

델파이 4 IDE 의 내부 (Internals of Delphi 4's IDE)

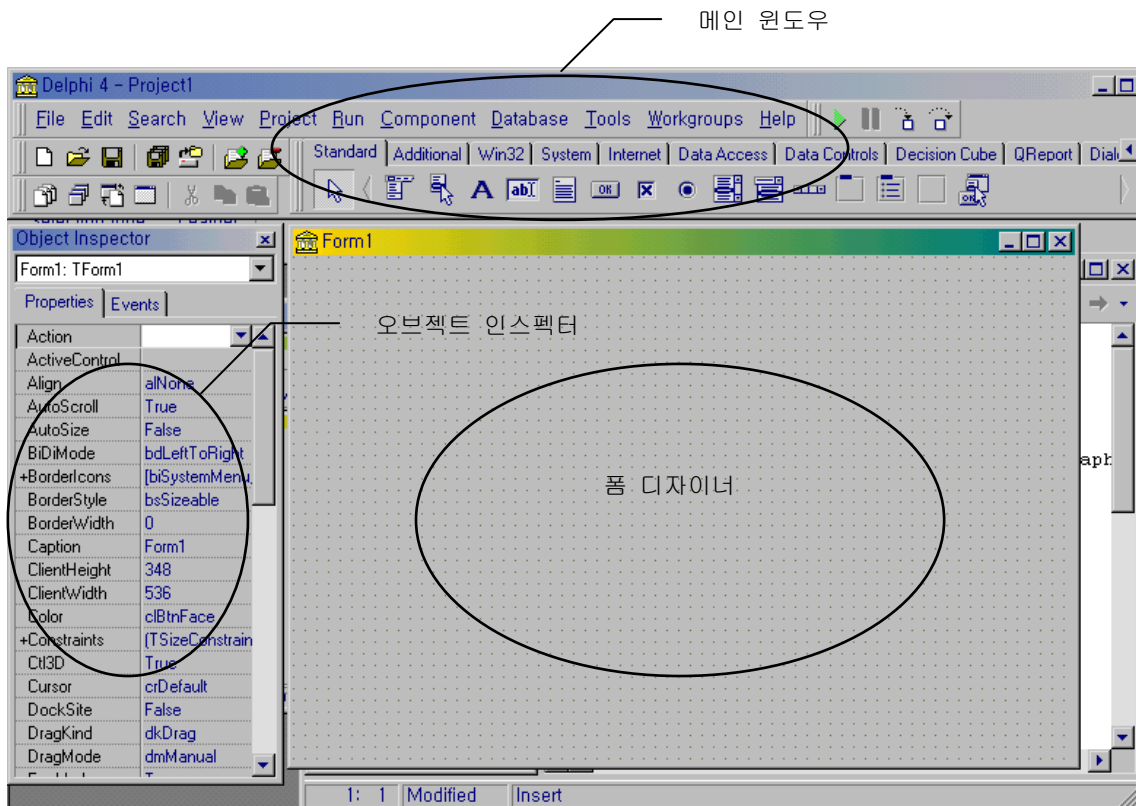
이번 장에서는 델파이의 통합개발 환경에 대하여 알아보도록 한다. IDE(Integrated Development Environment)는 어플리케이션을 설계하고, 실행시키고, 테스트할 수 있도록 해주는 환경을 말하는 것으로, 프로그램을 쉽게 개발할 수 있도록 도와주는 기능을 한다. 과거에는 개발자가 통합개발 환경이 없이 텍스트 에디터로 소스를 편집해서, 컴파일러로 컴파일 하고, 전용 디버거로 디버깅을 했었지만 볼랜드에서 터보 C 를 내놓으면서 처음으로 통합된 개발 환경을 지원하게 되었다.

그 후, 통합개발 환경은 MS 에 의해서도 지원되면서 개발자에게는 점점 더 편리한 환경으로 변모해가고 있는데 델파이 3 까지는 다소 부족한 면이 많다고 느껴왔던 통합개발 환경이 델파이 4 에서는 많이 향상되어 ‘역시 볼랜드’ 라는 탄성이 나오게 하였다.

그럼 델파이 4 의 통합개발 환경에 대해서 알아보는데 시간을 가지도록 하자.

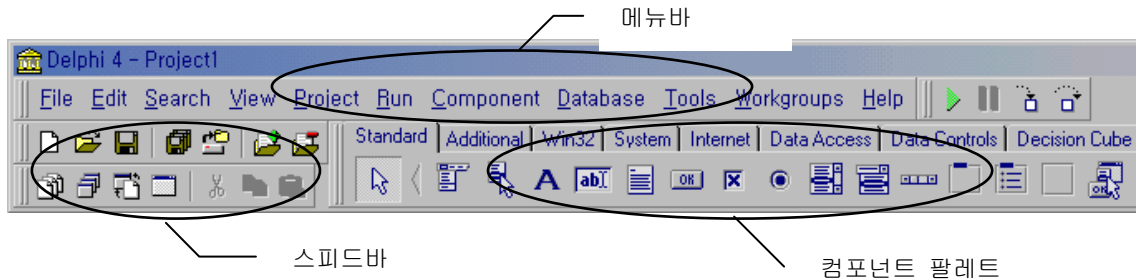
메인 윈도우

델파이를 실행시켜서, 프로그램이 모두 로딩되고 나면 다음과 같은 그림이 나타나게 된다.



전체적인 IDE 의 형태는 메인 윈도우와 오브젝트 인스펙터, 그리고 코드 에디터와 모듈 탐

색기와 폼 디자이너로 구성되어 있다. 일단 처음 실행하면 메인 윈도우와 오브젝트 인스펙터, 폼 디자이너가 보이게 된다. 델파이의 메인 윈도우는 크게 나누어 메뉴바(Menubar), 스피드바(Speedbar), 컴포넌트 팔레트(Component Palette)로 이루어진다.



델파이 4의 IDE에서 델파이 3와 바뀐 점을 든다면, 기본적으로 이런 윈도우들이 도킹을 지원한다는 것이다. MS 오피스 97에서부터 채용된 이런 형태의 툴바는 이제는 거의 표준이 되어간다는 느낌이다. 오피스 97과 마찬가지로 델파이 4의 메뉴바, 스피드바, 컴포넌트 팔레트도 마음대로 위치를 이동시킬 수도 있고, floating 윈도우로 나타나게 할 수도 있다.

스피드바 (Speedbar)

스피드바는 가장 자주 사용되는 기능들을 쉽고 빠르게 사용할 수 있도록 하기 위해 설계되었다. 기본적으로 가장 자주 사용될 것으로 예상되는 기능을 모아 놓은 것으로, 이들 기능은 델파이의 메뉴바를 통해 메뉴를 직접 선택하여 사용할 수도 있다. 이들 각각에 대해서는 나중에 주요 메뉴를 설명할 때 자세히 기술하도록 하겠다.

컴포넌트 팔레트 (Component Palette)

컴포넌트 팔레트는 VCL(Visual Component Library)에 포함되어 있는 구성요소를 가리킨다. 이들 항목들은 개발자가 원하는 대로 그룹을 형성할 수도 있지만, 기본적으로는 기능별로 구성되어 있다. 이들 그룹은 페이지 탭의 형태로 나뉘어 있다. 개발자는 컴포넌트 팔레트에서 사용하고자 하는 컴포넌트를 클릭해서 선택한 후, 이를 폼에 위치시키거나 더블 클릭하여 폼에 컴포넌트를 추가할 수 있다.

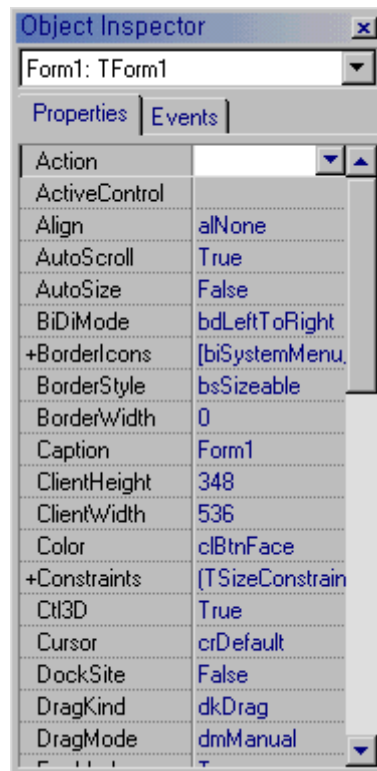
폼 디자이너 (Form Designer)

기본적으로 거의 모든 델파이 어플리케이션은 폼으로 구성된다. 델파이에서 폼은 다른 델파이의 컴포넌트들을 위치시킬 수 있는 장소로 사용된다. 개발자는 마우스를 가지고 폼의 위치와 크기 등을 마음대로 조절할 수 있으며, 컴포넌트를

폼에 올려 놓고 자신에 입맛에 맞도록 디자인할 수 있게 된다.

오브젝트 인스펙터 (Object Inspector)

오브젝트 인스펙터는 각 컴포넌트의 속성을 변경시키거나, 객체가 반응을 하게 되는 이벤트를 조정하는데 매우 편리한 인터페이스를 제공하고 있다. 오브젝트 인스펙터를 잘 살펴보면, 속성(Properties) 탭과 이벤트(Event) 탭으로 구성되어 있음을 알 수 있다.



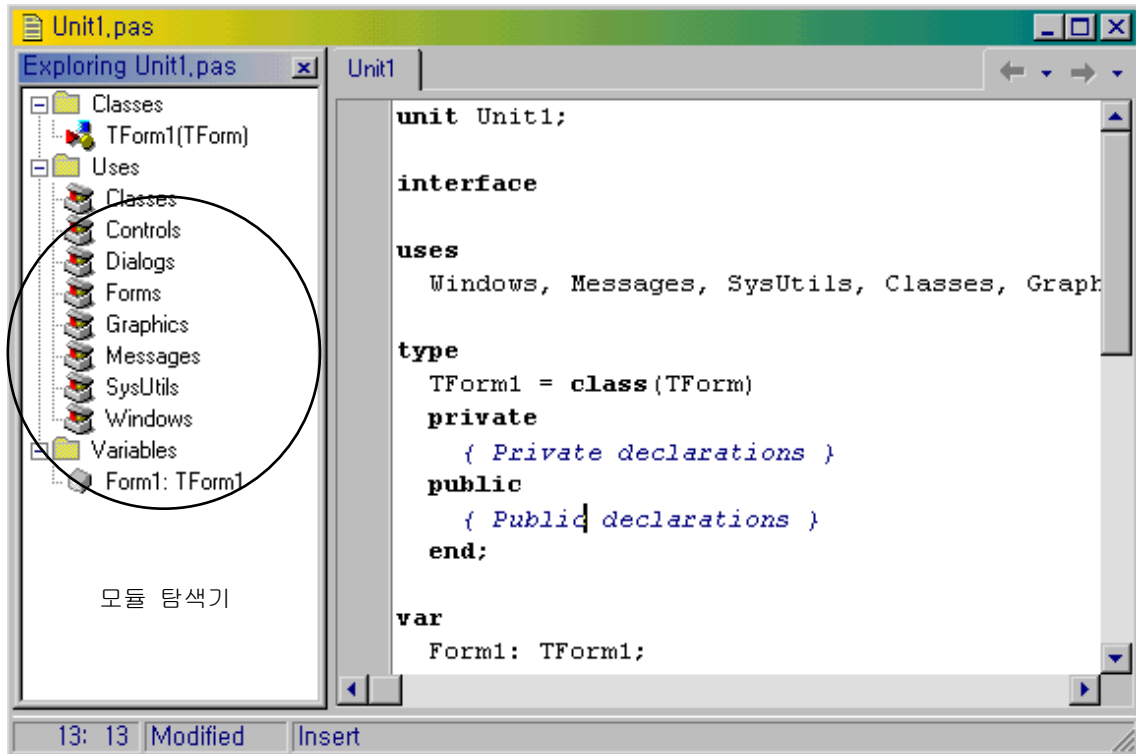
속성 탭에서는 객체의 프로퍼티를 살펴보거나 이를 마음대로 수정할 수 있는 기능을 제공하고 있다. 속성 이름의 옆에 + 기호가 나타나는 것은 그 아래에 하위 속성들이 있음을 나타낸다.

예를 들어, 폼을 선택한 다음에 오브젝트 인스펙터에서 Font 속성을 살펴보면 그 옆에 + 기호가 있음을 볼 수 있다. 그리고 Font 속성을 더블 클릭하거나 + 기호를 클릭하면 글꼴에 대한 Color, Height, Name 등의 하위 속성들이 나타나는 것을 볼 수 있다. 이러한 형식은 객체의 속성을 변경시키는데 매우 간단하면서 효과적인 방법을 제공하고 있다.

이벤트 탭에서는 개발자가 선택한 객체에 반응할 수 있는 이벤트를 선택하여, 이 이벤트가 발생할 때 어떤 동작을 취하라고 지정하는 것이 가능하다. 예를 들어, 어플리케이션에서 윈도우를 닫을 때에 어떤 작업을 실행하고자 한다면, 폼의 OnClose 이벤트를 사용하면 된다.

코드 에디터 (Code Editor)와 모듈 탐색기 (Module Explorer)

델파이 4 IDE 에서 가장 많은 변화가 있었던 부분을 꼽으라면 코드 에디터와 모듈 탐색기를 들 수 있다. 먼저 폼 뒤에 숨어 있는 코드 에디터를 살짝 클릭하면 다음과 같이 모듈 탐색기와 코드 에디터가 붙어서 나타나는 것을 볼 수 있다.



모듈 탐색기는 클래스의 생성을 자동화하고, 보다 쉽게 유닛 파일 들을 탐색할 수 있는 기능을 제공한다. 디폴트로 모듈 탐색기는 코드 에디터의 좌측에 도킹되어 있다.

모듈 탐색기를 닫으려면, 코드 에디터에서 떼어낸 후 우상부 코너를 클릭한다. 이를 다시 열고자 할 때에는 View|Module Explorer 메뉴를 선택하면 된다.

모듈 탐색기는 유닛에 정의된 모든 데이터 형과 클래스, 프로퍼티, 메소드, 전역 변수와 전역 루틴 등을 보여주는 트리 다이어그램 (tree diagram)을 포함하고 있다. 또한, 여기에는 uses 절에 들어있는 다른 유닛의 내용도 찾아볼 수 있게 되어 있다. 트리 뷰의 노드를 확장하거나 축소하며 뒤져볼 수 있다.

모듈 탐색기와 코드 에디터 사이를 토글하려면, Ctrl+Shift+E 키를 누르거나 또는 우측 버튼을 클릭하고 View Editor 메뉴를 선택한다.

모듈 탐색기는 점진적 검색(incremental searching)을 지원한다. 클래스, 프로퍼티, 메소드, 변수, 루틴 등을 검색하려면 단지 그 이름 만을 적어넣으면 된다. 모듈 탐색기에서 아이템

을 선택하면 커서가 코드 에디터에서 연관된 부분으로 이동해 가며, 코드 에디터에서 커서를 이동하면 모듈 탐색기에 적절한 아이템으로 하이라이트된 부분이 옮겨 진다.

또한, 모듈 탐색기의 클래스 완료(class completion), 모듈 탐색 (module navigation) 등의 기능을 이용하면 반복적인 코딩 작업을 자동화할 수 있다.

- 클래스 완료 (Class completion)

텔파이 4 의 클래스 완료 기능을 이용하면 새로운 클래스의 뼈대를 자동으로 만들어 낼 수 있으므로 코딩에 필요한 노력을 많이 줄일 수 있다.

유닛의 interface 섹션의 클래스 선언부에 커서를 위치시키고, Ctrl+ Shift+ C 키를 누른다. 이렇게 하면, 텔파이는 자동으로 프로퍼티에 해당되는 private read, write 필드에 해당되는 부분의 코드를 생성하고, implementation 섹션에 모든 클래스 메소드에 대한 뼈대 코드를 생성한다.

예를 들어, 다음의 코드를 interface 섹션에 작성했다고 하자.

```
type
  TMyButton = class(TButton)
    property Size: Integer;
    procedure DoSomething;
  end;
```

여기에 커서를 위치시키고, Ctrl+ Shift+ C 키를 누르면 interface 섹션에는 다음과 같은 코드가 생성된다.

```
type
  TMyButton = class(TButton)
    property Size: Integer read FSize write SetSize;
  private
    FButtonSize: Integer;
    procedure SetSize(const Value: Integer);
```

그리고, implementation 섹션에는 다음과 같은 코드가 생성된다.

```
{ TMyButton }
procedure TMyButton.DoSomething;
begin
```

```
end:

procedure TMyButton.SetSize(const Value: Integer);
begin
    FSize := Value;
end;
```

클래스 완료 기능은 implementation 섹션에 정의된 메소드에 대한 interface 선언부를 작성하게 할 수도 있다. 방법은 마찬가지로 implementation 섹션의 메소드 정의부에 커서를 위치시키고 Ctrl+ Shift+ C 키를 누르면 된다.

- 모듈 탐색 (Module navigation)

델파이 4 는 유닛 파일에서 Ctrl+ Shift 키와 각종 방향키를 누르면 쉽게 모듈 전체를 탐색할 수 있는 모듈 탐색 기능을 제공한다. 일단 유닛의 interface 섹션의 특정 메소드나 전역 프로시저의 prototype 에 커서를 위치시키고, Ctrl+ Shift+ Up 또는 Ctrl+ Shift+ Down 키를 누르면 그 프로시저나 함수의 구현 부분으로 이동한다. 마찬가지로 구현 부분에서 이들 키를 누르면 interface 섹션의 선언부로 이동하는 토크 키로 작동한다.

- 코드 브라우저 (Code browser)

코드 에디터에서 Ctrl 키를 누르면서 마우스를 특정 클래스, 변수, 프로퍼티, 메소드 등의 여러가지 identifier 이름 위로 지나가게 하면 마우스 포인터가 손모양으로 변하면서 포인터 위치의 identifier 가 하이라이트 되면서 밑줄이 쳐진다. 이를 클릭하면, 코드 에디터는 그 identifier 의 선언부로 위치를 옮겨간다. 이때 유닛의 interface 섹션에 선언된 메소드나 루틴의 선언부를 찾으려할 때에는 모듈 탐색 기능을 이용해서 Ctrl+ Shift+ Arrow 키를 이용하면 된다.

메뉴바 (Menubar)

델파이 환경에서 명령을 실행시키는 방법은 기본적으로 메뉴바의 메뉴를 이용하는 방법과 스피드바를 이용하는 방법, 그리고 마우스의 오른쪽 버튼을 클릭하면 나오는 스피드 메뉴를 선택하는 3 가지 방법이 있다.

여기에서 델파이 4 에서 제공되는 모든 메뉴에 대한 설명을 하는 것은 지면 낭비일 뿐이므로, 델파이의 도움말 파일을 참고하기 바람에 주요 메뉴에 대해서만 설명하도록 하겠다.

- File 메뉴

File 풀다운 메뉴에는 프로젝트와 소스 코드 파일에 대한 여러가지 작업 명령들을 포함하고 있다. 프로젝트와 관련이 있는 명령은 New, New Application, Open, Reopen, Save Project As, Save All, Close All 등이 있다. 이것들 외에도 프로젝트에 대해서는 Project 풀다운 메뉴가 특별히 따로 만들어져 있다. 소스코드 파일에 관계되는 명령은 New, New Form, Open, Reopen, Save, Save As, Close, Print 이다. 대부분의 명령들이 직관적으로 금방 알 수 있으므로 설명은 생략하고, 몇 가지 명령에 대한 것만 더 알아보도록 하자.

Reopen 메뉴 명령은 최근에 작업했던 프로젝트나 소스 코드 파일을 열 때 사용하는 것으로 오피스 등의 제품을 사용할 때 보는 history 와 비슷한 역할을 한다.

New 명령은 Object Repository 에 저장되어 있는 아이템을 재사용할 때 사용하는 메뉴로, New Items 대화 상자를 열게 된다. 여기에는 델파이의 각종 위저드를 불러내거나 새로운 어플리케이션의 형태와 기존의 폼을 상속하는 폼, 쓰레드나 DLL, 델파이 컴포넌트와 각종 액티브 X 와 관련된 아이템 들을 만들어낼 수 있다. Object Repository 에 대한 내용은 나중에 따로 설명하도록 하겠다.

Print 명령은 소스 코드나 폼을 인쇄할 수 있는 명령이다.

- Edit 메뉴

Edit 풀다운 메뉴에는 Undo 와 Redo, Cut, Copy, Paste 와 같이 전형적인 명령들과 폼이나 코드 에디터 윈도우를 위한 몇 가지 특별한 명령들이 포함되어 있다.

이런 명령들은 윈도우 어플리케이션에서 흔히 쓰이는 것들이기 때문에 몇 번만 직접 사용해 보면 어떤 기능을 하는지 쉽게 알 수 있을 것이다.

그 밖에 폼에 관련된 많은 명령어 들이 있다. 폼을 위한 명령어 들은 폼의 스피드 메뉴 (마우스 오른쪽 버튼을 클릭할 때 나타나는 팝업 메뉴)에도 나타나는 것들로 컨트롤을 그리드에 맞추어 정렬하게 하거나, 컨트롤의 앞뒤로 보내는 메뉴, 여러 개의 컨트롤 들을 정렬 하거나, 탭 순서를 설정하는 등의 메뉴가 포함된다. 이들 각각에 대한 설명은 도움말을 참고하기 바란다.

참고로 Lock Control 명령의 경우 스피드 메뉴에 나타나지 않는데, 이 명령은 폼 위에서 컴포넌트의 위치가 잘못해서 바뀌지 않도록 하는 역할을 한다. 예를 들어, 어떤 컴포넌트를 더블 클릭하려 했는데 잘못해서 그만 위치를 옮겨 버릴 수도 있는데, 이럴 경우 폼에는 Undo 기능이 없으므로 상당히 곤란할 경우가 있다. 이럴 때에는 폼을 일단 디자인해서 더 이상 바뀔 것이 없다면 그 다음에 컨트롤을 잠궈두면 이런 실수를 막을 수 있다.

- Search 메뉴

Search 메뉴에는 기본적인 Search(찾기)와 Replace(바꾸기) 명령과 여러 개의 파일에서 찾기를 할 수 있는 Find in Files 명령이 있다. 또한, 찾고자 하는 문자열을 하나씩 적어나가면서 매칭되는 소스 코드를 찾아주는 Incremental Search 명령도 있다.

Find in Files 명령은 찾기를 원하는 문자열을 라디오 버튼을 체크 함에 따라 프로젝트의 소스 파일들과 또는 모든 열려있는 파일들, 또는 특정 디렉토리 안의 모든 파일들 중에서 찾을 수 있도록 해 준다. 검색 결과는 코드 에디터 윈도우 밑에 있는 메시지 영역에 표시되며, 표시된 내용을 더블 클릭하면 그 파일의 내용이 있는 곳으로 코드 에디터가 옮겨가게 된다.

Incremental Search 기능은 상당히 편리하게 사용할 수 있는데, 이 명령의 단축키인 Ctrl+E 는 외워두었다가 써먹으면 좋을 것이다. 이 명령은 일단 단축키를 누르고 나서 찾고자 하는 문자열을 찍어나가면, 여기에 맞는 부분으로 계속 이동해 나간다.

Find Error 명령은 컴파일러 에러가 아닌 특정한 런타임 에러를 찾을 때 쓰이는 명령으로 어떤 단독 실행 프로그램을 실행시키는데 심각한 에러에 부딪히게 되면, 델파이는 어떤 내부 주소를 가리키는 숫자를 표시하게 된다. 즉, 컴파일된 코드의 논리적 주소를 표시하는 것인데, 이 값을 Find Error 대화 상자에 입력하면 델파이가 프로그램을 다시 컴파일해서 지정한 주소를 찾아준다. 만약 이 주소를 찾게 되면, 델파이 코드 에디터에 해당 소스 코드 라인을 찾아서 표시한다. 그런데, 에러가 소스 코드에 있는 것이 아니라 라이브러리나 시스템 코드의 문제로 발생하는 경우도 있다. 이런 경우에는 Find Error 명령으로 오류를 발견할 수 없게 된다.

Browse Symbol 명령은 컴파일된 프로그램에서 정의한 모든 심볼들을 살펴볼 수 있도록 하는 명령으로 Object Browser 를 불러서 이를 표시해 준다.

● View 메뉴

View 풀다운 메뉴에는 델파이 환경의 각 윈도우들을 표시하기 위해 사용된다. 델파이 환경의 윈도우라면 프로젝트 관리자(Project Manager), 정지점(Breakpoint) 리스트, 모듈 탐색기와 컴포넌트 리스트 등 여러가지가 있게 되는데, 이들 각각은 직접 실행해보면 어떤 윈도우를 가리키는 것인지 알 수 있을 것이다.

프로젝트 관리자와 정지점에 대한 부분은 델파이 4 에서 많이 바뀐 부분의 하나인데 프로젝트 관리자는 조금 뒤에 설명할 것이다.

View 메뉴에는 이렇게 서로 다른 윈도우를 표시하기 위한 명령들 이외에 여러가지 명령들이 포함되어 있다. Toggle Form/Unit 메뉴는 작업하고 있는 폼과 그 폼의 소스 코드 사이를 토글해주는 명령으로, 상당히 자주 쓰이게 되므로 이 명령의 단축키인 F12 는 외워두기 바란다. 또한, 편리한 명령으로는 New Edit Window 가 있는데, 이 명령을 선택하면 델파이 두번째 코드 에디터 윈도우를 열어 준다. 에디터 화면을 두 개 만들어 놓으면 서로

다른 파일을 각각 보이게 해 놓을 수 있고, 한 파일의 서로 다른 부분을 보이게 할 수도 있기 때문에 대단히 편리하다.

마지막 Toolbars 메뉴는 델파이 4 에서 추가된 명령으로 서브 메뉴를 살펴보면, 델파이 4 의 여러 툴바를 보이게 하거나, 숨기게 할 수 있다.

● Project 메뉴

Project 폴다운 메뉴는 프로젝트를 관리하고 컴파일하는 명령들을 가지고 있다. Add to Project 와 Remove from Project 명령은 폼이나 파스칼 소스 코드 유닛을 프로젝트에 추가하거나 제거할 때 사용된다.

Import Type Library 명령은 타입 라이브러리를 읽어올 때 사용하는 것으로 이 책의 후반부에서 자세히 다루게 되므로 설명을 생략하겠다. Add to Repository 는 폼을 Object Repository 에 등록하고자 할 때 사용하는 명령으로 자주 사용되는 폼의 모양을 디자인하고, 이를 계속 재사용할 때 유용하다. View Source 명령은 프로젝트의 소스를 보여준다.

델파이 4 에서는 프로젝트와 관련된 부분이 많이 향상되었으며, 여기에 관련된 명령들이 Project 폴다운 메뉴에 많이 추가 되었다. 이들을 설명하기 위해서 프로젝트 관리자의 변화된 부분에 대해서 조금 더 알아보도록 하자.

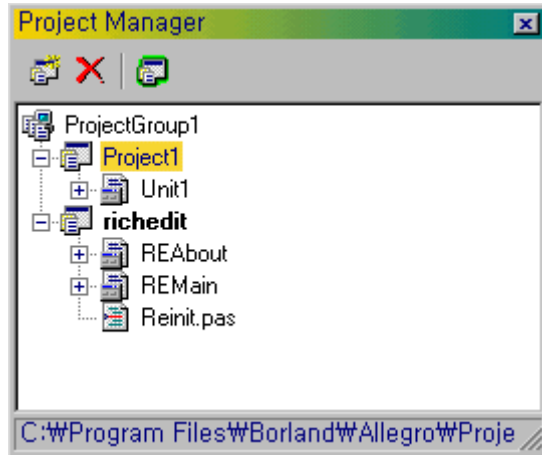
델파이 4 의 프로젝트 관리자는 프로젝트 그룹에서 여러 개의 프로젝트 들을 쉽게 관리할 수 있게 해 준다. 프로젝트 그룹은 상호연관성이 있는 프로젝트 들 (DLL 과 이를 사용하는 어플리케이션, 멀티-tiered 어플리케이션에서 각각의 tier 등)을 유기적으로 관리하는데 편리하게 사용할 수 있다.

프로젝트 관리자를 이용하면 연관된 모든 프로젝트의 파일 들을 볼 수 있으며, 이들을 디스플레이 하고, 파일을 추가 삭제하고 컴파일하는 등의 여러가지 조작을 할 수 있다. 또한, 이들을 한꺼번에 컴파일 할 수도 있다.

프로젝트 관리자의 메인 화면을 살펴보면 프로젝트 그룹이나 프로젝트에 속해있는 모든 파일 들을 트리의 형태로 관찰할 수 있다. 여기에서 트리의 루트는 프로젝트 그룹이며, 프로젝트 그룹에는 각각의 프로젝트를 나타내는 아이콘을 포함하고 있다. 프로젝트의 종류는 DLL, EXE, 패키지 또는 리소스 등일 수 있다. 프로젝트가 프로젝트 그룹의 일부이면, 프로젝트 관리자는 프로젝트 그룹에 있는 모든 프로젝트 들에 대한 정보를 제공한다. 우측의 파일 뷰에는 프로젝트 내의 모든 소스 파일(.pas, .rc 파일 등)과 이진 객체 파일 (.res, .lib, .obj 파일 등)을 들을 보여 준다.

각각의 프로젝트 파일은 .dpr 확장자를 가지고 있다. 프로젝트 관리자를 이용해서 파일을 추가, 제거하면 델파이는 프로젝트 파일을 자동으로 업데이트 해 준다. 프로젝트 그룹 파일은 .bpg 확장자를 가지고 있으며, 프로젝트 그룹에 프로젝트를 추가, 삭제 할 때마다 내용이 바뀌게 된다.

델파이 4 의 프로젝트 관리자의 실제 모습은 다음과 같다.



Create New Target 과 Open New Target 명령은 델파이 4 에서 새롭게 추가된 것으로 새로운 어플리케이션이나 DLL, 패키지 등의 아이템을 프로젝트 그룹에 추가할 때 사용한다. 또한, 델파이 4 에서는 복수 프로젝트를 관리할 수 있는 프로젝트 관리자에 부합하여, 컴파일과 관련된 명령 들이 많이 추가되었다. Compile, Build, Syntax Check 등의 기존 명령 말고도 Make 명령이 추가 되었는데, 이 명령들의 뒤에는 대상이 되는 프로젝트의 이름이 같이 디스플레이된다. 또한, 프로젝트 그룹에 있는 모든 프로젝트를 한꺼번에 컴파일할 때 사용할 수 있는 Complier All Projects, Build All Projects 명령이 추가되었다.

Information 명령도 프로젝트 그룹안에서 컴파일된 특정 프로젝트에 대한 정보를 보여주어야 하므로, Information for [프로젝트 이름]의 형태로 바뀌었다.

Web Deploy, Web Deploy options 명령은 액티브 X 폼과 컨트롤에 대한 것으로 이 책의 후반부에서 자세히 다루게 될 것이다.

프로젝트 메뉴의 가장 마지막 명령은 Options 메뉴이다. 여기에서는 컴파일러와 링커 옵션, 어플리케이션 객체의 여러가지 옵션을 설정할 수 있다.

- Run 메뉴

Run 메뉴는 주로 디버깅에 관련된 내용을 많이 담고 있다. 델파이 환경에서 Run 명령을 선택하면 작성된 어플리케이션은 델파이의 내장 디버거 내에서 실행된다. 물론 이 기능을 환경 옵션에서 해제할 수도 있다. 어쨌든 Run 명령은 델파이를 사용할 때 가장 자주 사용하게 되는 명령이므로 F9 단축키는 외워두는 것이 좋을 것이다.

Parameters 명령은 커맨드 라인을 실행시키려고 하는 프로그램에 전달하고, DLL 을 디버그 할 때에는 실행 파일의 이름을 제공하기 위해 파라미터를 설정할 수 있게 한다. 이 명령도 델파이 4 에서 향상된 것 중에 하나인데, 기존의 파라미터 설정 탭에 원격으로 디버깅을 할 수 있게 하기 위해 호스트 어플리케이션과 원격지 패스 등을 설정할 수 있도록 변경되었다. Run 풀다운 메뉴에서 디버깅에 관련된 명령 이외에는 액티브 X 개발에 관련된 명령이 몇

가지 있다. Register ActiveX Server 와 UnRegister ActiveX Server 명령은 현재 프로젝트에 의해 정의되어 있는 액티브 X 컨트롤에 대한 윈도우 레지스트리 정보를 추가하거나 삭제하는 역할을 한다. 또한, Install MTS Objects 메뉴를 통해 마이크로소프트 트랜잭션 서버를 지원하는 객체를 설치할 수 있다.

- Component 메뉴

Component 메뉴의 명령들은 컴포넌트를 작성하고 이것들을 패키지에 넣거나 패키지를 델파이에 설치하는데 주로 사용된다. New Components 명령은 간단한 컴포넌트 위저드를 호출하여 컴포넌트를 새로 작성하는데 도움을 주며, Install Components, Import ActiveX Library, Install Packages 명령은 새로운 델파이 컴포넌트, 패키지 또는 액티브 X 컨트롤을 환경에 추가하여 사용할 수 있도록 해준다. 이들에 대한 더욱 자세한 내용은 제 4 부에서 다루게 된다.

Create Component Template 명령은 하나 이상의 컴포넌트를 폼에서 선택하고 이 명령을 호출하면 새로운 컴포넌트의 이름, 팔레트에서의 페이지, 아이콘을 입력하는 대화 상자가 나타나는데 이를 이용해 새로운 컴포넌트 템플릿을 구성할 수 있게 해준다.

- Database 메뉴

Database 풀다운 메뉴에는 Database Form Wizard, SQL Explorer, SQL Monitor 등의 데이터베이스 관련 도구를 호출할 수 있는 메뉴가 모여 있다. 이들을 선택하면 데이터베이스 도구들이 실행되는데, 여기에 대한 자세한 내용은 13 장의 내용을 참고하기 바란다.

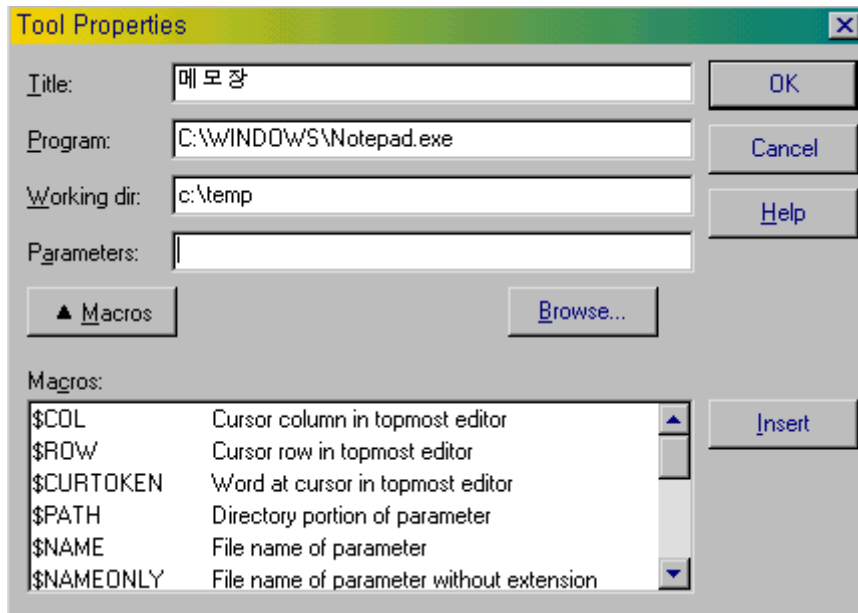
- Tools 메뉴

Tools 풀다운 메뉴는 외부 프로그램과 툴들을 실행시키기 좋게 모아놓은 것과 델파이 개발 환경의 옵션을 설정하는 명령, Object Repository 의 초기화를 위한 명령이 포함되어 있다.

Environment Options 대화 상자에는 포괄적인 환경 설정, 패키지와 라이브러리 설정, 많은 에디터 옵션, 컴포넌트 팔레트 설정, Object Browser 설정, 코드 인사이트 설정 등을 할 수 있는 많은 페이지가 있다. 이들 각각에 대한 것은 도움말을 참고하기 바란다.

또한, Configure Tools 명령을 이용하면 자신이 자주 쓰는 외부의 도구를 등록했다가 쉽게 불러낼 수 있다. 여기서 간단히 메모장(Notepad)을 등록해 보도록 하자.

Configure Tools 명령을 선택하면 대화 상자가 나타나는데, 여기서 Add 버튼을 클릭하면 등록할 도구의 속성을 설정할 수 있는 대화 상자가 보인다. 속성을 다음과 같이 설정하도록 하자.

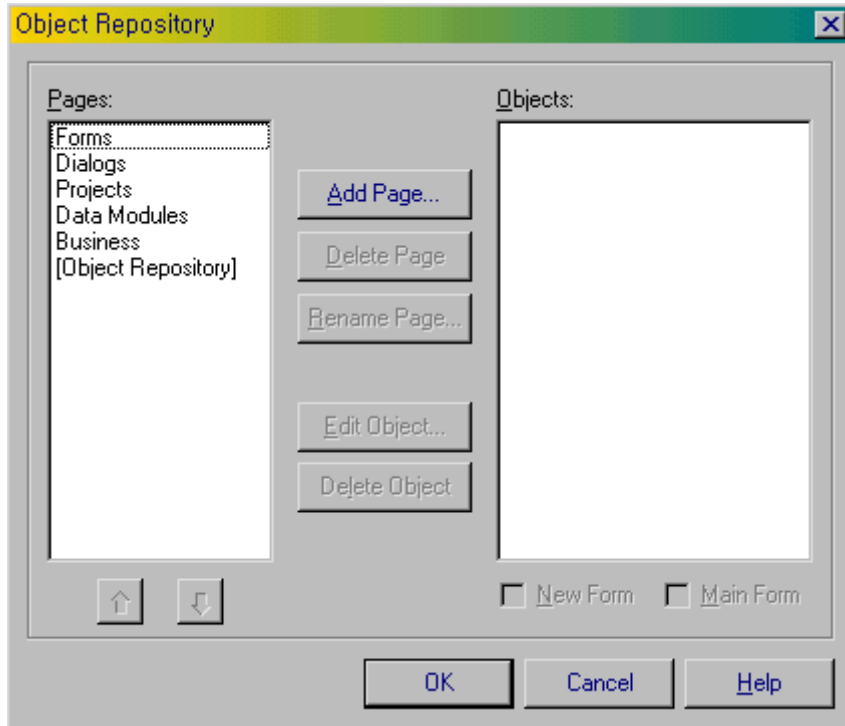


Browse 버튼을 클릭해서 추가하고자 하는 도구의 실행 파일을 선택하고, 이 도구가 메뉴에 나타나게 될 타이틀과 작업 디렉토리를 설정하고 OK 를 선택하면 Tools 메뉴에 메모장이 추가되며, 이를 선택하면 메모장이 실행된다.

- 그 밖에도 Workgroup 과 Help 메뉴가 있는데, Workgroup 메뉴에는 델파이의 버전 컨트롤 프로그램인 PVCS 를 실행시키는 명령이 포함되어 있으며 Help 메뉴에서는 도움 말을 불러올 수 있다.

객체 저장소 (Object Repository)

델파이 1.0 의 Gallery 는 템플릿의 저장과 폼 위저드의 기능으로 사용되었다. Gallery 는 델파이 2.0 이후부터 객체 저장소로 바뀌게 되었다. Tools 메뉴에서 Repository 를 선택하면 다음과 같은 대화상자가 나타나게 된다.



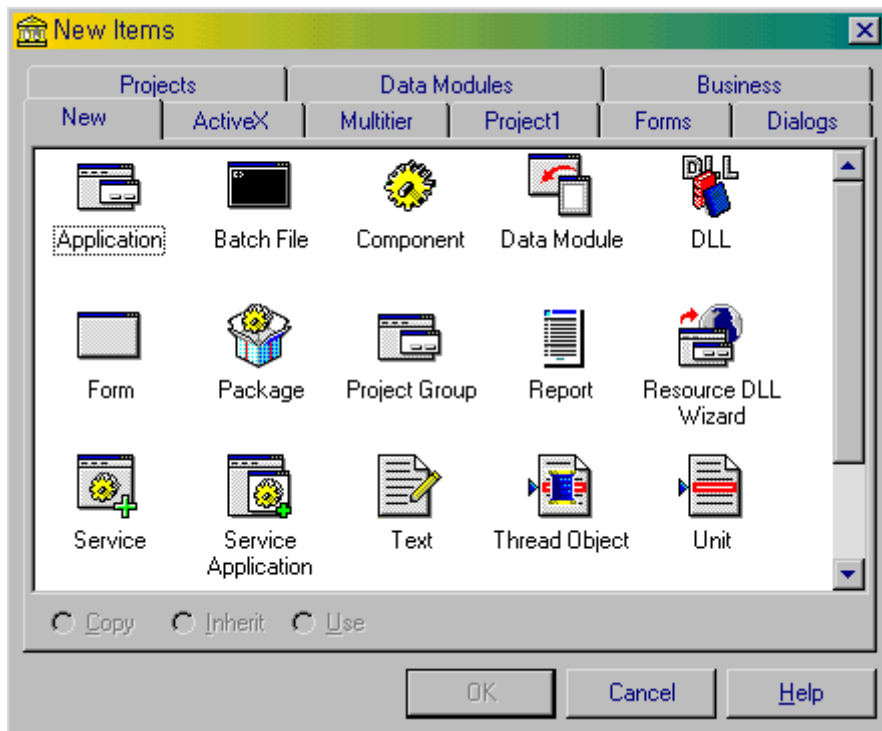
페이지 리스트 박스에는 Forms, Dialogs, Projects, Data Modules, Object Repository 의 5 가지 선택 항목이 나타난다. 이 대화상자를 사용하면 선택 항목을 편집하거나 새로운 폼이나 프로젝트를 생성할 때 사용되는 기본 값을 변경시킬 수도 있다. 여기서는 개발자 자신의 페이지를 생성하거나 Object Repository 대화 상자에서 저장소(Repository)로 추가시킬 수 있다.

이러한 객체 저장소의 기능은 델파이에서 코드의 재사용 기능을 더욱 강력하게 만들어 주고 있다. 대부분의 가장 일반적으로 사용되는 기능들은 이미 기본적으로 제공되고 있지만, 개발자 자신의 것들을 생성하여 추가하면 그 활용도는 훨씬 높을 것이다.

델파이에서 File|New 메뉴를 선택했을 경우에 델파이는 객체 저장소를 연다. 객체 저장소의 여러 페이지 중에서 New 페이지와 ActiveX 페이지에는 여러 가지 형태의 새로운 아이템을 만들 수 있도록 한다.

다음 그림의 객체 저장소 대화 상자 밑에 있는 라디오 버튼을 눌러서 기존의 아이템을 복사할 것인지, 아니면 상속할 것인지, 그대로 사용할 것인지를 지정하게 되는데 위저드의 경우처럼 이런 내용을 선택할 수 없는 경우에는 라디오 버튼의 기능이 비활성화 상태로 나타난다.

객체 저장소에 제공되는 기본 아이템에 대해 모두 설명하는 것은 지면 관계상 생략하도록 하고, 델파이 4 에서 새롭게 제공되는 것이 어떤 것들이 있는지 알아보도록 하자. 델파이 4 에서는 객체 저장소에 새로운 위저드를 많이 지원하는데, 여기에는 다음과 같은 것들이 있다. 이들 각각에 대한 자세한 내용은 해당되는 장을 참고하기 바란다.



- CORBA Data Module Wizard:
데이터 모듈이 CORBA 를 지원하도록 해주는 위저드이다.
- MTS Data Module Wizard:
데이터 모듈이 마이크로소프트 트랜잭션 서버를 지원하도록 하는 위저드이다.
- Project Group: 새로운 프로젝트 그룹을 생성한다.
- Resource DLL Wizard:
세계화를 위해서 필요한 위저드로 문자열을 여러 나라의 언어로 지정할 수 있는 쉬운 방법을 제공하며, 하나의 프로젝트를 여러 나라 버전으로 저장 관리할 수 있게 해준다.
- Service Wizard: 윈도우 NT 서비스를 생성해주는 위저드이다.
- Service Application Wizard: 윈도우 NT 서비스 어플리케이션을 생성해준다.
- COM object Wizard: COM 객체를 생성해주는 위저드이다.
- MTS Automation Object Wizard: MTS 자동화 객체를 자동으로 생성해준다.

정 리

이번 장에서는 델파이 4 의 IDE 에 대해서 간단하게 알아보았다. 더욱 자세하게 설명할 수도 있겠으나, 이 책에서는 주로 다른 책에서는 다루지 못한 여러 가지 테크닉 들을 많이 소개하려고 하기 때문에 단순히 도움말을 찾아보면 알 수 있는 내용들은 되도록 생략하였다.

더 자세한 사항은 도움말을 직접 참고하기 바란다.

텔과이 4 는 그동안 개발자들이 불편해하던 많은 부분을 해결한 멋진 IDE 를 제공하고 있다. 복수로 프로젝트를 관리할 수 있게 되었고, 모듈 탐색기를 통해 interface 섹션과 implementation 섹션 사이의 이동과 코드 에디터 내부에서의 탐색 기능의 효율을 높였다.