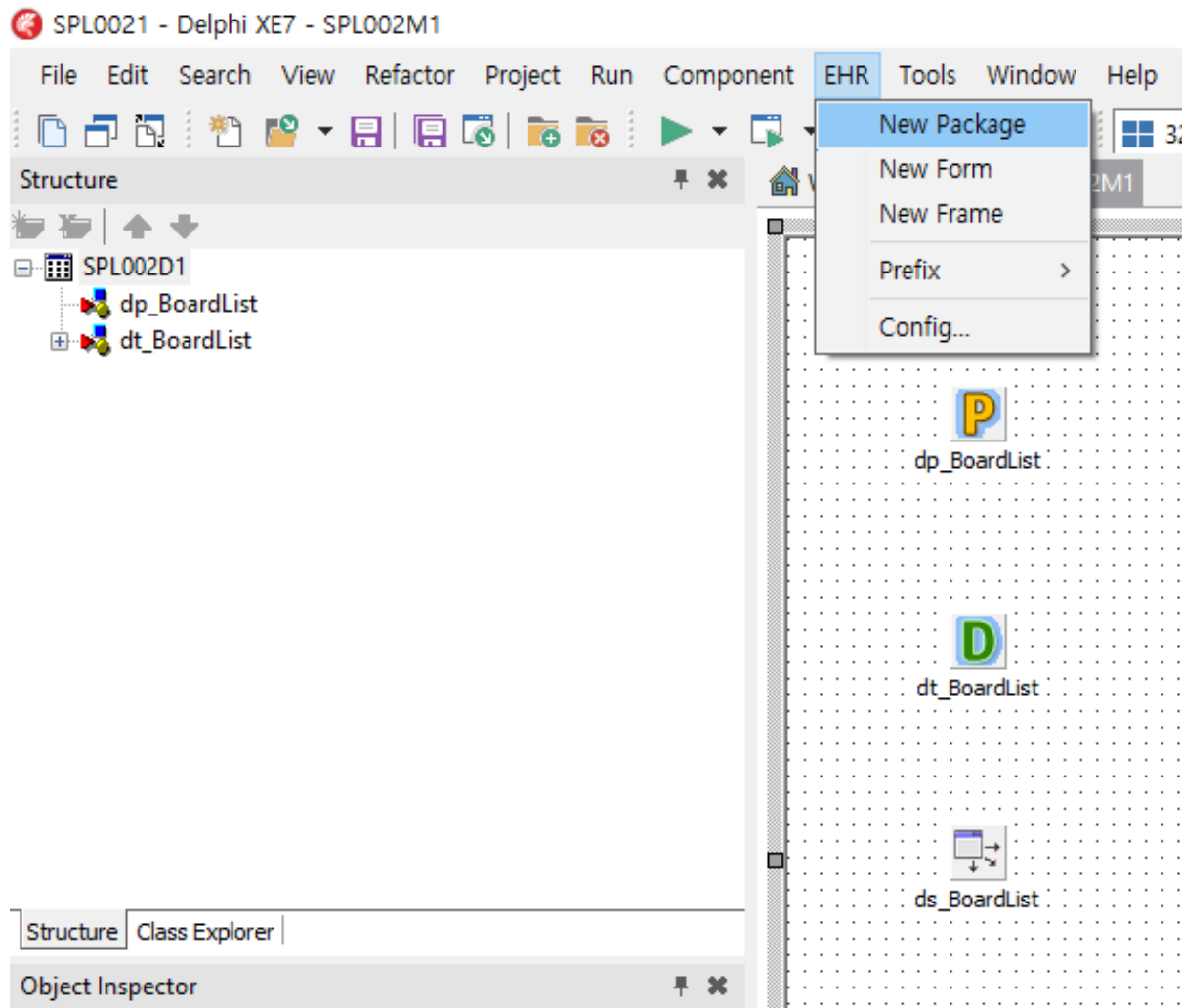


클라이언트 CRUD 개발 가이드

1. 패키지 생성

EHR - New Package 를 선택하여 패키지를 생성합니다.



업무별 패키지이름(SPL002L1), 디렉토리, 폼 캡션(SPL002F1) 등을 입력하고 생성을 클릭합니다.

New Package

Package Name

SPL002L1

Directory

D:\WNEXMED_EHR\DEV\SRC\WSAMPLE

Full Path

D:\WNEXMED_EHR\DEV\SRC\WSAMPLE\SPWSPL

Form Caption

SPL002F1

Description

게시판 샘플

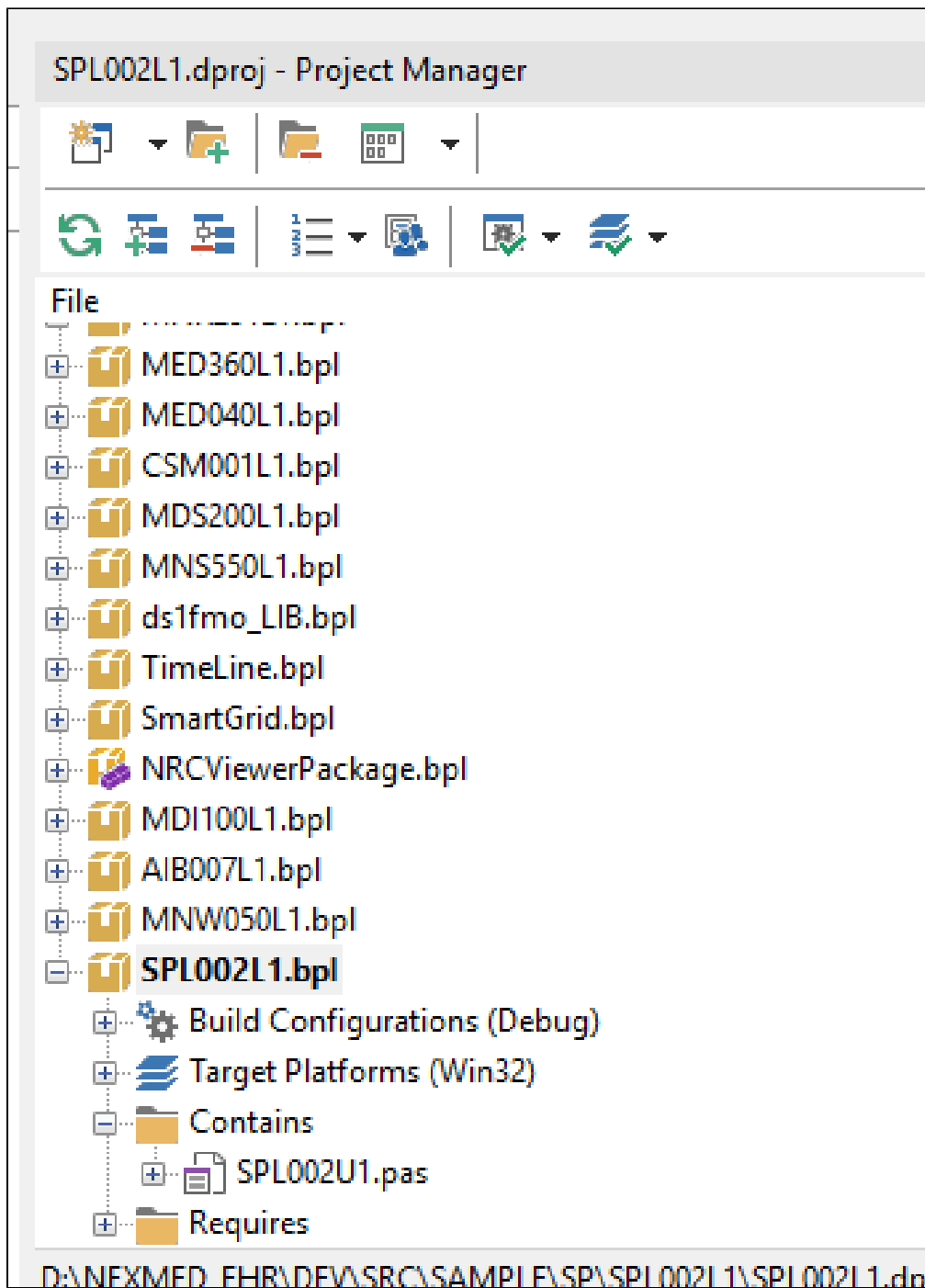
Category

서브시스템명 :
업무분류명(대) :
업무분류명(중) :

생성

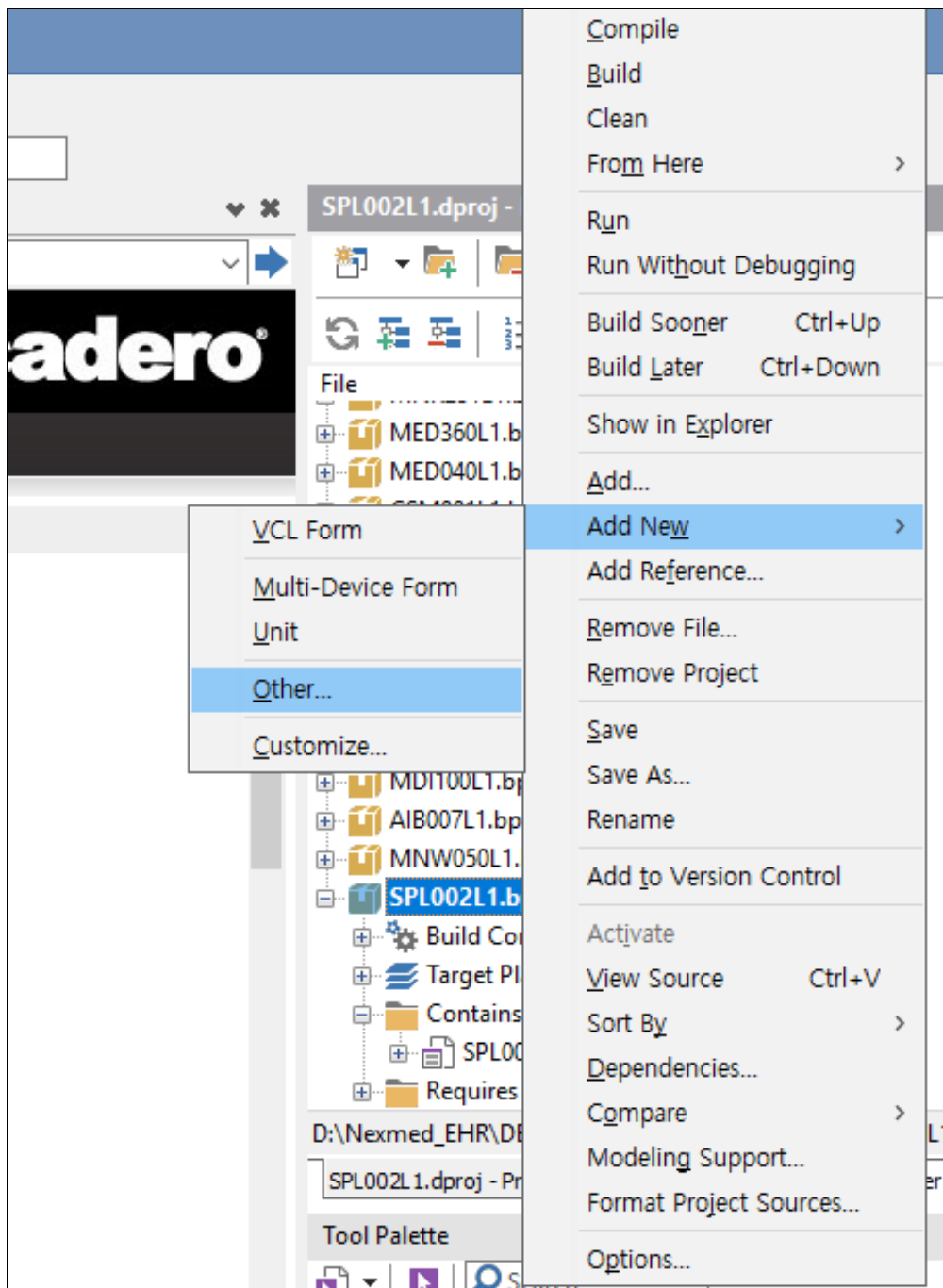
취소

Project Manager에서 패키지가 생성된 것을 확인합니다.

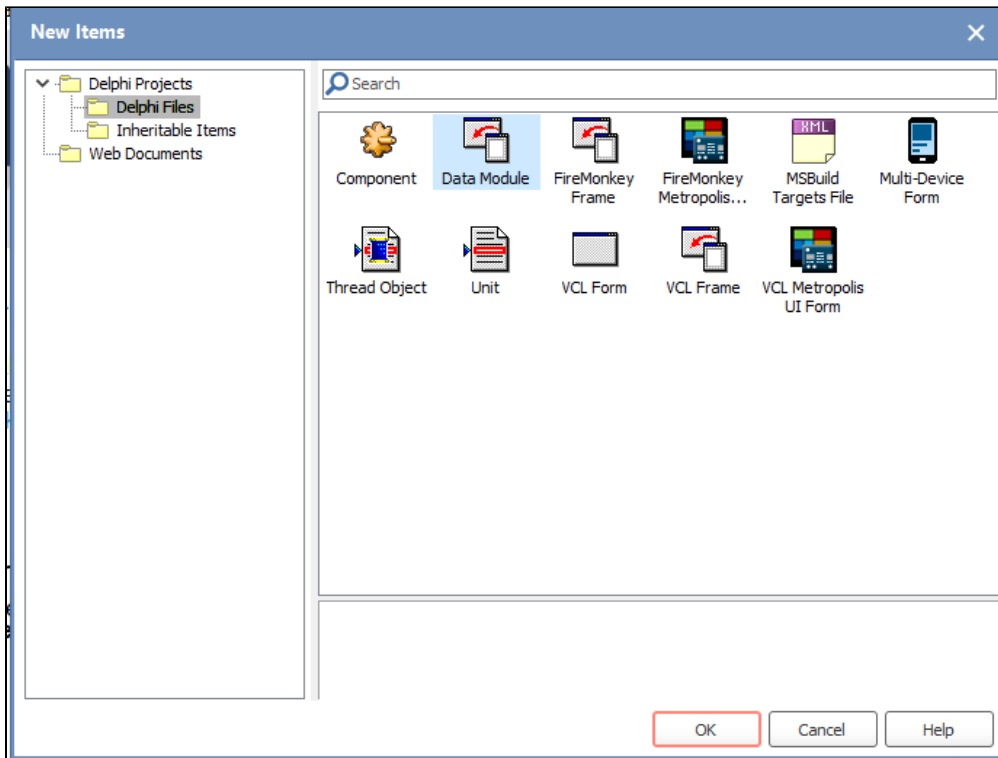


2. 데이터 모듈 생성

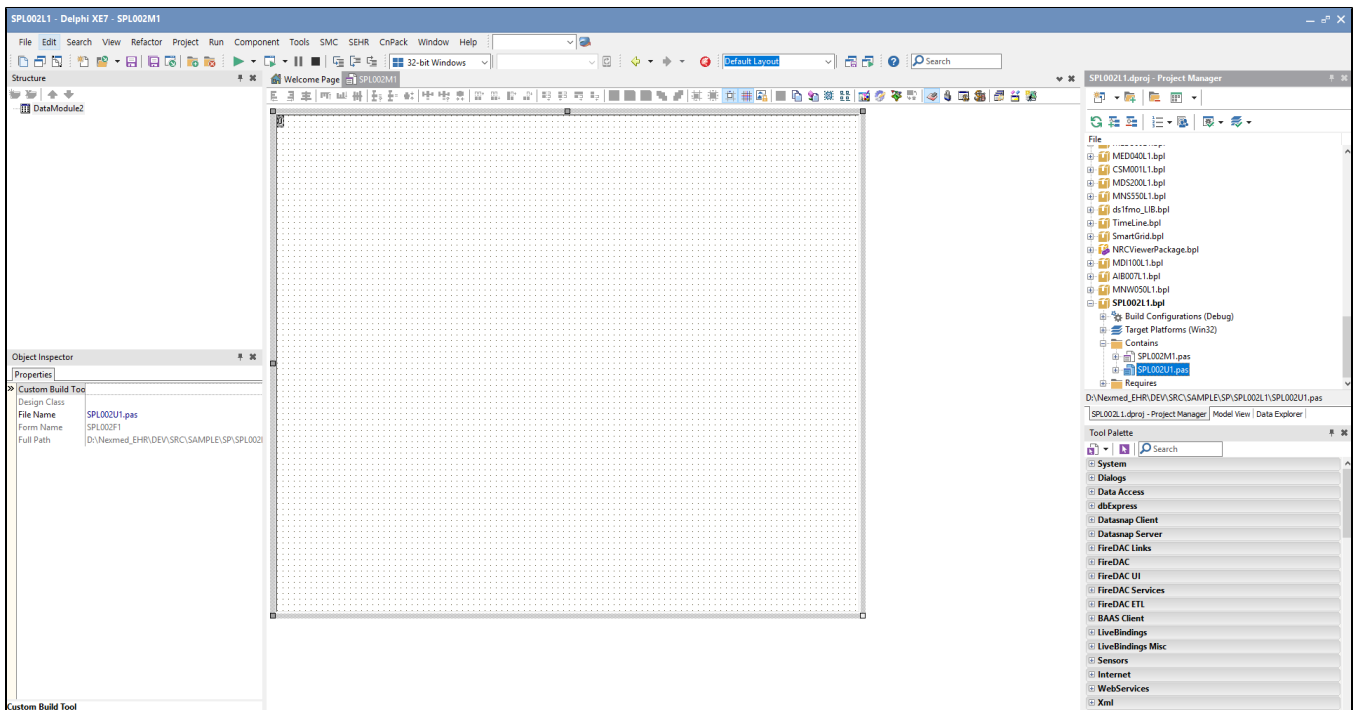
개발중인 프로젝트를 선택 후 - Add New - Other를 선택합니다.



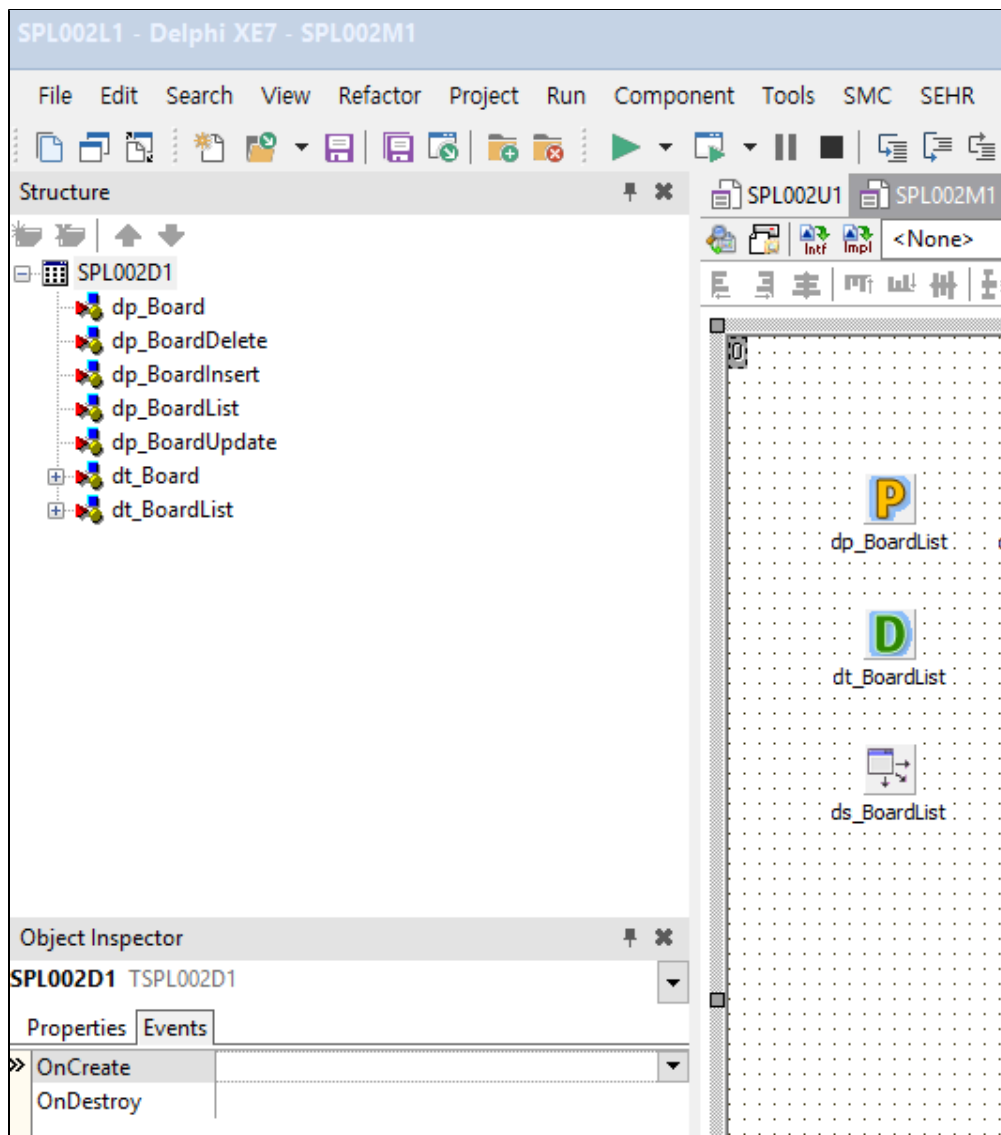
Dephi Projects - Delphi Files - Data Module을 선택 후 OK를 클릭합니다.



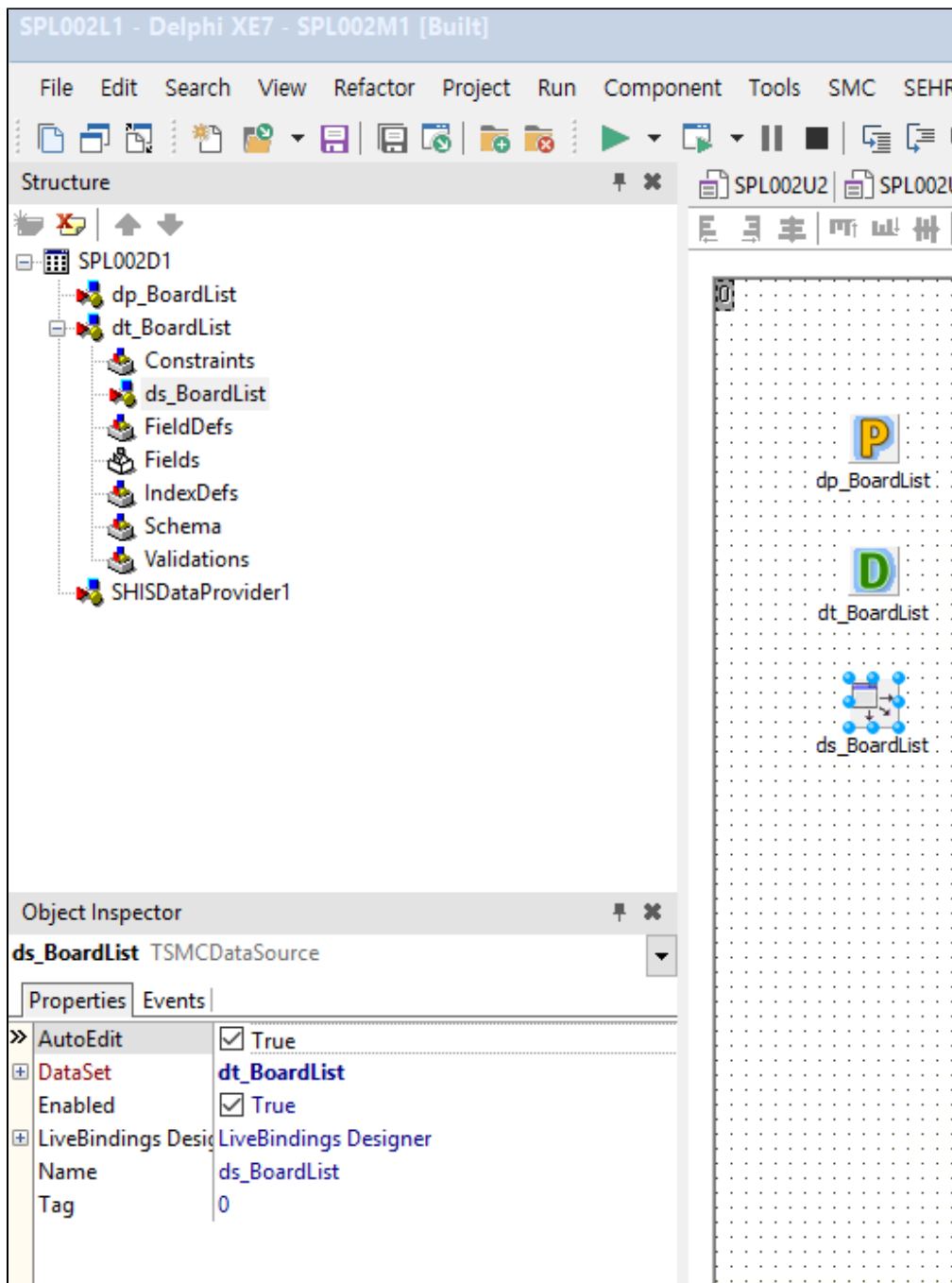
데이터 모듈이 생성된 것을 확인 후 이름을 바꿉니다.(파일명 : XXX00XM1.pas, 데이터 모듈 이름 : XXX00XD1)



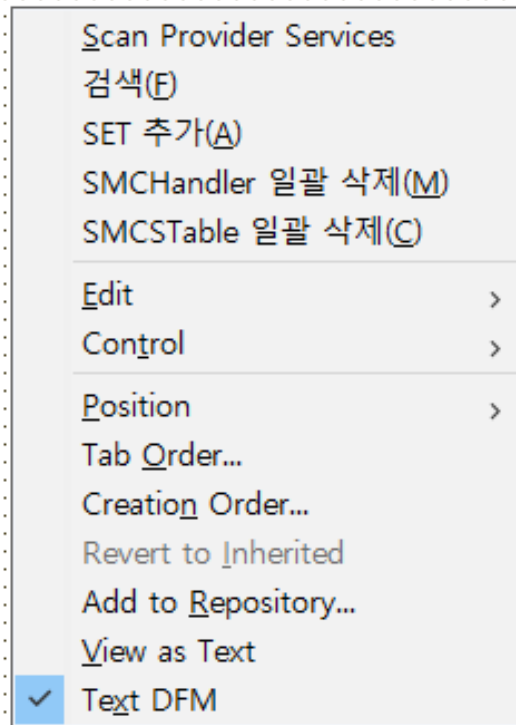
[[Tool Palette](#)]에서 TSHISDataProvider, TSHISDataSet, TSMCDataSource를 추가합니다.



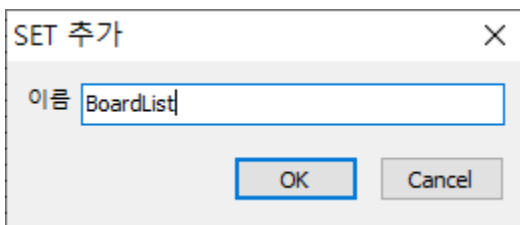
DataSource의 DataSet속성에 위에서 추가한 데이터셋을 연결합니다.



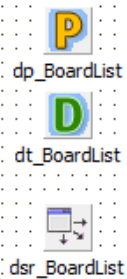
※ 아래와 같이 데이터 모듈 선택 후 마우스 우클릭 > SET 추가(A)를 선택하고, 이름을 입력하면 해당 이름의 SET이 한번에 추가 됩니다.



SET에 추가할 이름을 입력 후 OK 버튼을 클릭합니다.



위에서 입력한 이름의 SET이 생성된 것을 확인합니다. (이 경우 데이터 셋이 연결 되므로, 추가작업 불필요)



폼 파일의 소스에디터를 열어 다음과 같이 코딩합니다.

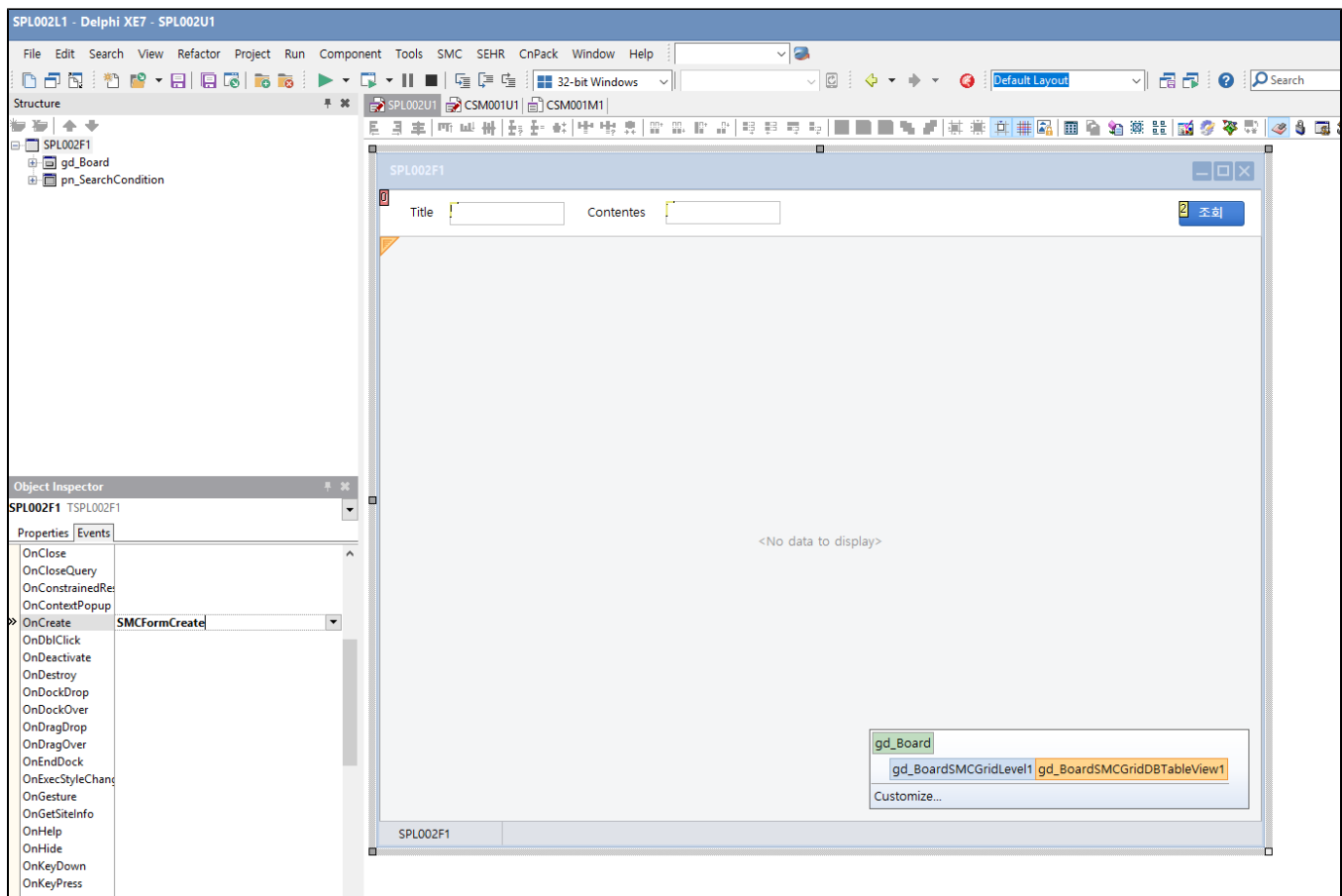
- uses 에 추가

```
uses
  Windows, Messages, SysUtils, Classes, Forms, Controls, StdCtrls,
  Dialogs, Graphics, SMCMessageDialog, SMCForm, Buttons, ExtCtrls,
  Variants, DateUtils, cxGraphics, cxControls, cxLookAndFeels,
  cxLookAndFeelPainters, dxSkinsCore, dxSkinsDefaultPainters, SMCLayoutControls,
  cxContainer, cxEdit, cxTextEdit, SMCInputControls, SMCViewControls, Vcl.Menus,
  cxButtons, SMCButtonControls, cxStyles, dxSkinscxPCPainter, cxCustomData,
  cxFilter, cxData, cxDataStorage, cxNavigator, Data.DB, cxDBData, cxGridLevel,
  SMCGridControls, cxClasses, cxGridCustomView, cxGridCustomTableView,
  cxGridTableView, cxGridDBTableView, cxGrid, SPL002M1;
```

- private 에 추가

```
    gd_Board: TSMCGrid;
private
    SPL002D1: TSPL002D1;
public
    { Public declarations }
end;
```

- [Events]탭의 onCreate - FormCreate 이벤트 생성 후 DataModule Create



```

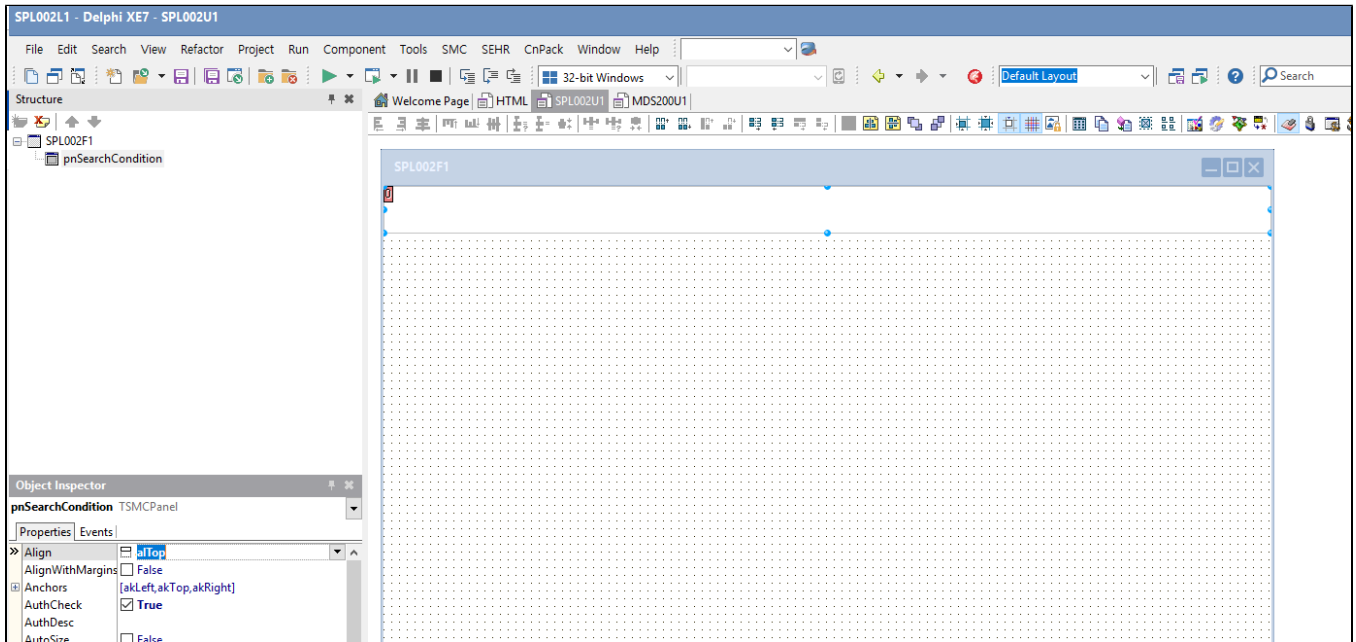
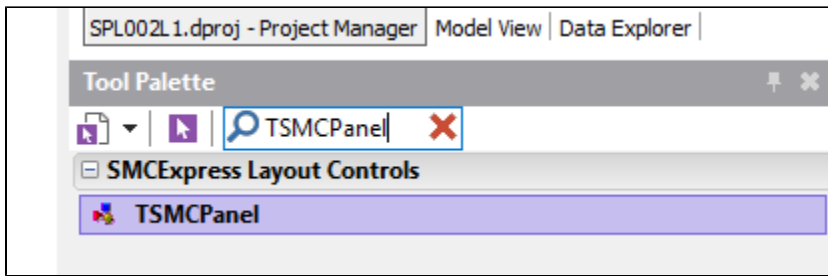
procedure TSPL002F1.SMCFormCreate(Sender: TObject);
begin
    if not Assigned(SPL002D1) then
    begin
        SPL002D1:= TSPL002D1.Create(Self);
        SPL002D1.Name:= '';
    end;
end;

```

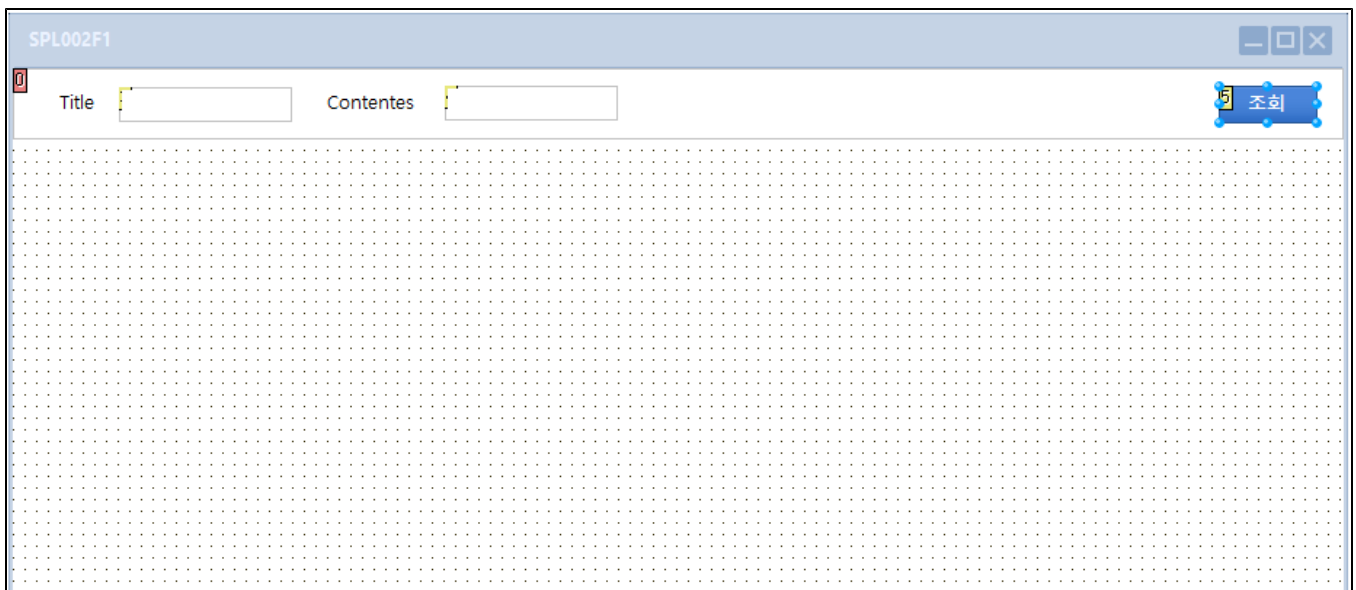
2. 조회 (다건)

1) 조회조건 컨트롤을 추가

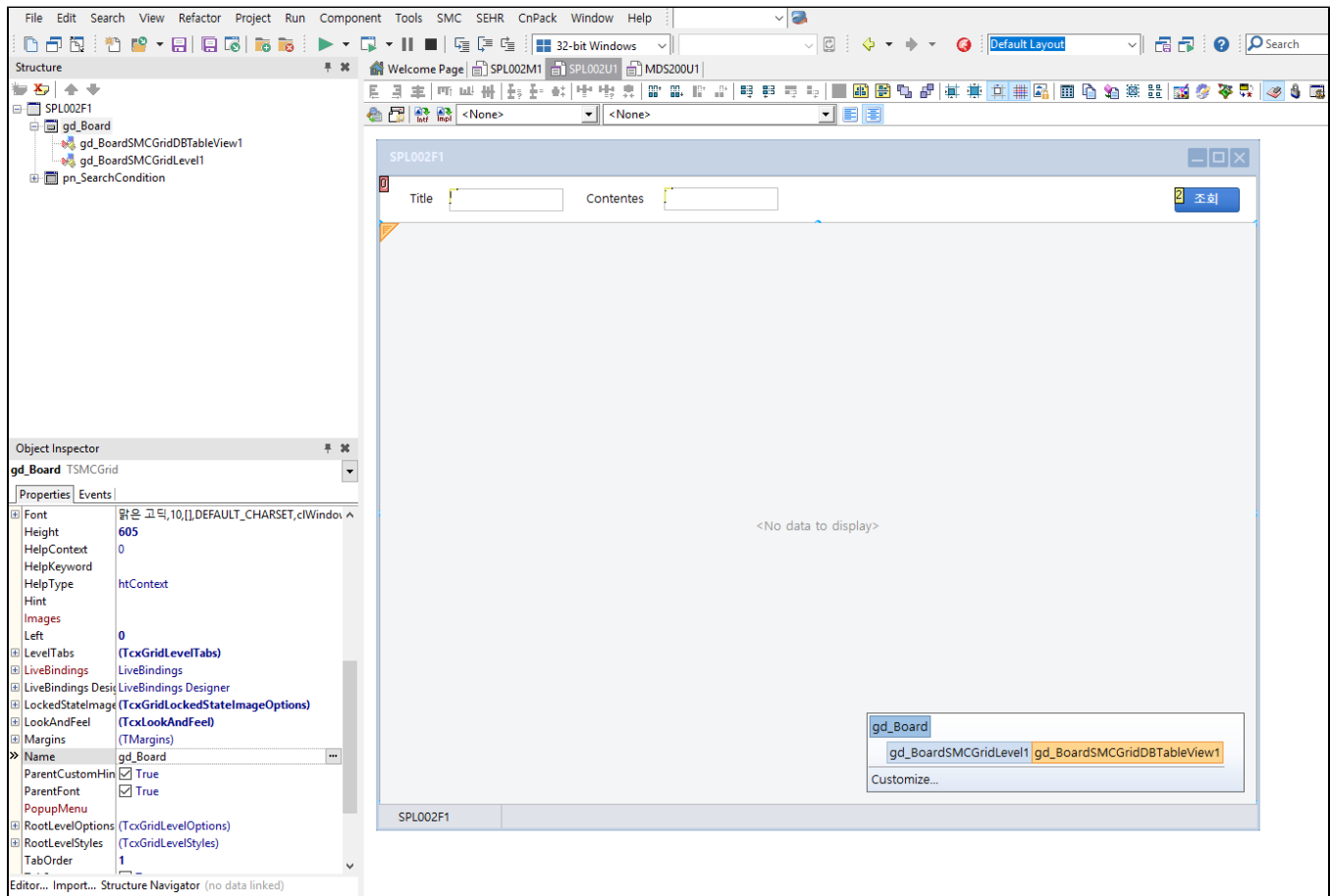
TSMCPanel을 추가하고 Align을 alTop으로 바꿉니다.



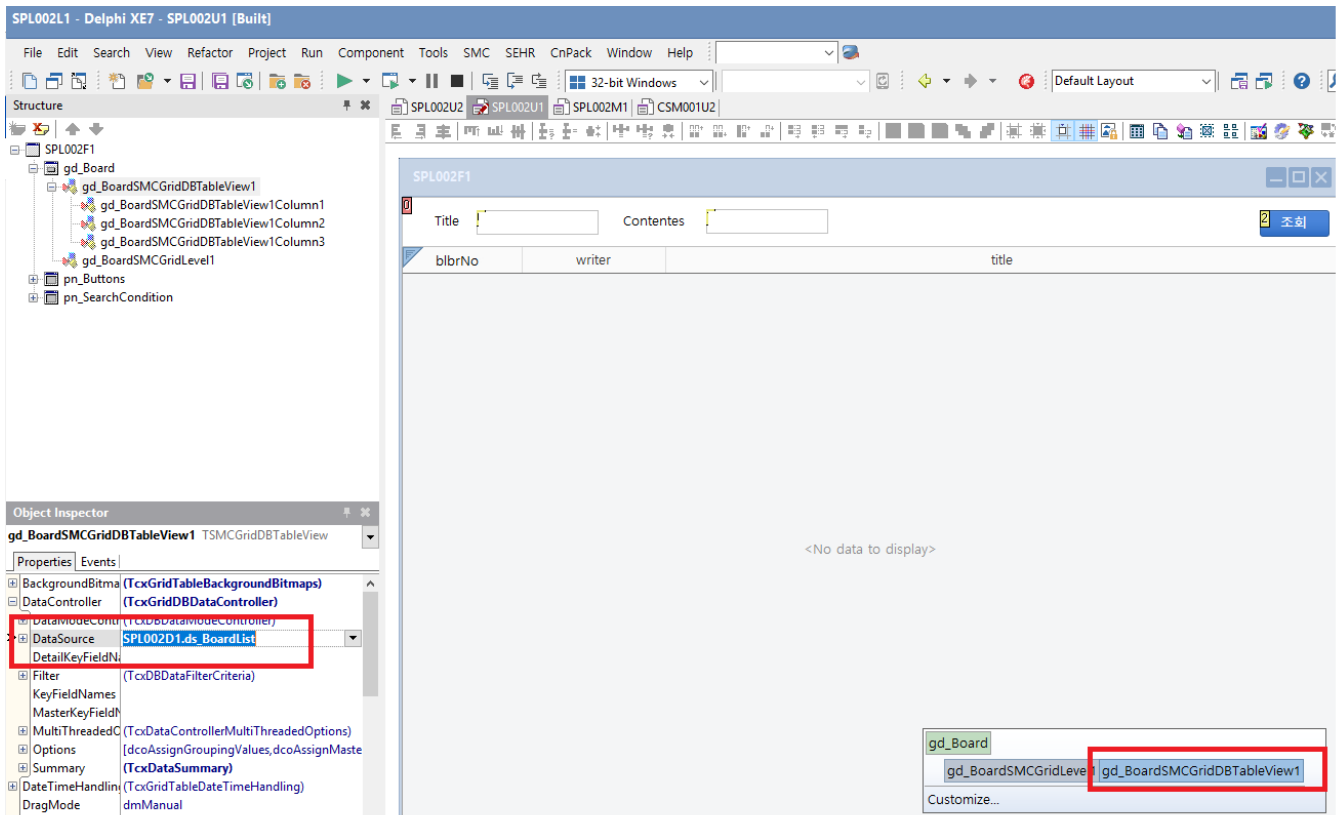
Title, Contents를 조회조건으로 입력하기 위해 TSMCTextEdit, TSMCLabel, TSMCQueryButton을 추가합니다.



게시판 목록을 조회할 TSMCGrid를 추가하고 Align을 alClient으로 바꿉니다.

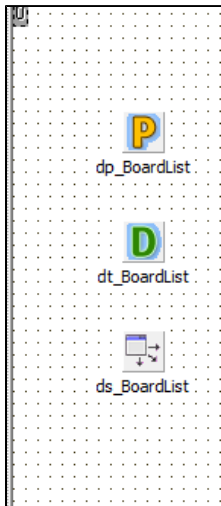


그리드 뷰 선택 후 DataController - DataSource 속성을 데이터모듈에 추가한 DataSource로 변경합니다.



2) 필드 매핑

데이터 모듈 - DataProvider를 선택 후 더블클릭합니다.



조회하고자 하는 서비스를 입력 후 조회버튼을 클릭합니다. (이름 : ExmGetBoardListSID)

Provider dp_BoardList Editor

Properties

이름: ExmGetBoardListSID

대기모드: ☐ 예 ☒ 아니오

공통서비스: ☐ 예 ☒ 아니오

Null 을 0 으로 대체: ☒ 예 ☐ 아니오

실행 모드

대기시간

데이터 크기

Request Response

이름: ExamBoardSVO

설명: ExamBoardSVO

종류: 업무

버전: 1.0

☐ 선택된 필드만 사용 ☐ 선택필드만보이게 ☒ 검색된 필드만보기 검색 ☐ 한글 사용 LogView Data로 체크

Name	Description	Type	Type2	Size	Scale	Count	Reference	UseYn
blbrNo	BLBR_NO	I	N	10	0	1		<input type="checkbox"/>
contents	CONTENTS	S	V	256	0	1		<input type="checkbox"/>
examBoardDVOList	examBoardDVOList	R	S			30		<input type="checkbox"/>
title	TITLE	S	V	256	0	1		<input type="checkbox"/>
writer	WRITER	S	V	256	0	1		<input type="checkbox"/>

F2 눌러서 수정모드 F3 눌러서 수정완료 BLOB 인 경우 Type 을 M으로 수정해주세요
List 객체가 아닌 단건 VO 를 사용하신다면 Count 를 1 로 해주세요.

저장 취소

필드 확인 후 저장 버튼을 클릭합니다.

※ 조회하고자 하는 목록항목의 다건(30), 단건(1) 카운트를 확인합니다.

Provider dp_BoardList Editor

Properties

이름

ExmGetBoardListSID

대기모드

☐ 예
☒ 아니오

공통서비스

☐ 예
☒ 아니오

Null 을 0 으로 대체

☒ 예
☐ 아니오

실행 모드

대기시간

데이터 크기

RequestResponse

이름

ExamBoardSVO

설명

ExamBoardSVO

종류

업무

버전

1.0

☐ 선택된 필드만 사용
☐ 선택필드만보이게
☒ 검색된 필드만보기

검색

☐ 한글 사용

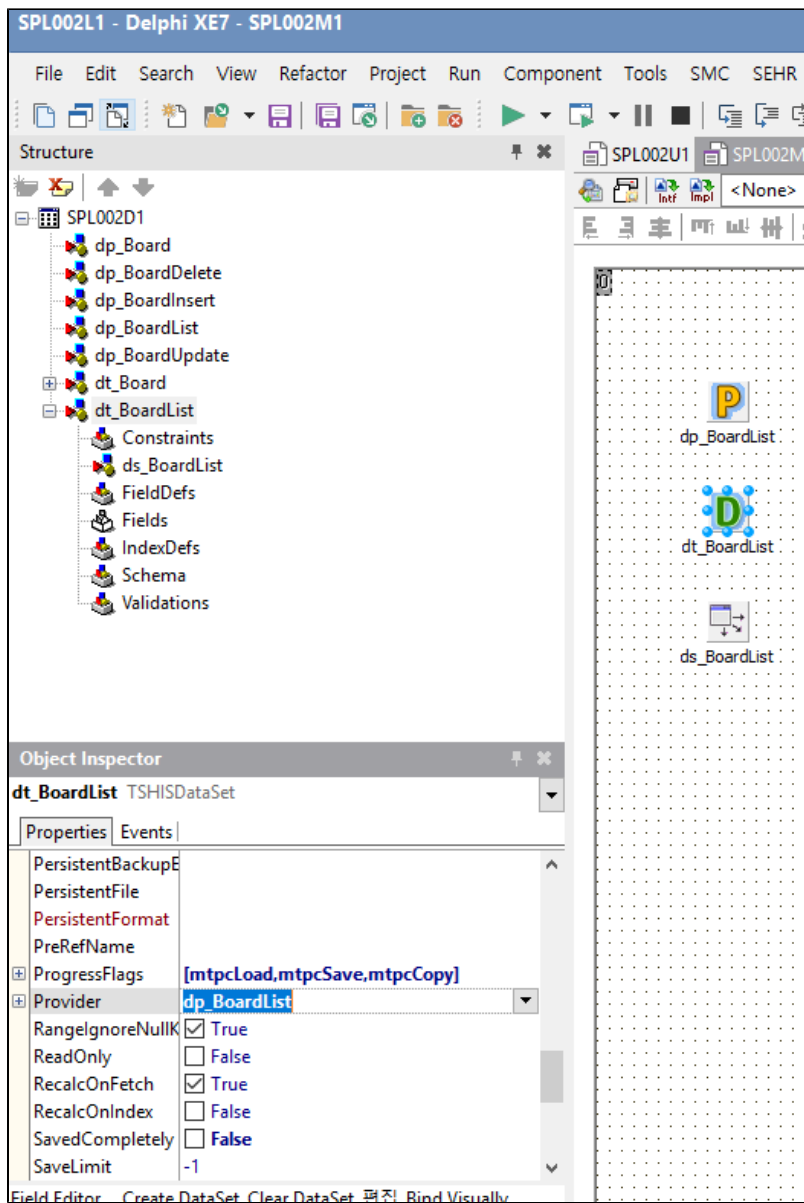
LogView Data로 체크

Name	Description	Type	Type2	Size	Scale	Count	Reference	UseYn
blbrNo	BLBR_NO	I	N	10	0	1		<input type="checkbox"/>
contents	CONTENTS	S	V	256	0	1		<input type="checkbox"/>
examBoardDVOList	examBoardDVOList	R	S			30		<input type="checkbox"/>
title	TITLE	S	V	256	0	1		<input type="checkbox"/>
writer	WRITER	S	V	256	0	1		<input type="checkbox"/>

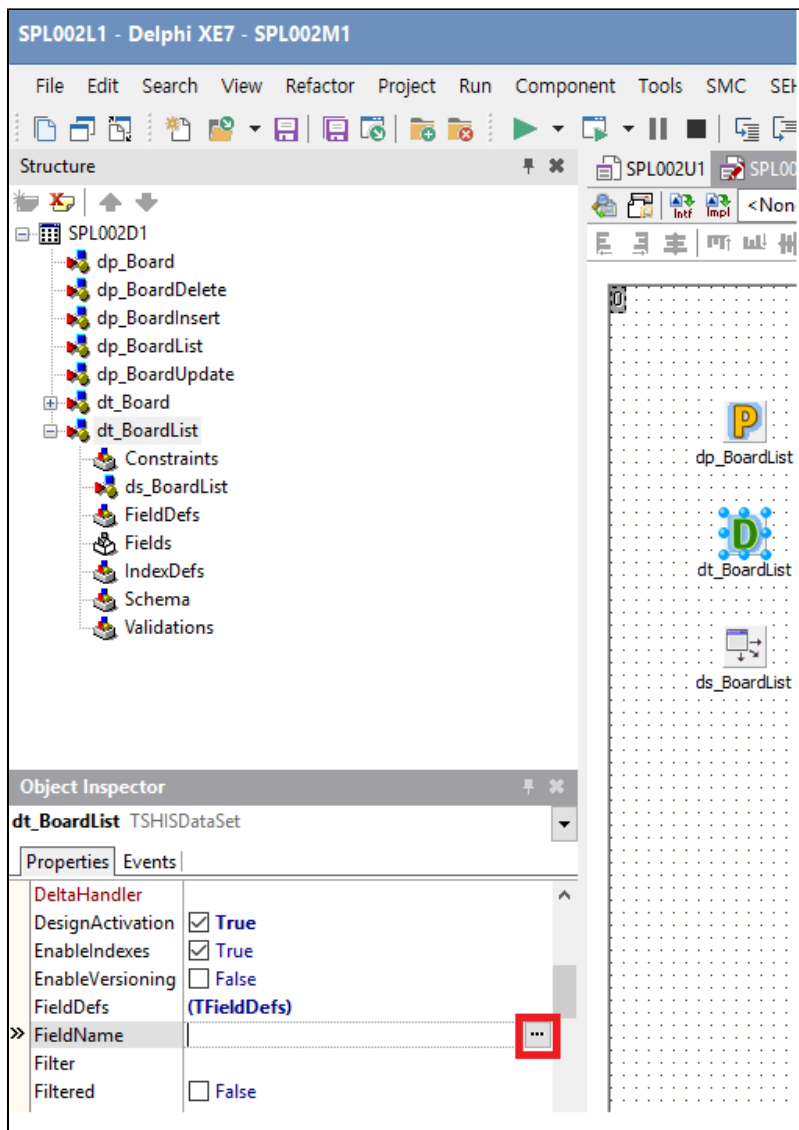
F2 눌러시면 수정모드 F3 눌러시면 수정완료 BLOB 인 경우 Type 을 M으로 수정해주세요
List 객체가 아닌 단건 VO 를 사용하신다면 Count 를 1 로 해주세요.

저장
취소

데이터셋 선택 후 Provider 필드에 프로바이더를 입력합니다.



FieldName 속성 선택 후 조회버튼을 클릭합니다.



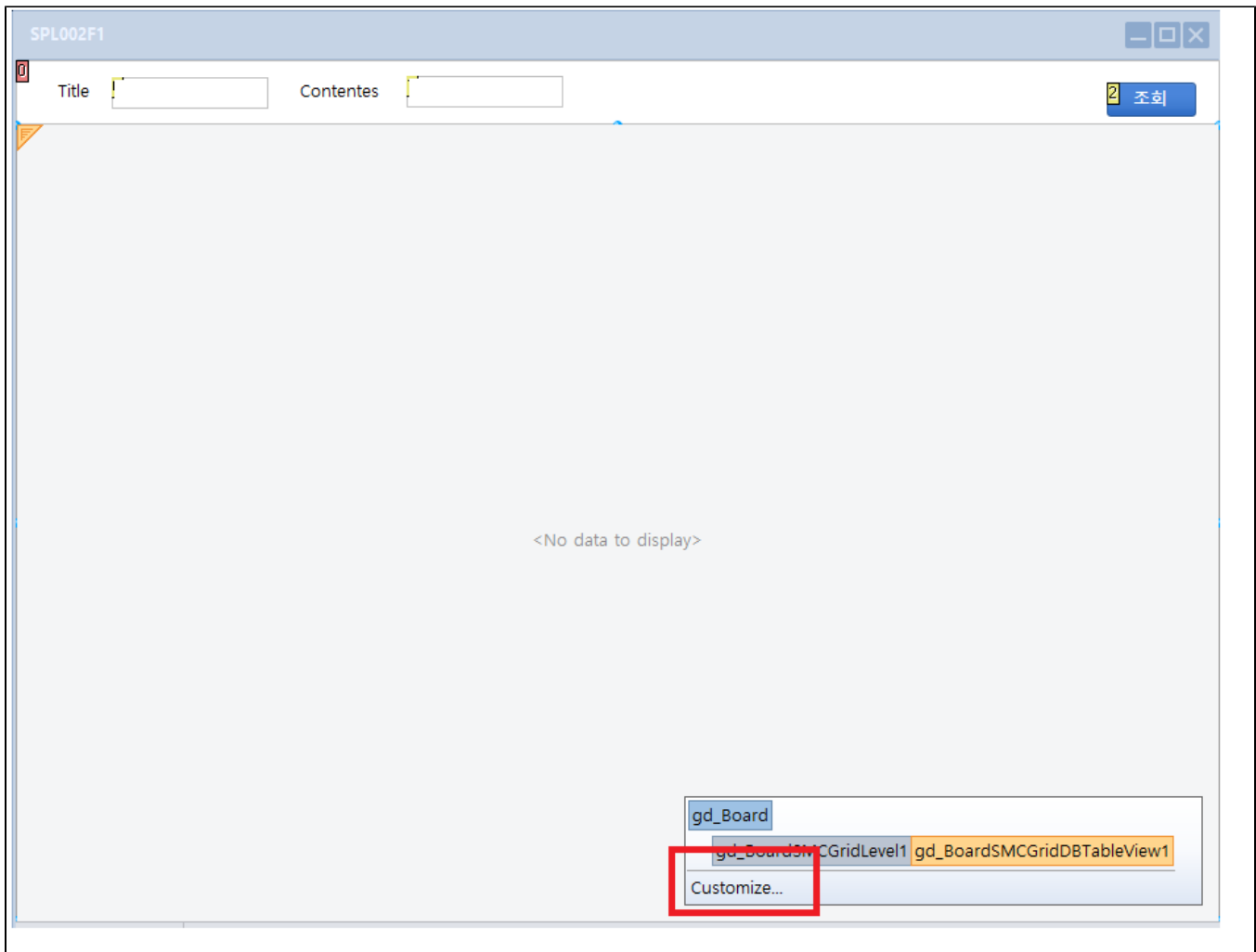
조회결과를 받아올 목록 이름을 선택 후 저장버튼을 클릭합니다.

frmDataSetFieldNameEditor

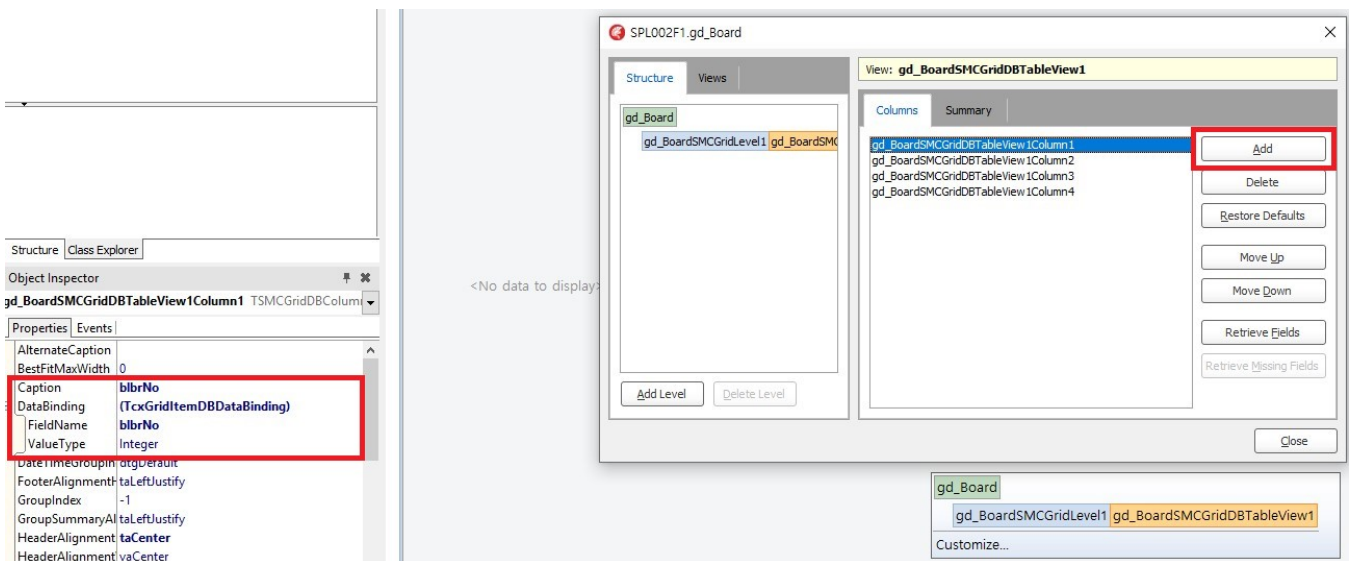
이름	설명
Message	
title	TITLE
contents	CONTENTS
blbrNo	BLBR_NO
writer	WRITER
examBoardDVOList	examBoardDVOList
blbrNo	BLBR_NO
title	TITLE
contents	CONTENTS
writer	WRITER
gvLginId	GV_LGIN_ID
gvUserIp	GV_USER_IP
gvClntPrgmId	GV_CLNT_PRGM_ID
gvSrvrPrgmId	GV_SRVR_PRGM_ID

저장 취소

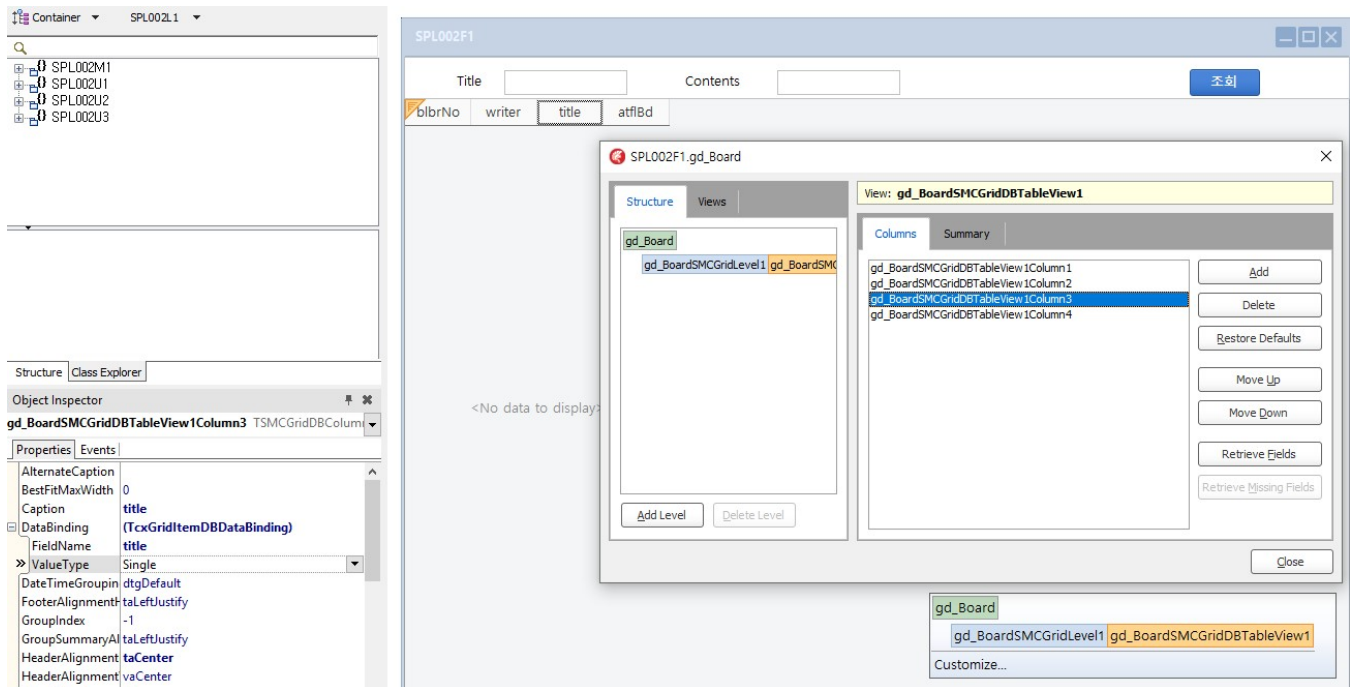
그리드 선택 후 Customize를 클릭합니다.



Add버튼을 클릭하여 컬럼을 추가한 후 매핑되는 필드, 데이터 타입을 선택합니다.



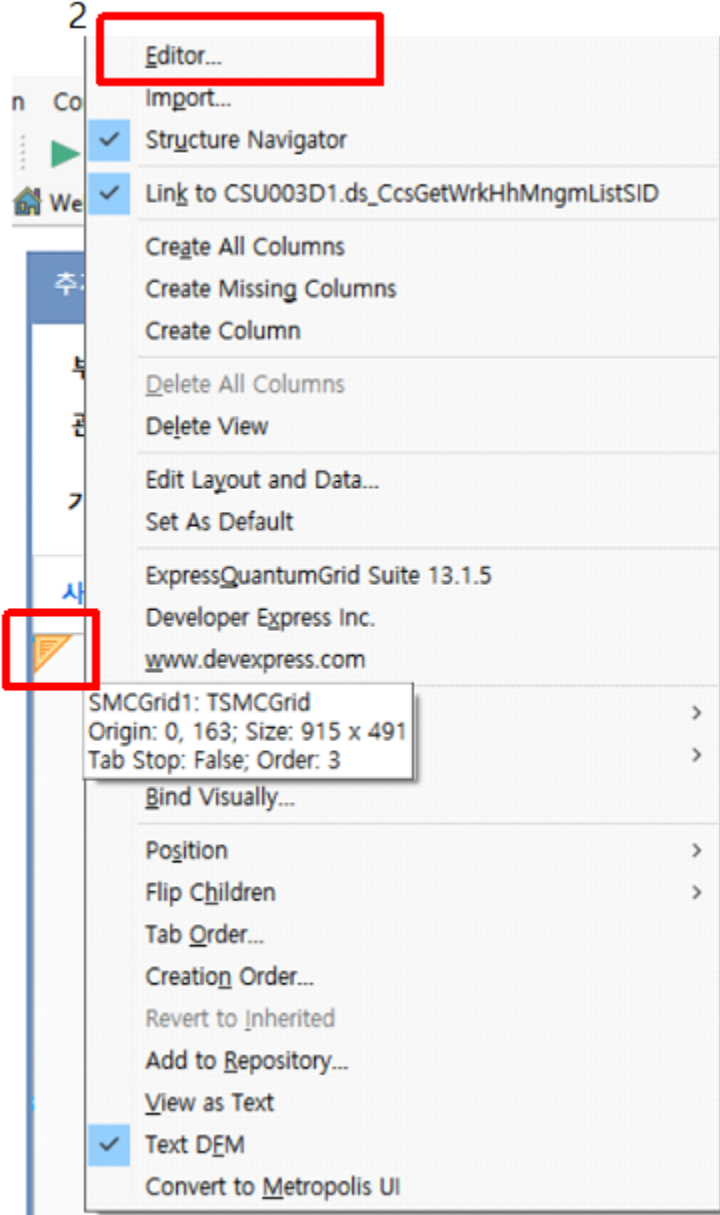
나머지 필드도 입력합니다.



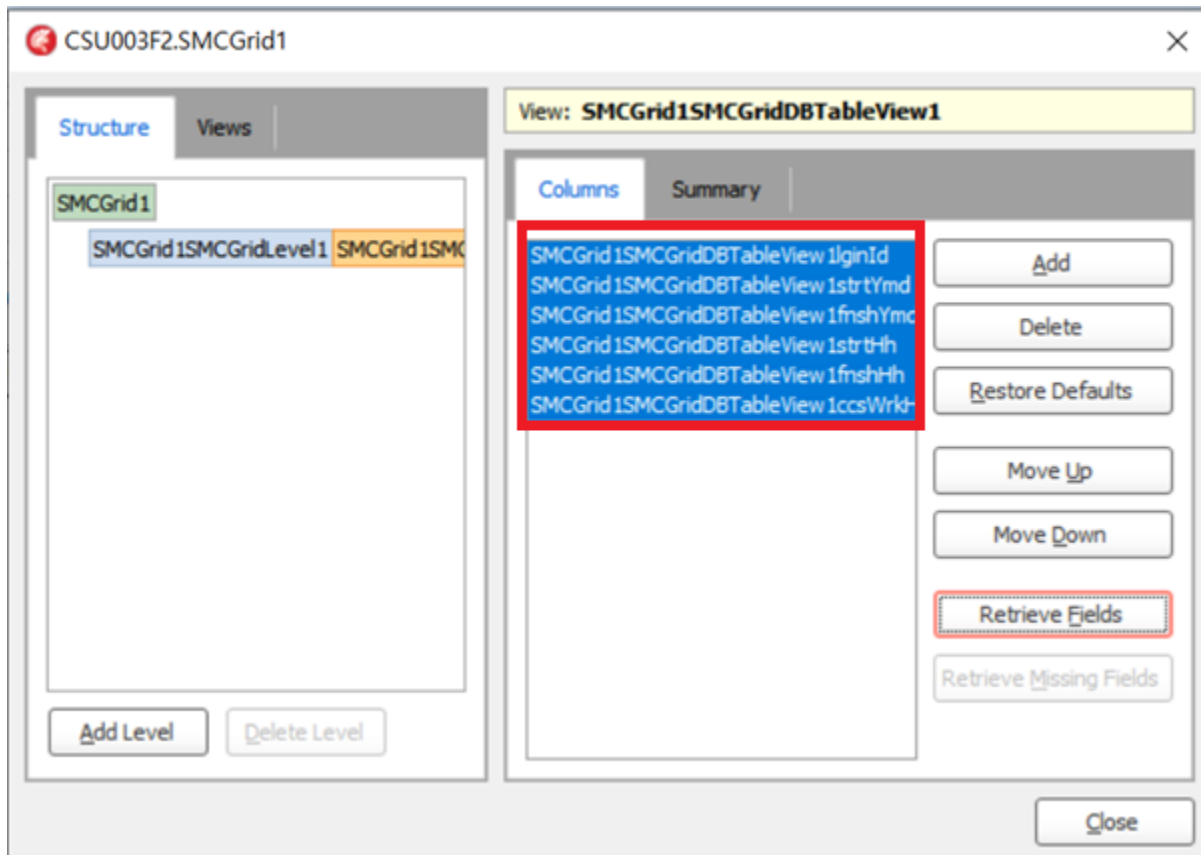
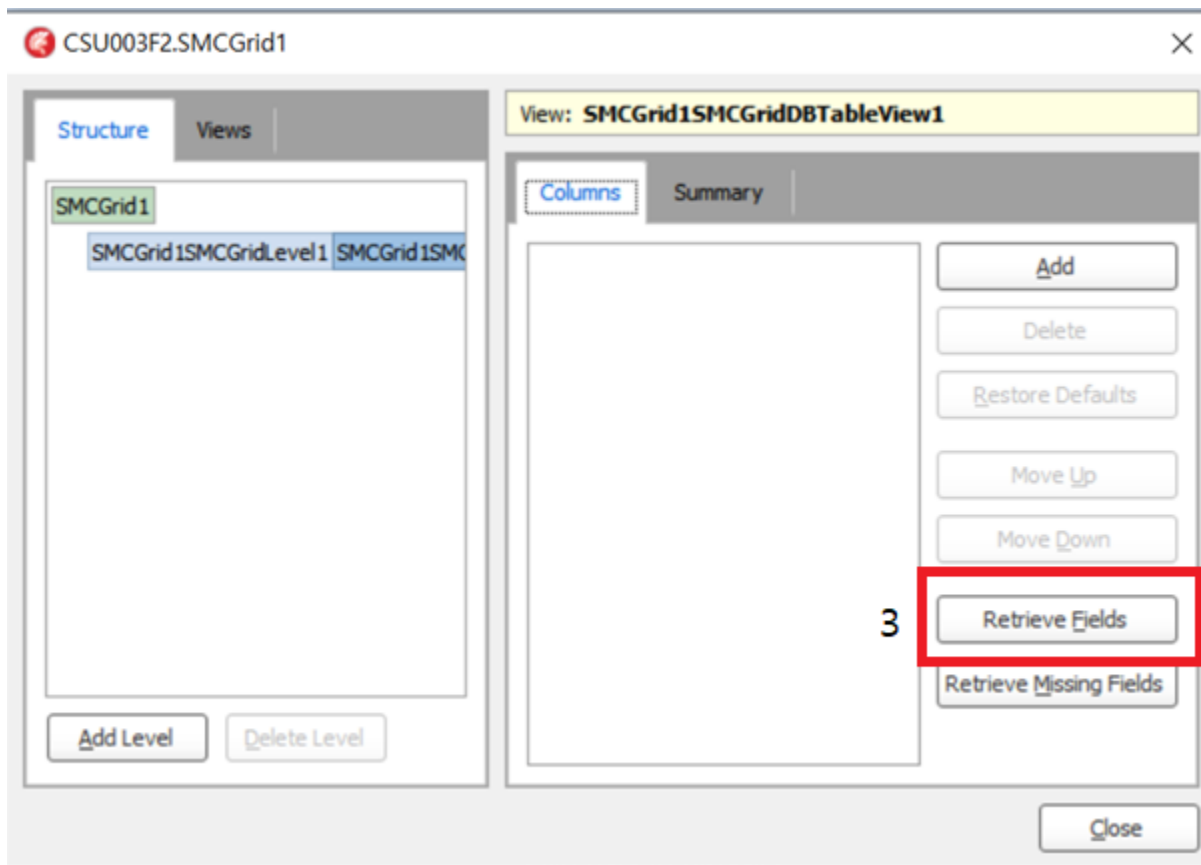
※ 아래와 같이 Editor 기능을 사용하면 자동으로 필드 Mapping 이 가능합니다.

1. TSMCGrid 왼쪽 상단 우클릭 > 2. Editor 클릭 > 3. Retrieve Fields 클릭 > 4. 해당 DataSet 에 있는 필드가 자동으로 추가됨

2

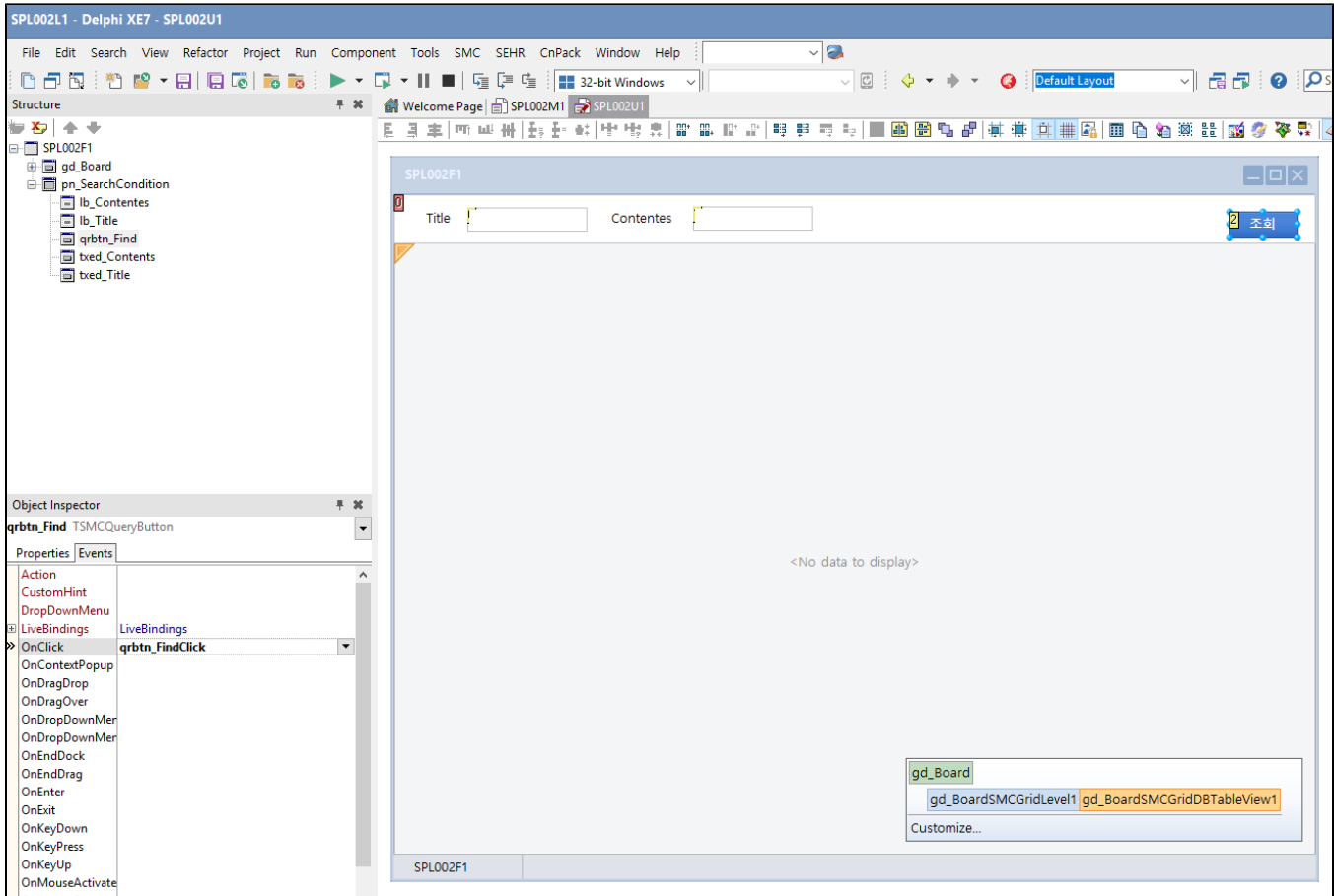


1



조회버튼의 클릭 이벤트를 추가하여 조회기능을 코딩합니다.

3) 조회기능 코딩



```
procedure TSPL002F1.qrbtn_FindClick(Sender: TObject);
begin

    SPL002D1.dp_BoardList.Request.FieldByName('title').asString := string(txed_Title.Text).Trim;
    SPL002D1.dp_BoardList.Request.FieldByName('contents').asString := string(txed_Contentes.Text).Trim;

    SPL002D1.dp_BoardList.Start(Self);

end;
```

이클립스 서버 정보에서 port, module path를 확인하여 설치폴더(C:\Nexmed_EHR\DEV\SYS\ENV) - SYS\ENV\Config.ini를 수정합니다.

Overview

General Information

Specify the host name and other common settings.

Server name:

Host name:

Runtime Environment:

Configuration path:

[Open launch configuration](#)

Server Locations

Specify the server path (i.e. catalina.base) and deploy path. Server must be published with no modules present to make changes.

☒ Use workspace metadata (does not modify Tomcat installation)

☐ Use Tomcat installation (takes control of Tomcat installation)

☐ Use custom location (does not modify Tomcat installation)

Server path:

[Set deploy path to the default value \(currently set\)](#)

Deploy path:

Server Options

Enter settings for the server.

☒ Serve modules without publishing

☐ Publish module contexts to separate XML files

☒ Modules auto reload by default

☐ Enable security

Publishing

Timeouts

Ports

Modify the server ports.



Port Name	Port Number
Tomcat admin port	8005
HTTP/1.1	80
AJP/1.3	8009

MIME Mappings

Overview | Modules

Web Modules

Configure the Web Modules on this server.

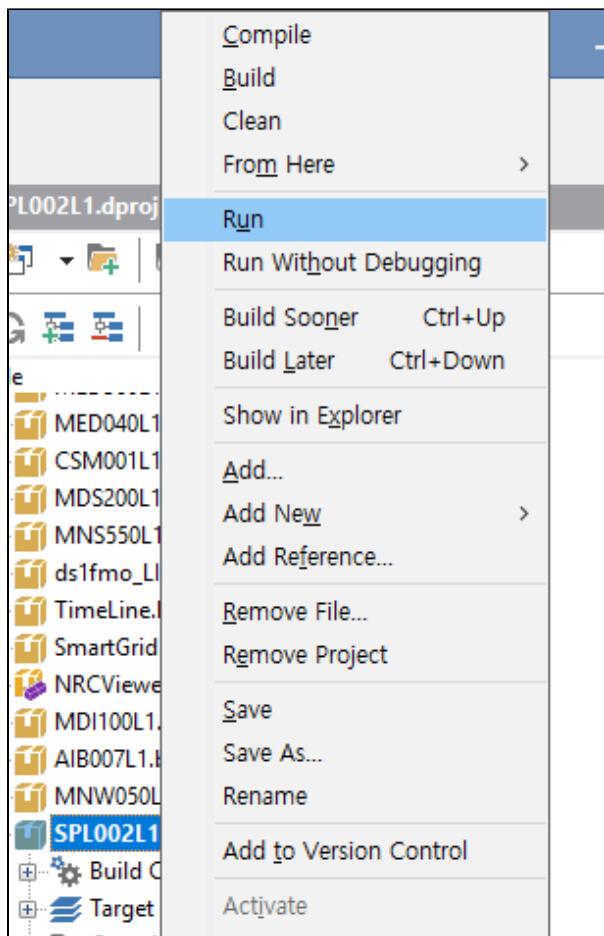
Path	Document Base	Module	Auto Reload
 /DOM_EHR_SAMPLE	DOM_EHR_SAMPLE	 DOM_EHR_SAMPLE	Enabled

Overview | Modules

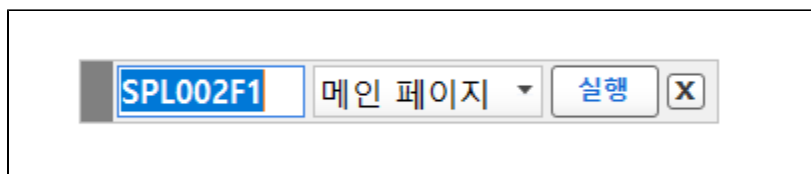

```
Config.ini - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

;Local
[SHIS]
SERVERTYPE=Local
[LOCAL]
SERVERNAME=127.0.0.1
SERVERPORT=80
NORMALSERVICE=/DOM_EHR_COM/integration/service
NORMALSERVICE=/DOM_EHR_SAMPLE/integration/service
BPLADDR=CJWoBZ4vCounDouoCW
BPLPORT=E300CC
BPLID=TsLYT6bbSW
BPLPASSWORD=TsLYT6bbSZ0m
BPLPATH=BrDeON9bH65q0IzbQ7810snfPMvqBqL8KYy
MQTTIP=CJWoBZ4vCounDouoDG
MQTTPORT=CJWuOm
SSL=N
ENCRYPT=N
ENCRYPTEDIGNORELIST=
ReportWaterMark=0
LogLevel=3
ExceptionLevel=3
UseDisableGhosting=1
AUTOLOGOUTTIME=300
USECOMCODEDAT=Y
KMIIP=CJWoBZ4vCounDouoCG
KMIPORT=Dp0mCG
CERTIP=
CERTPORT=Dp0mCG
CERTPROVIDER=1
DURIP=Q7HqS3e1BtHbStGpBcjfRNDmRsCk0sykQt8IRMzaTMnbSozDPMHf0s5iIMvcRovXSsru
DURSOAPIP=Q7HqS3e1Bt110ovhQMrpRsviQMvbBcD1BcjoBm
[DEV]
SERVERNAME=CJWoBZ4vCounDouoCG
```

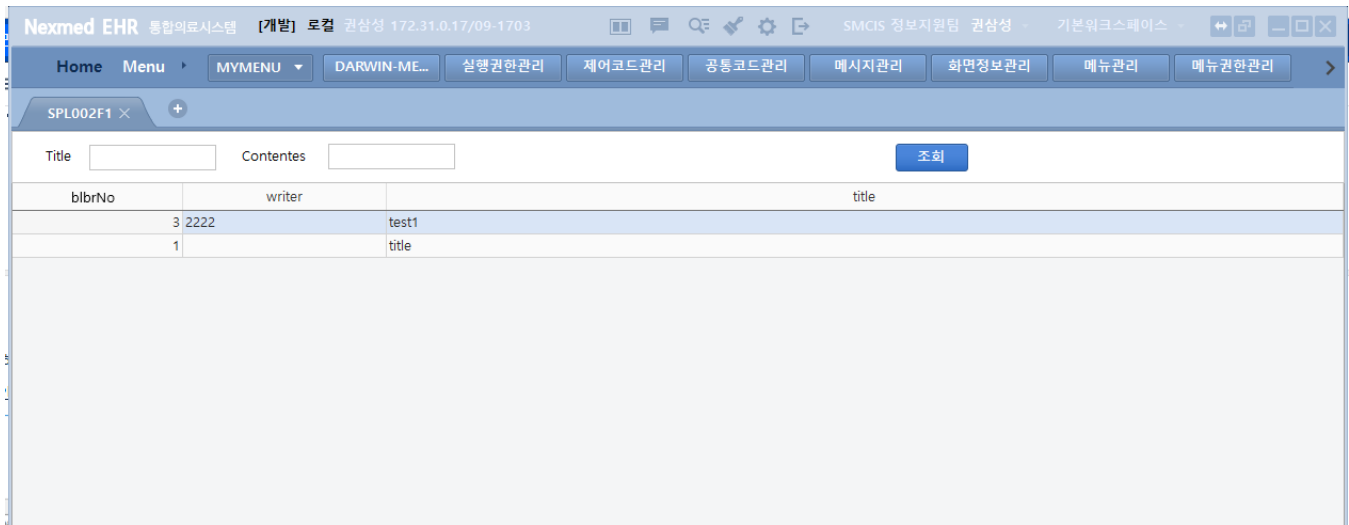
프로젝트를 선택한 후 마우스 우클릭 후 - run을 선택합니다.



임의의 계정으로 로그인 한 후 화면실행창에서 화면명을 입력하고 실행버튼을 클릭합니다.



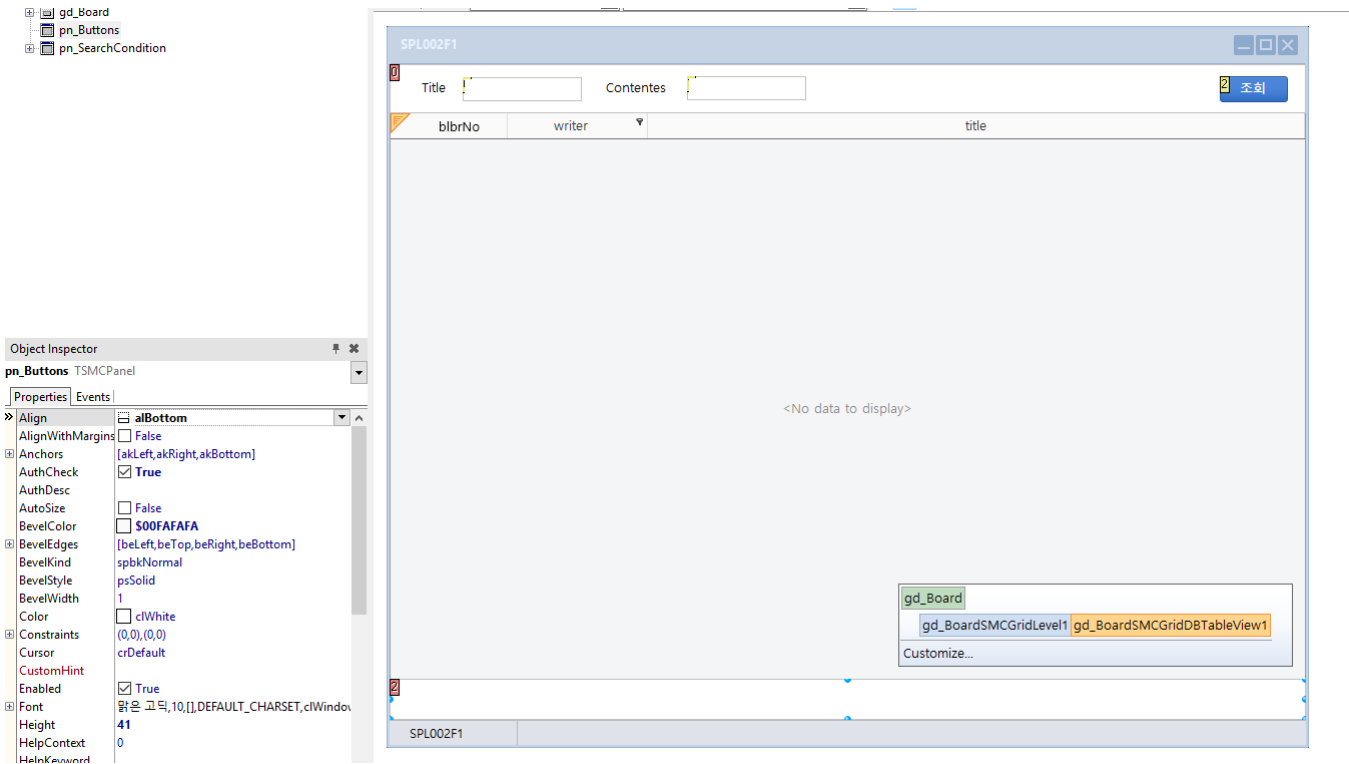
조회버튼을 클릭하여 조회결과가 나오는 것을 확인합니다.



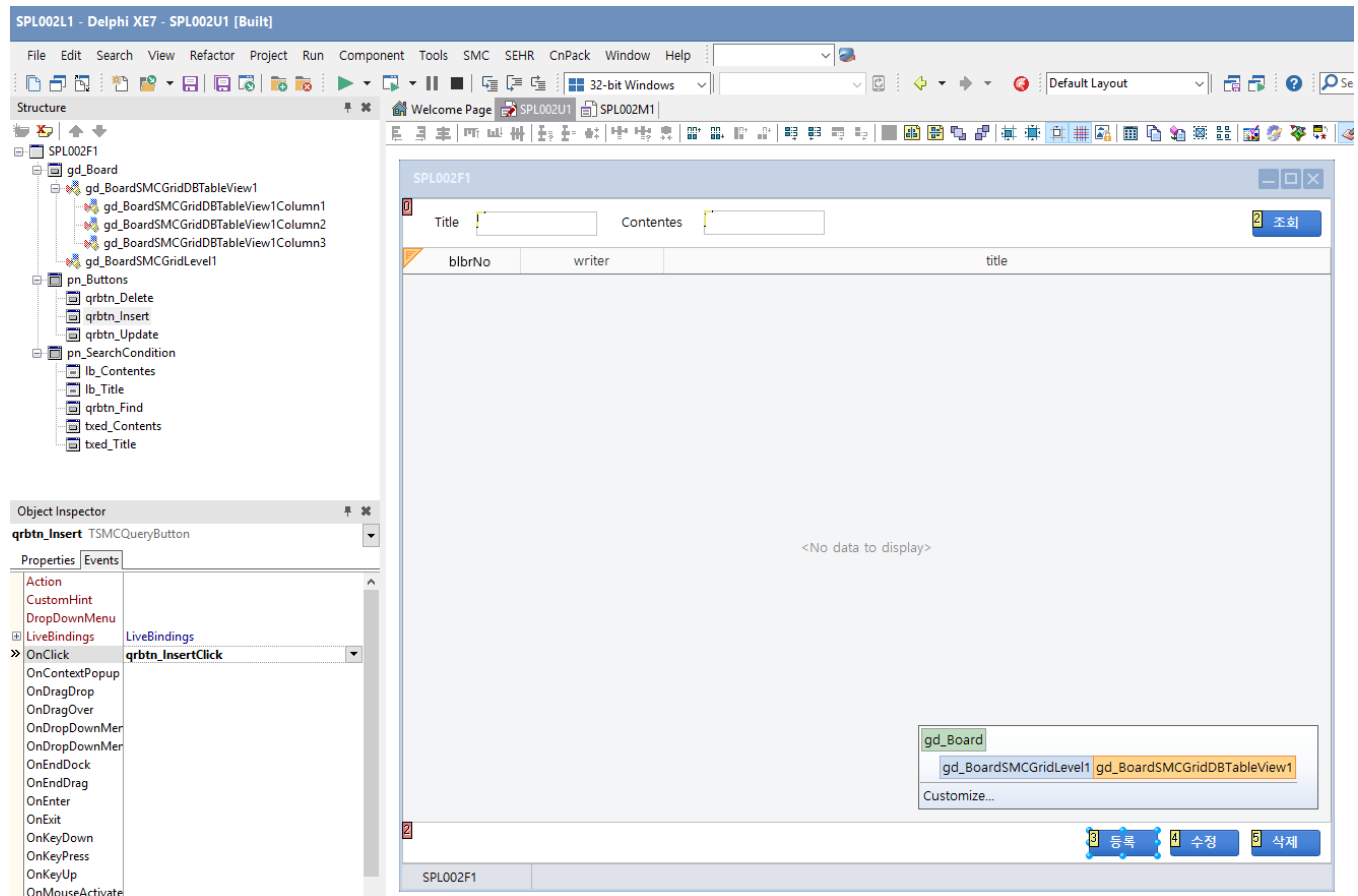
3. 입력

1) 버튼 컨트롤 추가

TSMCPanel을 추가하고 Align을 alBottom으로 바꿉니다.



세개의 TSMCQueryButton(등록, 수정, 삭제)을 추가합니다.



등록 후 조회하기 위해 조회버튼부분의 코딩을 다음과 같이 변경합니다

```
private
    SPL002D1: TSPL002D1;
    procedure findBoard; //
public
```

```
procedure TSPL002F1.findBoard;
begin

    SPL002D1.dp_BoardList.Request.FieldName('title').asString := string(txed_Title.Text).Trim;
    SPL002D1.dp_BoardList.Request.FieldName('contents').asString := string(txed_Contents.Text).Trim;

    SPL002D1.dp_BoardList.Start(Self);

end;
```

```
procedure TSPL002F1.qrbtn_FindClick(Sender: TObject);
begin

    findBoard;

end;
```

등록버튼의 클릭 이벤트를 추가하여 등록화면 호출 부분을 코딩합니다.

```
procedure TSPL002F1.qrbtn_InsertClick(Sender: TObject);
var
    SPL002F2 : TSPL002F2;
begin

    SPL002F2:= TSPL002F2.Create(Self);
    try

        if SPL002F2.ShowModal = mrOk then
            begin

                findBoard;

            end;

        finally
            SPL002F2.Free;
        end;

    end;
```

2) DataProvider 추가

데이터 모듈에 TSHISDataProvider를 추가하고 더블클릭 - 입력 서비스 입력 - 조회버튼을 클릭합니다. (이름 : ExmInsBoardSID)

Provider dp_BoardInsert Editor

Properties

이름

ExmInsBoardSID

대기모드

☐ 예
☒ 아니오

공통서비스

☐ 예
☒ 아니오

Null 을 0 으로 대체

☒ 예
☐ 아니오

실행 모드

대기시간

데이터 크기

Request

Response

이름

ExamBoardSVO

설명

ExamBoardSVO

종류

업무

버전

1.0

Name	Description	Type	Type2	Size	Scale	Count	Reference
title	TITLE	S	V	256	0	1	
contents	CONTENTS	S	V	256	0	1	
blbrNo	BLBR_NO	I	N	256	0	1	
writer	WRITER	S	V	256	0	1	
> examBoardDVOList	examBoardDVOList	R	S			30	

F2 눌러서면 수정모드
F3 눌러서면 수정완료
BLOB 인 경우 Type 을 M으로 수정해주세요
List 객체가 아닌 단건 VO 를 사용하신다면 Count 를 1 로 해주세요.

저장

취소

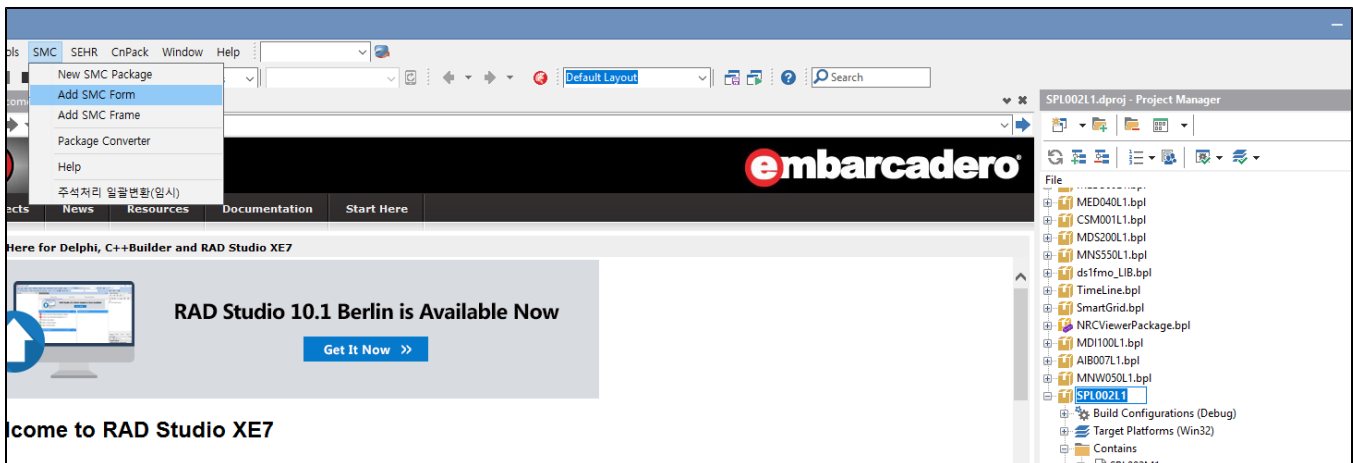
3) 등록화면 추가(SPL002U2)

목록조회화면의 uses에 등록화면 unit이름을 추가합니다.

uses

```
Windows, Messages, SysUtils, Classes, Forms, Controls, StdCtrls,  
Dialogs, Graphics, SMCMessageDialog, SMCForm, Buttons, ExtCtrls,  
Variants, DateUtils, cxGraphics, cxControls, cxLookAndFeels,  
cxLookAndFeelPainters, dxSkinsCore, dxSkinsDefaultPainters, SMCLayoutControls,  
cxContainer, cxEdit, cxTextEdit, SMCInputControls, SMCViewControls, Vcl.Menus,  
cxButtons, SMCButtonControls, cxStyles, dxSkinscxPCPainter, cxCustomData,  
cxFilter, cxData, cxDataStorage, cxNavigator, Data.DB, cxDBData, cxGridLevel,  
SMCGridControls, cxClasses, cxGridCustomView, cxGridCustomTableView,  
cxGridTableView, cxGridDBTableView, cxGrid, SPL002M1, SPL002U2;
```

프로젝트 선택 후 메뉴 - SMC - Add SMC Form을 선택합니다.



폼정보를 입력한 후 생성버튼을 클릭합니다.

New Form

Unit Name

SPL002U2

Form Caption

계시물 등록

Description

계시물 등록

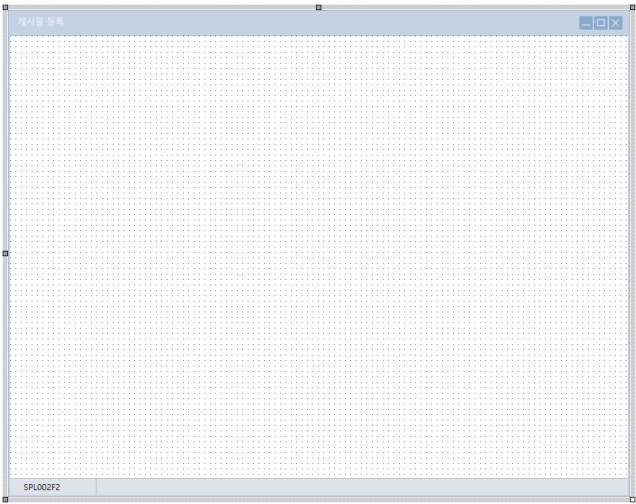
Category

서브시스템명 :
업무분류명(대) :
업무분류명(중) :

생성

취소

폼이 생성된 것을 확인합니다.



Title, Contents를 입력하기 위해 TSMCTextEdit, TSMCLabel, TSMCCommandButton을 추가합니다.

게시물 등록

Writer

Title

Contents

3

0 저장

1 종료

SPL002F2

uses에 DataModule을 추가합니다.

```

uses
  Windows, Messages, SysUtils, Classes, Forms, Controls, StdCtrls,
  Dialogs, Graphics, SMCMessageDialog, SMCForm, Buttons, ExtCtrls,
  Variants, DateUtils, cxGraphics, cxControls, cxLookAndFeels,
  cxLookAndFeelPainters, cxContainer, cxEdit, dxSkinsCore,
  dxSkinsDefaultPainters, Vcl.Menus, cxButtons, SMCButtonControls,
  SMCLayoutControls, cxMemo, SMCInputControls, cxTextEdit, SMCViewControls,
  SPL002M1;

```

데이터 모듈을 변수로 추가합니다.

```
private
    SPL002D1: TSPL002D1;
public
```

Form Create에 다음과 같이 코딩합니다.

```
procedure TSPL002F2.SMCFormCreate(Sender: TObject);
begin

    if not Assigned(SPL002D1) then
    begin
        SPL002D1:= TSPL002D1.Create(Self);
        SPL002D1.Name:= '';
    end;

    txed_Writer.Text := GlobalVar.lginId; //

end;
```

4) 등록기능 코딩

저장버튼의 클릭이벤트에 다음과 같이 코딩합니다.

```
procedure TSPL002F2.cmbtn_SaveClick(Sender: TObject);
begin

    SPL002D1.dp_BoardInsert.Request.FieldByName('writer').AsString := string(txed_Writer.Text).Trim;
    SPL002D1.dp_BoardInsert.Request.FieldByName('title').AsString := string(txed_Title.Text).Trim;
    SPL002D1.dp_BoardInsert.Request.FieldByName('contents').AsString := string(mm_Contents.Text).Trim;

    SPL002D1.dp_BoardInsert.Start(Self);

    ModalResult := mrOk;

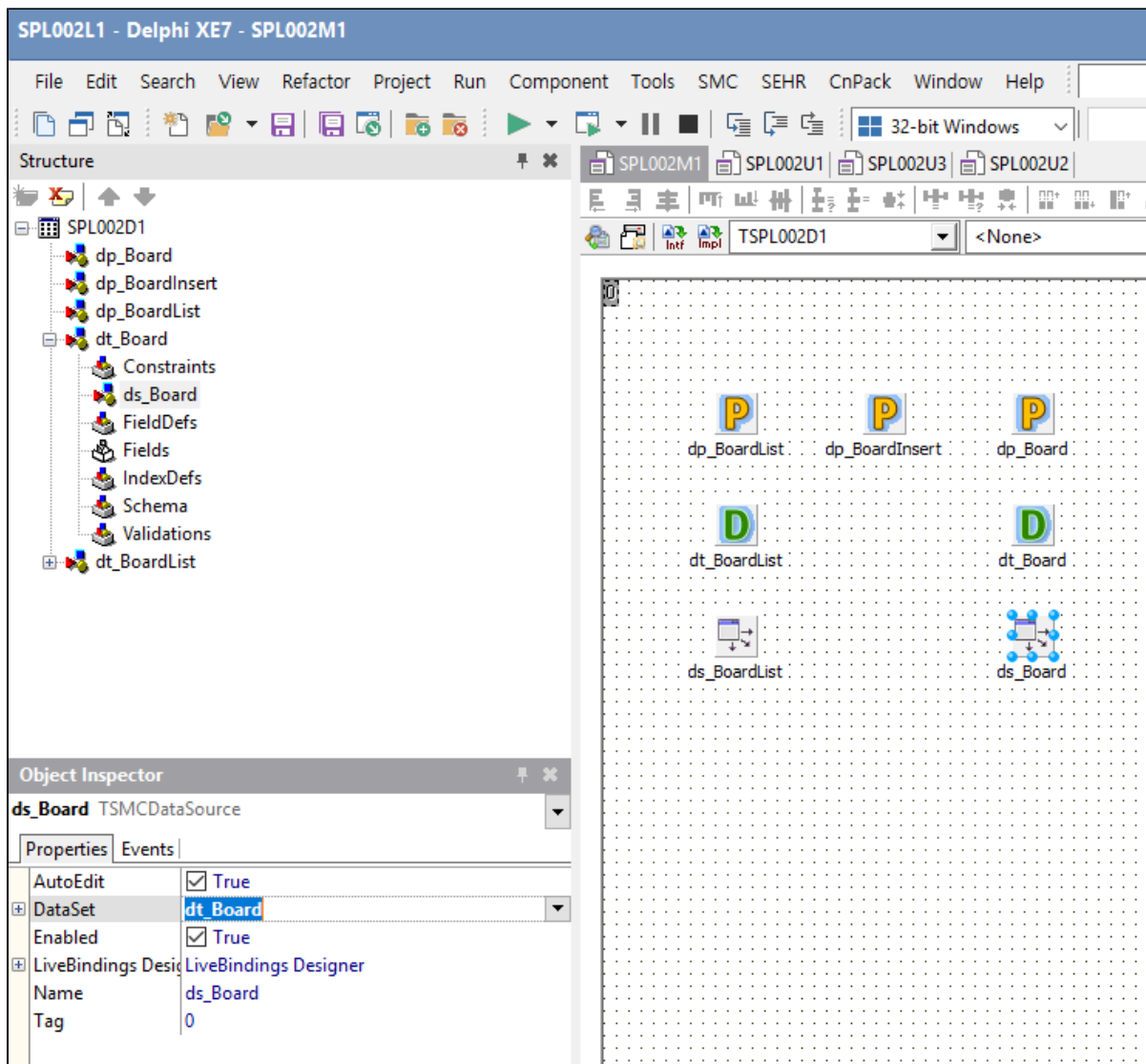
end;
```

4. 조회 (단건)

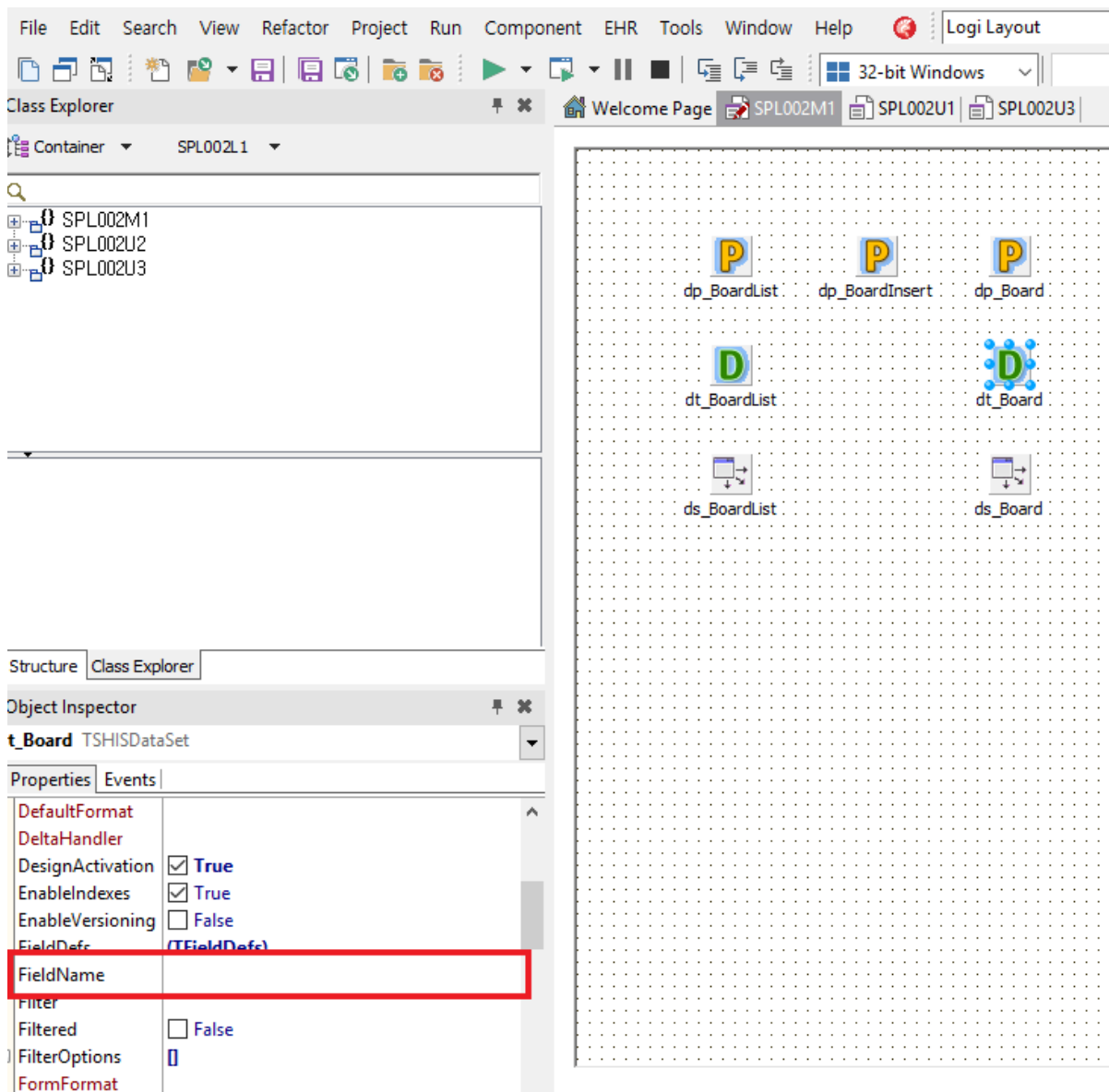
1) Data관련 컴포넌트 추가

데이터 모듈에 TSHISDataPrvider, TSHISDataSet, TSMCDataSource를 추가합니다.

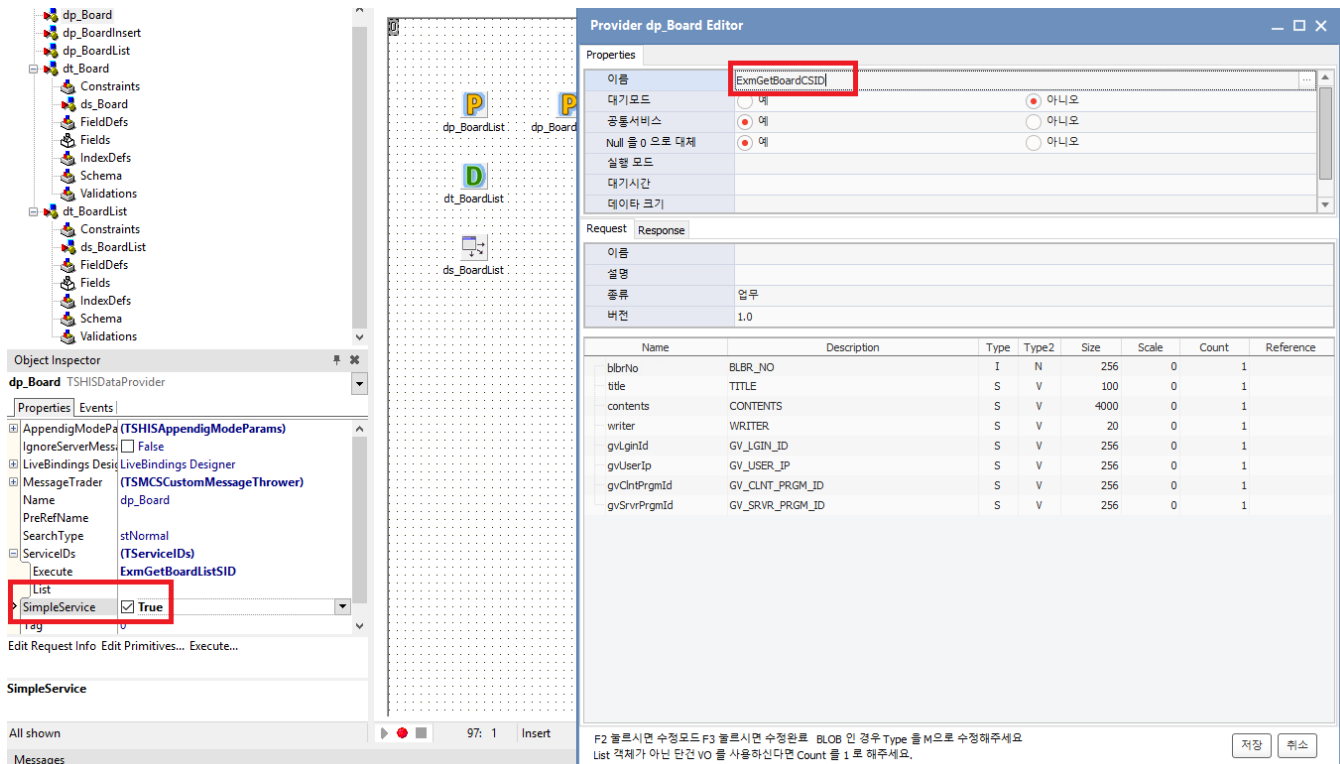
DataSet의 Provider필드, DataSource의 DataSet필드를 바꿉니다.



TSHISDataSet의 FeildName가 비워있는지 확인합니다.



TSHISDataProvider를 선택 & SimpleService속성을 true로 한후 더블클릭 - 조회 서비스 입력 - 조회버튼을 클릭합니다. (이름 : ExmGetBoardCSID)



2) 수정화면 호출

목록조회화면의 uses에 등록하면 수정화면 unit이름을 추가합니다.

```

uses
  Windows, Messages, SysUtils, Classes, Forms, Controls, StdCtrls,
  Dialogs, Graphics, SMCMessageDialog, SMCForm, Buttons, ExtCtrls,
  Variants, DateUtils, cxGraphics, cxControls, cxLookAndFeels,
  cxLookAndFeelPainters, dxSkinsCore, dxSkinsDefaultPainters, SMCLayoutControls,
  cxContainer, cxEdit, cxTextEdit, SMCInputControls, SMCViewControls, Vcl.Menus,
  cxButtons, SMCButtonControls, cxStyles, dxSkinscxPCPainter, cxCustomData,
  cxFilter, cxData, cxDataStorage, cxNavigator, Data.DB, cxDBData, cxGridLevel,
  SMCGridControls, cxClasses, cxGridCustomView, cxGridCustomTableView,
  cxGridTableView, cxGridDBTableView, cxGrid, SPL002M1, SPL002U2, SPL002U3;

```

수정버튼의 클릭이벤트에 다음과 같이 코딩합니다.

```

procedure TSPL002F1.qrbtn_UpdateClick(Sender: TObject);
begin

    if SPL002D1.dt_BoardList.RecordCount <= 1 then
        Exit;

    try
        SendLinkData.AllClear;
        SendLinkData.PackageName      := 'SPL002L1';
        SendLinkData.ClassName        := 'SPL002F3';
        SendLinkData.Destination      := lddModalForm;
        SendLinkData.BroadcastType    := ldbcNone;
        SendLinkData.Integers['blbrNo'] := SPL002D1.dt_BoardList.FieldByName('blbrNo').AsInteger;

        //
        SendLinkData.CreateOnlyOne := False;

        //
        SendLinkData.ShowForm := True;

        // (, )
        SendLinkData.SenderObject := Sender;

        // ( ldfpParentFormCenter)
        SendLinkData.FormPosition := ldfpParentFormCenter;

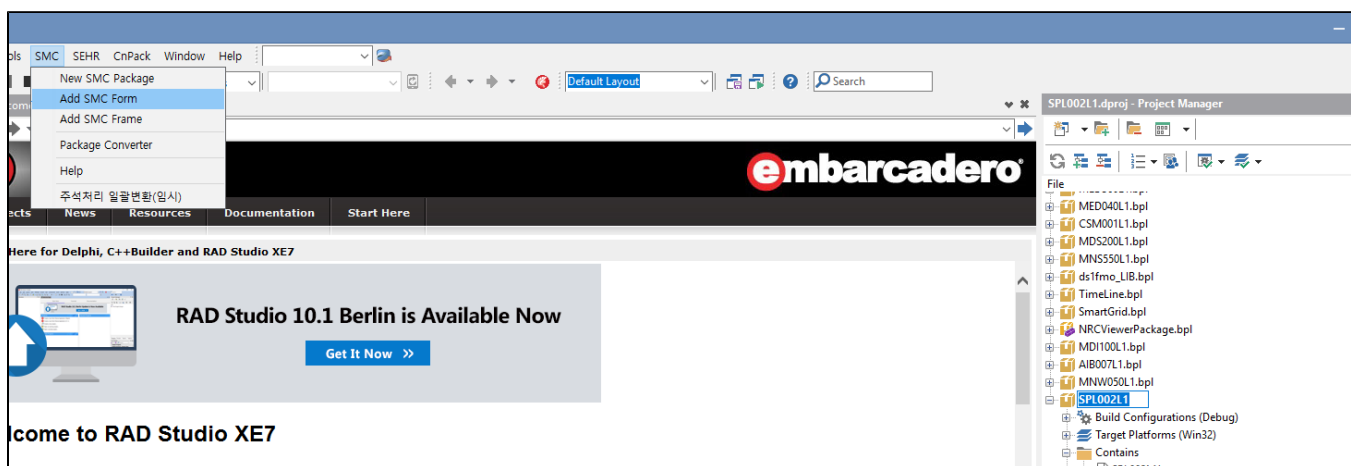
        // Method Call
        SendLinkData.Execute;
    except
        ShowMessage(' '); //
    end;

end;

```

3) 수정화면 추가(SPL002U3)

프로젝트 선택 후 메뉴 - SMC - Add SMC Form을 선택합니다.



폼정보를 입력한 후 생성버튼을 클릭합니다.

New Form

Unit Name

SPL002U3

Form Caption

게시물 수정

Description

게시물 수정

Category

서브시스템명 :

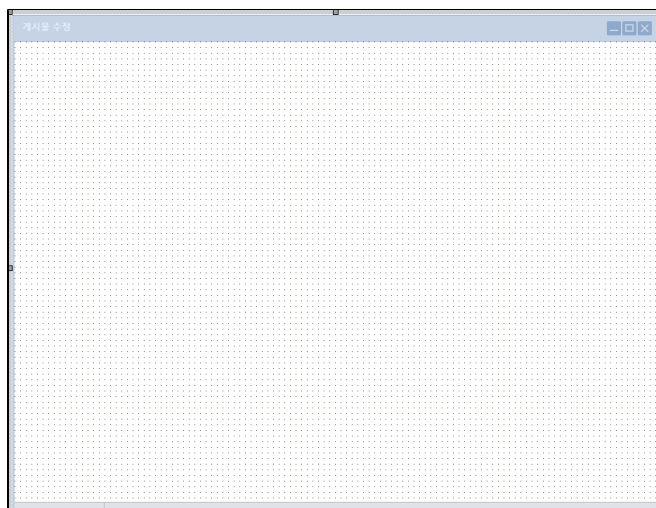
업무분류명(대) :

업무분류명(중) :

생성

취소

폼이 생성된 것을 확인합니다.



uses에 DataModule을 추가합니다.

```

uses
  Windows, Messages, SysUtils, Classes, Forms, Controls, StdCtrls,
  Dialogs, Graphics, SMCMessagesDialog, SMCForm, Buttons, ExtCtrls,
  Variants, DateUtils, cxGraphics, cxControls, cxLookAndFeels,
  cxLookAndFeelPainters, cxContainer, cxEdit, dxSkinsCore,
  dxSkinsDefaultPainters, Vcl.Menus, cxButtons, SMCButtonControls,
  SMCLayoutControls, cxMemo, SMCInputControls, cxTextEdit, SMCViewControls,
  SPL002M1;

```

데이터 모듈을 변수로 추가합니다.

```
private  
    SPL002D1: TSPL002D1;  
public
```

Title, Contents를 입력하기 위해 TSMCTextEdit, TSMCLabel, TSMCCommandButton을 추가합니다.

게시물 수정

Writer

Title

Contents

3

0 수정 1 종료

SPL002F3

게시물 번호를 처리하기 위한 public 변수를 추가합니다.


```

    procedure SMCFormCreate(Sender: TObject);
private
    { Private declarations }
public
    blbrNo : Integer;
end;

```

Form Create 이벤트에 다음과 같이 코딩합니다.

```

procedure TSPL002F3.SMCFormCreate(Sender: TObject);
begin
    if not Assigned(SPL002D1) then
    begin
        SPL002D1:= TSPL002D1.Create(Self);
        SPL002D1.Name:= ' ';
    end;
end;

```

Form LinkDataReceive이벤트에 다음과 같이 코딩합니다.

```

procedure TSPL002F3.SMCFormLinkDataReceive(Sender: TObject; AReceiveData,
    AReturnData: TLinkData);
begin
    blbrNo := AReceiveData.Integers['blbrNo'];

    SPL002D1.dp_Board.Request.FieldByName('blbrNo').AsInteger := blbrNo;
    SPL002D1.dp_Board.Start(Self);

    txed_Writer.Text := SPL002D1.dt_Board.FieldByName('writer').AsString;
    txed_Title.Text := SPL002D1.dt_Board.FieldByName('title').AsString;
    mm_Contents.Text := SPL002D1.dt_Board.FieldByName('contents').AsString;

end;

```

4. 수정

1) DataProvider 추가

데이터 모듈에 TSHISDataProvider를 추가하고 더블클릭 - 수정 서비스 입력 - 조회버튼을 클릭합니다. (이름 : ExmUpdBoardSID)

Provider dp_BoardUpdate Editor

Properties

이름	ExmUpdBoardSID	
대기 모드	<input type="radio"/> 예	<input checked="" type="radio"/> 아니오
공통서비스	<input type="radio"/> 예	<input checked="" type="radio"/> 아니오
Null 을 0 으로 대체	<input checked="" type="radio"/> 예	<input type="radio"/> 아니오
실행 모드		
대기시간		
데이터 크기		

RequestResponse

이름	ExamBoardSVO	
설명	ExamBoardSVO	
종류	업무	
버전	1.0	

Name	Description	Type	Type2	Size	Scale	Count	Reference
title	TITLE	S	V	256	0	1	
contents	CONTENTS	S	V	256	0	1	
blbrNo	BLBR_NO	I	N	256	0	1	
writer	WRITER	S	V	256	0	1	
examBoardDVOList	examBoardDVOList	R	S			30	

F2 눌러시면 수정모드 F3 눌러시면 수정완료 BLOB 인 경우 Type 을 M으로 수정해주세요
List 객체가 아닌 단건 VO 를 사용하신다면 Count 를 1 로 해주세요.

저장

취소

2) 수정 기능 코딩

수정 화면의 수정 버튼 이벤트에 다음과 같이 코딩합니다.

```
procedure TSPL002F3.cmbtn_SaveClick(Sender: TObject);
begin

    SPL002D1.dp_BoardUpdate.Request.FieldByName('writer').AsString := string(txed_Writer.Text).Trim;
    SPL002D1.dp_BoardUpdate.Request.FieldByName('title').AsString := string(txed_Title.Text).Trim;
    SPL002D1.dp_BoardUpdate.Request.FieldByName('contents').AsString := string(mm_Contents.Text).Trim;

    SPL002D1.dp_BoardUpdate.Start(Self);

    ModalResult := mrOk;

end;
```

5. 삭제

1) DataProvider 추가

데이터 모듈에 TSHISDataProvider를 추가하고 더블클릭 - 수정 서비스 입력 - 조회버튼을 클릭합니다.

Provider dp_BoardDelete Editor

Properties

이름	[ExmDelBoardSID]	
대기 모드	<input type="radio"/> 예	<input checked="" type="radio"/> 아니오
공통서비스	<input type="radio"/> 예	<input checked="" type="radio"/> 아니오
Null 을 0 으로 대체	<input checked="" type="radio"/> 예	<input type="radio"/> 아니오
실행 모드		
대기시간		
데이터 크기		

RequestResponse

이름	ExamBoardSVO
설명	ExamBoardSVO
종류	업무
버전	1.0

Name	Description	Type	Type2	Size	Scale	Count	Reference
title	TITLE	S	V	256	0	1	
contents	CONTENTS	S	V	256	0	1	
blbrNo	BLBR_NO	I	N	256	0	1	
writer	WRITER	S	V	256	0	1	
> examBoardDVOList	examBoardDVOList	R	S			30	

F2 눌러시면 수정모드 F3 눌러시면 수정완료 BLOB 인 경우 Type 을 M으로 수정해주세요
List 객체가 아닌 단건 VO 를 사용하신다면 Count 를 1 로 해주세요.

저장

취소

2) 삭제 기능 코딩

목록 화면의 삭제 버튼 이벤트에 다음과 같이 코딩합니다.

```
procedure TSPL002F1.qrbtn_DeleteClick(Sender: TObject);
begin

    if SPL002D1.dt_BoardList.RecordCount <= 1 then
        Exit;

    SPL002D1.dp_BoardDelete.Request.FieldByName('blbrNo').AsInteger := SPL002D1.dt_BoardList.FieldByName
('blbrNo').AsInteger;

    SPL002D1.dp_BoardDelete.Start(Self);
    findBoard;

end;
```