



표준 파일^{File} 다이얼로그^{Dialog}들

파일을 손쉽게 열거나, 만들고 저장할 수 있습니다.

앞의 모듈에서, 파일을 가지고 작업을 하는 가장 전형적이고 일반적인 방식으로 파일을 다루었다. 하지만, 델파이에서는 읽기나 쓰기를 할 파일을 손쉽게 표준 파일 열기와 표준 파일 저장 다이얼로그 창 즉, 파일^{File} 다이얼로그^{Dialog} 창들을 통해 선택할 수 있다.

표준 파일 다이얼로그 창 컴포넌트들은 툴 팔레트의 *Dialogs*^[다이얼로그] 그룹에 들어있다. 이것을 폼에 올리면 사용할 수 있게 된다. 폼에 올려놓았다고 해서 프로그램이 실행되는 동안 (런타임) 에 폼 안에 표시되는 것은 아니다.

OpenDialog^{열기 대화창} 컴포넌트(파일 열기 아이콘 모양)를 사용하면 파일에서 정보를 읽을 수 있다. *SaveDialog*^{저장 대화창} 컴포넌트(디스크 모양 아이콘)를 사용하여 기존 파일 또는 새 파일에 쓰기를 해보자. 새 파일의 이름은 키보드로 입력한다.

이 컴포넌트들에는 *FileName* 프로퍼티, 그리고 *Execute*^{실행}라는 불리언 메소드가 있다.

파일 다이얼로그 창을 가지고 작업하는 알고리즘은 다음과 같다.

```

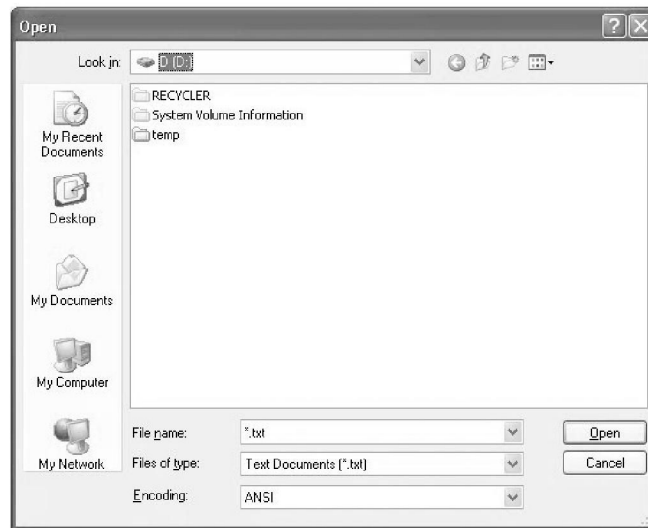
var
    Okf1, Okf2: Boolean;
    Stf1, Stf2: String;
begin
    //읽기를 할 파일을 선택한다. 잘 선택되었는가?
    //만약 읽기를 할 파일이 잘 선택되었으면
    begin
        //다이얼로그에서 파일 이름을 찾는다.
        //파일 변수를 파일에 할당한다.
        //읽기를 위해 파일을 연다.
        <파일에 있는 정보를 처리한다: 읽기, 처리하기 등>
        //파일을 닫는다
    end;
    //쓰기를 할 파일을 선택한다. 잘 선택되었는가?
    //만약 쓰기를 할 파일이 잘 선택되었으면
    begin
        //다이얼로그에서 파일 이름을 찾는다.
        //파일 변수를 파일에 할당한다.
        //쓰기를 위해 파일을 연다.
        < 파일에 정보를 쓴다 >
        //파일을 닫는다.
    end;
end;
end;

```

프로그램의 첫 번째 줄은 다음과 같다.

```
Okf1:= OpenDialogEx1.Execute;
```

프로그램을 실행하면, 다음과 같이 표준 *OpenDialog*^{열기 대화창} 다이얼로그 창이 나타난다. 이 다이얼로그 창에서 사용할 파일을 찾아서 선택한다.



Open 열기 버튼을 클릭하면 읽기 위해 선택한 파일이 열린다. 이때 *Execute* 메소드는 열린 파일의 이름을 `OpenDialogEx1` 컴포넌트의 *FileName* 프로퍼티에 넣고 `True`를 반환한다.

프로그램을 계속해보자.

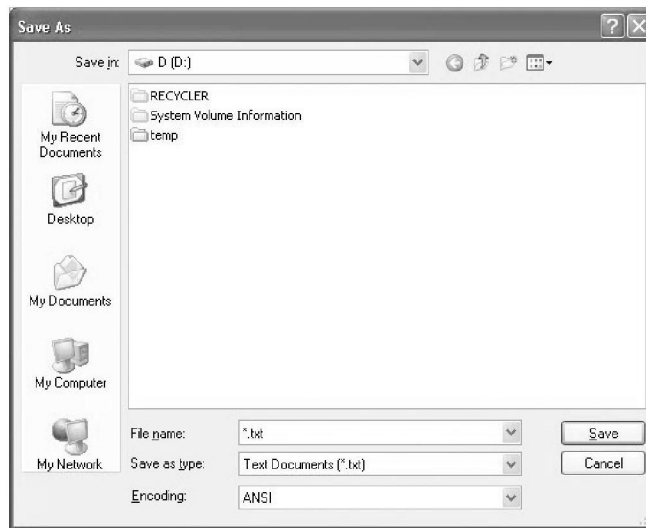
```
if Okf1 then
//만약 읽기를 할 파일이 잘 선택되었으면
begin
    Stf1:= OpenDialogEx2.FileName;
    //다이얼로그에 있는 파일이름을 잡아서 Stf1에 넣는다.
    AssignFile(f, Stf1); //파일 변수를 파일과 연결한다.
    Reset(f); //읽기를 하기 위해 파일을 연다.
    .....; //파일 안에 있는 데이터를 처리한다.
    Closefile(f); //파일을 닫는다.
end;
```

위 예문과 같이, 파일을 열었으면 이제, “.....” 자리에, 우리가 문제를 해결하기 위해 필요한 연산자들을 넣어서 파일 안에 있는 데이터를 처리한다. 파일 안의 정보 읽기, 읽은 데이터 처리하기 등 이미 앞의 모듈에서 해본 것들이다.

파일에 정보를 쓰는 작업을 하려면 아래와 같이 한다.

```
Okf2:=SaveDialogEx2.Execute;
//쓰기를 할 파일을 선택한다. 잘 선택되었는가?
```

이 명령이 실행되면, 이 표준 **파일 저장**^{Save} 다이얼로그가 나타난다. 이 창에서 파일을 찾아서 선택하거나 **File Name**^{파일 이름}을 넣는 텍스트박스에 새로 만들 파일의 이름을 적어 넣으면 된다. 지정된 파일의 이름은 `SaveDialogEx1.FileName` 프로퍼티에 들어간다.



Cancel^{취소} 버튼을 클릭하면 이 동작이 취소된다.

파일에 정보를 기록하는 코드는 다음과 같다.

```
Okf2:=SaveDialogEx2.Execute;
//쓰기를 할 파일을 선택한다. 잘 선택되었는가?
if Okf2 then
//만약 쓰기를 할 파일이 잘 선택되었으면
begin
    Stf2:= SaveDialogEx2.FileName;
    //다이얼로그에서 파일이름을 잡아서 stf2에 넣는다.
```

```

AssignFile(g, Stf2); //파일 변수를 파일과 연결한다.
Rewrite(g); //쓰기를 하기 위해 파일을 연다.
..... .; //파일에 데이터를 써넣는다, 데이터를 처리한다 등.
Closefile(g); //파일을 닫는다.
end;

```

이제 여러분은 다이얼로그 모드에서 파일을 가지고 작업하는 방법을 알게 되었다.

실습

Exercise 1.

메모장^{Notepad}에서 텍스트 파일을 하나 만들어 저장한 후에, 표준 파일 다이얼로그 들을 사용하여 다음과 같은 프로그램을 작성해보자. 만들어 놓은 텍스트 파일에서 문자 수가 짝수인 모든 문자열을 복사하여 File2에 넣는다. 문자 수가 홀수인 모든 문자열을 복사하여 File3에 넣는다. (File2와 File3는 프로그램에서 만든다)

Exercise 2.

메모장^{Notepad}에서 텍스트 파일을 하나 만들고 저장한다. 이 파일에는 각 줄마다 정수들이 들어있다. 이 숫자들 사이에는 공백이 하나 이상 있어서 서로 구분된다. 표준 파일 다이얼로그들을 사용하여, 이 숫자들 전체의 합계를 계산하여 새 파일에 기록하는 프로그램을 적어보자.

Exercise 3.

메모장^{Notepad}에서 텍스트 파일을 하나 만들고 저장한다. 이 파일에는 각 줄마다 무작위로 정수들과 단어들이 들어 있다. 이것들 사이에는 공백이 하나 이상 있어서 서로 구분된다. 표준 파일 다이얼로그들을 사용하여, 이 텍스트 파일에 있는 숫자들 전체의 합계를 계산하여 새 파일에 기록하는 프로그램을 적어보자.