비즈니스 데이터를 위한 간단한 웹 페이지를 개발하는 것은 매우 빠르고 비주얼합니다. 구조의 모든 핵심 기술을 알지 못해도 완전한 기능과 전문가적인 외양을 갖춘 웹 사이트를 쉽게 작성할 수 있습니다.

써드파티 웹 서버 확장

볼랜드의 InternetExpress 기술을 사용하는 것 이외에, HTML 기반의 서버측 애플리케이션 개발을 다른 접근 방식으로 지원하는 써드파티 컴포넌트와 툴들을 가지고 델파이를 사용할 수 있습니다. 이 범주에 속하는 툴들은 많으므로 여기서는 가장 인기 있는 것들만 언급하기로 합니다.

- HREF(http://www.href.com)는 HTML snippet을 조작하는 고급 기술인 WebHub 프레임워크를 제공하고 데이터베이스 데이터에 기초한 웹 콘텐츠를 생성합니다. WebHub는 사용자 세션을 처리하고, 코드 개발을 서버측 기술과 분리하며, 기존의 윈도우 유저 인터페이스를 웹에 적응시키는 대신에 개발자가 적절한 웹 중심의 접근으로 이동할 수 있도록 해줍니다.
- Nevrona Design은 반대의 접근 방식을 사용하는 ND-IntraWeb (http://www.nevrona.com/intraweb)을 제공합니다. 일련의 사용자 정의 유저 인터페이스 컴포넌트를 사용하여 윈도우 프로그램이나 웹 브라우저에서도 잘 작동할 수 있는 유저 인터페이스를 구축할 수 있도록 합니다.
- Marotz 델파이 Group은 윈도우 DNA 애플리케이션의 개발을 포괄하고 단순화하는 컴포넌트 셋을 제공하는 ASP Express(http://www.asp-express.com)를 제공합니다. 이것은 ASP, MSXML, COM 및 기타 기술을 사용하며, 모든 것을 델파이 코드와 쉽게 결합하도록 합니다.

이상을 요약하면 다음과 같습니다.

- 사용자 정의 서버측 솔루션은 애플리케이션을 델파이로 개발할 수 있도록 즉시 사용 가능한 복잡한 프레임워크를 제공합니다. 이러한 기술의 일부는 대형의 복잡한 웹 사이트 개발에 성공적으로 사용되었습니다.
- 서버측 프로토콜을 감싸는 컴포넌트를 사용하는 것과 비슷하게 사용자의 웹 서버에서 전체 소프트웨어를 완전히 제어할 수 있으며, 다른 사람들이 발생시킨 버그가 시스템에 침투하는 위험이 없습니다.

인터넷에서 활동중인 델파이

인터넷과 웹 개발 분야에서의 델파이의 능력은 이 문서에서 살펴본 것처럼 여러 기술적인 장점으로 평가할 수도 있으나, 그 성공적인 결과를 기반으로 평가할 수도 있습니다. 델파이는 간단한 셰어웨어로부터 거대한 e-커머스에 이르기까지 많은 인터넷 제품의 개발에 사용되었습니다. 언론은 모두 다른 언어와 환경에 대해 언급하고 있지만, 델파이는 인터넷 영역에 침투하여 이제는 널리 보급되어 사용되고 있습니다.

볼랜드는 수많은 성공 스토리를 제공하고 있지만 (http://www.borland.com/delphi/cases/ 참고), 그중 더욱 눈여겨볼 만한 것이 몇 개 있습니다. 적어도 데이터베이스 지향 부분에 있어서 델파이로 구동되는 웹 사이트 가운데 다음과 같은 것들이 있습니다.

- Autobytel.com, 미국의 온라인 자동차 딜러
- The National Trust (http://www.nationaltrust.org.uk), 영국의 역사적 건물을 보존하기 위한 기관
- iVillage.com, 여성 네트워크 웹 사이트
- Dulux Trade Paints (http://www.dulux.com), 페인트 색상을 선택하기 위한 웹 사이트
- CalJobs (http://www.caljobs.ca.gov), 고용주의 취업 목록과 구직자의 이력서를 연결시키는 캘리포니아 주정부의 인터넷 시스템
- Travel.World.NET, 오스트레일리아 World.NET이 운영하는 완전한 여행 서비스 관리 시스템

인터넷 유틸리티 프로그램에 관하여 말하자면, 시중에 나와 있는 인터넷 관련 셰어웨어 프로그램의 거의 절반이 델파이로 구축되었습니다. 여러 분야 가운데에서 몇 가지 중요한 것을 예로 들면 다음과 같습니다.

- 강력한 이메일 관리자인 The Bat! (http://www.ritlabs.com) 및 프리웨어 메일 클라이언트인 Mail Warrior (http://pages.infinit.NET/kaufman)는 많은 소프트웨어 웹 사이트에서 상위를 랭크하고 있습니다.
- 환영받는 HTML 편집기인 HomeSite (http://www.allaire.com/homesite)
- MERAK Mail Server (http://www.icewarp.com)
- 가장 인기 있는 두 가지 IRC 클라이언트인 Pirch(http://www.pirchat.com)와 Virc(http://www.megalith.co.uk/virc/)

이 외에도 뉴스그룹 리더, FTP 프로그램, XML 편집기, 채팅 프로그램, 그리고 거의 모든 인터넷 프로토콜에 대해 클라이언트 및 서버를 위한 애플리케이션들이 존재합니다. 델파이의 개발이 성공한 다른 분야는 고품질 ASP 컴포넌트의 개발입니다. 마이크로소프트의 웹 사이트(http://msdn.microsoft.com/workshop/server/components/catalog.asp)에 언급된 가장 인기 있는 ASP 애드온들 가운데 많은 것들이 델파이로 작성되었으며, 그 중에는 Dimac(http://www.dimac.net)이 작성한 ASP 컴포넌트와 "ASP 개발자를 위한 최고의 SMTP 컴포넌트"로 불리우는 유명한 JMail 컴포넌트가 있습니다. Baltic Solution 웹사이트(http://www.balticsolutions.com/bwd)의 "Built with Delphi"를 보면 (비공식이긴 합니다만) 델파이로 구동되는 인터넷 애플리케이션과 웹 사이트의 목록을 더 많이 찾아볼 수 있습니다.

결론: 델파이의 장점

위와 같은 장문의 설명을 하여도 델파이를 핵심적인 웹 개발툴로 사용해야 하는 이유를 몇 마디로 요약하는 것은 쉽지 않습니다. 분명히 말할 수 있는 것은, 델파이가 윈도우와 웹을 위한 빠른 개발 환경(RAD)에서 구축된 고성능의 애플리케이션을 가져다 준다는 것입니다. 델파이는 최적의 클라이언트/서버 지원과 고품질의 객체 지향적 코드를 작성하는 능력을 가지고 있어 복잡한 애플리케이션 구조를 개발하고 저수준의 프로그래밍 작업을 깊이 파고들 수 있게 해줍니다.

델파이는 어떤 기업에서나 모든 인터넷 요구에 맞는 훌륭한 도구입니다. 또한, 윈도우 및 리눅스 플랫폼에서 동시에 사용할 수 있는 툴로서, "델파이 방식"으로 프로그래밍 하는 것을 좋아하는 사람들을 위한 이상적인 솔루션입니다.