

CodeGear™ Delphi® 2009

리뷰 가이드



목차

목차	2
도입	3
Delphi 2009 일반 개요	3
Delphi 2009 에서 새로워진 점	4
사전 필요 환경	4
최소 시스템 사양	4
에디션	5
프로페셔널.....	5
엔터프라이즈.....	5
아키텍트.....	6
설치	6
통합 개발환경 (IDE).....	6
일반사항	6
코드 작성 - 코드 에디터 (Code Editor)	7
비주얼한 UI 디자인 - 폼디자이너	8
애플리케이션 항목 관리 - 프로젝트 매니저	9
내부 동작 들여다보기 - 디버거 (Debugger).....	9
데이터 액세스 - 데이터 익스플로러	11
비주얼 컴포넌트 라이브러리 (VCL).....	11
Delphi 2009 의 새 기능	12
데이터베이스 디자인과 모델링	13
새로운 Delphi 언어 기능	13
제네릭 (Generics)	13
익명 메소드 (Anonymous Methods).....	13
새 VCL 기능	14
리본 컨트롤.....	14
추가된 새 컴포넌트	15
기존 컴포넌트의 업데이트.....	17
새로운 멀티티어 데이터베이스 아키텍처 - DataSnap 2009.....	17
새로운 IDE 기능	18
리소스 매니저	18
빌드 설정 (Build Configuration)	18
클래스 익스플로러 (Class Explorer).....	19
번역 툴 (Translation Tools).....	20
향상된 COM/ActiveX 지원	21
Delphi 2007 에서 새로워졌었던 기능	22
Blackfish™ SQL.....	22
Blackfish SQL 추가 자료.....	22
비스타 지원.....	22
글래스 효과.....	23
비스타 다이얼로그	23
AJAX 와 VCL for the Web	23
디비 익스프레스 4 (dbExpress 4).....	24
Delphi 2009 의 주요 추가 기능들	24
통합 개발환경 (IDE).....	24
코드 에디터 (Code Editor).....	24
비주얼 디자인 관련 기능들	28
폼디자이너 (Form Designer).....	28

모델링 뷰 (Modeling View).....	30
프로젝트매니저 (Project Manager).....	30
빌드 설정 (Build Configuration)	31
디버거 (Debugger).....	32
통합된 유닛 테스트 (Integrated Unit Testing)	34
리팩토링 (Refactorings).....	35
오브젝트 파스칼 언어 (Object Pascal Language).....	37
비주얼 컴포넌트 라이브러리 (VCL)	37
데이터베이스 액세스 (Database Access)	37
웹 개발	38
VCL for the Web	38
멀티 티어 개발	38
추가 자료	38
Delphi 로 만들어진 애플리케이션	38
Delphi 2009 출시 이벤트	39
코드레이지 (CodeRage) I 온라인 컨퍼런스 발표 내용	39
코드레이지 (CodeRage) II 발표 내용	39

도입

Delphi 2009 리뷰를 위해 시간을 할애해 주셔서 감사합니다. 이 가이드는 검토 시 제품의 특징점 및 기능 등 충분한 정보를 드리기 위해 작성 되었습니다.

이 가이드는 세 부분으로 나뉘어져 있습니다. 1 부에서는 Delphi 2009 를 개괄적인 오버뷰로 기본적인 목적과 기능 설명하기 위해 Delphi 2009 기능을 높은 수준에서 단순화하여 살펴 봅니다. 2 부에서는 Delphi 2009 의 새로운 주요 기능을 요약하여 “최신의 대단한” 기능으로 어떤 것이 Delphi 2009 에 있는지를 논합니다. 예를 들면 업그레이드를 할 만한 주요 기능들입니다. 3 부에서 광범위한 기능들 중 대표적인 것에 대해 더욱 자세하게 설명하였습니다.

Delphi 2009 일반 개요

코드기어 Delphi 2009 는 신속한 애플리케이션 개발 즉 RAD 방식으로 윈도우 애플리케이션을 개발하기 위한 범용적인 목적의 개발툴입니다. Delphi 2009 는 x86 OS 에서 작동하는 Win32 바이너리 코드를 만들어 냅니다.

Delphi 2009 를 이용하면 개발자들은 거의 모든 타입의 윈도우 바이너리를 개발해낼 수 있습니다. 독립 실행 파일 (EXE), 다이내믹 링크 라이브러리 (DLL), OCX, COM 오브젝트, 타입 라이브러리, 제어판 애플릿 윈도우 서비스 애플리케이션, 콘솔 애플리케이션 등 풍부하고 복잡한 UI 를 가진 클라이언트 애플리케이션에서부터 간단한 커맨드라인 애플리케이션까지 다양하게 구축할 수 있습니다. 또한 관계형 데이터베이스 시스템 (RDBMS)와 직접 통신하는 클라이언트 애플리케이션, 미들웨어 애플리케이션 서버, 웹애플리케이션, 웹사이트, 웹서비스, 윈도우 애플리케이션 ,ActiveX 컨트롤, 멀티 쓰레드 애플리케이션을 작성할 수 있고 복잡한 임베디스 시스템에서 작동시킬 수도 있습니다. 간단히 말해서, Delphi 2009 는 윈도우 애플리케이션을 개발하는 개발자들이 필요로 하는 모든 요구사항들을 충족시킵니다. 코드기어 Delphi 로 개발함에 있어 어떤 한계를 찾아내기는 그리 쉽지 않습니다.

Delphi 2009 에서 새로워진 점

Delphi 2009 의 주요 새로운 기능:

새로운 비주얼 컴포넌트 라이브러리 (VCL) 컴포넌트들이 추가 되었습니다. 여기에는 일반적으로 리본 컨트롤이라고 불리는 완전한 오피스 2007 UI 구현도 포함됩니다. 이제 개발자들은 시각적으로 매력적인 세련된 UI 를 가진 윈도우 데스크톱 애플리케이션을 개발할 수 있습니다. 또한 비스타의 새로운 API 들을 지원하기 위해 많은 기존의 VCL 컴포넌트가 강화되었습니다.

두가지 새로운 언어 기능, 제네릭과 익명 메소드가 도입되어 더욱 풍부하고 쉽게 코드에 표현할 수 있도록 해줍니다. 새로운 RTL 클래스들인 리스트, 컬렉션, 딕셔너리, 스택, 큐에서도 이 새로운 언어 기능들을 활용합니다. 또한 Delphi 2009 에서는 멀티 쓰레딩 클래스의 실행 과정에 익명 메소드를 넘길 수 있도록 지원이 추가되었습니다. 멀티 티어 데이터베이스 애플리케이션 개발 프레임워크에 새로운 아키텍처가 적용되었습니다. 비즈니스 룰과 데이터 액세스를 미들 티어 애플리케이션 서버로 격리시키고, 강력하면서도 가벼운 클라이언트 데스크톱 또는 브라우저 애플리케이션을 개발하여 애플리케이션 서버에 액세스할 수 있습니다.

Delphi 2009 에는 세계화에 대한 준비가 되어있습니다. 유니코드 기능이 포함된 스트링 타입이 기본 스트링이 되어, 개발자들이 완전한 유니코드 기반 애플리케이션을 개발할 수 있습니다. 개발자들은 기존의 혹은 새 애플리케이션을 쉽게 지역화 할 수 있습니다. 이제 세계 어느 지역의 데이터도 수용할 수 있으며 개발한 애플리케이션을 글로벌하게 보급할 수 있습니다. 통합 개발환경 (IDE)에 새로운 생산성 향상 기능을 추가하여 Delphi 의 훌륭한 IDE 를 더욱 새로운 수준으로 끌어 올렸습니다. 개발자들은 빌드 설정, 프로젝트 옵션 셋, 윈도우 리소스를 손쉽게 관리할 수 있습니다.

위 새로운 기능은 아래에서 더 자세히 살펴볼 것입니다.

사전 필요 환경

Delphi 2009 설치를 위해 필요한 사전 필요 환경들은 다음과 같습니다:

- Microsoft .NET Framework 2.0
- Microsoft Direct Access Objects 2.8
- Microsoft Internet Explorer v6.0 SP1
- Microsoft XML Core Services (MSXML) v4.0 SP2
- Microsoft Visual J# .NET v2.0 Redistributable

여러분의 컴퓨터에 위 항목들이 설치되어 있지 않을 경우 Delphi 인스톨러가 자동으로 설치해줄 것입니다.

.NET 프레임워크는 IDE 의 실행을 위해서만 필요하며, Delphi 로 빌드한 산출물인 네이티브 애플리케이션은 .NET 프레임워크에 전혀 의존하지 않습니다.

최소 시스템 사양

Delphi 2009 구동을 위해 다음과 같은 시스템 사양이 요구됩니다:

- 최소한 Intel Pentium or compatible, 1.4Ghz (권장: 2Ghz+ 이상)
- 1GB RAM (2GB+ 권장)
- 3GB 하드 디스크 공간 (Delphi 용)
- 750MB 하드 디스크 공간 (사전 필요 환경 설치용)
- DVD-ROM 드라이브
- 1024x768 이상 해상도 모니터
- 마우스 혹은 기타 포인팅 장치

Delphi 2009 설치를 지원하는 윈도우 플랫폼은 다음과 같습니다:

- Microsoft Windows 2000 (SP4 이상, 보안 업데이트 권장)
- Microsoft Windows XP Home 혹은 프로페셔널 (SP3 이상)
- Microsoft Windows Vista SP1 (관리자 권한 필요)
- Microsoft Windows Server 2003 (SP1)
- Microsoft Windows Server 2008

Delphi 2009 설치 전에 다음과 같은 항목들의 설치가 필요합니다:

- XP 와 Server 2003 등 모든 플랫폼에서 최신 서비스 팩과 보안 업데이트 권장.
- Microsoft 인터넷 익스플로러 6.0 SP1 이상
- 완전한 설치를 위해서는 약 3 GB 의 디스크 공간이 필요. 인스톨러에서는 약간의 추가 임시 공간도 필요로 합니다.
- 최저 Pentium 혹은 호환 CPU 1.4GHz (권장: 2GHz 이상)
- 1 GB RAM (2GB+ 권장)
- DVD 드라이브
- VGA 이상 해상도 모니터
- 마우스 등 기타 포인팅 장치

주의: Delphi 2009 영문판은 인터넷 익스플로러가 영문판 6.0 이하인 경우 6.0 SP1 으로 업데이트 합니다. 한국어 등 각 국가별 언어의 OS 를 사용하고 있는 경우 윈도우 업데이트를 실행하여 적절한 인터넷 익스플로러의 지역판을 설치하십시오.

에디션

Delphi 2009 에는 서로 다른 요구를 맞추기 위해 프로페셔널, 엔터프라이즈, 아키텍트의 세가지 에디션이 있습니다.

프로페셔널

Delphi 2009 프로페셔널 에디션은 제한적인 로컬 데이터베이스 연결만 필요한 개발자들을 위한 범용 RAD 개발툴입니다. 프로페셔널 에디션에는 IDE 의 모든 기능, Delphi 언어, VCL 소스 코드를 포함한 VCL 이 들어 있습니다. 하지만 오직 InterBase, MySQL, Blackfish SQL 의 로컬 데이터베이스 액세스만 가능합니다.

프로페셔널 에디션의 VCL for the Web은 제한된 버전으로서, 5 개까지의 커백션만 가능하며 스탠드얼론 서버만을 개발할 수 있습니다.

프로페셔널 에디션은 독립 소프트웨어 벤더 (ISV)나 원격 데이터베이스 액세스가 필요 없는 프리랜서, 그리고 일반적인 윈도우 애플리케이션이나 유틸리티를 만드는 개발자들에게 적당할 수 있습니다.

엔터프라이즈

Delphi 2009 엔터프라이즈 에디션은 RDBMS 에 저장된 엔터프라이즈 급 데이터에 대한 액세스가 필요한 개발자를 위해 설계되었습니다. 엔터프라이즈 에디션은 다음의 9 가지 데이터베이스 엔진에 로컬 또는 원격 네이티브 연결을 지원합니다:

- InterBase 7.5.1, 2007, 2009
- Blackfish SQL for .NET/Java
- Oracle 10g 와 11g
- Microsoft SQL Server 2000, 2005, 2008
- DB2 UDB 8.X
- MySQL 4.0.x, 5.0.x
- Informix 9x

- Sybase Adaptive Server 엔터프라이즈 12.5
- Sybase SQL Anywhere 9

또한, 엔터프라이즈 에디션은 VCL for the Web의 풀 버전을 포함하고 있으며, 연결 개수의 제한이 없고 스탠드얼론, IASPI, Apache 기반 웹 애플리케이션의 개발을 지원합니다.

엔터프라이즈 에디션은 [UML \(Unified Modeling Language\)](#)을 통해 고급 모델링을 지원합니다. 클래스 다이어그램의 리버스 엔지니어링 기능은 양방향 코드 에디팅으로 더욱 강화되었으며, 전체 프로젝트를 하イレ벨에서 오버뷰할 수 있습니다. 자동 문서 생성 기능으로 프로젝트를 쉽게 살펴볼 수 있게 하는 동시에 핵심 개발팀을 넘나드는 커뮤니케이션이 가능합니다.

엔터프라이즈 에디션은 기업 데이터 액세스가 필요한 기업체의 개발자 또는 다양한 이기종 데이터베이스가 필요한 애플리케이션을 생성하는 ISV에 적합합니다. 컨설턴스 및 엔터프라이즈 급 개발자를 지원하는 VAR도 해당됩니다.

아키텍트

Delphi 2009 아키텍트 에디션은 엔터프라이즈 버전의 모든 것을 포함하고 있으며 데이터 모델링과 설계를 위한 엠바카데로 ER/Studio 디벨로퍼 에디션이 추가되어 있습니다. 엠바카데로 ER/Studio은 업계 선두의 데이터 모델링 툴로서, 기업의 데이터 자산을 탐색, 문서화, 재활용 하는 것을 쉽게 해줍니다. 데이터베이스 전반을 아우르는 기능들을 지원함으로써 데이터 아키텍트는 기존 데이터베이스를 쉽게 리버스 엔지니어링하고 분석, 최적화 할 수 있습니다. ER/Studio의 강력한 협업 기능은 조직의 표준화를 강화하여 생산성을 향상시킵니다.

아키텍트 에디션은 엔터프라이즈 에디션에 관심이 있으면서도 생산성 높은 데이터베이스 설계와 관리를 원하는 개발자에게 적합합니다.

설치

Delphi 2009 는 DVD 또는 웹 다운로드를 이용하여 설치할 수 있습니다. DVD를 컴퓨터에 넣고 인스톨 애플리케이션을 구동하는 방식이 일반적입니다.

인터넷으로 다운로드된 경우(ESD; Electronic Software Delivery), 고객은 작은 런처 프로그램을 받게 되며, 이것을 실행하면 Delphi 2009 설치 과정에 필요한 모든 파일들을 웹에서 다운로드하여 Delphi 2009를 설치합니다. 등록 사용자들은 <http://www.codegear.com>에서 추가 항목들을 다운로드할 수 있습니다.

통합 개발환경 (IDE)

Delphi 2009를 작동시키면, 개발자들은 통합 개발환경 즉 IDE를 사용하게 됩니다. IDE라는 하나의 애플리케이션 안에는 개발자가 애플리케이션을 개발하는 데 필요한 모든 기능이 함께 들어 있습니다. 에디터, 폼 디자이너, 프로젝트 매니저, 디버거 등 모든 기능을 단일 환경에서 서로 넘나들며 애플리케이션을 쉽고 빠르게 개발할 수 있습니다..

일반사항

IDE는 윈도우 사용자에게 친숙한 UI를 제공합니다. 드롭다운 메뉴, 툴 버튼을 포함한 설정 가능 툴바를 통해 기본 기능을 제공합니다. IDE 안에 있는 많은 다양한 윈도우들은 도킹이 가능하며, 개발자가 자신의 작업 환경을 완전히 커스터마이징 할 수 있습니다. 데스크톱 레이아웃을 저장할 수도 있습니다. 데스크톱은 디버깅과 같은 특정 용도에 맞추어 지정할 수 있습니다. 애플리케이션은 IDE 안에서 실행 및 디버깅될 수 있습니다. 개발자는 거의 모든 관점에서 IDE의 거의 모든 옵션을 각각의 애플리케이션과 프로젝트 별로 설정할 수 있습니다. IDE 전체가 커스터마이징 될 수 있게 설계되었으므로 보다 효율적이고 빠른 개발이

가능하게 됩니다.

코드 작성 - 코드 에디터 (Code Editor)

더 이상의 텍스트 에디터는 없습니다. Delphi 2009 의 코드 에디터는 코드 생성과 타이핑을 광범위하게 지원하기 위한 수많은 기능을 통해 개발자의 생산성을 현격하게 향상시킵니다. 이 에디터는 탭 형식 윈도우로 되어 있으므로 한꺼번에 여러가지 파일을 열고 편집할 수 있습니다. 코드 에디터에는 다음과 같은 개발자가 기대해왔던 모든 생산성 향상 기능을 포함하고 있습니다:

- 코드의 문법 강조(Syntax highlighting)를 통해 각 코드 부분들이 무엇인지 분명하게 해줍니다 – 주석, 스트링, 식별자, 키워드, 예약어는 각각 쉽게 알아볼 수 있는 색상으로 나타납니다.
- 코드 완성(Code Completion)은 개발자가 코딩을 하는 과정에서 적절한 식별자 이름의 힌트를 제공합니다.
- 라이브 템플릿(Live Templates)을 통해 개발자는 몇번의 키 타이핑만으로도 큰 코드 단위를 불러내어 해당 코드의 적절한 부분을 간단히 채워 넣을 수 있습니다. 라이브 템플릿 스크립트는 완전히 편집 가능하므로 개발자들은 코드 에디터에 상상할 수 있는 모든 것을 넣을 수 있습니다.
- 클래스 완성 (Class completion)은 클래스 선언의 구현 스텝을 자동으로 생성합니다.



그림 1 - 코드 에디터

- 리팩토링을 통해 코드를 어려없이 더욱 읽기 쉽고 깔끔하게 재작성할 수 있습니다.
- 코드 인사이트 (CodeInsight)는 팝업 윈도우를 통해 해당 루틴에 필요한 파라미터 가이드를 제공합니다.
- 에러 인사이트 (ErrorInsight)는 코드의 실행시 에러를 즉각 밑줄로 알려줍니다.
- 헬프인사이트 (HelpInsight) 는 에디터의 팝업 윈도우를 통해 코드 안의 모든 식별자에 대해 기본적인 문서화와 선언 정보를 제공합니다.
- 블록 완성 (Block Completion) 은 모든 코드 블록들이 적절하게 열리고 닫히도록 해줍니다. 예를 들어 개발자가 begin 을 타이핑하고 엔터 키를 치면, 대응되는 end 가 자동으로 추가되어 코드가 적절하게 구성되도록 하기 때문에 개발자의 집중이 흐트러지지 않습니다.
- “문맥 지능적인” 코드로 인해 코드 전반에 걸친 네비게이션이 보다 쉬워집니다. 컨트롤 키를 누른 상태에서 식별자를 클릭하면 해당 식별자의 선언 또는 구현부로 갈

수 있습니다. 스택기반 모델을 사용하므로 개발자는 코드 작업 순서 앞 뒤로 이동할 수 있습니다. 간단한 키 조작으로 개발자는 클래스 메소드의 선언부와 구현부 사이를 이동할 수 있습니다. 인텔리마우스 스크롤을 지원하므로 쉽게 큰 파일을 네비게이션할 수 있습니다.

- 라인 번호 표시로 위치 정보를 바로바로 제공합니다. 북마크 설정이 가능하여 개발자는 쉽게 코드상의 특정 위치로 돌아올 수 있습니다.
- 다수의 라인을 한꺼번에 선택했을 때, 에디터의 마진 영역에서 싱크 에디트 (SyncEdit) 아이콘을 사용할 수 있습니다. 싱크 에디트 (SyncEdit) 모드에 들어가게 되면 텍스트에서 하이라이트된 블록 전체에 쉽고 빠르게 검색 및 변경이 가능합니다. 싱크 하이라이트된 코드 블록에서 리네임 리팩토링 (Rename Refactoring) 기능을 사용하면 전체 프로젝트에 대해서 문맥 지능적으로 찾아 바꾸기를 하기 때문에 동일 항목에 해당하는 식별자들만이 변경됩니다.
- 반복적인 타이핑 작업을 실행하기 위한 키 매크로를 녹화 및 재생할 수 있습니다.

요약하자면, 코드 에디터는 더 쉽고 효율적이고 효과적인 코드 타이핑이 가능하도록 설계되었습니다.

비주얼한 UI 디자인 - 폼디자이너

코딩 작업을 제외하면, 대부분의 개발 작업 시간은 컴포넌트들로 폼을 레이아웃하여 UI를 구성하는 데에 소요됩니다. IDE의 폼 디자이너는 이런 작업들을 위한 강력한 폼 디자이너를 제공합니다. Delphi 2009의 폼디자이너는 정확하게 윈도우와 똑같이 보이므로, 위지윅 (WYSIWYG) 방식의 폼 레이아웃이 가능합니다. 개발자는 툴 팔레트에서 컴포넌트를 드래그&드롭 방식으로 원하는 대로 폼에 끌어다 놓을 수 있으며 툴 팔레트 자체는 완전히 사용자 설정이 가능합니다. 컴포넌트는 비주얼 가이드라인에 따라 쉽게 간격 조정과 위치 정렬이 됩니다. 비주얼 가이드라인은 컴포넌트가 폼에 드래그 되는 동안 이미 위치해 있는 다른 컴포넌트들과의 정렬을 위해 컴포넌트 간의 공간 등을 표시해주기 위해 나타나는 컬러 라인입니다.

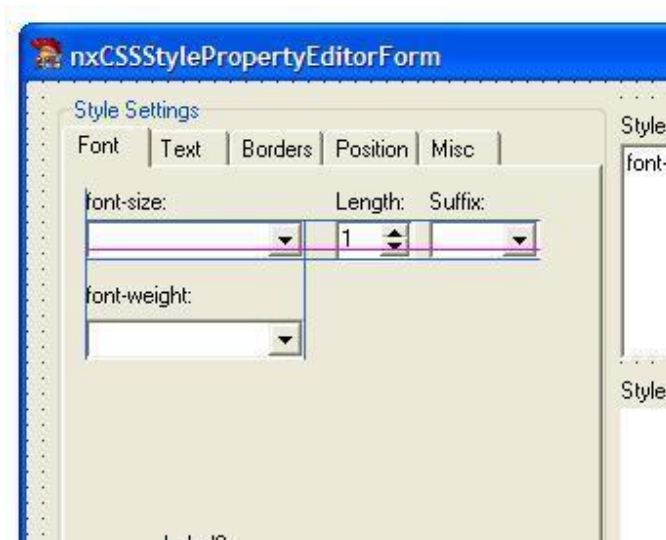


그림 2- 비주얼 가이드라인을 활용하여 쉽게 폼에 컨트롤을 정렬

일단 컴포넌트가 폼에 위치하면, 오브젝트 인스펙터 (Object Inspector)에서 속성들을 설정할 수 있습니다. 오브젝트 인스펙터는 컴포넌트의 모든 속성과 이벤트를 나열하여 개발자가 쉽고 빠르게 컴포넌트의 외형을 조작하고 컴포넌트에서 발생할 수 있는 다양한 이벤트를 볼일 수 있게 해줍니다.

예를 들어, 개발자는 오브젝트 인스펙터 (Object Inspector) 를 이용하여 컴포넌트의 위치와 크기를 조절할 수 있습니다. 또한 툴바에 버튼을 추가할 수 있고 에디트 박스의 텍스트를 바꾸거나 폼의 배경색을 변경할 수 있습니다. 게다가 `OnClick`, `OnMouseOver`, `OnKeyDown`, 등 특정 이벤트가 발생했을 때 코드가 작동하도록 할 수 있습니다.

툴 팔레트는 IDE 에 설치된 모든 VCL 컴포넌트들을 가지고 있는 곳입니다. Delphi 2009 와 함께 제공되는 모든 표준 컴포넌트가 기본적으로 들어있으며, 필요한 써드 파티 컴포넌트를 추가할 수도 있습니다.

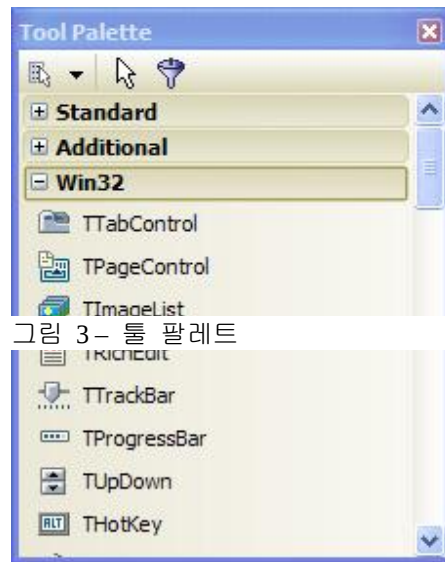


그림 3- 툴 팔레트

툴 팔레트는 완전히 재설정할 수 있습니다. 컴포넌트는 원하는 대로 그룹핑될 수 있고, 필터 검색을 통해 쉽게 찾을 수 있습니다. 레이아웃과 컬러 설정은 사용자 정의가 가능합니다. 컴포넌트를 선택하고 폼에 떨어뜨리는 것은 마우스 혹은 키보드로 가능합니다.

애플리케이션 항목 관리 – 프로젝트 매니저

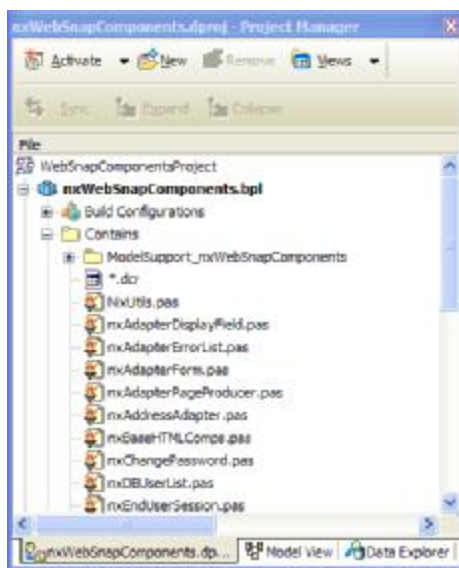


그림 4 -- 프로젝트매니저

애플리케이션은 수많은 폼과 코드 파일로 인해 금방 복잡해집니다. 대부분의 애플리케이션이 수많은 바이너리 및 프로젝트로 구성되어 있습니다. Delphi 2009 의 프로젝트 매니저는 해당 프로젝트를 IDE 안에서 바로 관리할 수 있게 해줍니다. 프로젝트 매니저는 파일과 폼을 프로젝트 안에 정돈하고, 프로젝트는 프로젝트 그룹 안에 정돈합니다. 프로젝트는 프로젝트 매니저에서 바로 컴파일 및 빌드 될 수 있습니다. 파일과 폼은 IDE 안에서 바로 열 수 있습니다. 프로젝트는 재정렬 될 수 있으므로 애플리케이션의 요구에 따라 명시된 순서로 컴파일 될 수 있습니다. 개발자가 파일과 폼을 관리하는 데 필요한 모든 것이 프로젝트 매니저에서 관리됩니다.

또한 프로젝트 매니저는 하나의 프로젝트에 다수의 빌드 설정을 설정 관리할 수 있습니다. 또한 이러한 설정은 보다 쉽게 할 수 있도록 특정 옵션 셋을 재활용 할 수 있도록 관리되고 저장됩니다.

내부 동작 들여다보기 – 디버거 (Debugger)

모든 개발은 문제에 봉착하게 마련입니다. 버그나 예상치 못한 애플리케이션 오작동이 발생할 경우, 개발자는 애플리케이션 내부 작동을 살펴볼 수 있어야 합니다. 이를 통해서 문제가 발생할 때 내부에서 무슨 일이 벌어지고 있는지를 들여다 보아야 합니다. Delphi 2009 는 IDE 에 통합되어 있는 디버거를 통해 애플리케이션이

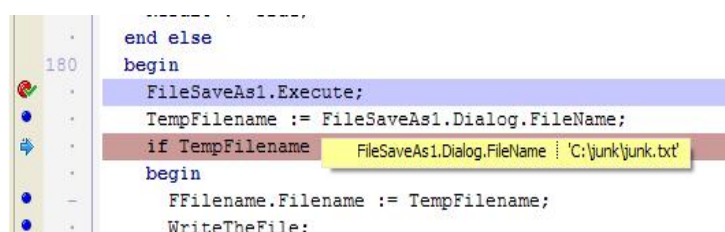


그림 5 -- 브레이크포인트에서 디버거가 정지합니다

동작할 때 그 내부를 깊이 접근하여 살펴볼 수 있도록 해줍니다.

애플리케이션이 IDE 안에서 작동될 때, 디버거는 이를 전달받아서 개발자가 실행되는 프로세스를 통제하고 전체 프로세스와 관련된 모든 정보에 접근할 수 있도록 해 줍니다. 개발자는 코드 안 어느 곳에든 브레이크포인트를 설정하여 실행을 정지시킬 수 있습니다. 브레이크포인트는 설정 가능하고 매번 또는 특정 횟수 이후에 작동하거나, 사용자 코드 안에서 특정 조건의 변수와 함수 상황에 의해 작동되도록 설정할 수 있습니다. 암시적인 브레이크포인트는 예외 발생시 항상 작동될 수 있습니다. 일단 실행이 중단되면, 디버거는 해당 범위 내의 모든 정보에 접근할 수 있도록 합니다. 개발자는 특정 변수의 값을 추적하기 위해 와치(Watch)를 설정할 수 있습니다. 코드를 한 단계씩 들어가면서 라인 단위로 실행 시키고 언제든지 변수의 값을 살펴볼 수 있습니다. 디버거는 현재의 호출 스택, 로딩된 모듈, 애플리케이션에 연관된 모든 쓰레드의 상태를 보여줍니다. 만약 더 자세한 사항이 필요할 경우 디버거의 CPU 레벨의 뷰를 표시하면 실행되는 정확한 어셈블리 코드를 볼 수 있습니다.

데이터 액세스 – 데이터 익스플로러

많은 애플리케이션이 데이터 액세스를 필요로 합니다. IDE의 패널 중 하나인 데이터

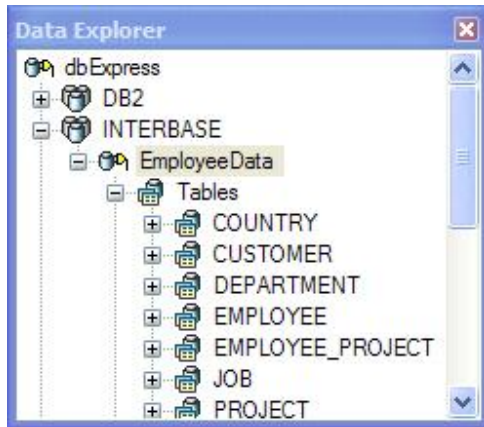


그림 6 – 데이터 익스플로러

익스플로러는 데이터베이스 데이터에 빠르고 강력하게 액세스할 수 있게 해줍니다. 데이터 익스플로러는 Delphi의 데이터베이스 액세스 기술인 디비 익스프레스 (dbExpress)에서 지원하는 모든 RDBMS에 대해 커백션을 생성할 수 있습니다. 일단 커백션이 생성되면 데이터 익스플로러를 통해 해당 데이터베이스의 데이터와 메타 데이터를 살펴볼 수 있게 됩니다. 커백션과 데이터를 사용할 수 있게 되고 나면 데이터 익스플로러에서 폼 디자이너로 커백션과 테이블들을 드래그하여 올려 놓는 방식으로 데이터 액세스 컴포넌트를 생성할 수 있습니다.

또한 데이터익스플로러의 비주얼 쿼리 빌더 (Visual Query Builder)를 사용하면 데이터를 살펴보거나 쿼리 작성을 할 수 있습니다.

비주얼 컴포넌트 라이브러리 (VCL)

Delphi 2009는 거의 모든 유형의 애플리케이션을 개발할 수 있지만, 그 중에서도 윈도우 클라이언트 또는 독립 데스크톱 애플리케이션을 개발할 때 특히 강력합니다. 바로 Delphi 2009는 애플리케이션 개발 프레임워크인 비주얼 컴포넌트 라이브러리 (VCL)을 제공하기 때문입니다. VCL은 윈도우 애플리케이션 구축을 캡슐화 하기 위해 Win32 API의 광범위한 영역을 모두 포괄하는 클래스 라이브러리입니다. 이것은 컴포넌트 기반, 즉 개발자들이 폼 디자이너 상에서 컴포넌트를 구축하여 활용할 수 있도록 설계되었습니다.

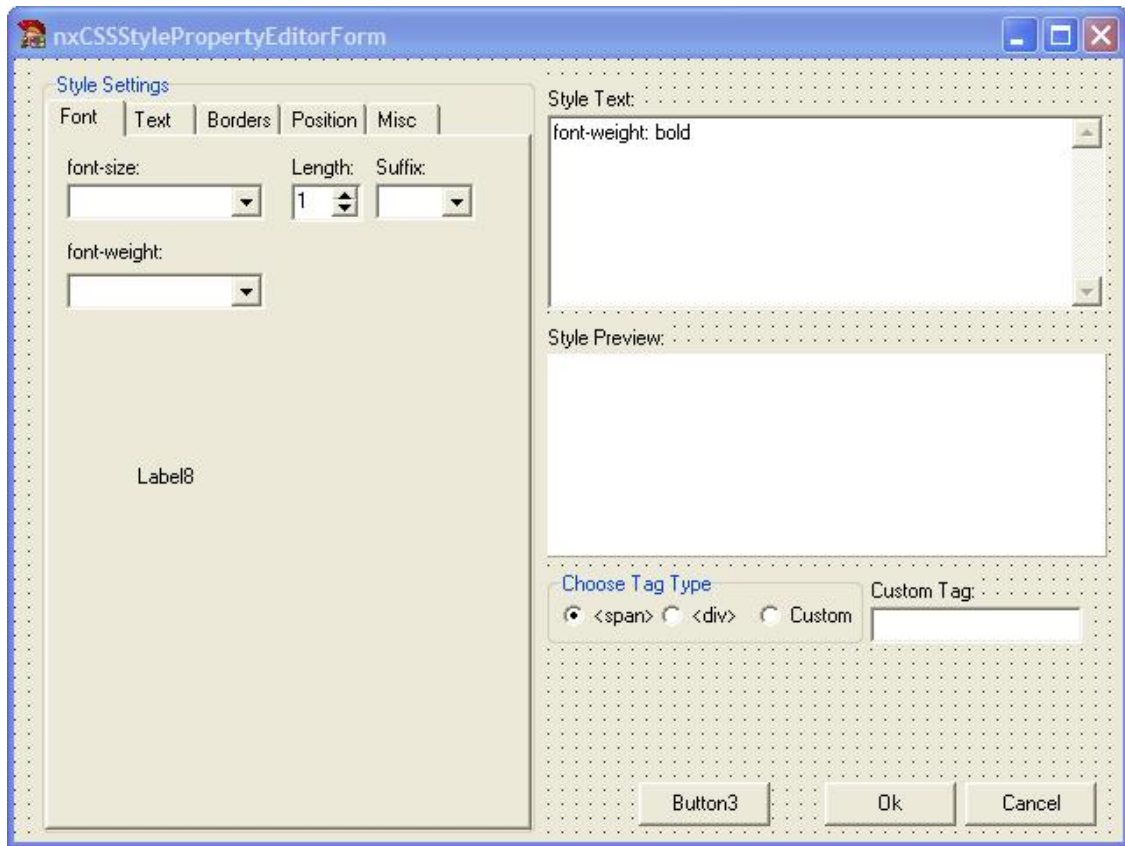


그림 7- 폼 위에 여러가지 VCL 컨트롤들이 올라가 있습니다.

VCL 은 Delphi 2009 에서 윈도우 애플리케이션 개발하는 기반이 됩니다. VCL 은 MS 윈도우 기반의 한 윈도우를 TForm 로 캡슐화 하여 제공합니다. (오래된 명명 규칙, 즉 네이밍 컨벤션에 따라 오브젝트 파스칼의 클래스는 'T'로 시작하게 됩니다). IDE 의 폼 디자인어는 "캔버스 (canvas)"를 제공하는데 개발자는 그 위예다가 UI 를 디자인 하게 됩니다. VCL 은 모든 윈도우 표준 UI 컨트롤을 래핑하여 TButton, TEdit, TLabel, TCheckbox 와 같은 컴포넌트로 제공합니다. VCL 프레임워크는 간단한 상속을 통해 확장될 수 있으므로 사용자들은 자신들이 개발한 컴포넌트를 사용하여 IDE 를 손쉽게 보강하여 사용할 수 있습니다. (그 결과 상업용, 프리웨어, 오픈 소스의 풍부하고 광범위한 커뮤니티에서 써드파티 개발자들이 VCL 개발자들을 위해 더 많은 기능의 컴포넌트를 광범위하게 제공하고 있습니다)

VCL 이 놀라울 만큼 견고하다는 것은 지금까지 수년간에 걸쳐 많은 다양한 플랫폼에서 VCL 을 채택되어 사용되면서 이미 입증된 바입니다. VCL 은 16-bit 윈도우에서 시작하였지만 윈도우 95 을 지원하기 위해 곧바로 32-bit 로 넘어갔으며, 과거에 이미 리눅스와 .NET 지원이 적용되기도 했었습니다.

Delphi 2009 의 새 기능

리뷰 중에 가장 관심있는 대상은 이번 업그레이드에서 새롭게 등장한 기능들일 것입니다. 이 장에서 코드기어 RAD 스튜디오 2007 (CodeGear RAD Studio 2007) 이후 Delphi 2009 에서 새롭게 추가된 기능을 살펴 봅니다.

데이터베이스 디자인과 모델링

Delphi 2009 아키텍트 에디션은 데이터베이스 디자인과 모델링을 위한 모든 솔루션을 포함하고 있습니다. 개발자들은 아키텍트 에디션에 포함되어 있는 ER/Studio 디벨로퍼 에디션을 사용하여 엔티티와 릴레이션 또는 데이터베이스의 물리적 모델을 생성할 수 있습니다. 이렇게 만들어진 모델은 .ER/Studio 디벨로퍼 에디션이 지원하는 다양한 데이터베이스로 익스포트 될 수 있습니다..

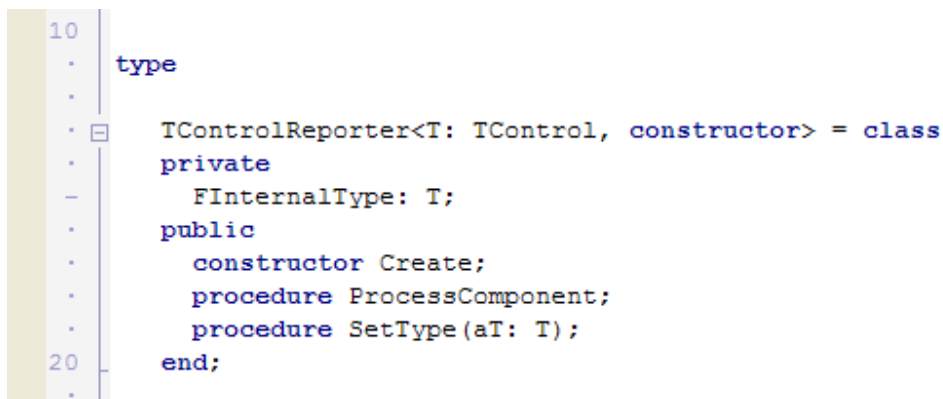
ER/Studio 디벨로퍼 에디션에 대한 온라인 데모는 아래의 코드기어 디벨로퍼 네트워크(CDN)에 있습니다:

<http://windemo1.codegear.com/Tiburon/LaunchReplays/ERStudio/ERStudio.html>

새로운 Delphi 언어 기능

제네릭 (Generics)

제네릭을 이용하면 명시적 타입을 일일이 명시하지 않고도 타입을 사용하는 코드를 작성할 수 있습니다. 제네릭은 흔히 파라미터 타입이라고 불리는 일반적인, 즉 제네릭(generic)한 클래스를 작성하며 구체적으로 명시하지 않은 타입에서 작동합니다. 클래스에서 제네릭을 이용하는 한 예는 리스트입니다. 리스트 클래스의 코드를 작성하는 시점에서 리스트에 포함될 아이템들의 타입을 지정할 필요가 없습니다.



```
10  . type
    .
    .
    . TControlReporter<T: TControl, constructor> = class
    . private
    .     FInternalType: T;
    . public
    .     constructor Create;
    .     procedure ProcessComponent;
    .     procedure SetType(aT: T);
    . end;
20  .
```

그림 8 - 제약조건(Constraint)을 가진 제네릭 클래스

Delphi 2009의 제네릭은 제네릭 메소드와 제약조건 (constraint) 모두를 지원합니다. 제약조건 (constraint)은 제네릭 타입의 한계를 특정 셋으로만 지정하는 데 사용됩니다. 예를 들어 제약조건을 통해 제네릭 타입을 오직 생성자를 가진 것들 또는 특정 인터페이스에 대한 구현 등으로만 한정할 수 있습니다.

또한 Delphi 2009에서는 제네릭 리스트, 컬렉션, 스택, 큐 등을 위한 새로운 런타임 라이브러리 지원을 제공합니다.

익명 메소드 (Anonymous Methods)

익명 메소드 (Anonymous Methods)는 개발자들이 파라미터로 코드 블록을 전달할 수 있게 하는 코드 구조입니다. 이것은 이름이 지정되지 않은 프로시저나 함수를 말합니다. 익명 메소드는 코드 블록을 변수에 할당하거나 다른 메소드의 파라미터로 지정할 수 있는 것으로 취급합니다. 또한 익명 메소드는 해당 메소드가 정의된 컨텍스트에서 변수를 참조하고 값을

```

10 type
    // method reference
    TProc = reference to procedure(x: Integer);

- procedure Call(const proc: TProc);
begin
    proc(42);
end;

20 - procedure DoSimpleAnonymousMethod;
var
    proc: TProc;
begin
    // anonymous method
    proc := procedure(a: Integer)
begin
    Writeln(a);
end;

30     Call(proc);
end;

```

새 VCL 기능

Delphi 2009 는 리본 컨트롤이라고 불리는 Office 2007 UI 컨트롤을 완벽하게 구현하고 있습니다. 이는 기존의 VCL 아키텍처를 사용하여 순수하게 오브젝트 파스칼로만 구축되어 있으므로 개발자들은 세련되고 강력한, 그리고 사용이 편리한 GUI 애플리케이션을 별도의 코딩 없이 리본 컨트롤을 사용하여 구축할 수 있습니다. 또한 Delphi 의 강력한 TActionManager 의 기반 위에 만들어졌으므로 기존의 애플리케이션을 이 새로운 UI 패러다임을 사용할 수 있도록 마이그레이션 하는 것이 매우 쉽습니다.

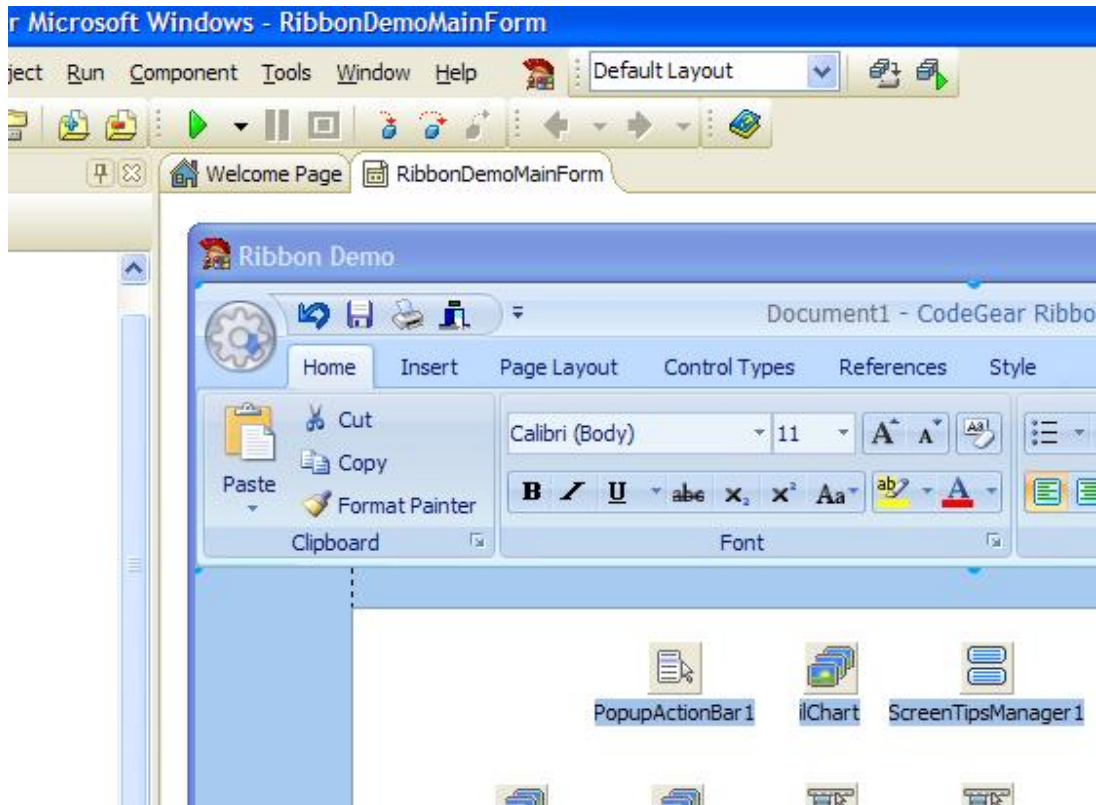


그림 10 -리본 컨트롤을 사용하는 Delphi 폼

추가된 새 컴포넌트

TPNGImage

Delphi 2009 는 이제 PNG (Portable Network Graphics) 이미지를 완벽하게 지원합니다. PNG 이미지는 TImage 와 TImageList 컴포넌트에서 사용될 수 있습니다. 개발자들은 PNG 이미지를 폼에 표시하거나 툴바 또는 메뉴, 버튼 등에서 사용할 수 있습니다.

TCategoryPanelGroup

TCategoryPanelGroup 컴포넌트는 하나의 단독 컴포넌트 안에서 접었다 폈다를 할 수 있는 패널들의 컬렉션입니다. “아웃룩의 툴바”와 유사한 방식 입니다. TCategoryPanelGroup 에는 접히는 패널을 얼마든지 추가할 수 있으며, 각 패널에는 다른 컴포넌트들을 개수와 타입에 제한 없이 포함할 수 있습니다.

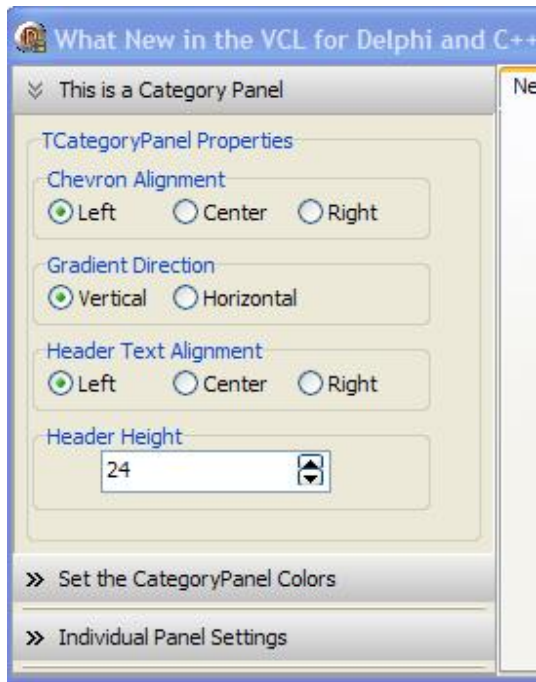


그림 11 –TCategoryPanelGroup 에 3 개의 CategoryPanel 가 있으며 2 개는 접혀 있습니다.

TBalloonHint

Delphi 2009 는 이제 유연하고 설정 가능한 컨트롤 힌트 시스템을 지원합니다. TCustomHint 클래스를 상속하면 새로운 힌트 모양들을 보다 쉽게 생성할 수 있습니다. Delphi 2009 는 기본적으로 TBalloonHint를 제공합니다. 모든 VCL 컴포넌트는 이제 CustomHint 라는 속성을 가지며, TCustomHint 를 상속받은 컴포넌트를 가리킬 수 있습니다.

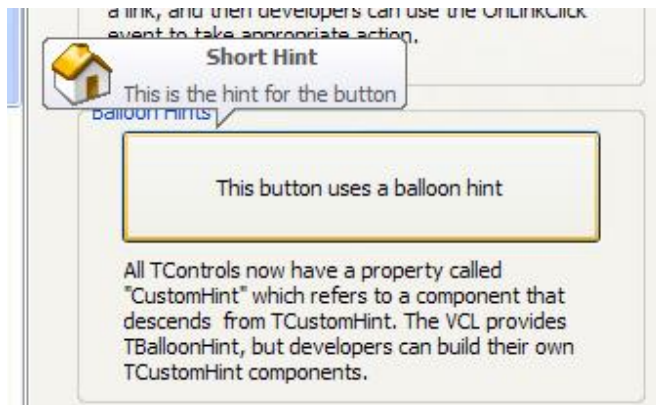


그림 12 – BalloonHint 가 표시된 VCL 컨트롤

BalloonHint는 제목과 메인 텍스트를 지정하여 표시할 수 있으며, 이미지도 표시할 수 있습니다.

TButtonEdit

TButtonEdit는 에디트 컨트롤을 확장한 것으로서, 컨트롤 안쪽 양면에 이미지를 넣을 수 있습니다. 이 컨트롤의 이미지를 클릭했을 때 발생하는 이벤트도 제공됩니다.

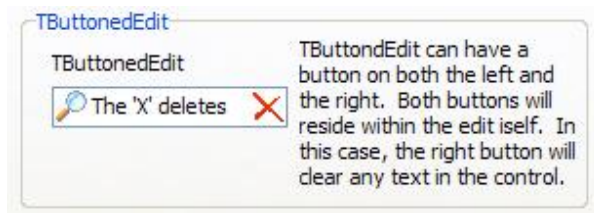


그림 13 – 검색을 위한 TButtonEdit. 빨간색 X는 텍스트를 지우기 위해 사용 가능.

기존 컴포넌트의 업데이트

기존의 컴포넌트 또한 많이 업데이트 되었습니다:

- TImageList 컴포넌트는 TImage 컴포넌트가 지원하는 모든 이미지 타입을 포함할 수 있습니다.
- TButton의 버튼이 강화되어 이미지를 표시할 수 있게 되었으며 버튼의 상태에 따라 해당 이미지를 보여줄 수 있습니다. 비스타에서 TButton은 이제 CommandLink와 SplitButton 스타일을 지원합니다.
- TTreeView는 이제 이미지 확장을 지원합니다. 즉 노드가 접히거나 열렸을 때 서로 다른 이미지를 표시할 수 있습니다.
- TListView는 이제 비스타 그룹을 지원합니다.
- TRichEdit는 이제 윈도우 리치에디트 (RichEdit) 2.0 스펙을 지원합니다.
- TProgressBar는 이제 테마룩, 마키 그리고 스무스 모드 스타일을 지원합니다. 그리고 비스타에서는 일시정지(Pause) 및 중지 (Stopped) 상태가 지원 됩니다.
- TEdit는 이제 패스워드 문자를 커스터마이징 할 수 있으며 TextTip 속성을 제공합니다. TextTip은 TEdit가 비어있고 포커스가 없을 때 “힌트”를 텍스트로 표시합니다.

VCL의 새 기능에 대한 전체 데모는 다음 코드기어 디벨로퍼 네트워크 (CDN)에 있습니다:

<http://windemo1.codegear.com/Tiburion/LaunchReplays/DelphiVCL/DelphiVCL.html>

새로운 멀티티어 데이터베이스 아키텍처 – DataSnap 2009

Delphi는 최초로 멀티 티어 데이터베이스 개발 프레임워크가 도입된 개발툴들 중 하나입니다. 데이터스냅(DataSnap)은 개발자들이 단일 애플리케이션에서 데이터를 제공하고 비즈니스 룰을 관리할 수 있게 해주는 애플리케이션 서버를 미들티어 애플리케이션을 개발할 수 있게 해주었습니다. 미들티어는 데이터베이스 연결의 파수꾼 역할로서, 데이터 액세스를 제공하고 그 데이터의 처리 및 변경 작업에 비즈니스 룰을 강제 합니다. 또한 데이터스냅 (DataSnap)은 미들티어에 액세스하는 강력한 클라이언트 솔루션을 제공하고, 또한 클라이언트에서 데이터를 관리하고 조작하기 위한 강력한 메모리 데이터셋(in-memory DataSet)을 제공합니다.

Delphi 2009에서는 데이터스냅 아키텍처를 대폭 업데이트 하였습니다. 이전 버전까지는 데이터스냅이 COM 기술을 사용하였지만, Delphi 2009에서는 이러한 의존성을 제거하고 강력하면서도 가벼운 구현인 서버 메소드 (Server Methods)로 대체하였습니다. 서버 메소드 (Server Methods)를 통해 개발자는 미들 티어에 속해있는 메소드를 자연스럽게 클라이언트에서 사용할 수 있습니다. 개발자는 클라이언트 바이너리 안에서 실행되는 코드와 동일하게 서버 메소드를 호출할 수 있습니다. 서버 메소드는 스트링, 정수, 데이터셋, 데이터리더, 커백션, OLEVariant 등 디비익스프레스 (dbExpress) 타입 시스템의 모든 타입을 파라미터로 전달할 수 있습니다. 이것은 클라이언트와 미들 티어 사이의 데이터 전달을 위한 매우 강력한 수단이 됩니다.

이 새로운 데이터스냅은 기존의 데이터스냅 서버 또한 지원합니다.

새로운 IDE 기능

리소스 매니저

대부분의 윈도우 애플리케이션은 윈도우 리소스들을 포함합니다. 비트맵, 커서, 폰트 등의 데이터가 컴파일된 바이너리 안에 포함될 수 있습니다. Delphi 2009 는 이제 리소스 매니저 (Resource Manager)를 제공하여 개발자들이 프로젝트에 리소스를 쉽게 추가하고 관리할 수 있게 해줍니다. 또한 개발자는 모든 표준 윈도우 리소스 유형을 추가하고 이름을 지정할 수 있으며, 코드 안에서 TResourceStream 클래스를 사용하여 리소스를 추출할 수 있습니다.

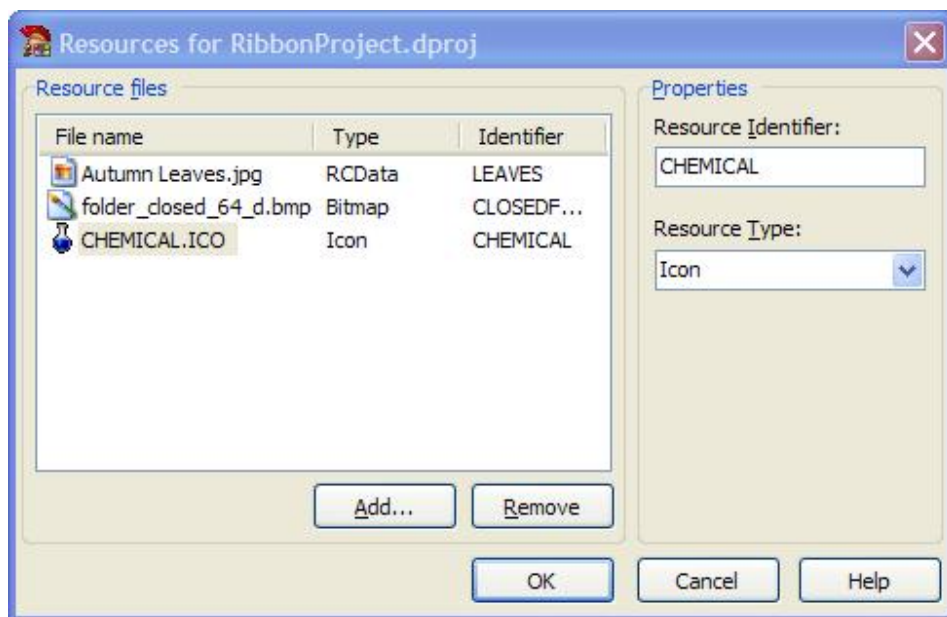


그림 14 – 리소스 매니저

빌드 설정 (Build Configuration)

많은 경우에 Delphi 프로젝트는 하나의 애플리케이션을 생성하기 위해 수많은 개별 프로젝트를 포함하게 됩니다. 개발자들은 여러 DLL, EXE, 패키지를 한데 모아 하나의 통합된 솔루션을 만들 수 있습니다. 이렇게 서로 다른 프로젝트와 코드 파일 모두를 관리하는 것은 어려운 일이 될 수 있습니다. 특히 빌드할 때에는 더욱 그렇습니다.

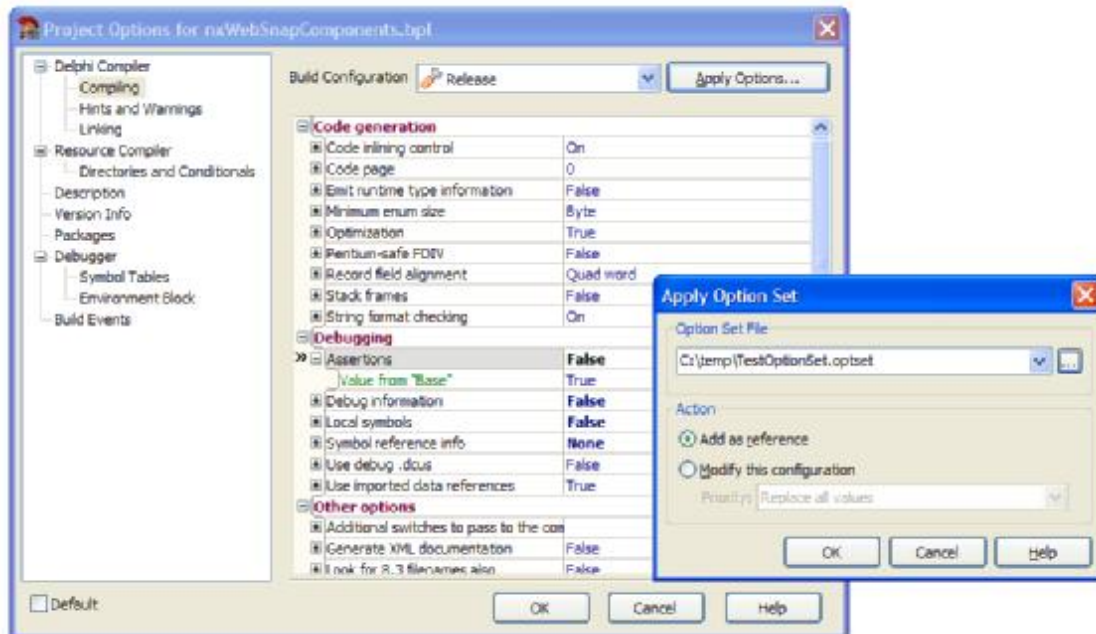


그림 15 – 상속된 설정을 가진 프로젝트옵션 대화상자와 옵션셋의 적용

프로젝트는 여러 목적에 맞게 서로 다르게 빌드되어야 할 필요가 종종 발생합니다. 테스트, 디버깅, 필드 테스트, 릴리즈 용 빌드 작업은 각각 서로 다른 설정과 컴파일러 옵션 셋팅을 필요로 합니다. Delphi 2009 는 IDE 와 커멘드라인 모두에서 동일하게 작동하는 유연한 빌드 설정 관리 시스템을 제공합니다.

Delphi 2009 는 개발자들이 프로젝트 옵션셋을 생성하여 빌드 옵션을 설정하고 관리할 수 있도록 합니다. 옵션셋은 다른 옵션셋으로부터 상속받아 원하는 대로 상속하고 오버라이딩 할 수 있습니다. 옵션셋은 파일로 저장되므로 특정 프로젝트 또는 다른 설정에도 사용될 수 있습니다.

클래스 익스플로러 (Class Explorer)

Delphi 2009 의 또다른 특징은 새로운 클래스 익스플로러 (Class Explorer) 입니다. 클래스 익스플로러는 Delphi 의 클래스 모델링 기능을 바탕으로 만들어졌으며 전체 프로젝트 내부의 클래스 구조를 볼 수 있게 해줍니다. 개발자는 클래스를 보는 방식으로 상향식(top-down) 또는 하향식(bottom-up)을 선택할 수 있습니다. 또한 클래스 익스플로러는 기존 클래스 라이브러리에 클래스, 메소드, 프로시저를 추가할 때에도 사용될 수 있습니다.

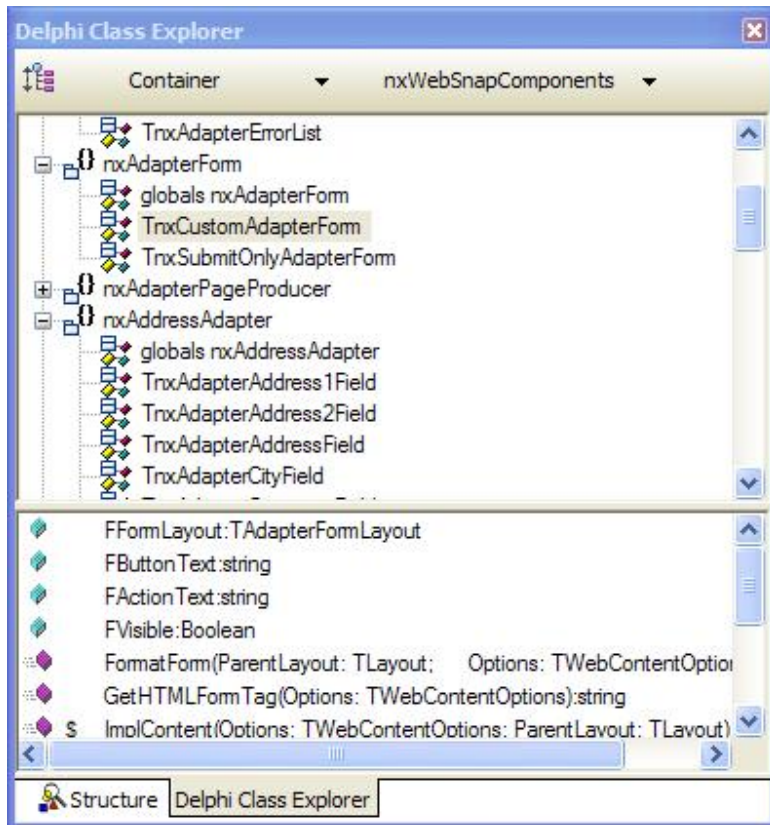


그림 16 - 클래스 익스플로러 (Class Explorer)

번역 툴 (Translation Tools)

세계화 (Globalization)는 Delphi 2009 의 중요한 테마이며, 개발자들은 새로운 세계화 기능을 활용하여 자신의 애플리케이션을 다른 언어로 번역하고 싶을 것입니다. Delphi 2009 는 업데이트되고 개선된 통합 번역 환경 (ITE) 과 외부 번역 매니저 (ETM)를 포함합니다.

ITE 는 IDE 에 통합되어 있는 도구로서, 개발자들이 애플리케이션의 번역 버전을 쉽게 만들 수 있게 해줍니다. ITE 에서는 여러 언어의 프로젝트를 원하는 대로 습니다.

ITE 는 메인 프로젝트를 스캔하여 번역 가능한 모든 스트링, 캡션, 텍스트를 추출합니다. 그런 다음 개발자 혹은 번역자가 스트링을 번역할 수 있도록 해주는 IDE 에 통합된 툴을 사용합니다. 번역된 내용들은 해당 프로젝트 안의 리소스 DLL 로 들어가게 됩니다. 일단 리소스 DLL 이 만들어지면 그 프로젝트가 실행될 때 적절하게 번역된 스트링이 표시되게 됩니다.

ETM 은 ITE 의 작업 결과와 함께 번역자 혹은 번역 서비스 회사 등으로 보내줄 수 있는 재배포 가능한 도구입니다. ETM 을 이용하여 애플리케이션의 번역을 쉽게 아웃소싱할 수 있습니다.

빌드 설정과 리소스 매니저 등 새로운 IDE 기능에 대한 전체 데모는 아래의 코드기어 디벨로퍼 네트워크 (CDN)에 있습니다:

<http://windemo1.codegear.com/Tiburon/LaunchReplays/Delphi2009IDE/Delphi2009IDE.html>

ITE 와 ETM 데모는 아래의 코드기어 디벨로퍼 네트워크 (CDN)에 있습니다:

<http://windemo1.codegear.com/Tiburon/LaunchReplays/ASCIInew/ASCIInew.html>

향상된 COM/ActiveX 지원

Delphi 2009 는 완전히 새로운 아키텍처가 적용된 COM 과 ActiveX 개발 프레임워크를 포함합니다. 이 새로운 프레임워크는 마이크로소프트 IDL 스펙의 서브셋인 새롭게 “축소된 IDL (Reduced IDL)” 언어에 기반을 두고 있습니다. 이것은 Delphi 의 COM 오브젝트를 정의하기 위해 특별히 설계되었습니다. RIDL 파일과 해당 PAS 파일의 조합은 COM 과 ActiveX 오브젝트가 이제는 순전히 텍스트로 정의될 수 있음을 의미합니다. 이 두개의 파일은 실제 프로젝트 자체의 외부에 타입라이브러리 파일 (*.TLB)로 컴파일 될 수 있습니다. 이것은 TLB 파일이 프로젝트의 컴파일된 결과물 뿐이지 프로젝트 자체의 일부가 아니라는 것을 의미합니다. 이런 방식으로 Delphi COM 과 ActiveX 오브젝트는 소스코드 관리 툴 안에서 적절하게 관리되고 병합되고 저장될 수 있습니다. TLB 파일은 이제 더 이상 COM/ActiveX 프로젝트 정보를 저장할 필요가 없습니다.

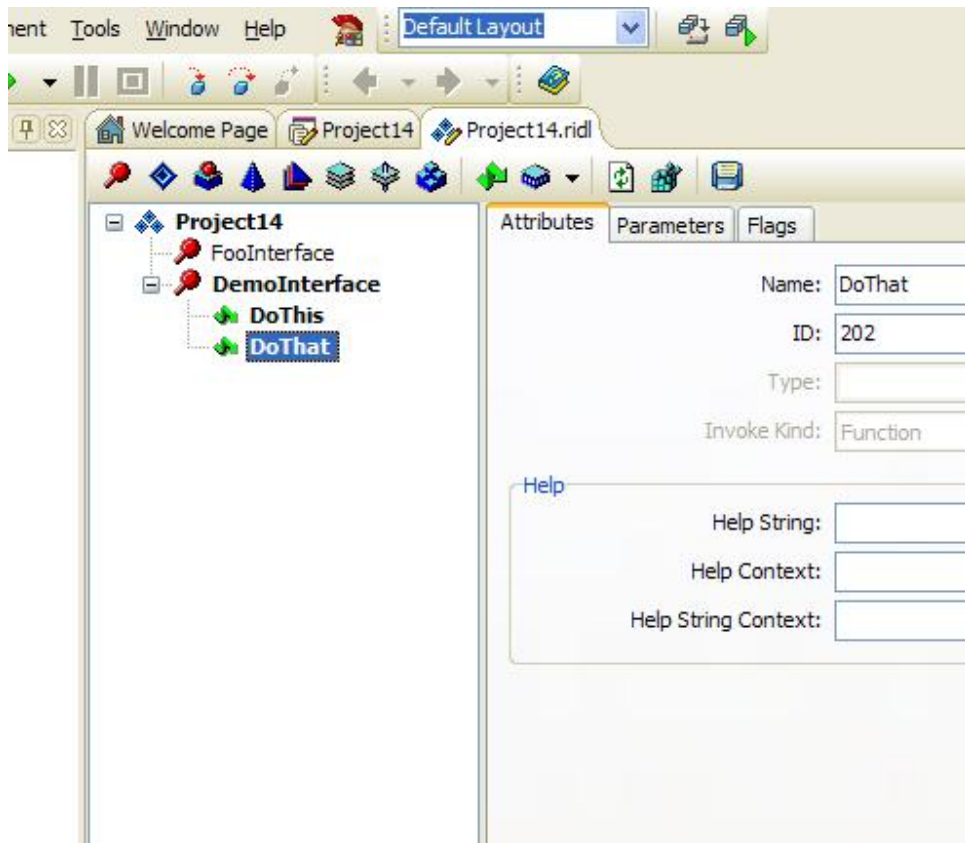


그림 17 - *.RIDL 파일을 편집하는 타입 라이브러리 에디터

Delphi 2009 의 새로운 COM 과 ActiveX 기능에 대한 데모는 아래의 코드기어 디벨로퍼 네트워크 (CDN)에 있습니다:

<http://windemo1.codegear.com/Tiburon/LaunchReplays/COM/COM.html>

Delphi 2007 에서 새로워졌었던 기능

다음은 Delphi 2007 에서 새롭게 추가되었던 기능들입니다. 물론 이것들 모두가 Delphi 2009 에 들어있습니다.

Blackfish™ SQL

Delphi 2007 은 Blackfish SQL Delphi 에디션을 도입하였습니다. Blackfish SQL 은 매니지드 코드로 된 SQL-92 호환 관계형 데이터베이스입니다. Blackfish 는 매우 유연한 RDBMS 입니다. 이것은 독립형 애플리케이션의 임베디드 데이터베이스로 사용될 수 있을 뿐만 아니라 본격적인 엔터프라이즈 데이터베이스 솔루션으로 쉽게 확장될 수 있습니다. 애플리케이션이나 웹 솔루션의 프로세스 안에서 작동하거나 서버나 윈도우 서비스로 동작할 수도 있습니다. 배포는 너무나 간단합니다. 바이너리, 데이터베이스 파일, 라이선스 파일을 XCOPY 하기만 하면 됩니다.

Blackfish SQL 은 .NET 프레임워크를 활용함으로써, RDBMS 로서 완벽한 기능을 제공하면서도 임베디드 시스템에서 엔터프라이즈 시스템으로까지 유연하게 확장 가능합니다.

Blackfish 는 다음과 같은 3 가지 서로 다른 방식으로 동작할 수 있습니다:

1. 윈도우 서비스
2. 독립형 실행
3. 인프로세스 어셈블리 (in-process assembly)

애플리케이션에서는 Blackfish 에 원격 또는 로컬로 연결할 수 있습니다. 원격 연결인 경우 해당 애플리케이션은 정보 전달을 위해 TCP/IP 스택을 사용할 것입니다. 로컬 연결일 경우에는 해당 애플리케이션은 Blackfish 어셈블리를 바인딩합니다.

Blackfish 는 설치될 때 시스템 시작과 함께 자동 실행되는 윈도우 서비스로 설치될 것입니다. 이를 통해 Blackfish 는 각 개발자 장비에서 항상 사용될 수 있습니다. Blackfish SQL 은 디폴트로 2508 포트를 사용합니다.

Blackfish SQL 추가 자료

- 애플리케이션 구축과 배포에 대한 보다 자세한 사항은 다음 아티클에 있습니다:
Developing and Deploying with Blackfish SQL and Delphi
- Blackfish 개발자의 Blackfish SQL 데모는, 여기에 있습니다.
- [Developing and Deploying with Blackfish SQL and Delphi](#)
- Blackfish SQL 개발자 가이드는 여기에서 다운로드 받으세요.
- Blackfish 개발 팀장인 스티브 쇼니시 (Steve Shaughnessy) 의 Blackfish 블로그는 여기에 있습니다.
- 레그네 코그니올리 (Leonel Tognioli) Blackfish SQL R&D 개발자의 발표입니다.
Introduction to Blackfish SQL

비스타 지원

비스타 운영체제의 출시로 비스타의 새로운 기능들을 지원하는 수많은 새로운 API 들이 추가되었습니다. Delphi 2009 는 이런 새로운 비스타 API 들을 직접적으로 이용할 수 있게 한 최초의 툴입니다. 새 기능들을 VCL 컴포넌트들과 속성들로 래핑함으로써, Delphi 는 개발자들에게 비스타의 새로운 기능들의 많은 부분들을 즉시적으로 그리고 쉽게 액세스할 수 있게 해줍니다. Delphi 2009 는 새로운 비스타 API 지원을 VCL 의 많은 컴포넌트에 추가하여 수준을 한단계 더 끌어 올렸습니다.

물론 개발된 애플리케이션이 포함한 비스타 지원 부분은 오직 비스타에서 실행될 경우에만 예상 대로 완벽하게 동작할 수 있습니다. XP 나 윈도우 2000 에서 작동할 경우 이와 같은 비스타 고유의 기능은 “자연스럽게 하향조정” 되어 XP 기본형으로 표시됩니다.

글래스 효과

Delphi 2007 에서는 윈도우 에어로(Aero) 인터페이스를 지원이 도입되었습니다. VCL 에서 폼의 테두리에 “글래스” 프레임을 만들어주는 속성들을 TForm 에 추가했습니다. 이렇게 해서 네이티브 윈도우 개발자들은 자신의 네이티브 애플리케이션에서 이 기능을 활용할 수 있습니다. 이 기능을 구현하는 방법은 대단히 쉬워서, TForm.GlassFrame.Enabled 속성을 True 로 설정하기만 하면 됩니다. 개발자들은 폼의 바깥쪽 주변의 글래스 프레임의 너비를 지정할 수도 있으며, SheetOfGlass 속성을 True 로 설정하여 폼 전체에 글래스 효과를 낼 수도 있습니다.

글래스 효과를 사용하는 애플리케이션이 비스타가 아닌 시스템에서 실행되면 글래스 효과는 아무 동작도 하지 않습니다.

비스타 다이얼로그

새 비스타 운영체제에서는 또한 운영체제에 포함된 표준 다이얼로그들도 강화되었습니다. Delphi 는 이런 새로운 다이얼로그들을 TFileOpenDialog, TFileSaveDialog, TTaskDialog 컴포넌트로 캡슐화했습니다. 이런 컴포넌트들은 새로운 비스타 다이얼로그 컨트롤들에서 제공하는 기능들을 캡슐화한 것들로서, Delphi 개발자들이 매우 쉽게 이들 다이얼로그들을 자신의 애플리케이션에 통합할 수 있게 해줍니다.

이 기능은 오직 비스타 기반 시스템에서만 지원되므로, 이런 새로운 다이얼로그들을 사용하는 애플리케이션이 비스타가 아닌 시스템에서 실행되고 새 다이얼로그를 호출하면 특수한 예외가 발생합니다. 개발자들은 이것을 체크하여 해당 운영 OS 에 맞게 지원되는 동작을 불러낼 수 있습니다.

AJAX 와 VCL for the Web

Delphi 2007 for Win32 에서는 VCL 에 새로운 부분으로 VCL for the Web 이 추가되었습니다. [AtoZed Software](#) 의 IntraWeb 기술을 기반으로 한 VCL for the Web 은 개발자들이 (웹 사이트가 아닌) 웹 애플리케이션을 표준 Delphi 클라이언트를 개발하는 방식과 동일한 방식으로 개발할 수 있게 해줍니다. VCL for the Web 은 Win32 용 VCL 과 매우 비슷하게 동작하는 컴포넌트 기반의 기술입니다. 개발자들은 폼에 컴포넌트를 드롭하고 속성들을 설정한 후 일반적인 Delphi 폼에서처럼 웹 페이지를 디자인할 수 있습니다. 하지만, 이 VCL for the Web 애플리케이션은 브라우저에서 실행됩니다.

VCL for the Web 은 웹 사이트가 아닌 웹 애플리케이션을 생성하기 위해 특별히 맞춰진 최초이자 유일한 툴입니다. 이 기술은 현재 시장에서 구할 수 있는 어떤 다른 툴보다 더 빠르고 더 쉽게 웹 애플리케이션을 개발할 수 있게 해줍니다. VCL for the Web 은 강력한 HTML 엔진을 기반으로 하므로 개발자들이 일반적인 윈도우 애플리케이션을 디자인하는 것과 동일한 방법으로 웹 애플리케이션을 디자인할 수 있게 해줍니다. 드래그&드롭 접근 방식을 이용함으로써, 개발자들은 컨트롤들을 폼(HTML 페이지와 폼이 합쳐진 것이라고 볼 수 있습니다)에 드롭하고 이벤트를 만들고 속성을 설정할 수 있습니다.

VCL for the Web 은 또한 자동으로 비동기 XML 및 Javascript 기술(AJAX)을 광범위하게 활용합니다. VCL for the web 은 가능한 경우에는 자동으로 필요한 AJAX 코드를

애플리케이션에 삽입하여 클라이언트 측 처리의 이점을 이용할 수 있게 해줍니다. 단순히 디자인타임에 Async 이벤트에 코드를 연결하기만 함으로써 개발자들은 Delphi 코드만으로 AJAX 기반 이벤트를 클라이언트 브라우저에 생성할 수 있습니다.

VCL for the Web 데모는 아래의 코드기어 디벨로퍼 네트워크 (CDN)에 있습니다:
<http://windemo1.codegear.com/Tiburón/LaunchReplays/D2009Intraweb/D2009Intraweb.html>

디비 익스프레스 4 (dbExpress 4)

데이터베이스 지원과 데이터로의 컴포넌트 기반 액세스는 Delphi의 첫 등장 때부터 Delphi의 특징들 중 핵심적인 부분이었습니다. 오랜 전통을 이어, Delphi 2007에서는 새로운 기반 데이터 액세스 아키텍처인 디비익스프레스 4 (dbExpress 4, DBX4)가 포함되었습니다. 이전의 디비익스프레스 3와 컴포넌트 레벨에서 완벽하게 하위 호환되는 DBX4는 완전히 새로 구축된 Delphi 데이터베이스 액세스 레이어로서, 동일한 소스로 데이터를 다루는 통일된 수단을 제공합니다.

DBX4는 데이터베이스 드라이버를 개발하는 과정을 대단히 단순화시켜주므로, Delphi 개발자들이 더 많은 범위의 데이터베이스 서버들을 액세스할 수 있도록 해줍니다. 완전히 오브젝트 파스칼로 작성된 DBX4는 네이티브 및 매니지드 코드 사이에서 크로스 플랫폼 개발이 가능하게 해주며, 연결 풀링(connection pooling)과 명령 추적(command tracing) 기능도 포함하고 있습니다. 추가로, DBX4는 개발자들에게 델리게이트 모델을 공개하여 데이터 액세스 과정에 끼어 들어 데이터의 읽기 및 쓰기 과정에 추가 기능을 할 수 있도록 해줍니다. 또한 DBX4는 이전 버전에 비해 현저하게 성능이 높아졌습니다.

Delphi 2007에서 새로워진 또다른 점은 지원되는 9개의 데이터베이스 모두에서 데이터베이스 메타데이터를 읽고 쓸 수 있도록 풍부한 메타데이터 클래스 셋이 있다는 것입니다.

디비익스프레스 4 (dbExpress 4)는 저수준의 아키텍처이므로, Delphi 2007 사용자는 애플리케이션 수준에서는 거의 차이점을 느낄 수 없습니다. dbExpress 컴포넌트들, 즉 데이터를 액세스하고 데이터 컨트롤들에 데이터를 제공하는 VCL 컴포넌트들은 그다지 변경되지 않았습니다. 그 대신, DBX4 아래의 코드는 더 효율적이고 확장성 있게 변경되었습니다. 일반적인 사용자들은 큰 변화를 느끼지 못하겠지만, 속도 개선과 데이터에 대해 더 폭넓은 액세스를 할 수 있게 된 것은 분명히 알아챌 것입니다.

Delphi 2009의 주요 추가 기능들

이 장에서는 Delphi 2009의 많은 추가 기능 중 Delphi 2009를 살펴볼 때 가장 관심이 갈 수 있는 몇가지에 대해 살펴봅니다.

통합 개발환경 (IDE)

코드 에디터 (Code Editor)

Delphi 2009의 코드 에디터는 코드 개발을 위한 강력한 환경입니다. 이 코드 에디터에는 코드를 작성하는 지겨운 작업들을 줄여주는 여러 기능들이 포함되어 있어, 개발자들이 빠르게 적절한 메소드 이름을 찾고, 템플릿으로 공통 코드 블록을 삽입하고, 코드 에러를 쉽게 알아볼 수 있게 해줍니다.

라이브 템플릿 (Live Templates)

라이브 템플릿(Live Templates)은 개발자들이 몇 번의 키 타이핑만으로 공통 코드 블록을 빠르고 쉽게 작성할 수 있게 해주는 코드 에디터 기능입니다. 라이브 템플릿은 자기 설명적(self-describing)이고 지능적인 코드 삽입 기능을 제공하며, 템플릿의 변수 부분들을 인터랙티브하게 오갈 수 있습니다. 라이브 템플릿은 짧은 기호를 큰 코드 블록으로 확장할 수 있게 해주며, 개발자가 직접 필요한 대로 커스터마이징할 수도 있습니다.

```
for i := 0 to List.Count - 1 do
begin loop counter
  <tab> or <shift + tab> to navigate
end;
```

그림 18 -- 라이브 템플릿이 for 루프를 자동으로 만들어주고 있습니다.

위의 그림 18 에서, 개발자는 단지 ‘for’라고 타이핑하고 스페이스 바를 누르기만 함으로써 해당 템플릿을 호출하였습니다. 그러면 템플릿이 나타나고 코드 템플릿을 채우기 위한 입력 지점을 알려줍니다. 개발자는 탭 키를 눌러 한 입력 지점으로부터 다른 입력 지점으로 이동할 수 있으며, 이동한 지점에서 필요한 정보를 채워넣을 수 있습니다. 이 라이브 템플릿 시스템은 또한 각각의 입력 지점에서 필요한 정보에 대한 힌트도 제공합니다.

라이브 템플릿은 템플릿이 어떻게 동작할지를 지정하는 단순한 XML 파일로 되어있습니다. 따라서 개발자들은 특정 목적을 위한 자신만의 라이브 템플릿을 쉽게 만들어낼 수 있습니다. 또한 라이브 템플릿은 스크립트 엔진을 이용할 수 있으므로 거의 모든 기능을 수행할 수 있습니다.

코드 완성 (Code Completion)

코드 완성 (Code Completion)은 개발자들이 클래스를 이용하려고 할 때 그 클래스에 대한 모든 정보를 제공하는 코드 에디터 기능입니다. 더불어 현재 코드의 범위 내에서 선언된 식별자들에 대한 모든 정보도 제공합니다..

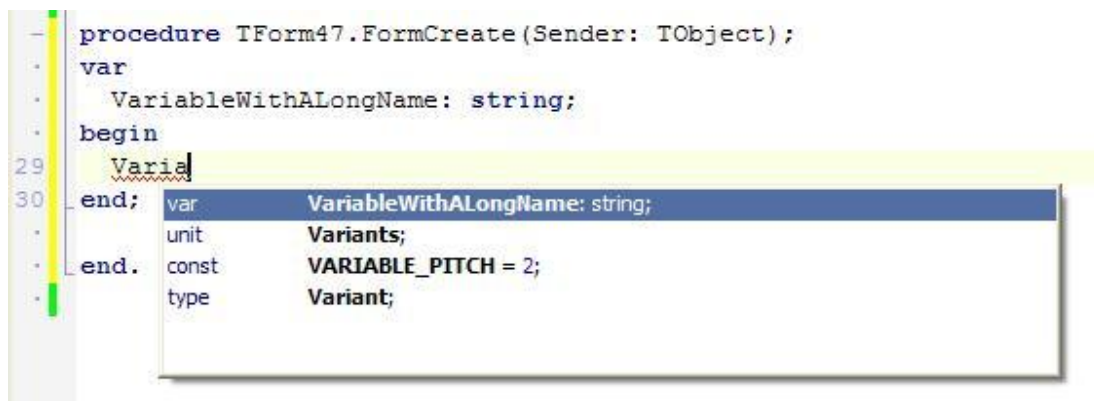


그림 19 – 코드 완성 (Code Completion) 이 복잡한 식별자를 쉽게 완성해 줍니다

중요한 애플리케이션들의 대부분은 많은 식별자들을 포함하게 되며, 그런 식별자들은 설명적인 의미를 가지고 있어 길이가 긴 경우가 많습니다. 코드 컴플리션은 그런 식별자들을 빠르고 쉽게 타이핑할 수 있게 해주며, 동시에 정확한 식별자가 입력될 수 있게 해줌으로써 타이핑 실수도 줄여줍니다.

그림 19 에서 보여주는 것처럼, 코드 컴플리션 기능이 호출되면 개발자가 타이핑을 시작한 식별자를 쉽게 찾아낼 수 있습니다. 선택한 후 엔터 키를 누르기만 하면 코드 에디터에 식별자 전체가 추가됩니다.

이에 더하여, 코드 컴플리션은 클래스나 레코드의 적절한 메소드와 필드를 빠르게 찾을 수 있게 해주어 코드 작성을 도와줍니다.

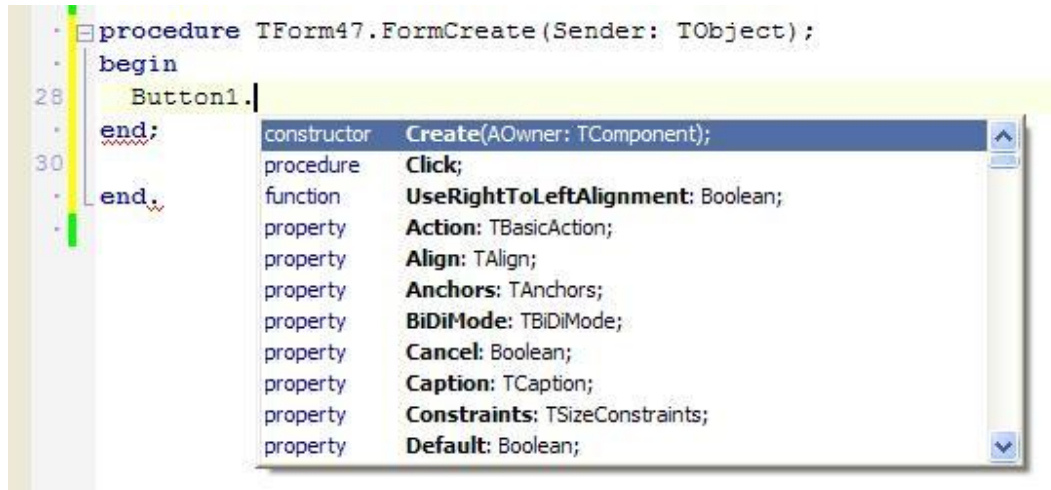


그림 20 – 코드완성 (Code Completion) 이 TButton 변수가 사용할 수 있는 모든 메소드를 나열합니다.

그림 20 에서, 개발자는 Button1 식별자를 입력하고 마침표를 타이핑한 후 CTRL+SPACE 키를 누름으로써 코드 컴플리션을 호출했습니다. 그러자 코드 컴플리션이 TButton 클래스에 속한 모든 메소드와 필드들을 나열해주었습니다. 개발자는 원하는 메소드나 필드를 계속 타이핑할 수도 있고(이 경우 타이핑하는 내용에 따라 코드 컴플리션 리스트가 필터됨), 혹은 원하는 항목을 찾기 위해 리스트를 스크롤해볼 수 도 있습니다. 원하는 항목을 찾아낸 후에는 단지 엔터 키를 누르기만 하면 선택된 항목을 코드 에디터에 추가할 수 있습니다.

블록 완성 (Block Completion)

블록 완성 (Block Completion)은 코드 블록이 항상 정확히 닫히도록 도와줌으로써 코드의 구조를 개선할 수 있도록 도와줍니다. 예를 들면, Delphi 에서는 모든 begin 에는 그에 해당하는 end 문이 따라와야 합니다. 모든 case 문도 end 를 필요로 합니다. 블록 컴플리션은 이런 문들이 개발자의 노력 없이도 정확하게 닫히도록 해줍니다. 개발자가 다음과 같이 타이핑하면:

begin<엔터 기를 누름>

블록 컴플리션은 end 를 추가하여 블록을 닫고 커서를 다음과 같이 ‘|’ 위치에 이동시킵니다:

```
begin
|
end;
```

에러 인사이트 (Error Insight)

에러 인사이트(Error Insight)는 개발자에게 코드 에디터에서 문법 에러가 있음을 비주얼하게 보여줍니다. 에러 인사이트는 코드의 문제를 인식하여 맞춤법 검사기에서의 “붉은 구불구불한 선” 기능과 비슷하게 개발자에게 알려줍니다.

```

- procedure TForm47.FormCreate(Sender: TObject);
-   var
-       Misspelled: string;
-   begin
-       ThisVariableDoesNotExist := 42;
30       ThereIsNoProcedureByThisName(aBadParameterName);
31       Misspelled := 'Misspelled';|
-   end;

```

그림 21 – 에러 인사이트 (Error Insight) 가 코드 에러를 강조합니다

헬프 인사이트 (Help Insight)

헬프 인사이트(Help Insight)는 식별자에 대한 정보를 알려주는 “툴팁” 형식의 팝업을 보여줍니다. 이런 툴팁은 개발자의 코드에서 세 개의 연속된 슬래시(///)로 주석 처리된 XML 태그로 정의할 수 있습니다. 개발자들은 자신의 코드를 주석 처리하고 그 주석이 IDE 에 의해 처리되어 해당 식별자 위에 마우스 커서가 위치했을 때 툴팁 헬프가 나타나도록 할 수 있습니다. 개발자는 툴팁 자체의 스타일을 CSS(Cascading Style Sheets)를 이용하여 형식을 지정할 수 있습니다.

```

- type
-   /// <summary>
-   ///   This is a class designed to demonstrate the Help Insight feature. This comment
8   ///   will show up in the Help Insight Tool-tip
-   /// </summary>
-   ///
10  THelpInsightClass = class
-   private
-       FSomeString: string;
-       procedure SetSomeString(const Value: string);
-   public
-       /// <summary>
-       ///   This is the SomeString property. It holds a string value. This comment
-       ///   will show up in the Help Insight Tool-tip
-       /// </summary>
20  property SomeString: string read FSomeString write SetSomeString;
-   end;
-
- implementation
-
-   { THelpInsightClass }
-
-   procedure THelpInsightClass.SetSomeString(const Value: string);
-   begin
-       FSomeStr
30  end;
-   end.

```

그림 22 – 헬프 인사이트를 위해 형식을 바르게 지정한 주석과 그 결과로 나타나는 헬프 인사이트 툴팁. 클래스 선언 위의 텍스트가 툴팁 윈도우의 텍스트와 일치합니다..

비주얼 디자인 관련 기능들

폼디자이너 (Form Designer)

Delphi 2009 의 폼 디자이너(Form Designer)는 Delphi 개발자에게 비주얼하고 이벤트 기반의 개발 인터페이스를 제공합니다. 폼 디자이너는 폼 위의 컴포넌트들을 비주얼하게 조작할 수 있게 해줍니다. 개발자는 툴 팔레트로부터 컴포넌트를 드래그&드롭하고 폼의 디자인을 조절하여 디자인타임에 유저 인터페이스를 만들어낼 수 있습니다. 컴포넌트들은 런타임에서 동작하는 것과 매우 비슷하게 동작하고 보여집니다. 오브젝트 인스펙터를 이용하여 컴포넌트들의 속성들을 설정하여 해당 컴포넌트를 조작할 수 있으며, 속성이 변경되는 즉시 비주얼하게 반영됩니다. 폼 디자이너 자체도 디자인을 위한 그리드나 비주얼 가이드라인 등 폼에 컴포넌트들의 배열하는 작업을 단순하고 쉽게 해주는 중요한 디자인 도움 기능을 제공합니다.

비주얼 가이드라인 (Visual Guidelines)

많은 애플리케이션들은 수많은 컨트롤들을 가진 폼들을 포함하고 있으며, 그로 인해 그 컨트롤들을 배열하는 일은 개발자들에게 어려운 작업들 중 하나입니다. 레이블은 자신이 설명하는 다른 컨트롤과 정확하게 배열되어야 하는 경우가 종종 있습니다. 잘 디자인된 폼에서는 컨트롤들이 적절하게 배열되어야 하며 무작위로 흩어져 있어서는 안됩니다. Delphi 의 폼 디자이너는 컨트롤들과 컨트롤 안의 텍스트가 정확히 배열되었는지를 보여주는 비주얼 가이드라인(Visual Guidelines) 기능을 제공합니다. 이 기능은 개발자들이 단지 마우스로 컨트롤을 움직이는 것만으로 폼을 빠르고 쉽게 완성할 수 있게 해줍니다.

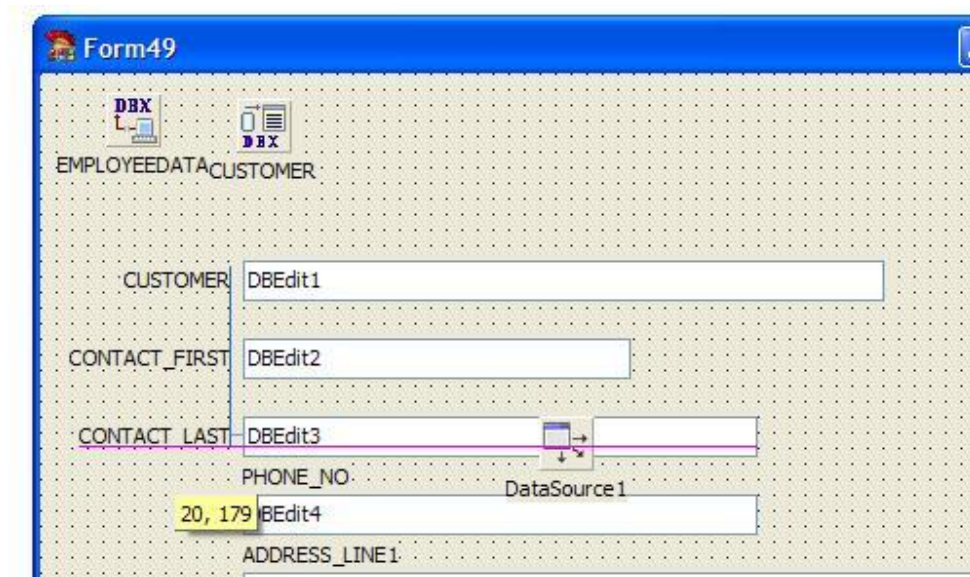


그림 23 -- 비주얼 가이드라인을 이용하여 레이블들과 에디트 박스들을 정렬합니다.

비주얼 가이드라인 (Visual Guidelines)은 세가지 유형의 정렬 표시를 제공합니다 (위 그림 23 을 보세요). 파란색 라인은 컴포넌트의 Top, Bottom, Left, Right 가 알맞게 정렬되었다는 것을 표시합니다. 진홍색 라인은 컨트롤 내부의 텍스트를 기준으로 표시합니다. 그리고 옅은 회색 라인은 컨트롤들이 자신의 Margin 및 Padding 속성에 따라 정확히 배치되었다는 것을 표시합니다.

오브젝트 인스펙터 (Object Inspector)

오브젝트 인스펙터 (Object Inspector)는 두 가지 기능을 가지고 있습니다. 컨트롤의 속성 값을 보여주고 설정할 수 있게 해주며, 또한 컴포넌트에 발생하는 이벤트와 그 이벤트에 연결된 코드 사이의 연결 관계를 제공합니다..

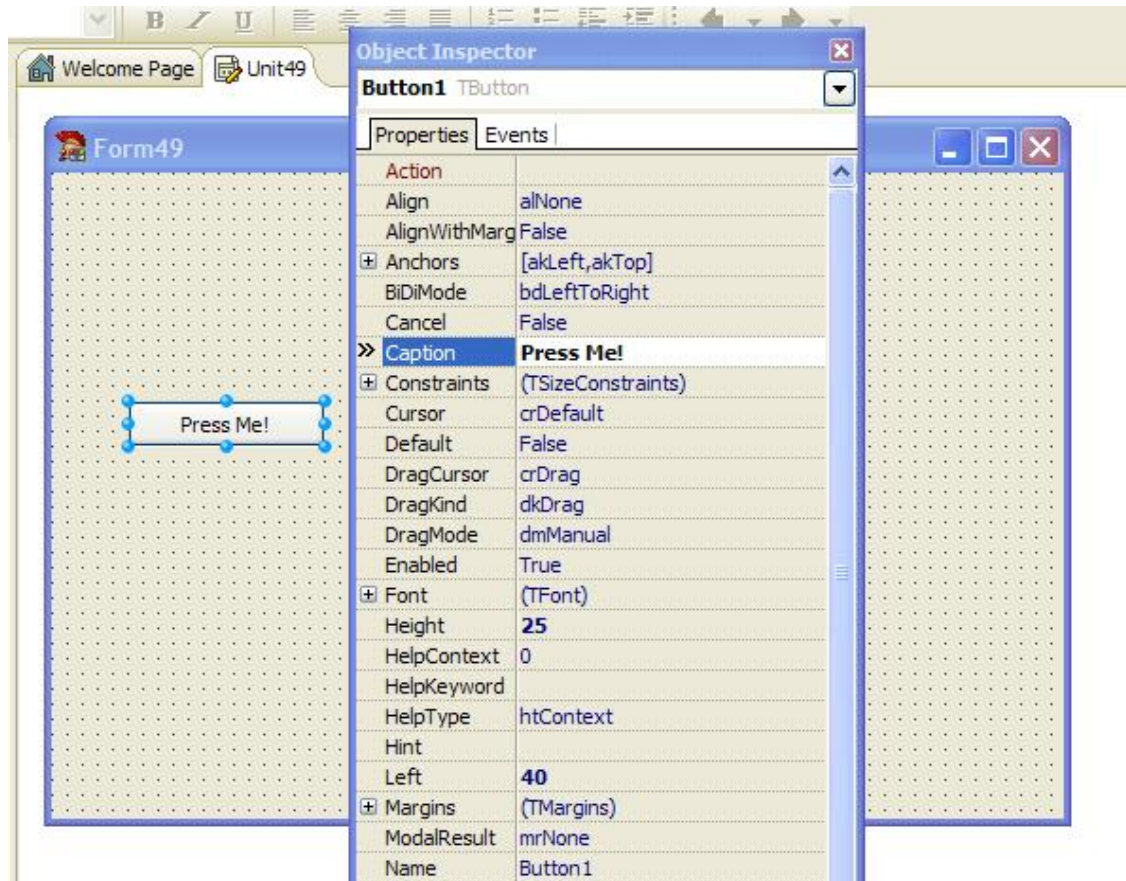


그림 24 -- 오브젝트 인스펙터가 TButton 컴포넌트의 속성을 보여줍니다

오브젝트 인스펙터는 동적이며 자동으로 선택된 컨트롤의 속성들을 보여줍니다. 하나 이상의 컨트롤이 선택되면 모든 컨트롤들에 공통된 속성들을 보여줍니다. 오브젝트 인스펙터는 기본값으로부터 변경된 속성들을 볼드체로 표시하여 알려줍니다. 속성들을 알파벳 순서 혹은 분류별로 정렬할 수 있습니다. 추가로, 오브젝트 인스펙터의 모양은 완전히 스타일 가능합니다.

모델링 뷰 (Modeling View)

모델링 뷰 (modeling view)는 프로젝트로부터 바로 리버스 엔지니어링 될 수 있습니다. 이것은 계층적 모델 뷰 (Model View) 를 생성하여 프로젝트 클래스들을 트리 형태로 나타냅니다. 또한 UML 클래스 다이어그램으로 표시될 수도 있습니다 이 모델과 다이어그램은 프로젝트 자동 문서화 생성에 사용될 수 있습니다..

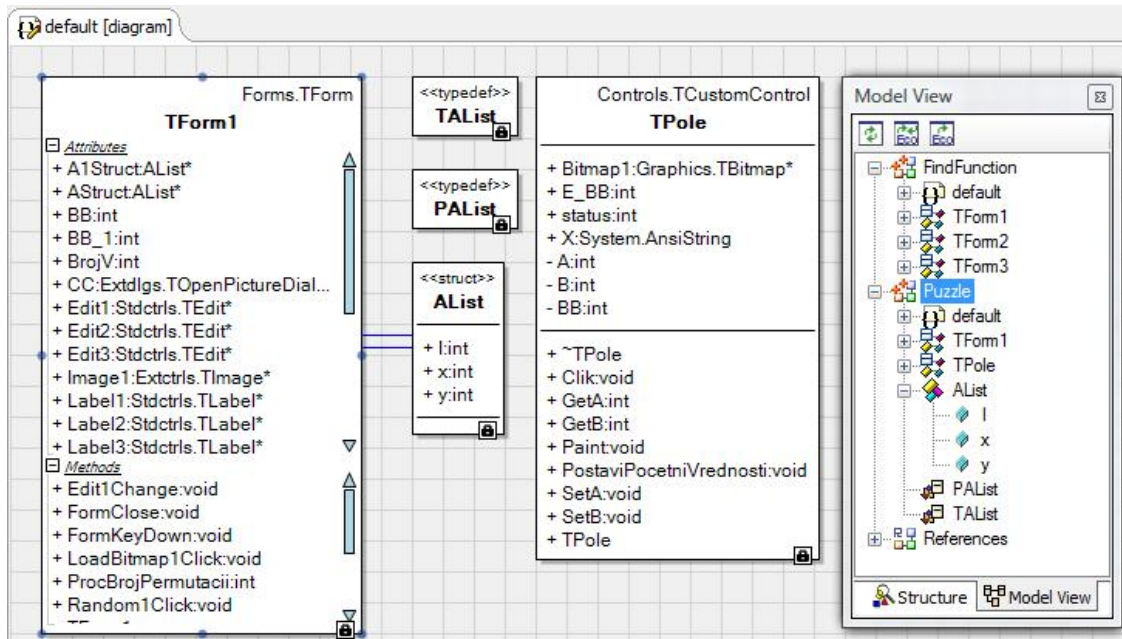


그림 25 – 모델뷰 (Model View) 에서 UML 클래스 다이어그램이 생성됩니다

프로젝트매니저 (Project Manager)

애플리케이션 프로젝트와 코드, 파일들, 그리고 그와 함께 필요한 다른 리소스들을 관리하는 일은 모든 개발툴에 있어 핵심적인 부분입니다. Delphi 는 Delphi 개발자들이 코드 파일들, 폼 파일들, 리소스들, 그리고 다른 항목들을 하나의 개발 프로젝트로 관리할 수 있게 해주는 강력한 프로젝트 매니저를 제공합니다.

프로젝트 매니저는 프로젝트의 계층적인 뷰를 제공합니다. 루트는 하나 혹은 그 이상의 프로젝트들을 포함하는 프로젝트 그룹입니다. 하나의 프로젝트 그룹에 속한 프로젝트들은 그 사이의 의존 관계를 고려하여 특정 순서대로 재정렬될 수 있습니다. 한 프로젝트 그룹 내의 모든 프로젝트들은 프로젝트 매니저에서 나타나는 순서대로 컴파일할 수 있습니다.

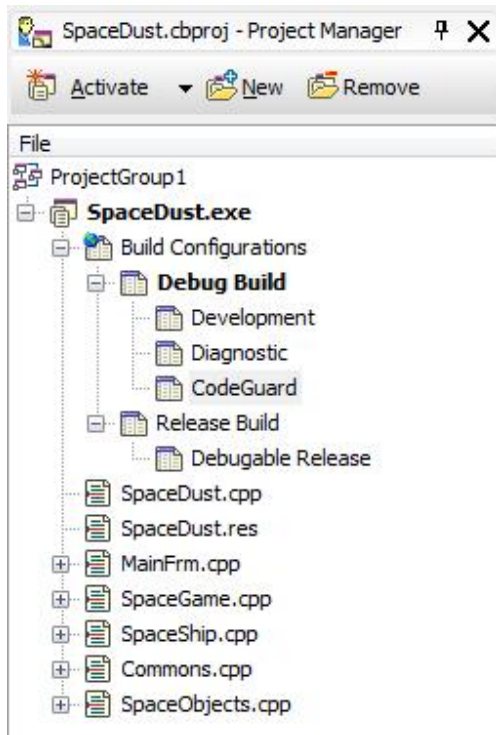


그림 26 – 프로젝트 매니저 안에 있는 빌드 설정 (Build Configurations)

빌드 설정 (Build Configuration)

개발자는 프로젝트 최적화, 경로 검색, 기타 옵션 등을 컨트롤하기 위해 다수의 빌드 설정을 만들 수 있습니다. 빌드 설정은 기본(base) 설정으로부터 옵션을 상속하고 특정 변경된 설정을 적용하여 재정의될 수 있습니다.

빌드 설정은 프로젝트 매니저를 통해 액세스 될 수 있으며, 컨텍스트 메뉴에서 한번의 빌드 명령으로 프로젝트의 모든 설정을 빌드할 수 있습니다.

프로젝트 옵션은 옵션 셋 (Option Set) 으로 저장될 수 있으므로, 동시에 다수의 프로젝트의 빌드 설정 간에 공유할 수도 있습니다..

빌드 설정 매니저(Build Configuration Manager) 는 현재 프로젝트 그룹 내의 모든 프로젝트에 적용될 수 있는 모든 명명된 설정을 나열합니다. 이름으로 설정을 선택하고 ‘모두 선택 (Select All)’을 클릭하고, ‘적용 (Apply)’하여 전체 프로젝트를 위해 필요한 설정을 쉽게 활성화할 수 있습니다.

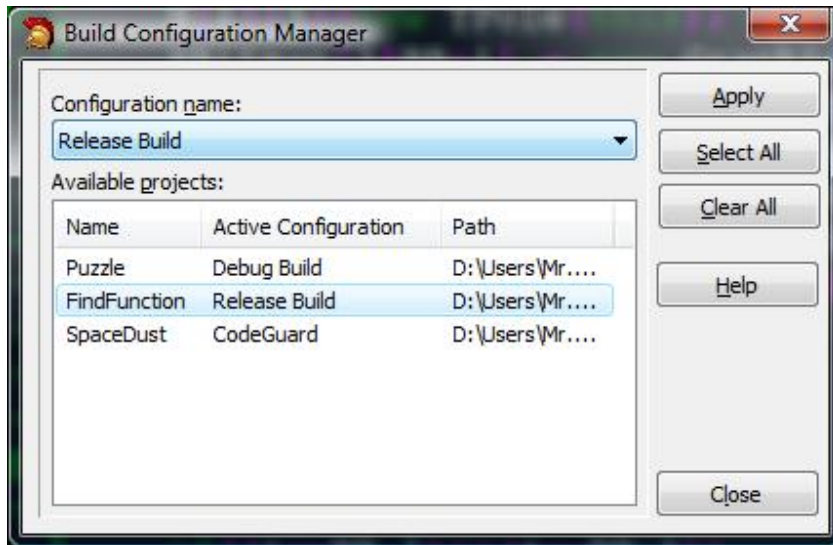


그림 27 – 3 개의 프로젝트를 보여주는 빌드 설정 매니저

디버거 (Debugger)

Delphi 2009 에는 완벽하고 완전한 기능을 갖춘 디버거가 통합되어 있습니다. Delphi 의 디버거는 애플리케이션의 실행 동안 완벽한 제어를 할 수 있게 해주며, 실행되고 있는 머신 코드에 대해 모든 정보를 제공합니다. 개발자들은 모든 변수들의 상태를 추적할 수 있으며, 코드 라인들을 한 라인씩 진행하면서 실행을 제어할 수 있습니다. 디버거가 브레이크포인트에서 실행을 일시 중지하면, 범위 내의 모든 변수들을 빈틈없이 쉽게 살펴볼 수 있습니다. 항상 스택 추적(stack trace) 기능을 사용할 수 있으며, 일반적으로는 스택에 대한 디버그 정보를 살펴볼 수 있어 개발자들이 역 방향으로 추적하여 어떻게 특정 지점이나 상태에 도착했는지 추적할 수 있게 해줍니다. 요약하자면, Delphi 의 디버거는 문제를 찾으려 할 때 실행중인 애플리케이션의 동작을 들여다보는 데에 개발자에게 필요한 모든 기능과 자유를 제공합니다.

브레이크포인트 (Breakpoint)

디버깅 기법들 중 가장 일반적이고 1 차적인 방법은 브레이크포인트(breakpoint)를 이용하는 것입니다. 브레이크포인트는 특정 지점에서 프로그램의 실행을 일시적으로 정지시키도록 디버거에게 지시하는 신호입니다. 디버거가 브레이크포인트에서 코드 실행을 정지시켰을 때 애플리케이션이 종료되거나 어떤 다른 방법으로 프로그램의 실행을 끝내지 않습니다. 대신, 디버거는 실행을 일시 정지시키고 개발자들이 애플리케이션의 동작과 상태를 들여다볼 수 있도록 해줍니다. 개발자가 애플리케이션의 상태를 볼 수 있는 방법들에 대해서는 아래에서 설명합니다.

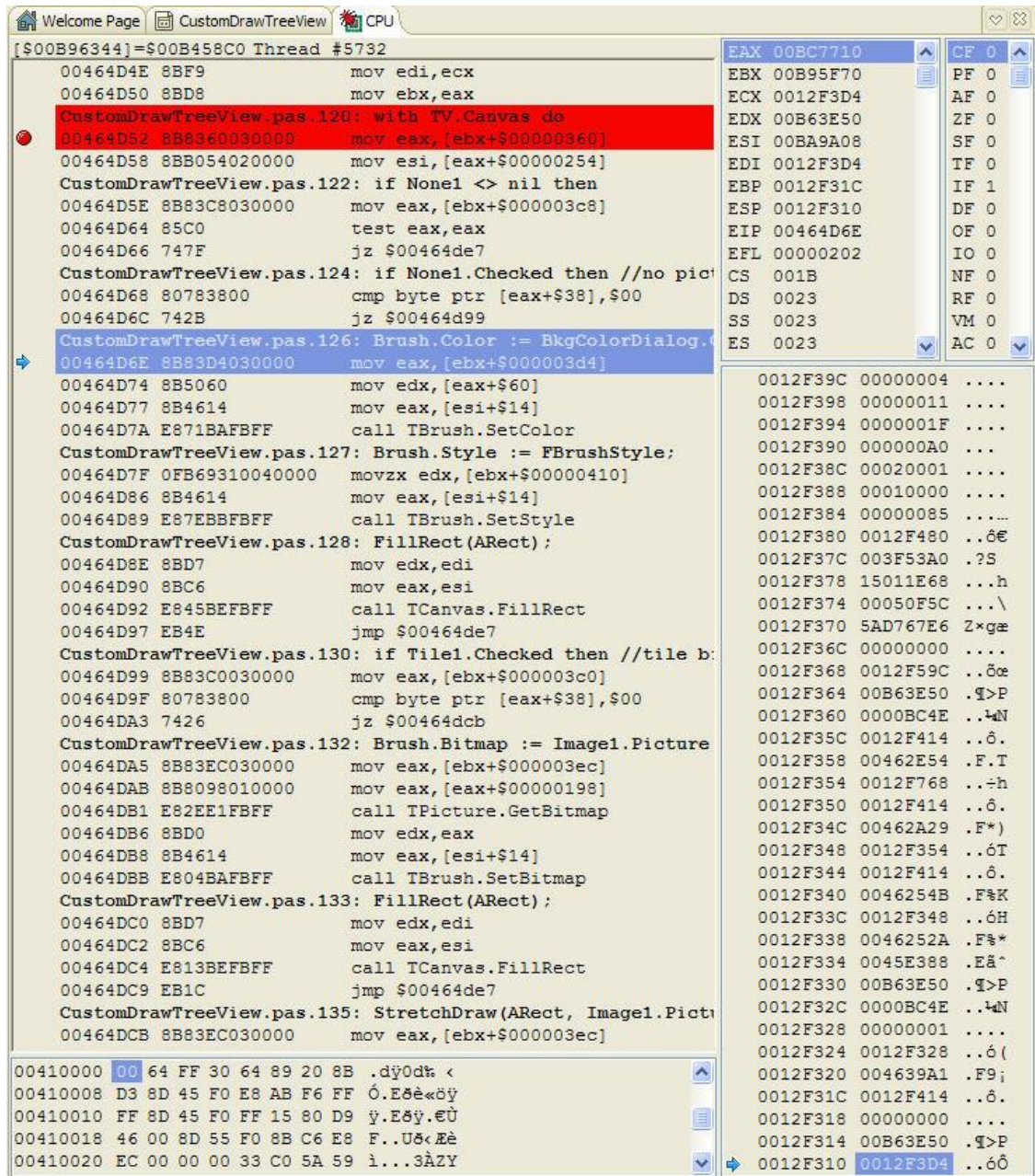


그림 29 – CPU 뷰에서 애플리케이션에 대한 완벽한 저수준 정보들을 보여주고 있습니다

통합된 유닛 테스트 (Integrated Unit Testing)

테스트 주도 개발(Test-driven Development; TDD)은 최근 몇 년 동안 점점 더 많은 인기를 끌고 있으며, 유닛 테스트는 그 방법론 중에서 주요한 부분입니다. Delphi 2009 는 [DUnit](#) (DUnit 은 Delphi 코드에 대해 유닛 테스트를 수행하기 위한 Delphi 기반의 라이브러리입니다). 개발자들은 특정 라이브러리 코드를 IDE 에서 처리하도록 지정할 수 있으며, 그러면 Delphi 는 선택된 클래스들의 public 및 published 메소드들을 위한 테스트 프로젝트와 비어 있는 테스트 케이스를 만들어주게 됩니다. 추가적인 메소드들이 테스트된 클래스에 추가되면, 유닛 테스트 위저드는 새 메소드들을 인식하고 이들 새 메소드들에 대한 새 스텝(stub) 메소드들

추가합니다. 따라서, 개발자들은 코드 라이브러리에 대한 유닛 테스트를 빠르고 쉽게 생성할 수 있습니다.

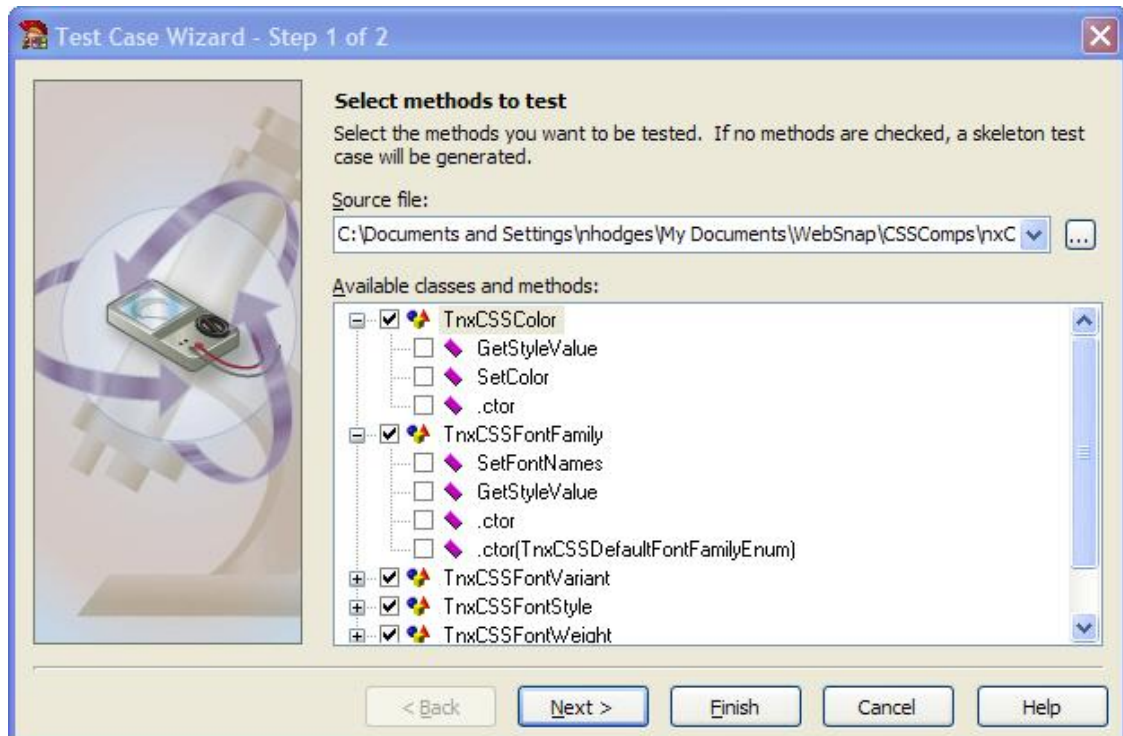


그림 30 – 유닛 테스트 위저드가 코드 라이브러리에 대한 유닛 테스트들을 만들기 위해 정보를 수집하고 있습니다

리팩토링 (Refactorings)

리팩토링(Refactoring)은 기존의 코드 바디를 재구성하여 외부적인 동작의 변경 없이 내부의 구조를 바꾸는 작업입니다. 수작업 기반의 코드 리팩토링은 최초의 프로그래밍 당시부터 진행되어왔지만, 최근에 이르러서야 형식화되고 자동화되었습니다. Delphi 2009 는 개발자들이 코드를 자동으로 리팩토링을 할 수 있도록 도와주는 여러 기능들을 제공합니다.

예를 들어, Change Parameter 리팩토링은 빠르게 메소드 선언과 구현 부분을 빠르게 변경해줄 수 있습니다. 이 기능은 파라미터를 추가하거나 제거할 수 있을 뿐 아니라 기존 파라미터의 특성도 변경할 수 있습니다.

이 기능을 이용하기 위해서는 에디터에서 메소드나 함수, 프로시저를 선택한 후(클래스에서의 선언 혹은 구현부 어디에서도 가능) Refactor | Change Params 메뉴를 선택하면 됩니다. 그림 31 과 같은 Change Parameters 대화 상자를 이용하여 필요한 변경 작업을 하면 됩니다.

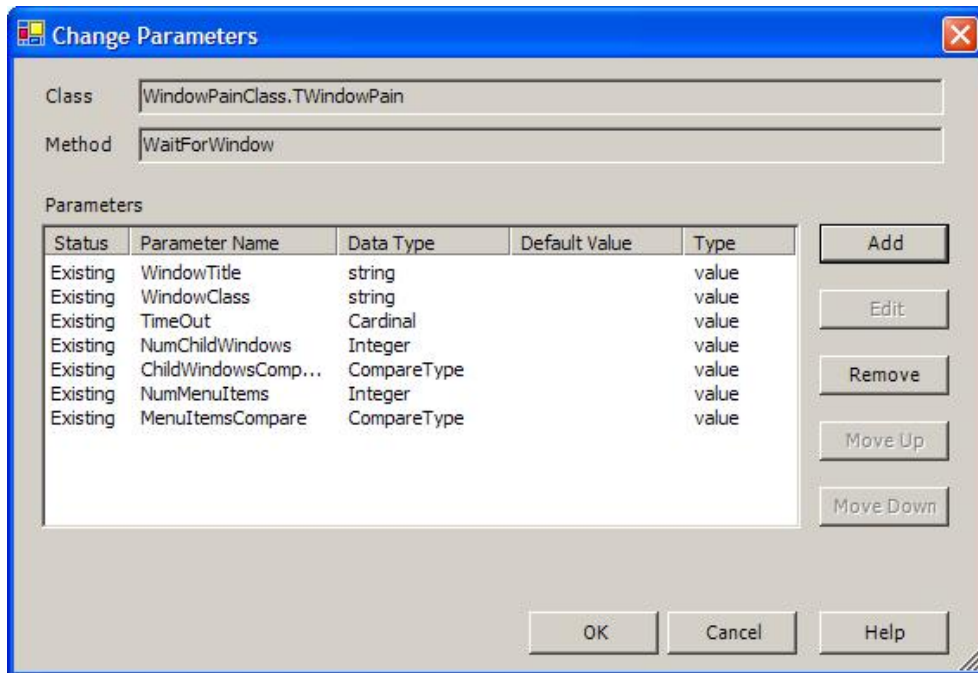


그림 31 - 체인지 파라미터 (Change Parameters) 대화 상자

이외에도 많은 패턴 관련 리팩토링 기능들이 Delphi 2009 에 포함되어 있습니다. Delphi 2009 의 모든 리팩토링 기능들을 아래의 표에서 요약해서 설명합니다:

리팩토링 이름	설명
Move	정적인 메소드를 한 클래스에서 다른 클래스로 이동시킵니다
Extract Interface	클래스의 선택한 메소드에 기반한 인터페이스를 만들고 그 클래스에서 구현되도록 선언해줍니다.
Extract Superclass	선택한 클래스 멤버를 그 클래스의 새로운 상위 클래스를 만들어 이동시킵니다.
Pull Members Up	선택한 멤버를 선택한 클래스의 바로 상위의 클래스로 이동시킵니다.
Push Members Down	선택한 멤버를 선택한 클래스의 바로 아래 상속된 클래스로 이동시킵니다.
Safe Delete	선택한 아이템이 애플리케이션의 어디에서도 사용되지 않을 경우 삭제해줍니다
Inline Variable	임시 변수의 인스턴스를 인라인 선언으로 대체합니다.
Introduce Field	기존의 로컬 변수를 지정한 클래스의 필드로 이동합니다.
Introduce Variable	표현식을 새 변수 이름으로 할당하는 문장을 추가하고 표현식을 변수로 바꾸고 선언합니다.
Rename	프로젝트 전체에서 지정한 식별자의 이름을 바꿉니다.
Declare Variable	사용되었지만 아직 선언되지 않은 로컬 변수를 선언해줍니다.
Declare Field	선택된 선언되지 않은 식별자를 지정한 클래스의 필드로 선언해줍니다.
Extract Method	선택된 코드를 기반으로 새 메소드를 생성해주며, 모든 필요한 파라미터를 전달하도록 해줍니다.

Extract Resource String

선택된 문자열을 기반으로 새 리소스 문자열을 만들고
문자열을 새 리소스 문자열의 이름으로 대체합니다.

Change Params

해당 메소드에서 파라미터를 추가, 변경, 제거합니다.

이 리팩토링 기능들 중 상당수는 프로젝트에서 모델링 지원이 활성화되어야만 사용
가능하다는 것을 주의해주시시오.

오브젝트 파스칼 언어 (Object Pascal Language)

오브젝트 파스칼 언어는 아주 오랜 동안 교육계에서 사용되어 왔습니다. 원래 프로그래밍
언어를 가르치는 데 사용되었었지만, 현재 오브젝트 파스칼 언어는 완전한 기능을 갖춘 객체
지향 개발 언어가 되었습니다. 오브젝트 파스칼은 언어 설계에 있어 최신의 트렌드 및
특징들의 발전에 보조를 맞추어왔습니다. 사실, 오브젝트 파스칼은 구조적 예외
핸들링(SEH)과 속성들을 구현한 최초의 주류 개발 언어입니다. 읽기 쉬운 문법과 쉽고
직관적인 코드 구성을 사용하는 Delphi는 배우기 쉬우면서도 동시에 극단적으로 강력한
언어입니다.

오브젝트 파스칼은 절차적 코드와 객체 지향 코드를 양쪽 모두 만들어낼 수 있습니다. 이
언어는 두 가지 방법을 매끄럽게 통합하여 개발자들이 양쪽 세계에서 최고가 될 수 있도록
해줍니다. 개발자들은 자신만의 클래스 계층을 만들고 사용할 수 있고, VCL을 사용하고
상속받아 새로운 클래스를 만들 수 있으며, 동시에 일반적인 절차적 코드 라이브러리도
만들어낼 수 있습니다. Delphi는 코딩 기법의 모든 영역을 커버하는 능력을 가지고 있습니다.

오브젝트 파스칼 언어에 대한 최고의 자료로 알려져 있는 것은 마르코 칸투(Marco Cantú)의
에센셜 파스칼 ([Essential Pascal](#)) (PDF 형식) 입니다.

비주얼 컴포넌트 라이브러리 (VCL)

VCL (Visual Component Library)는 완전한 기능을 갖추고 확장이 대단히 용이하며 강력한
컴포넌트 기반 클래스 라이브러리입니다. 가장 밑바닥에서부터 컴포넌트 기반으로 설계되어
비주얼 디자이너에서 사용되는 VCL은 1995년까지 그 역사가 거슬러올라갑니다. VCL은
지금까지 여러 플랫폼 변경을 거쳐왔으며, 여기에는 Win16으로부터 Win32 까지, 그리고
Win32로부터 닷넷 (.NET)까지가 포함됩니다.

데이터베이스 액세스 (Database Access)

Delphi는 단순하면서도 동시에 강력한 데이터 액세스 모델을 제공합니다. TDataset이라는
VCL 클래스를 기반으로 하는 Delphi의 데이터베이스 액세스는 매우 비주얼하며 매우
코딩하기 쉽습니다. 커넥션 컴포넌트를 데이터셋 컴포넌트에 간단히 연결함으로써 테이블
형태의 데이터를 쉽게 VCL 데이터 연결 컨트롤들에 공급할 수 있습니다. 사실, 코드를 전혀
작성하지 않고도 데이터를 표시할 수 있습니다.

그리고 데이터 필드들을 VCL의 TField 상속 클래스에 자동으로 바인딩 해줌으로써
개발자들은 이해하기 쉬운 코드로 쉽게 데이터를 가져오고 수정할 수 있습니다. 예를 들면
다음과 같습니다.:

```
CustomerTableCustomerNameField.AsString := 'Gary Johnson';
```

이런 방식으로, Delphi 개발자들은 비주얼하게 혹은 코드로 데이터를 쉽게 액세스할 수
있으며 코드 작업이 거의, 혹은 전혀 필요하지 않습니다. VCL 컴포넌트들은 데이터를
액세스하고 관리하는 데 있어 개발자들이 코드를 작성하도록 강요하지 않습니다..

웹 개발

VCL for the Web

Delphi 2009 에서 웹애플리케이션을 개발하기 위한 주된 방법은 VCL for the Web 입니다.

[AtoZed Software](#), 사의 IntraWeb 기술에 기반한 VCL for the Web 은, 디자인타임에는 Delphi 와 비슷한 느낌으로 개발하게 되지만 런타임에는 브라우저 기반의 웹 애플리케이션을 만들어내기 위해 특별히 설계된 컴포넌트 라이브러리입니다. VCL for the Web 은 Win32 용 VCL 과 매우 유사하기 때문에, 애플리케이션 개발 방식에 있어 일반적인 Delphi 방식에 익숙한 개발자들이 자신의 스킬을 브라우저에서 HTML 과 자바스크립트를 통해 제공되는 애플리케이션을 개발하는 데에 이용할 수 있도록 해줍니다.

VCL for the Web 은 또한 개발자들이 자바스크립트에 대한 지식이 전혀 없더라도 매우 쉽고 직관적으로 AJAX 기능을 구현할 수 있게 해줍니다. 개발자들은 서버에서 동작하는 오브젝트 파스칼 코드 기반 애플리케이션을 개발하여 클라이언트에서 실행시킬 수 있습니다..

멀티 티어 개발

VCL 은 멀티 티어 아키텍처를 개발하기 위해 DataSnap 이라는 강력한 아키텍처를 제공합니다. 다양한 통신 프로토콜을 채용하고 강력한 컴포넌트 세트를 제공함으로써 Delphi 개발자들은 데이터를 공급하는 애플리케이션 서버와 애플리케이션 서버로부터 데이터를 받거나 보내는 쉘 클라이언트를 적은 노력으로 쉽게 개발할 수 있습니다.

개발자는 TRemoteDataModule 클래스를 이용하여 HTTP, DCOM, 혹은 소켓 연결을 통해 인터페이스를 데이터 프로바이더 컨트롤로 간단히 리모팅할 수 있습니다. 다음으로 클라이언트 애플리케이션은 원격 데이터에 대한 레퍼런스를 얻어서 TClientDataset 이라는 데이터셋으로 임포트합니다. 데이터가 TClientDataset 에 들어오면 클라이언트 애플리케이션은 데이터에 일반적인 “Create, Read, Update, and Delete” (CRUD)의 모든 동작을 수행할 수 있으며, TClientDataset 이 모든 변경 내용을 캐시하게 됩니다. 클라이언트 애플리케이션은 심지어 서버로부터 완전히 연결을 끊을 수도 있으며(“서류가방 모델”), 이후에 다시 연결될 수 있습니다. 일단 다시 연결이 되면, TClientDataset 은 변경된 데이터를 서버로 다시 반영하기 위한 모든 코드를 제공합니다.

추가 자료

Delphi 로 만들어진 애플리케이션

Delphi 가 ISV 에게 매우 인기있는 개발 플랫폼인 이유는 손쉬운 배포 요구사항만으로 네이티브 개발을 제공하기 때문입니다. 널리 알려진 수많은 애플리케이션이 Delphi 로 작성되었습니다. 아래 참고 바랍니다:

- [FeedDemon \(NewsGator\)](#)
- [Skype](#) 윈도우 클라이언트
- [BeyondCompare](#) (Scooter Software)
- [The PortableApps Project](#)
- [FL Studio](#) (Digital Audio Workstation)
- [TestComplete](#) 및 [AQTime \(AutomatedQA\)](#)
- [FinalBuilder](#) (VSoft Technologies)
- [InstallAware](#) 인스톨 툴
- [MediaMonkey](#) 음악 관리 툴

- [Copernic Desktop Search](#)

Delphi 로 만들어진 애플리케이션들에 대한 더 많은 리스트는 [Delphi Wiki](#) 에서 찾을 수 있습니다.

Delphi 2009 가 바로 자기 자신을 가지고 만들어졌다는 점도 특기할 만한 사항입니다. [Delphi for PHP](#) 와 [Blackfish SQL](#) 등 코드기어의 다른 툴 또한 Delphi 2009 를 사용하여 개발되었습니다.

Delphi 2009 출시 이벤트

Delphi 2009 출시 일환으로, 엠바카데로는 Delphi 2009 의 새 기능과 특징점을 다루는 48 시간 온라인 콘텐츠를 만들었습니다. 발표내용 전체가 코드기어 디벨로퍼 네트워크 즉 CDN 에 있습니다.:

<http://dn.codegear.com/article/38478>

코드레이지 (CodeRage) I 온라인 컨퍼런스 발표 내용

2007 년 3 월, 코드기어는 [CodeRage](#) 라고 하는 온라인 컨퍼런스 발표를 하였습니다. 이 때 많은 코드기어 인사와 커뮤니티의 저명한 회원들이 Delphi 2009 에서 새로워진 기술에 대한 데모를 제공하였습니다. 일부 코드레이지(CodeRage) 중에서 다음 내용이 아마 관심이 갈 수 있을 것입니다. 다음은 컨퍼런스 이벤트 발표 플래쉬 파일의 압축 파일에 대한 링크입니다.

제목

[What's New in Delphi 2007](#)
[The New VCL for Delphi 2007](#)
[What's New in DBX4](#)
[What's New in the Delphi 2007 Debugger](#)
[Ajax Powered Web Applications with Delphi and IntraWeb 9.0](#)
[Web Services in Delphi 2007](#)
[MSBuild and the New Project Manager](#)

발표자

Nick Hodges, Delphi Product Manager
 Seppy Bloom, VCL R&D Engineer
 Steve Shaughnessy, Lead R&D Engineer for DBX4
 Chris Hesik, R&D Engineer for the Debugger UI

 Olaf Monien, AtoZed Software

 Bruneau Babet, Delphi R&D Engineer
 Dave Wilhelm— R&D Engineer for the IDE

코드레이지 (CodeRage) II 발표 내용

코드레이지(CodeRage) I 은 2007 년 11 월 코드기어 개발자를 위한 보다 큰 규모의 온라인 이벤트인 코드레이지 II 로 이어졌습니다. 이때 발표된 내용의 리스트는 다음과 같습니다. (주의 이 파일들은 HTML 파일에 임베드된 플래쉬 버전에 대한 압축 파일입니다.)

제목

[Delphi Product Address](#)
[Top Ten Things Added to Delphi since Delphi 7](#)
[Introduction to Blackfish SQL](#)
[Delphi 2007 and Vista](#)
[Delphi Dynamic 아키텍처 ures](#)

발표자

[Nick Hodges](#) – Delphi Product Manager
 Pawel Glowacki – Delphi EMEA Evangelist

 Leonel Tognioli -- CodeGear R&D Developer
[Marco Cantu](#) – WinTech Italia
 Marco Cantu

[with Packages](#)

[Introduction to Generics in Delphi](#)

Ray Konopka– [Raize Software](#)

[Creating Composite VCL](#)

Ray Konopka

[Controls](#)

[Building Advanced VCL.NET](#)

Holger Flick – Delphi Q&A Engineer

[ECO Applications and Migrating](#)

[Old Databases](#)

[Delphi Unit Testing Power Tools](#)

Craig Stuntz — Vertex Systems Corporation

[Effective UML Modeling in](#)

Alexander Evdokimov and Alexey Dubkov— CodeGear

[Delphi](#)

[**ClientDataset Tips and Tricks**](#)

Cary Jensen – Jensen Data Systems