

OZ Application Course

Business Application Development Solution OZ Application



(주)포시에스

목 차

Chapter 1 Application Overview

Lesson 1. Introduction	2
Lesson 2. Overview	4

Chapter 2 OZ Designer

Lesson 1. 쿼리 디자이너 사용자 인터페이스	13
Lesson 2. 리포트 디자이너 사용자 인터페이스	18

Chapter3 Making Dataset

Lesson 1. DataSet 작성	27
Lesson 2. 사용자 파라미터 이용	32

Chapter 4 Designing UserInterface

Lesson 1. Application Designer Start	35
Lesson 2. 기본 컴포넌트 제어	47
Lesson 3. 복합 컴포넌트 제어	61
Lesson 4. 공통 모듈 사용하기	68
Lesson 5. Data Action	76
Lesson 6. 오즈 리포트 연동	

Chapter 5 Using Viewer

Lesson 1. Application Viewer 호출	81
---------------------------------	----

Appendix

OZEdu.mdb를 ODBC로 이용하기	85
-----------------------	----

Chapter 1

OZ Application Overview

Lesson 1. Introduction

Lesson 2. Overview



Lesson 1. Introduction

Introduction

● 본 교재의 목적

본 교육 교재는 오즈 퀘리 디자이너(이하 '퀘리 디자이너'라 칭함)와 오즈 애플리케이션 디자이너를 사용하여 웹 화면 UI 프로그램을 작성하는 사용자들에게 다양한 화면을 작성하기 위한 기능의 사용법을 숙지 시키고자 제작되었습니다.

● 본 교재의 구성

- Chapter 1** 교육에 대한 소개 및 OZ Overview
- Chapter 2** 퀘리 디자이너와 어플리케이션 디자이너의 인터페이스
- Chapter 3** 퀘리 디자이너에서 데이터셋 작성하기
- Chapter 4** 어플리케이션 디자이너를 이용하여 화면 만들기
- Chapter 5** 브라우저에서 배포하기



Lesson 2. OZ Application Overview

Business Application Development Solution

개발 배경

인터넷의 급속한 확산으로
인한 업무 스타일의 변화

웹을 기반으로 하는
비즈니스 전환 가속화

웹 기반의 새로운 업무 및
비즈니스 대거 출현

웹 시스템 개발에 막대한
비용과 노력 투자

전문 웹 개발자 확보의
어려움과 비효율성

**보다 손쉽고 효과적인
비즈니스 애플리케이션
구축 방법은?**

Business Application Development Solution

제품 개요



A. 효율적이고 최적화된 개발 프로세스

- 전문 웹 개발툴 도입으로 개발 기간 단축
- 효율적인 자원활용으로
개발 생산성 증대와 소요비용 최소화

B. 고품질의 개발 결과를 및 유연한 확장성

- 표준화된 개발 플랫폼 적용으로
웹 기반 서비스의 품질 극대화
- 환경 변화에 신속하고 능동적인 대응

Business Application Development Solution

제품 컨셉

완벽한 웹 애플리케이션 개발 및 보고서 서비스 환경 구축

비즈니스 로직

- 데이터베이스 연결
- 질의 생성

프리젠테이션 로직

- UI 디자인
- 컴포넌트 생성
- 질의 데이터 조작
및 로직 생성

각종 보고서

- 업무 양식 디자인
- 분석 보고서 생성

운영환경

- 대규모 사용자
환경에 대한
안정적인 서비스
- 데이터 보안

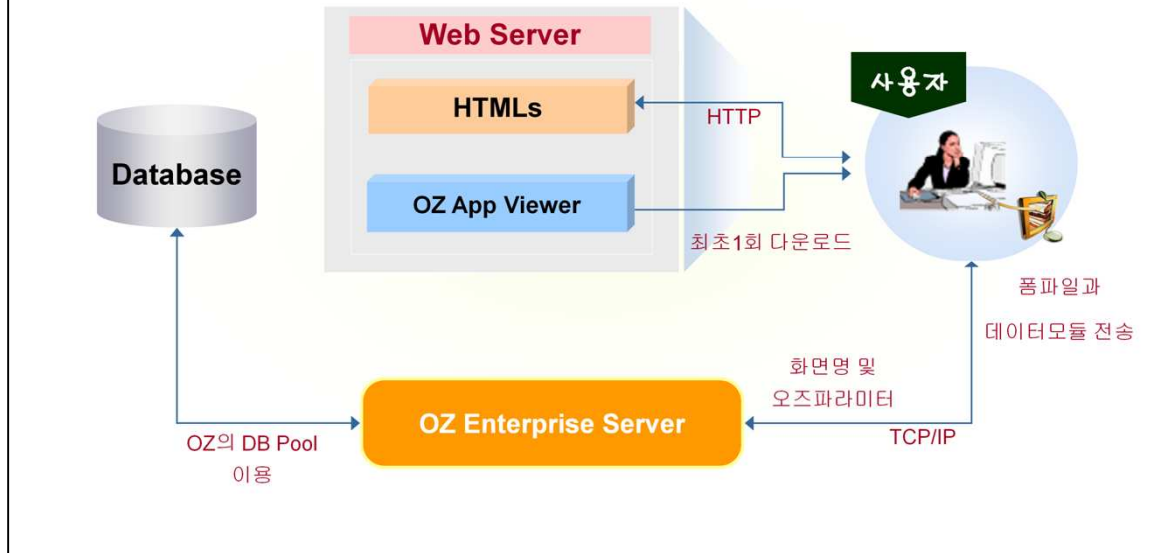
Introduction to OZ



- ❖ **OZ Query Designer**
오즈 리포트 디자이너에서 사용할 데이터셋을 만들고 관리하는 개발 툴입니다.
 - 데이터베이스 연동 : SQL(Multi-SQL 지원) 문장을 이용하여 데이터베이스와 연동하게 됩니다.
- ❖ **OZ Report Designer**
오즈 보고서로 보여주기 위한 보고서 폼 디자인 개발 툴입니다.
 - 객체 기반의 보고서 제작 툴로 양식 그리기 및 속성 정의
- ❖ **OZ Application Designer**
UI 를 구성하기 개발 툴입니다.
 - 폼 디자인 및 스크립트 작성 등을 통해 화면을 구성하고 클라이언트 단의 로직을 구현합니다.
- ❖ **OZ Server**
오즈 보고서 폼을 저장/관리/배포하며, DB에 접속하여 오즈 보고서에 필요한 데이터 모듈을 생성, 배포합니다.
 - 데이터베이스 질의 결과에 대한 데이터셋을 생성하고 이를 클라이언트에 전송
(전송 파일 : 보고서 폼과 데이터 모듈)
 - OZ Repository 관리
 - OZ Server는 독립 구동형인 Daemon Type과 WAS 연동형인 Servlet Type으로 구분
- ❖ **OZ Repository Manager**
오즈 서버에 저장되어 있는 보고서 폼(*.OZR)과 데이터셋 파일(*.ODI)을 관리하는 툴입니다.
- ❖ **OZ Viewer**
클라이언트 컴퓨터에 설치되어 오즈 서버로부터 받은 오즈 폼과 데이터 모듈을 바인딩하여 하나의 보고서로 보여줍니다.
 - ActiveX 형태로 웹 서버를 통해 다운로드
 - 서버에서 받은 보고서 폼과 데이터를 출력 양식으로 변환
 - OZ Report Viewer : 출력 양식의 미리보기 및 파일 저장, 프린트 출력
 - OZ Application Viewer : 사용자의 제어에 따른 이벤트를 받아 서버에 전송하여 데이터 조작 가능
- ❖ **OZ Scheduler Server**
주기적인 보고서 처리를 위한 Daemon으로 지정된 시간에 보고서를 생성하고 배포합니다.
- ❖ **OZ Enterprise Manager**
오즈 서버에 대한 시스템적 성능정보표시/스케줄러 서버가 수행할 Task의 등록 및 관리/복수개의 서버에 대한 중앙 집중적 관리를 합니다.

OZ Application Operation Flow

OZ Server Daemon Type



❖ Standalone 형식(독립 서버 구조)입니다.

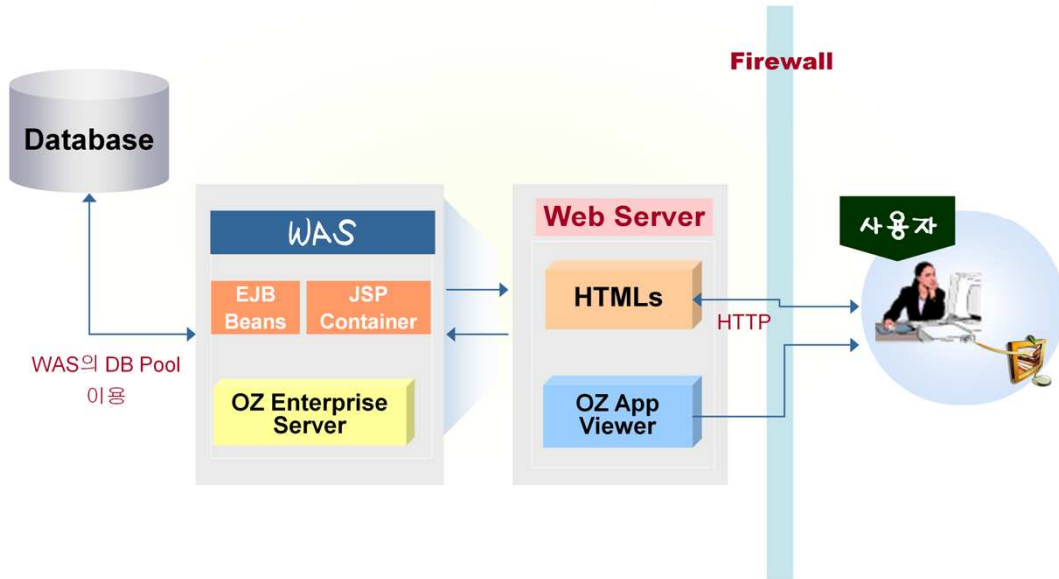
❖ OZ Enterprise Server는 Web Server와 같이 독립적으로 구동되며, 독립적인 포트를 이용합니다.

❖ Work Flow

- 웹 서버로부터 받은 html 에 Embedding 되어 있는 뷰어 정보를 확인
- 오즈 뷰어 다운로드(최초 1회만)와 오즈 서버에 보고서를(폼 파일과 데이터 모듈) 요청함
- 오즈 서버는 Repository 에 저장되어 있는 폼 파일과 데이터베이스와의 연동을 통해 생성한 데이터 모듈 파일을 사용자에게 전달함
- 사용자의 Action 에 따라 뷰어와 서버가 통신을 하면서 애플리케이션 구동

OZ Application Operation Flow

OZ Server Servlet Type



- ❖ WAS(Web Application Server)와 연동되는 구조
- ❖ 웹 서버가 이용하는 포트 이용하여 오즈 뷰어 다운 및 오즈 서버 연결
- ❖ OZ Enterprise Server의 자체 DB Pool 이용 또는 WAS의 DB Pool 이용
(오즈서버의 `conf/db.properties`파일에서 설정)

시스템 요구사항

Server

JAVA Runtime Environment(이하 JRE) 1.1.x 이상이 가동되는 모든 platform

Designer

Windows NT/2000(Service pack2 이상)/XP/2003

Pentium III, 500MHz 이상 권장, HDD 100MB

Minimum 256MB, Suggested 512 MB Memory

(Windows 95/98/ME 에서는 Unicode를 지원하지 않으므로 사용할 수 없습니다.

Viewer

뷰어의 탑재는 어느 웹 서버에서나 가능

ActiveX Viewer 사용

Chapter 2

OZ Designer

Lesson 1. 쿼리 디자이너 사용자 인터페이스

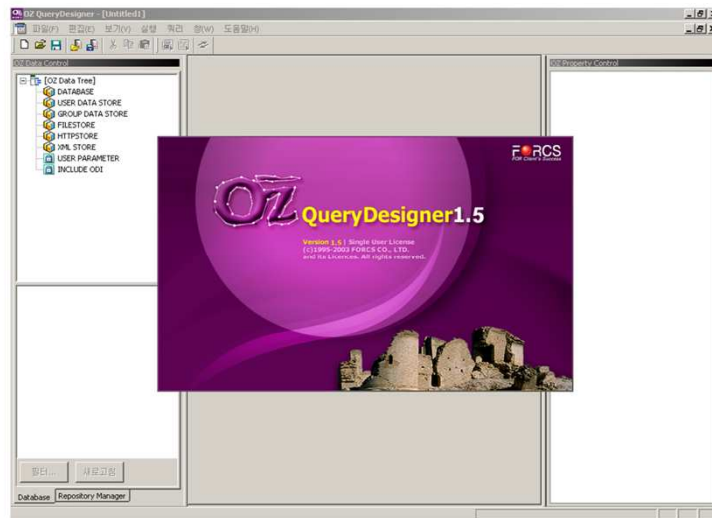
Lesson 2. 애플리케이션 디자이너 사용자 인터페이스

Lesson 1. 퀵리 디자이너 사용자 인터페이스

교육목표

- 퀵리디자이너 설치 및 실행 하기
- 퀵리디자이너의 구성 및 기능 습득

Query Designer 실행하기

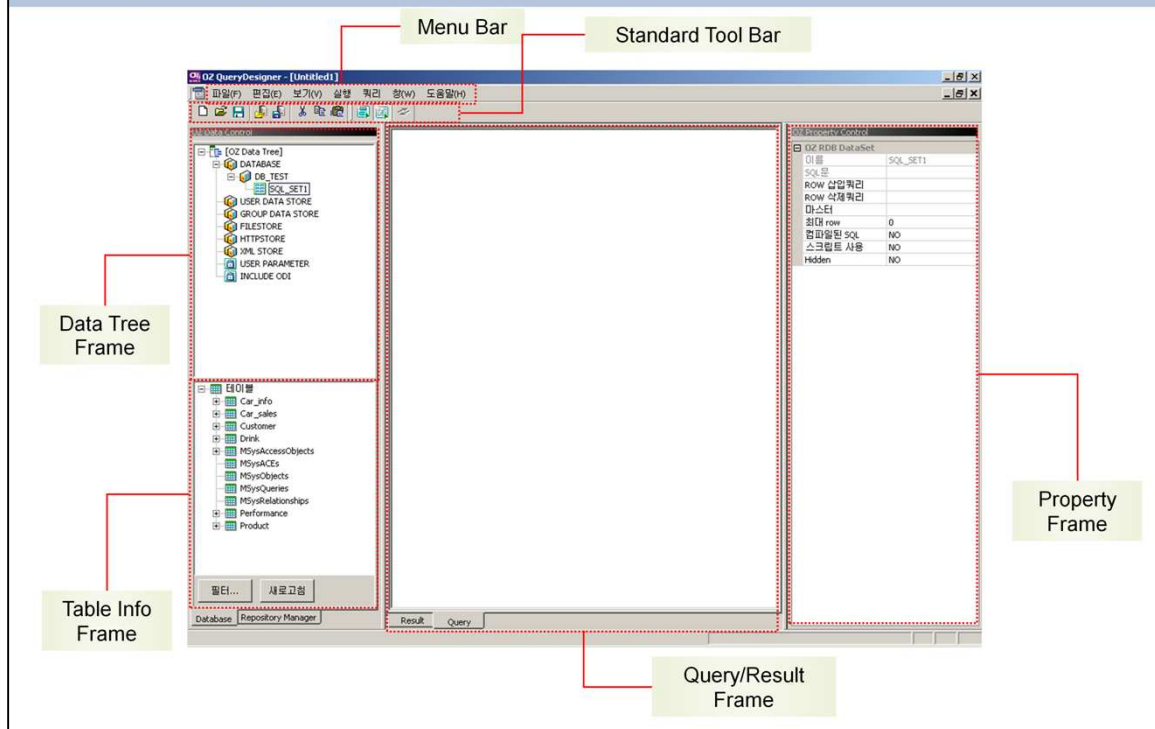


❖ Query Designer 실행 방법

- Query Designer 아이콘을 클릭해서 실행
- Report Designer의 [FILE] - [쿼리디자이너 실행] 으로 실행
- {오즈 3.0 설치 디렉토리}\ QueryDesigner\ bin\ OZQueryDesigner.exe를 실행

❖ Query Designer의 실행 중 로그는 실행파일이 있는 디렉토리에 console.txt에 기록됩니다.

Query Designer – User Interface

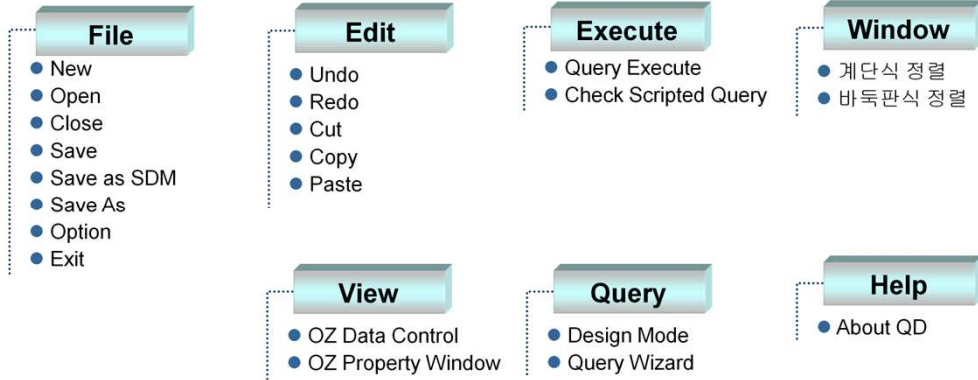


❖ Query Designer는 크게 4개의 부분으로 나눌 수 있습니다.

- Data Tree Frame은 사용자가 작성하는 Database 정보 및 Dataset 정보를 가지고 있습니다.
- Query/Result Frame은 사용자가 입력한 SQL 문장 및 실행 결과를 가지고 있습니다.
- Property Frame은 Database/Dataset의 속성값을 변경할 수 있습니다.
- Table Info Frame은 현재 연결된 Database의 조회 가능한 테이블 정보를 가지고 있습니다.

Query Designer – Menu Bar

파일(F) 편집(E) 보기(V) 실행 쿼리 창(W) 도움말(H)



Query Designer - Tool Bar



파일 관련 부분

새페이지, 열기, 저장

편집 부분

자르기, 복사, 붙이기

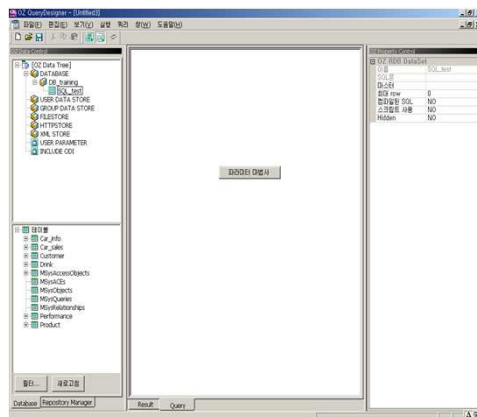
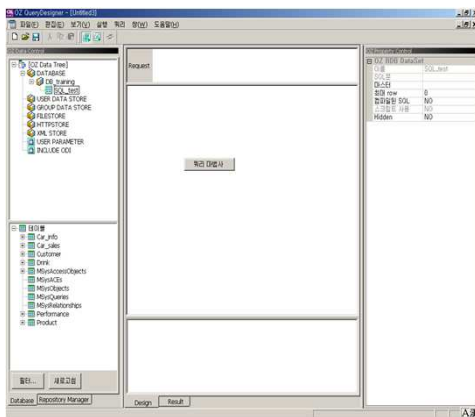
모드 전환

디자인뷰 전환, 매뉴얼 전환

실행 부분

쿼리문 실행

❖ 디자인뷰 모드와 매뉴얼 모드



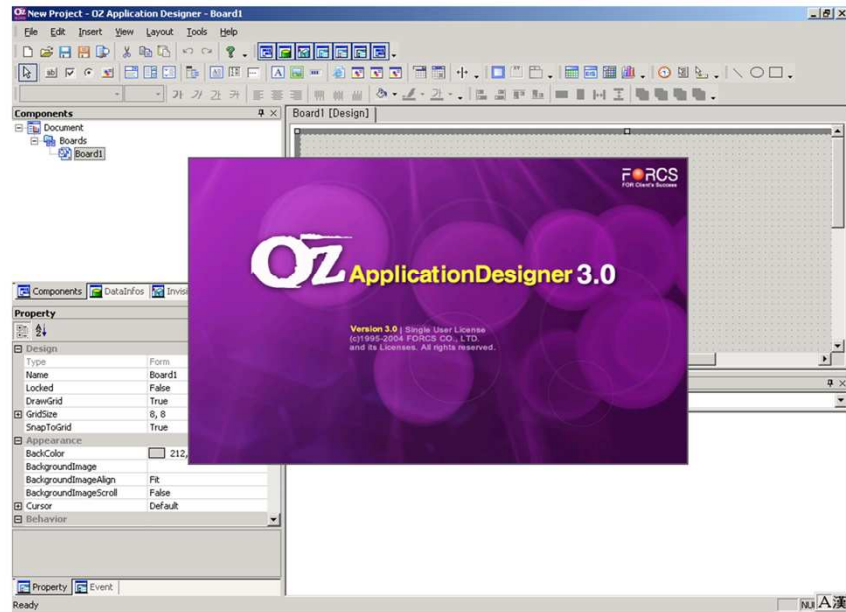
- 디자인 모드 실행 시 오른쪽 마우스를 클릭할 경우 쿼리 마법사창이 뜹니다.
- 매뉴얼 모드 실행 시 오른쪽 마우스를 클릭할 경우 파라미터 마법사창이 뜹니다.

Lesson 2. 어플리케이션 디자이너 사용자 인터페이스

교육목표

- 어플리케이션 디자이너 설치 및 실행 하기
- 어플리케이션 디자이너의 구성 및 기능 습득

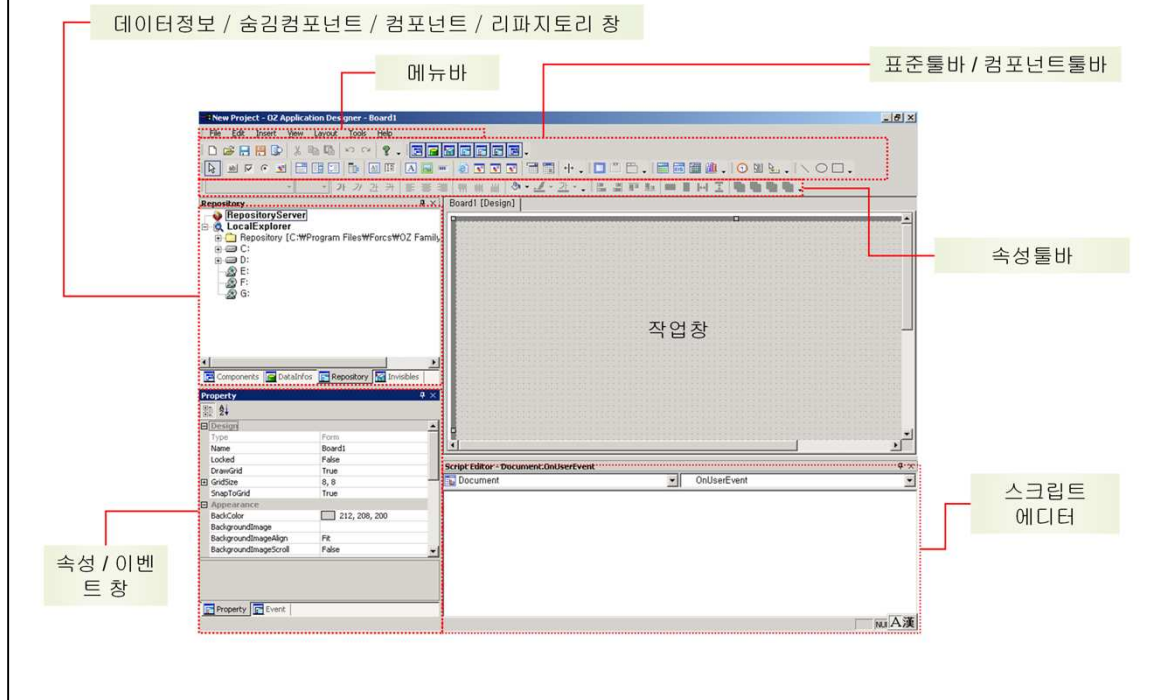
Application Designer 실행하기



❖ 실행

- Application Designer 아이콘을 클릭
- {오즈 Application 설치 디렉토리}\ bin\ OZADesigner.exe 실행

Application Designer - User Interface



❖ 메뉴바

오즈 애플리케이션의 메인 메뉴이며 풀다운(full down) 형식으로 제공합니다.

❖ 표준툴바

메뉴를 빠르게 사용할 수 있도록 아이콘 형태의 툴바로 제공합니다.

❖ 컴포넌트툴바

컴포넌트를 빠르게 추가할 수 있도록 컴포넌트 툴바를 제공합니다.

❖ 속성툴바

컴포넌트의 기본 속성을 빠르게 설정할 수 있도록 속성 툴바를 제공합니다.

❖ 리파지토리창

리파지토리에 대한 접속 및 품(oz)파일, 데이터정보(odt)파일을 체크인, 체크아웃하는 기능, 품파일에 데이터정보 파일을 추가하는 기능등을 수행할 수 있는 창입니다.

❖ 컴포넌트창

디자인에 사용된 컴포넌트들을 트리형태로 보여주는 창입니다.

❖ Invisible 컴포넌트창

디자인에 사용된 Invisible 컴포넌트들을 트리형태로 보여주는 창입니다.

❖ 데이터정보창

품에 바인딩된 데이터 파일의 데이터셋, 데이터필드, OZ 파라미터 등의 정보를 트리형태로 보여주는 창입니다.

❖ 속성창

컴포넌트의 속성을 설정할 수 있는 창입니다.

❖ 이벤트창

컴포넌트의 이벤트를 확인하고 스크립트를 작성하기 위한 창입니다.

❖ 작업창

Board 를 추가시키고 컴포넌트를 올려 놓을 수 있는 작업 공간입니다.

❖ 스크립트편집창

스크립트를 편집할 수 있는 창입니다.

Application Designer - Menu Bar

File **E**dit **I**nsert **V**iew **L**ayout **T**ools **H**elp

File

- Create New Project
- Open Project
- Close Project
- Save Project
- Save Project As
- Add New Board
- Remove Board
- Insert New ODI
- Insert New Style
- Preview
- Exit

Edit

- Undo
- Redo
- Cut
- Copy
- Paste
- Delete
- Select All
- Find
- Replace
- Export Script
- Import Script
- Preview Script

Insert

- Controls
- Invisibles
- Container Controls
- Complex Controls
- Drawings

Tools

- Default Value Editor
- Options...
- External Tools

View

- Component Explorer
- Invisible Explorer
- DataInfo Explorer
- Repository Explorer
- Property Grid
- Event Grid
- Script Editor

Help

- Help
- About OZ Application

Layout

- Style
- Align
- Make Same
- Make
- Spacing Equal
- Order

Application Designer - Tool Bar



파일 관련 부분

새 프로젝트, 열기, 저장, 새이름으로 저장

미리보기 부분

미리보기

편집 부분

자르기, 복사, 붙이기, Undo, Redo

도움말

도움말

작업 창 부분

Component Explorer, Invisible Explorer, DataInfo Explorer, Repository Explorer
Property Grid, Event Grid
Script Editor

Application Designer – Component Tool Bar



Controllable	Button, CheckBox, RadioButton
리포트 Component	ReportButton, ReportControl
리스트 Component	ComboBox, ListBox, CheckListBox, TreeView
텍스트 Component	TextBox, NumericUpDown, MaskedTextBox
View 용 Component	Label, PictureBox, ProgressBar, HTMLView
ActiveX Component	ActiveXControl, CustomControl
Date Component	DateTimePicker, MonthCalendar
Container	Splitter, Panel, GroupBox, TabControl

Application Designer – Component Tool Bar



Complex

Table, Grid, CrossTab, Chart

Invisible

ReportButton, ReportControl

Drawings

ComboBox, ListBox, CheckListBox, TreeView

Application Designer - Hotkey Tool Bar



폰트	컴포넌트, 라벨 등의 객체를 앞으로 또는 뒤로 이동
폰트 스타일	굵게, 이탤릭, 밑줄, 취소선
텍스트 Alignment	텍스트 정렬
색상 설정	컴포넌트의 채우기, 선색, 글색 설정
맞춤	왼쪽, 오른쪽, 위로, 아래로 정렬
높이,너비,수직,수평 간격조정	컴포넌트간 높이, 너비 조정, 간격 조정
컴포넌트 이동	컴포넌트를 왼쪽, 오른쪽, 위로, 아래로 이동

Chapter 3

Making DataSet



Lesson 1. DataSet 작성

Lesson 2. OZ Parameter

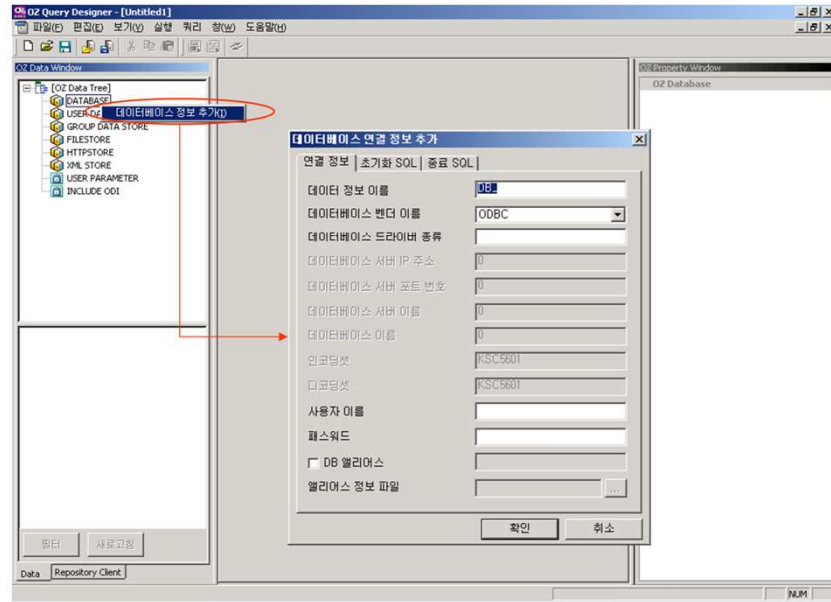
Lesson 1. Dataset 작성

교육목표

- 데이터베이스에 연결하기
- 데이터베이스에 SQL문을 입력하여 Dataset을 생성

데이터베이스 연결하기

새 문서에서 데이터베이스 연결을 위한 [데이터베이스 정보 추가]를 선택합니다.



- ❖ [데이터 정보 이름]는 사용자가 지정하는 Database의 앨리어스 이름입니다.
- ❖ 해당 DBMS 종류에 따라 필요 항목이 활성화 됩니다.

DBMS별 설정방법(계속)

< DB Alias를 이용한 연결>

< User-defined(MySQL JDBC)를 이용한 연결>

❖ DB Alias를 이용한 연결

: 오즈 서버의 db.properties를 쿼리 디자이너와 리포트 디자이너가 설치된 컴퓨터에 위치시킵니다.

- DB Alias : db.properties 파일에 설정된 DB 연결을 위한 Alias명 설정
- Alias Info File Name : db.properties의 경로 및 파일명
- db.properties

```
#
# sample configuration for Oracle
#
sales1.vendor=oracle
sales1.serverAddress=192.168.0.2
sales1.portNo=1521
sales1.sid=ORCL
sales1.user=oz
sales1.password=oz
sales1.maxconns=20
sales1.initconns=5
sales1.timeout=5
```

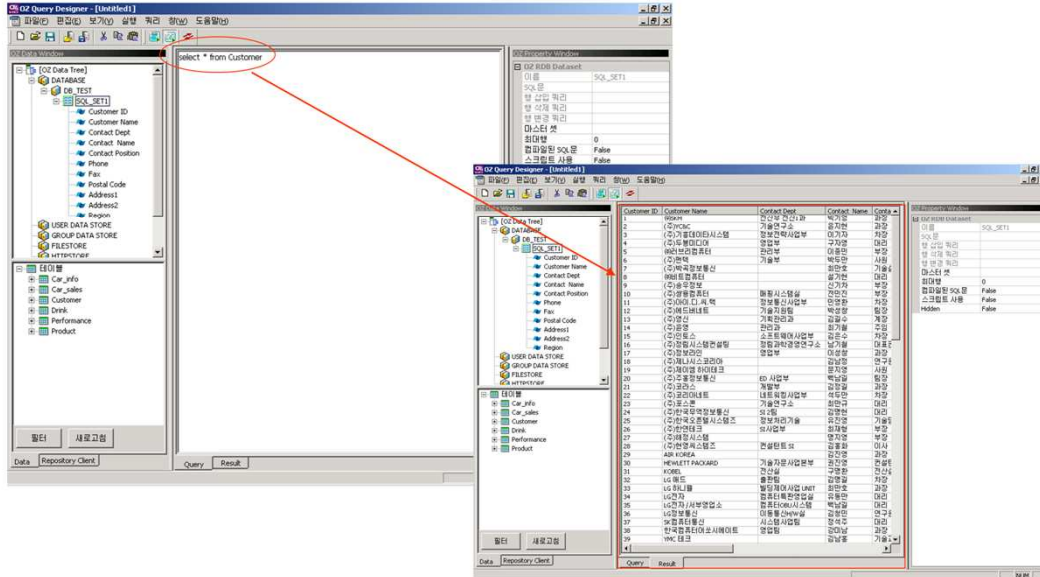
DB Alias

❖ User-Defined를 이용한 연결

현재 데이터베이스 밴드 리스트에 등록되어 있지는 않지만 JDBC를 제공하는 DB의 경우 User-Defined를 선택하고 [JDBC Info]탭에서 해당 JDBC 드라이버의 JDBC Driver Name, Connection URL을 입력하여 연결할 수 있습니다. 쿼리 디자이너를 시작하기 전에 해당 JDBC Driver 파일은 클래스 패스에 잡혀져 있어야 합니다.

SQL 실행문 생성 - 디자인뷰

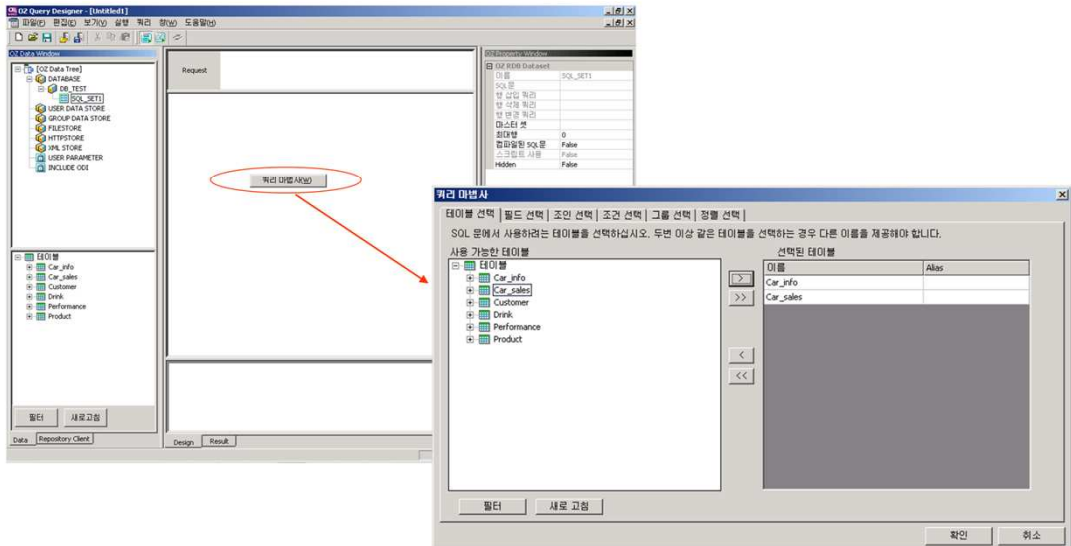
Query를 직접 입력하여 F5(단축키)로 실행시키면 결과창으로 전환됩니다.



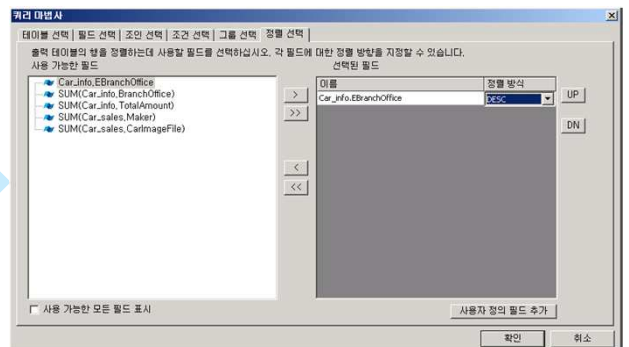
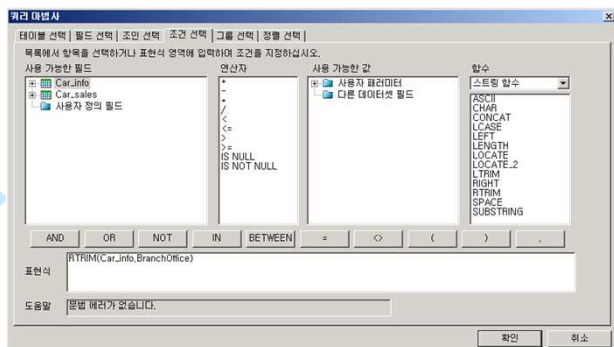
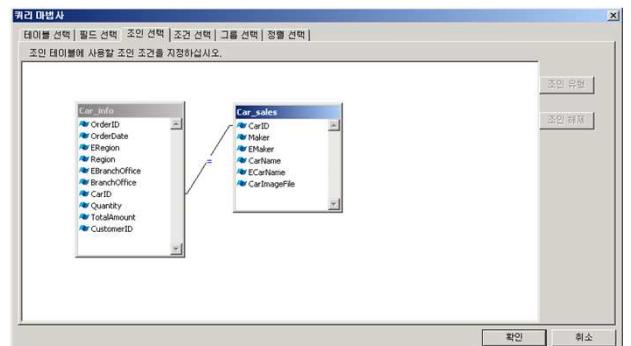
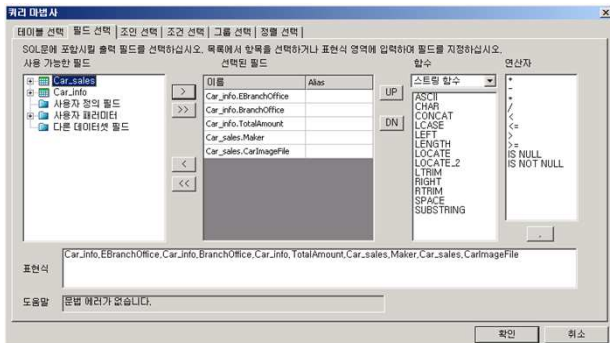
- ❖ [Add Query Dataset]을 이용하여 [Query Dataset] 윈도우 실행
- ❖ [Query Dataset ID]는 해당 쿼리의 사용자 정의 이름
- ❖ [Query Statement]에 쿼리를 직접 입력하거나 [Query Wizard]를 이용하여 쿼리 입력
- ❖ 입력한 쿼리는 [쿼리문 실행]을 통해 직접 확인할 수 있습니다.

SQL 실행문 생성 - 쿼리 마법사

쿼리 마법사는 Query 생성을 도와주는 마법사로서 쿼리 생성에 필요한 구문 필드를 선택하는 방식으로 쉽게 쿼리를 생성할 수 있도록 합니다.



❖ 쿼리 마법사를 이용한 자동 쿼리문 생성 과정



Lesson 2. OZ Parameter

교육목표

- OZ Parameter 생성 및 사용 방법

ODI Parameter

OZ User Parameter

보고서에 대한 사용자 정의 외부 패러미터 변수로 외부사용자 애플리케이션에서 넘어온 값을 사용하기 위한 패러미터 데이터 소스입니다.

OZ User Parameter의 생성



Chapter 4

OZ Application Designer

Lesson 1. Application Designer Start

Lesson 2. 기본 컴포넌트 제어

Lesson 3. 복합 컴포넌트 제어

Lesson 4. 공통 모듈 사용하기

Lesson 5. Data Action

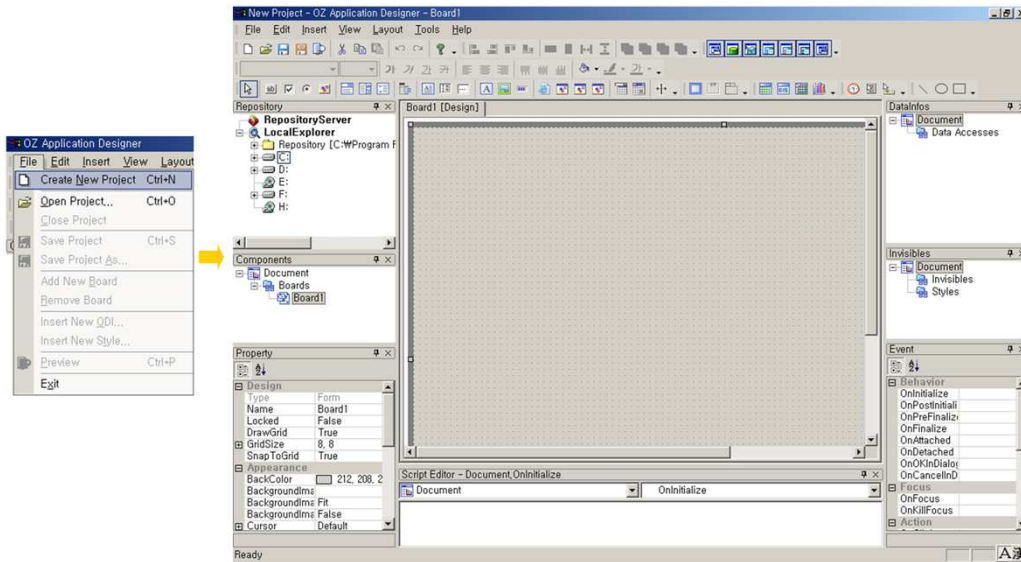
Lesson 6. 오즈 리포트 연동



Lesson 1. Application Designer 사용하기

OZApplication Design Start(1)

프로젝트 추가



❖ 애플리케이션 디자이너를 실행시킨 후 새로운 프로젝트를 만듭니다.

❖ 프로젝트 화면은 메뉴, 툴바, Repository, Components, DataInfos, Invisibles, Property, Event, Script Editor로 구성됩니다.

❖ Repository : 작업위치를 지정합니다. Repository Server는 OZ Server에 있는 Repository를 설정합니다.

❖ Components : Board에 추가된 컴포넌트를 트리구조로 보여줍니다. 최상위 컴포넌트는 Document입니다.

❖ DataInfos : DataSet 과 데이터 필드들을 트리구조로 보여줍니다.

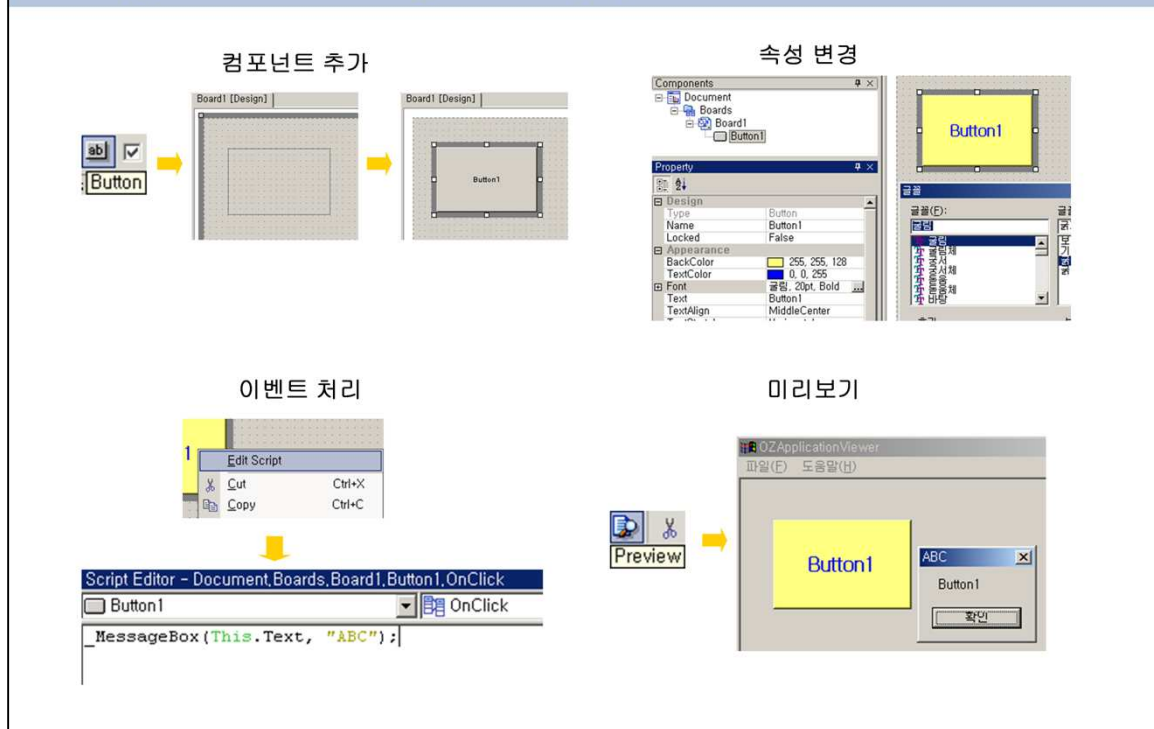
❖ Invisibles : Board에 추가된 컴포넌트 들 중에서 시각적으로 보이지 않는 컴포넌트를 트리구조로 보여줍니다.

❖ Properties : 컴포넌트에 대한 속성을 지정합니다.

❖ Event : 컴포넌트별 이벤트를 나타냅니다.

❖ Script Editor : 이벤트에 대한 스크립트를 작성하는 편집기입니다.

OZApplication Design Start(2)



❖컴포넌트 추가

- 컴포넌트를 툴바에서 선택한 후 Board위에서 드래그 합니다.

❖속성 변경

- Property 창에서 배경색, 폰트, 사이즈 등을 조절합니다.
- 속성을 선택한 후 '...' 버튼을 클릭하면 세부적으로 속성을 선택할 수 있습니다.

❖이벤트 처리

- 이벤트를 처리할 컴포넌트를 선택한 후 팝업메뉴에서 'Edit Script'를 선택합니다.
- Script Editor에서 이벤트를 선택한 후 스크립트를 작성합니다.
- 스크립트에 대한 도움말을 보려면 'F1'키를 클릭합니다.

❖미리보기

- 툴바에서 'Preview' 버튼을 클릭하면 뷰어창이 뜹니다. (Cntrl+P)

❖ 스크립트 주석 처리

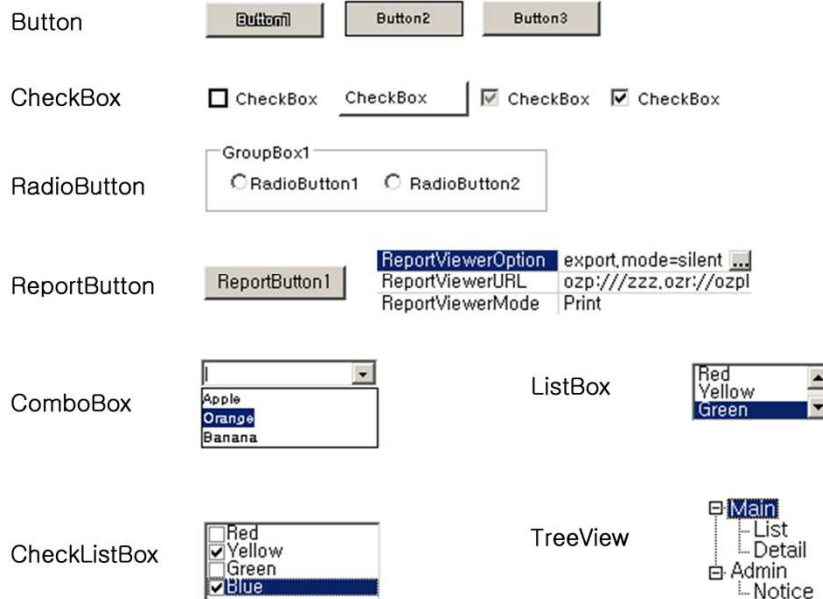
- // 와 /* */
- Cntrl+kc : 주석 처리
- Cntrl+ku : 주석 풀기

Lesson 2. 기본 컴포넌트 제어

교육목표

- 버튼, 텍스트박스 등 단순 컴포넌트들의 속성 및 스크립트 확인
- Event와 Script 사용하기

Component - Controls



❖ Button

- Effect, FlatStyle 등을 지정할 수 있습니다.

❖ CheckBox

- Appearance, CheckState, Check 등을 지정할 수 있습니다.

❖ RadioButton

- 컨테이너에 속해있는 모든 라디오 버튼은 그룹핑됩니다.

❖ ReportButton

- 오즈 리포트 뷰어를 띄우거나 바로 출력하도록 지정할 수 있습니다.

❖ ComboBox, ListBox

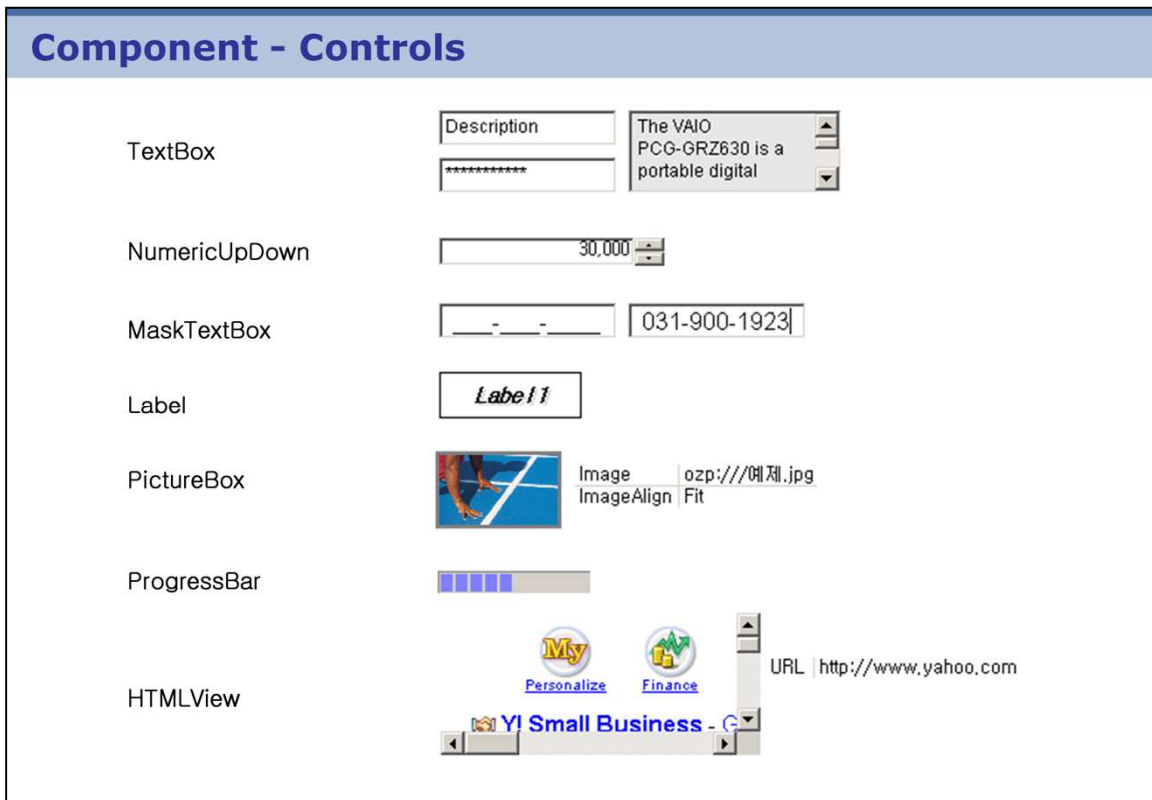
- 마법사를 이용하여 item을 추가할 수 있습니다.
- 각각의 item에 대한 text, value를 가집니다.

❖ CheckListBox

- CheckBox와 ListBox의 속성을 복합적으로 가지고 있는 컴포넌트 입니다.

❖ TreeView

- 마법사를 이용하여 데이터를 트리구조로 나타냅니다. Indent 구분은 Tab으로 지정됩니다.



❖ TextBox

- 멀티라인, WordWrap, 패스워드 문자열 입력, 스크롤바 등이 지원됩니다.

❖ NumericUpDown

- Numeric 데이터를 입력하기 위해 Hexa/Decimal 값 지정, 증가값, 천단위 구분자 지정 등을 지원합니다.

❖ MaskedTextBox

- 전화번호, 주민등록 번호 등의 데이터를 입력할 때 지정된 마스크 이외의 부분에 데이터가 입력되도록 하는 컴포넌트입니다.
- Mask, Literal, Default, PromptChar 속성을 사용합니다.

❖ Label

- Text를 입력합니다.

❖ PictureBox

- 이미지를 표현합니다.

❖ ProgressBar

- 프로그래스바를 표현할 때 사용합니다.

❖ HTMLView

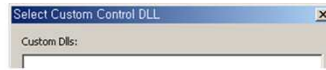
- 웹 페이지를 표현하는 컴포넌트입니다. 속성창에서 URL을 지정합니다.

Component - Controls

ActiveXControl



CustomControl



ReportViewer

ReportViewerOption connection, servlet=http://127.0.0.1/oz/server
ReportViewerURL ozp:///zzz.ozr://ozplocal://C:

DateTimePicker



MonthCalendar



Splitter



Horizontal
Navigator



Vertical
Navigator



❖ ActiveXControl/CustomControl

- 애플리케이션 디자이너에서 제공되지 않는 컴포넌트를 추가 할 수 있습니다.

❖ ReportViewer

- ReportViewer 컴포넌트 영역에 오즈 리포트를 호출하여 보여줍니다.
- 속성창에서 뷰어옵션 및 뷰어URL을 지정합니다.

❖ DateTimePicker

- 달력이 지원되는 날짜 입력 컴포넌트입니다.
- 날짜 포맷, MaxDate, MinDate등을 지정할 수 있습니다.

❖ MonthCalendar

- 달력컴포넌트 입니다.
- 컴포넌트 영역만큼 달력이 보여집니다.

❖ Splitter

- 전체 화면을 분할하고 분할된 영역 사이의 크기를 유동적으로 조절하고자 할 때 사용합니다.

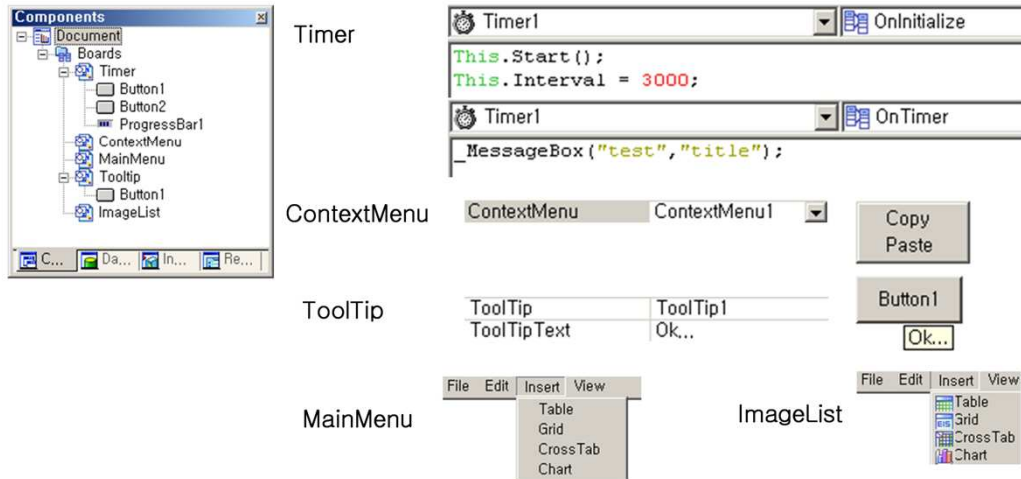
❖ Horizontal Navigator, Vertical Navigator

- Table 컴포넌트와 연동하여 Table에 표시될 Row수를 나누어 페이지를 나타내며 데이터를 조회할 때 이전/다음 페이지로 이동시키는 컴포넌트 입니다.

Component - Invisibles

Invisible Component

- ❖ Board에 포함되지 않아 보이지는 않지만 내부적으로 사용되는 컴포넌트입니다.
- ❖ Invisibles 창에서 확인할 수 있습니다.



❖Timer

- 타이머를 이용하여 Interval 기간마다 이벤트를 발생시킬 수 있습니다.

❖ContextMenu

- 특정 컴포넌트에 대한 팝업메뉴를 만들 수 있습니다.
- 마법사를 이용하여 메뉴 아이템을 추가합니다.

❖ToolTip

- 특정 컴포넌트에 대해 툴팁을 지정할 때 사용합니다.

❖MainMenu

- 특정 컴포넌트에 대해 툴팁을 지정할 때 사용합니다.

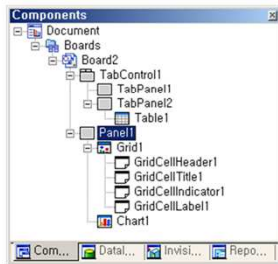
❖ImageList

- ContextMenu, MainMenu, TreeView 등의 각 아이템 앞에 이미지를 표시할 때 사용하는 컴포넌트입니다.

Component – Container Controls

Container Component

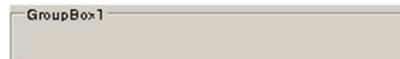
- ❖ 여러 컴포넌트를 그룹을 지을 때 사용되어집니다.
- ❖ 해당 컴포넌트 내에 포함된 모든 컴포넌트에 대한 **Enable, Visible** 속성을 제어할 수 있습니다.



Panel



GroupBox



TabControl

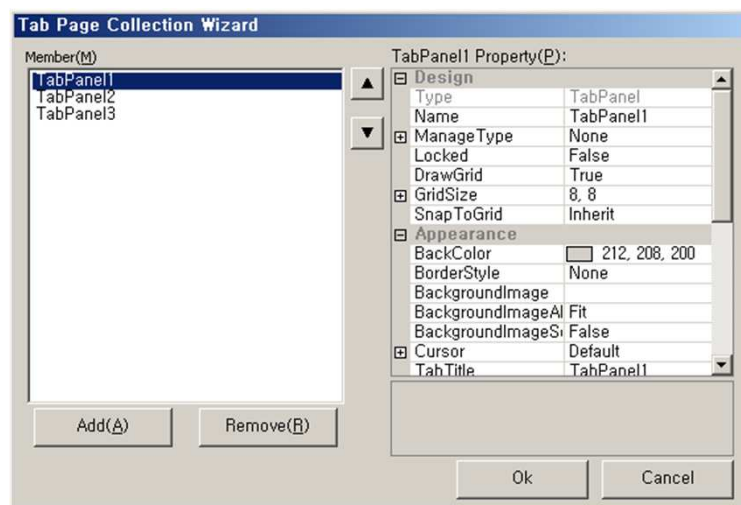


❖ Panel / GroupBox

- 다수의 컴포넌트를 자식 컴포넌트로 포함시킬 때 사용하는 컴포넌트입니다.

❖ TabControl

- 마법사를 이용하여 탭을 추가/삭제할 수 있습니다.
- 탭별로 컴포넌트들을 추가할 수 있으며 다른 탭으로 이동될 때 이전 탭의 데이터가 유지됩니다.



<Tab Control 마법사>

Events & Script

Event 선택

- ❖ 각 컴포넌트에 Event를 선택하고 스크립트를 작성합니다.
- ❖ JavaScript에서 지원하는 기본 기능과 language Spec이 사용됩니다.

OnInitialize	Initialize 직전에 발생
OnPostInitialize	Initialize 가 끝난 후 발생
OnPreFinalize	Finalize 직전에 발생
OnFinalize	Finalize 가 끝난 후 발생
OnSelectedIndexChanged	선택된 item 이 변경된 후 발생
OnItemCheckChanged	아이템의 check 상태가 변경된 후 발생
OnItemSelected	아이템이 선택된 후 발생
OnSelectedNodeChanged	선택된 node 가 변경된 후 발생

< Events의 예 >

❖ 시작/종료 이벤트 실행 순서

- 다음과 같은 순서로 이벤트가 호출 됩니다.

OnInitialize

Document -> Board1 -> Panel1 -> Button1 -> Button2

OnPostInitialize

Button2 -> Button1 -> Panel1 -> Board1 -> Document

OnPreFinalize

Document -> Board1 -> Panel1 -> Button1 -> Button2

OnFinalize

Button2 -> Button1 -> Panel1 -> Board1 -> Document

Component & Events

Script를 이용하여 제어

RadioButton

<체크되었는지 확인>

```
Script Editor - Document.Boards.Board1, Butt...
Button3
OnClick
var temp = RadioButton1.Checked;
if (temp) _MessageBox ("Checked");
else _MessageBox ("UnChecked");
```

DateTimePicker

<선택된 날짜 가져오기>

```
Script Editor - Document.Boards.Board1, ComboBox2, OnSelectedIndexChanged
DateTimePicker1
OnValueChanged
var date = DateTimePicker1.GetCurDateTime();
Label14.Text = date.GetYear() + "년 "
+ date.GetMonth() + "월 "
+ date.GetDay() + "일";
```

CheckBox

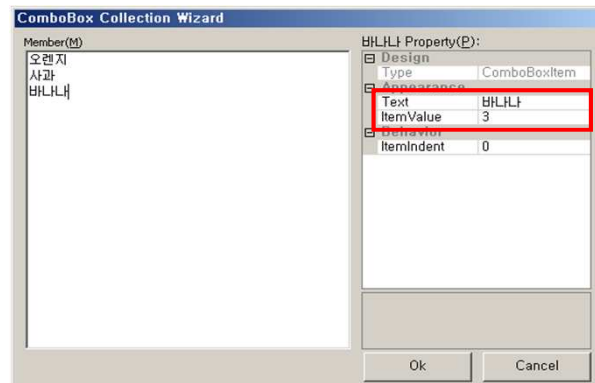
<선택된 Text와 Value
추가 및 가져오기>

```
Script Editor - Document.Boards.Board2, ComboBox3, OnInitialize
ComboBox3
OnInitialize
var comboItem1=ComboBox3.AddItem("꺾은선형");
comboItem1.ItemValue="VLine";
var comboItem3=ComboBox3.AddItem("꺾은선형입체");
comboItem3.ItemValue="VLine_3D";
```

```
Script Editor - Document.Boards.Board1, Button3, OnClick
ComboBox2
OnSelectedIndexChanged
Label12.Text = ComboBox2.Value;
Label13.Text = ComboBox2.GetItemObject(ComboBox2.SelectedIndex).ItemValue;
```

❖ CheckBox 이용

- 방법1 : 마법사를 이용하여 텍스트와 값을 직접 입력



- 방법2 : DataSet을 이용하여 ODI와 필드명을 지정

Property	
AllowDrop	False
ToolTip	
ToolTipText	
Layout	
Location	56, 176
Size	224, 21
Anchor	Top, Left
Dock	None
#DataSource	
ODIKey	car
DataSet	지역
Field	Region
ValueField	

연습문제

과제

여러 컴포넌트를 이용하여 회원 가입 양식 화면을 만듭니다.

조건

- ❖ TextBox, Button, Combobox, CheckBox, RadioButton 을 사용합니다.
- ❖ 다른 Board를 ShowDialog 함수를 이용하여 우편번호 검색 창을 띄웁니다.

```
var b = Form1.ShowDialog("Dialog_Form2", "Title", true, 100, 100);
```
- ❖ “등록” 버튼을 클릭하여 사용자의 Action 결과를 받아 옵니다.

The screenshot displays a software application window titled "OZApplicationViewer" with a menu bar containing "File" and "Help". The main window contains a registration form with the following fields and controls:

- * 야후! 아이디**: A text box containing "@forcs.com" and a "중복확인" (Check Duplicate) button. Below it, a note says "한글, 특수문자 ID를 사용할 수 없습니다." (Cannot use Korean characters or special characters for ID).
- * 이름(성명)**: A text box for the user's name.
- * 주민등록번호**: Two text boxes separated by a tilde (~) for the resident registration number.
- * 비밀번호**: A text box with a red note "6자 이상으로 대소문자 구분하여 입력" (Enter with uppercase and lowercase letters, 6 characters or more).
- * 비밀번호 확인**: A text box for confirming the password.
- * 본인확인 질문**: A dropdown menu with the text "[질문을 선택하세요]".
- * 본인확인 답**: A text box with a red note "한글 2자 이상, 그외 4자 이상, 특수문자 불가" (Korean 2 characters or more, others 4 characters or more, no special characters allowed). Below it, a note says "비밀번호 분실, 변경시 본인확인을 위해 필요한 정보이므로 기억해두세요" (This is necessary information for password recovery and change, so please remember it).
- * 주소**: Two text boxes for the address. A red box highlights a "우편번호검색" (Postcode Search) button next to the first address box. A red arrow points from this button to a separate dialog box.
- * 직장**: A dropdown menu with the text "[직종 선택]".
- * 이메일 수신 선택**: Radio buttons for "회사 메일" (Company Email) and "'다른 이메일 주소'에 입력한 곳으로 받기" (Receive at the address entered in 'Other Email Address'). There is also a checkbox for "중요 서비스 공지 받겠습니다." (I will receive important service notices).
- A "등록" (Register) button is at the bottom right of the form.

A separate dialog box titled "우편번호 검색" (Postcode Search) is shown in the foreground. It contains the following text and controls:

- Text: "거주하시는 지역의 '동' 이름을 입력해 주세요" (Please enter the 'Dong' name of the area you live in). Example: "예 : 삼성동, 역삼동" (Example: Samsung-dong, Yeoksam-dong).
- Text: "우편번호 검색" (Postcode Search) followed by a text box and "동(읍/면/리) 찾기" (Find Dong (Eup/Myeon/Li)).
- Text: "동(면) 중심으로 검색어를 입력해 주세요" (Please enter the search word centered on the Dong (Myeon)).

A red arrow points from the "우편번호검색" button in the main form to the dialog box, illustrating the requirement to open a separate board for postcode search.

Lesson 3. 복합 컴포넌트 제어

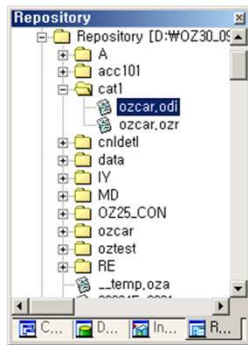
교육목표

- Table, Grid, CrossTab, Chart 컴포넌트의 구현
- 복합 컴포넌트의 마법사 사용방법

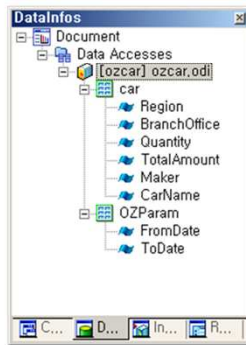
Component – Table(1)

테이블의 데이터셋 지정

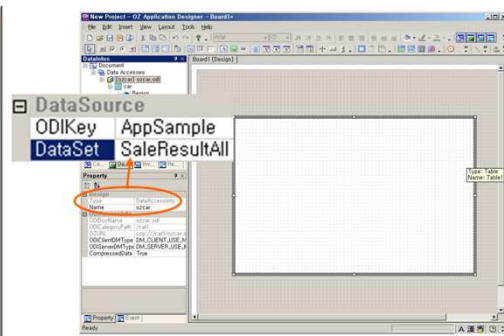
- ❖ Repository창에서 사용될 ODI파일을 더블클릭 하여 선택합니다.
- ❖ DataInfos창에서 선택된 ODI파일의 DataSet명, 필드정보, 파라미터를 확인할 수 있습니다.
- ❖ 테이블 속성창의 DataSource부분에서 ODIKey와 DataSet을 지정합니다.



<Repository창에서 odi호출>



<DataInfos에서 호출된 odi정보 확인>

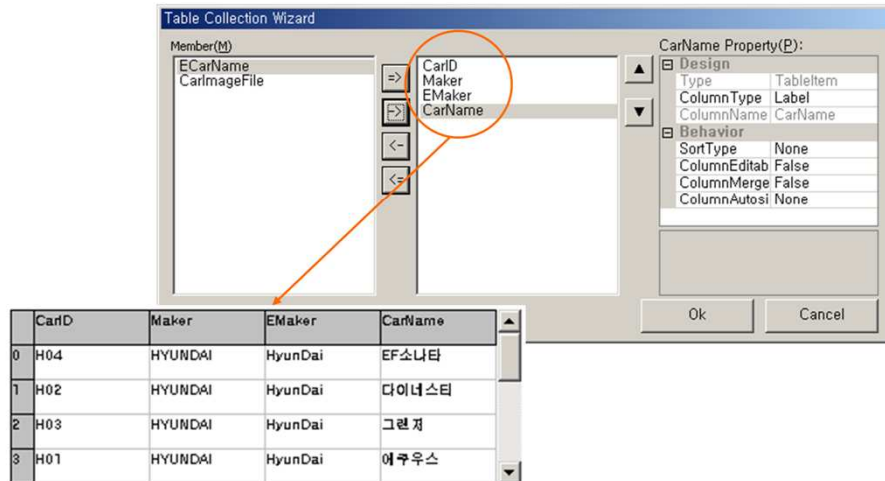


<테이블에 데이터 셋 지정>

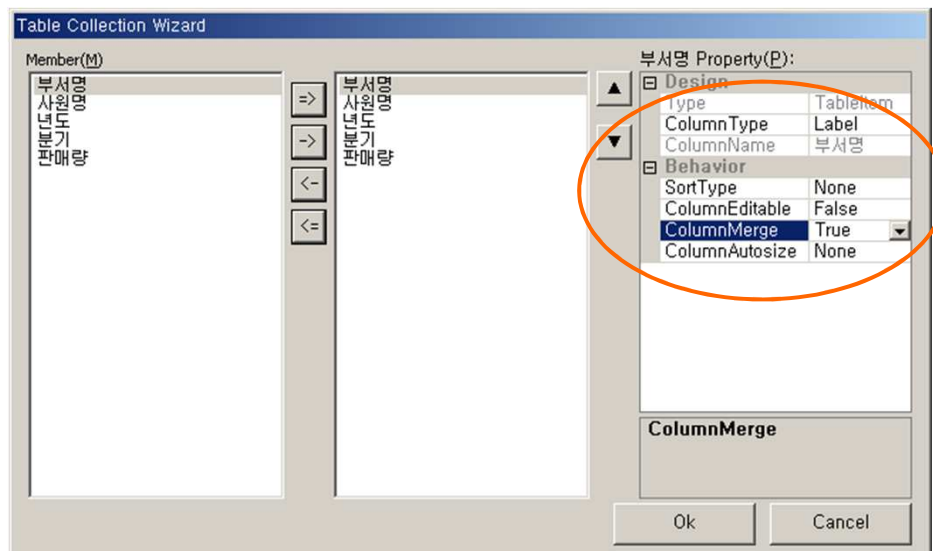
Component – Table(2)

Table Wizard

- ❖ 테이블마법사를 이용하여 나타낼 컬럼을 선택합니다.



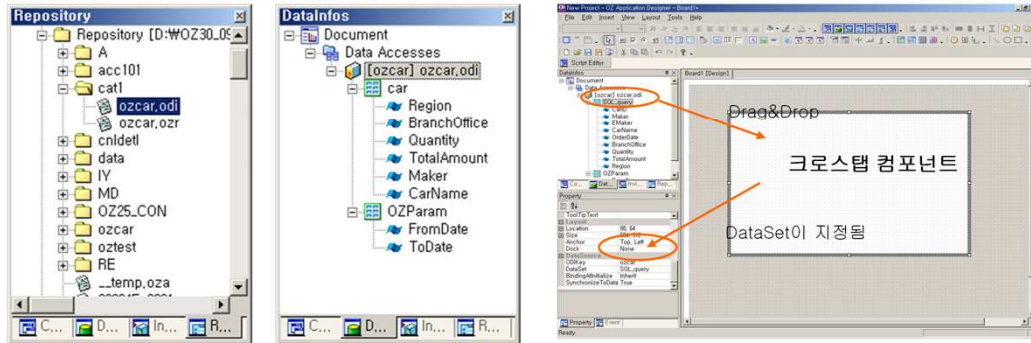
- ❖ 컬럼타입을 Label, PictureBox, Button, CheckBox, RadioButton, ComboBox로 지정할 수 있습니다.
- ❖ 컬럼단위로 Merge, Editing, SortType 기능을 지정할 수 있습니다.



Component – CrossTab(1)

CrossTab 데이터셋 설정

- ❖ ODI파일 호출은 테이블 컴포넌트 사용과 동일합니다.
- ❖ 데이터셋을 Drag&Drop하여 크로스탭의 DataSet의 속성값을 지정할 수 있습니다.



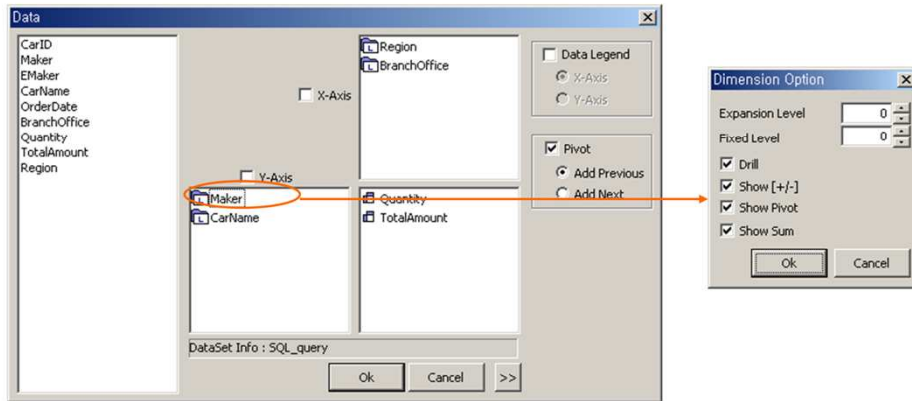
<Repository창에서 odi호출> <DataInfos에서 호출된 odi정보 확인>

<크로스탭에 데이터 셋 지정>

Component – CrossTab(2)

CrossTab Wizard

❖ 데이터 필드를 열, 행, 값에 Drag&Drop하여 지정합니다.



- ❖ 크로스탭은 마법사를 이용하여 기준이 되는 데이터 필드들을 X축, Y축에 배치하고 필드 단위로 합계, 평균 등의 값을 표현합니다.
- ❖ 뷰어에서 피팅팅을 이용하여 기준이 되는 데이터 필드를 변경할 수 있습니다.
- ❖ X축, Y축의 데이터 필드의 오른쪽 마우스를 클릭하여 Dimension Option창을 띄워 드릴다운 기능을 이용할 수 있습니다.

OZApplicationViewer											
File Help											
	Sum		+경기		+서울		+충청		+전라		+경상
Sum	30	690,080,000	6	156,160,000	6	136,110,000	6	150,850,000	6	136,750,000	110,210,000
+기아자동차	9	178,380,000	2	27,420,000	2	45,930,000	2	56,820,000	2	38,310,000	9,900,000
+Hyundai	13	350,590,000	3	111,220,000	2	41,060,000	3	69,470,000	2	70,610,000	58,230,000
+대우자동차	8	161,110,000	1	17,520,000	2	49,120,000	1	24,560,000	2	27,830,000	42,080,000

OZApplicationViewer											
File Help											
	Sum		-경기								
			Sum		수원		인천		Sum		
-기아자동차	Sum	30	690,080,000	6	156,160,000	3	88,130,000	3	68,030,000	6	136,110,000
	Sum	9	178,380,000	2	27,420,000	1	17,520,000	1	9,900,000	2	45,930,000
	크레도스	2	35,040,000	1	17,520,000	1	17,520,000	0	0	1	17,520,000
	엔터프라이즈	4	113,640,000	0	0	0	0	0	0	1	28,410,000
-Hyundai	세피아	3	29,700,000	1	9,900,000	0	0	1	9,900,000	0	0
	Sum	13	350,590,000	3	111,220,000	2	70,610,000	1	40,610,000	2	41,060,000
	EF소나타	3	62,810,000	0	0	0	0	0	0	1	17,200,000
	에쿠우스	3	121,830,000	2	81,220,000	1	40,610,000	1	40,610,000	0	0
	베르나	1	10,510,000	0	0	0	0	0	0	0	0
	다이너스티	2	60,000,000	1	30,000,000	1	30,000,000	0	0	0	0

<Drill Down 결과>

Component – Grid(1)

Grid를 이용하여 동적 표현

- ❖ 표시할 Data를 Grid 각각의 셀에 표시하는 Script Coding 합니다.

Grid는 Table와 달리 Wizard 기능이 없어 별도의 코딩이 필요함

GetCell(rowIndex, columnIndex)로 Cell 객체를 얻어 .Value 속성으로 데이터 표시

- ❖ Grid Title, Cell등의 모양을 디자인하는 Script Coding 합니다.

Grid는 Table와 달리 Wizard 기능이 없어 별도의 코딩이 필요함

GetCell(i,j), .GetTitle(j)로 Title 객체를 얻어 Text와 BackColor등 설정

The diagram illustrates the process of rendering a Grid component. On the left, a code editor window titled 'Grid1' shows the following script code for the 'OnPostInitialize' event:

```
This.AddColumn(2);
This.AddRow(1);
This.GetCell(0,0).Value = "etc";
This.GetCell(0,1).Value = "1,000";
This.GetTitle(0).Value = "Contents";
This.GetTitle(1).Value = "Value";
```

A blue arrow points from the code to a rendered table on the right. The table has two columns: 'Contents' and 'Value'. The first row contains the values 'etc' and '1,000'.

	Contents	Value
0	etc	1,000

- ❖ 가변적인 데이터를 Grid 컴포넌트를 이용하여 나타낼 수 있습니다.
- ❖ Grid의 PostInitialize 이벤트에서 스크립트로 Column, Row 개수를 지정한 후 표현할 데이터를 지정합니다.

Component – Grid(2)

엑셀과의 연관성

- ❖ Grid내의 데이터를 엑셀로 Copy & Paste가 가능합니다.

엑셀의 데이터를 오즈 어플리케이션 Grid에 복사가 가능하므로, DB에 데이터를 Insert할 때 기존 데이터를 쉽게 이용할 수 있습니다.

OZApplicationViewer

Region	BranchOffice	Quantity	TotalAmount	Maker	CarName
경기	수원	1	40,610,000	Hyundai	예쿠루스
경기	수원	1	17,520,000	기아자동차	그레도스
경기	수원	1	30,000,000	Hyundai	다이내스티
경기	인천	1	9,900,000	기아자동차	세피아
경기	인천	1	40,610,000	Hyundai	예쿠루스
경기	인천	1	17,520,000	대우자동차	리노스
경상	대구	1	23,860,000	Hyundai	그랜저
경상	대구	1	23,860,000	Hyundai	그랜저
경상	대구	1	17,520,000	대우자동차	리노스
경상	부산	1	24,560,000	대우자동차	메그너스
경상	부산	1	10,510,000	Hyundai	베르나
경상	부산	1	9,900,000	기아자동차	세피아
서울	강남	1	28,410,000	기아자동차	엔터프라이즈
서울	강남	1	23,860,000	Hyundai	그랜저
서울	강남	1	24,560,000	대우자동차	메그너스

Microsoft Excel - Book2

	A	B	C	D	E	F	G	H
1								
2								
3								
4		경기	인천	1	9900000	기아자동차	세피아	
5		경기	인천	1	40610000	Hyundai	예쿠루스	
6		경기	인천	1	17520000	대우자동차	리노스	
7		경상	대구	1	23860000	Hyundai	그랜저	
8		경상	대구	1	23860000	Hyundai	그랜저	
9		경상	대구	1	17520000	대우자동차	리노스	
10								
11								
12								
13								

- ❖ 가변적인 데이터를 Grid 컴포넌트를 이용하여 나타낼 수 있습니다.
- ❖ Grid의 PostInitialize 이벤트에서 스크립트로 Column, Row 개수를 지정한 후 표현할 데이터를 지정합니다.
- ❖ 그리드 내의 데이터를 엑셀로 Copy & Paste가 가능합니다.
(엑셀의 데이터를 카피하여 오즈 어플리케이션 화면을 이용하여 DB Insert 가 쉽게 가능합니다.)

Component –Grid(3)

Cell Merge

```
Grid1.SelectCellRange(0,4,1,5);
```

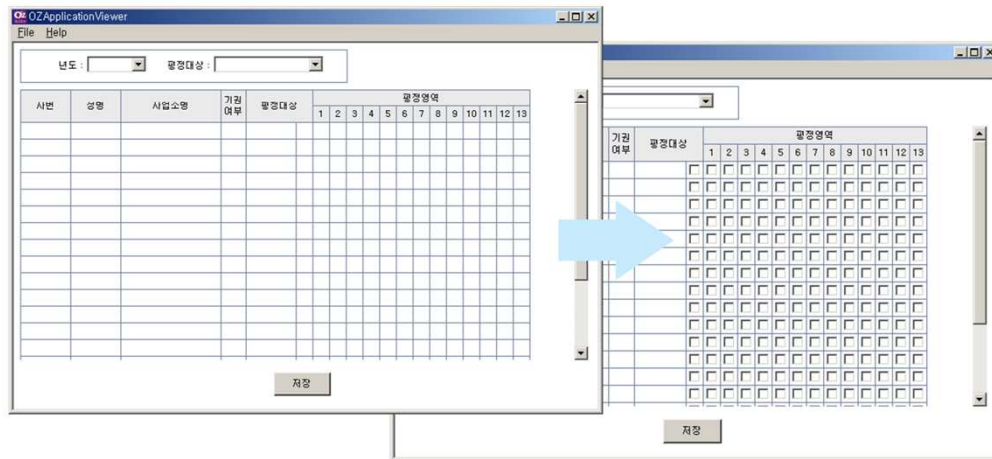
```
Grid1.MergeCellInRange();
```



Component – Grid(4)

다른 컴포넌트 삽입

```
var rows = 20;  
Grid1.AddRow(rows);  
Grid1.SelectCellRange(2,5,rows+1,18);
```



❖ 스크립트

```
var rows = 20;  
Grid1.AddRow(rows);  
Grid1.SelectCellRange(2,5,rows+1,18);
```

// 체크 박스를 위한 추가 스크립트

```
Grid1.ChangeCellTypeInRange(CellComponentTypeConst.CheckBox);  
for (var i=0;i<rows;i++)  
    for (var j=5;j<19;j++) {  
        Grid1.GetCell(i+2, j).CheckAlign = AlignmentConst.MiddleCenter;  
        Grid1.GetCell(i+2, j).SetCellFrameColor(110, 128, 154);  
    }  
}
```

Component – Grid(5)

Data 표현하기

- ❖ 표시할 Data를 Grid 각각의 셀에 표시하는 Script

```
Script Editor - Document.Boards.Board1.Grid1.OnInitialize
Grid1
//데이터 모듈 열어오기
var dataManager = _GetDataManager();
var dataModule = dataManager.GetDataModule("AppSample");
var dataSet = dataModule.GetDataSet("SaleResultAll");

//데이터 수 열어오기
var dataCnt = dataSet.GetRowCount();

//DataSet에서 설정한 전체 필드 수 열어오기
var fieldCnt = dataSet.GetColumnCount();

//DataSet에서 설정한 필드 이름 열어오기
var fieldNames = new Array();
for(i=0;i<fieldCnt;i++){
    fieldNames[i] = dataSet.GetColumnName(i);
}

//Grid에 필드수와 데이터수만큼 Column, Row 추가하기
Grid1.AddColumn(fieldCnt);
Grid1.AddRow(dataCnt);

//Grid의 각각의 Cell에 데이터 표시
for(i=0;i<dataCnt;i++){
    for(j=0;j<fieldCnt;j++){
        Grid1.GetCell(i,j).Value = dataSet.GetStringData(i, fieldNames[j]);
    }
}
```

- 위의 스크립트에서 ODI명("AppSample")과 DataSet명("SaleResultAll") 만 수정하여 해당 스크립트를 다른 그리드 컴포넌트에서 그대로 재사용이 가능합니다.

- ❖ Grid의 Title, Cell등의 모양을 디자인하는 Script

```
Script Editor - Document.Boards.Board1.Grid1.OnInitialize
Grid1
//Grid Cell 모양 디자인
for(i=0;i<dataCnt;i++){
    for(j=0;j<fieldCnt;j++){
        Grid1.GetCell(i,j).SetFont("돋움", "", 12); //글꼴, BOLD, 글씨Size
        if(fieldNames[j] == "판매량") {
            Grid1.GetCell(i,j).TextAlign = AlignmentConst.MiddleRight;
            Grid1.GetCell(i,j).SetSize(80,24);
            Grid1.GetCell(i,j).Pattern = "#,###,###";
            Grid1.GetCell(i,j).SetProperty(PropertyNameConst.InternalMargin, "0,0,7,0");
        }else {
            Grid1.GetCell(i,j).TextAlign = AlignmentConst.MiddleCenter;
            Grid1.GetCell(i,j).SetSize(70,24);
        }
    }
}

//Grid Title 모양 디자인
for(j=0;j<fieldCnt;j++){
    Grid1.GetTitle(j).Value = fieldNames[j]; //dataSet.GetColumnName(j);
    Grid1.GetTitle(j).TextAlign = AlignmentConst.MiddleCenter;
    Grid1.GetTitle(j).SetFont("돋움", "", 12); //글꼴, BOLD, 글씨Size
    Grid1.GetTitle(j).BackColor = "80,80,140";
    Grid1.GetTitle(j).TextColor = "250,250,200";
    if(fieldNames[j] == "판매량") {
        Grid1.GetTitle(j).SetSize(80,32);
    }else {
        Grid1.GetTitle(j).SetSize(70,32);
    }
}

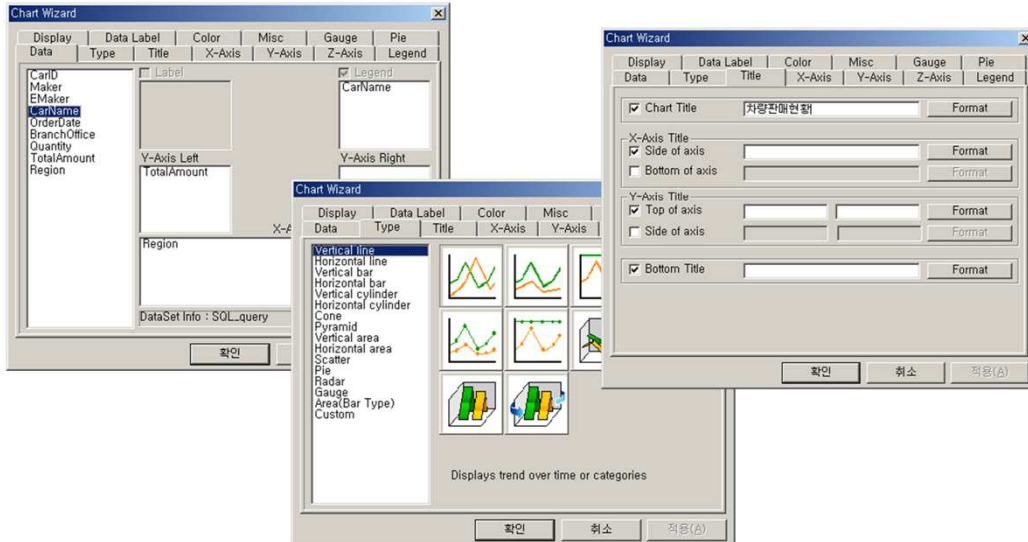
//Grid Indicator는 보이지 않음으로 속성에서 설정함

//Grid를 그린 후 ScrollBar를 Refresh해야 마지막 데이터까지 제대로 보임
Grid1.RefreshScrollBars();
```

Component – Chart(1)

Chart Wizard

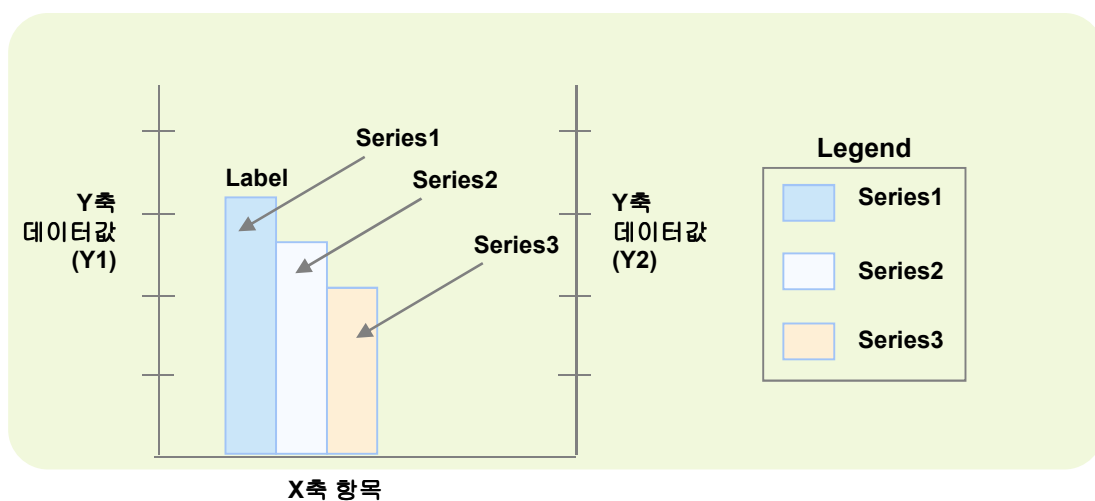
❖ 차트 마법사를 이용하여 X,Y축에 필드를 지정하고 차트의 옵션을 줍니다.



❖ Y축

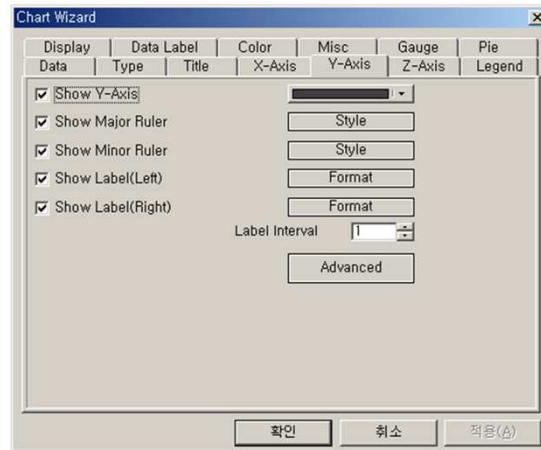
- 2중 Y축 : 양쪽에 Y축이 생성
- 필드의 데이터 타입이 int, double, float일 경우 가능
- sum, avg, max, min, non 함수 값 설정

❖ 차트의 구성요소



Component – Chart(2)

Chart Wizard – Y축



❖ Y축 탭

▪ Y축 최대/최소값 설정

: 설정하지 않을 경우 데이터의 가장 작은 값을 최소값으로, 가장 큰 값을 최대값으로 할당

▪ 원점 지정

❖ 최소값 내림처리/최대값 올림처리

동적으로 최소/최대값을 설정할 경우 실제 데이터의 최소값에서 특정 자리수를 버림처리하고, 최대값에서 특정 자리수를 올림처리하여 차트를 나타낼 수 있습니다.

오즈 파라미터 이용하여 조회

ODI 파라미터 Refresh

❖ 입력 받은 값을 이용하여 조회 버튼 등으로 차트, 크로스탭, 테이블 등에 나타난 값을 조회합니다.

```
var datamanager = _GetDataManager();  
var datamodule = datamanager.GetDataModule("ozcar"); // odi 이름  
var dataset = datamodule.GetDataSet("SQL_query");    // 데이터 셋 이름  
    //GetDataSet("SQL_query", null, true);  
    //true값이면 무조건 DB접근하여 Refresh Data  
  
datamodule.UpdateOZParam("FromDate", "00-01-01");  
datamodule.UpdateOZParam("ToDate", "00-01-31");  
  
dataset.RefreshDataSet();
```

연습문제

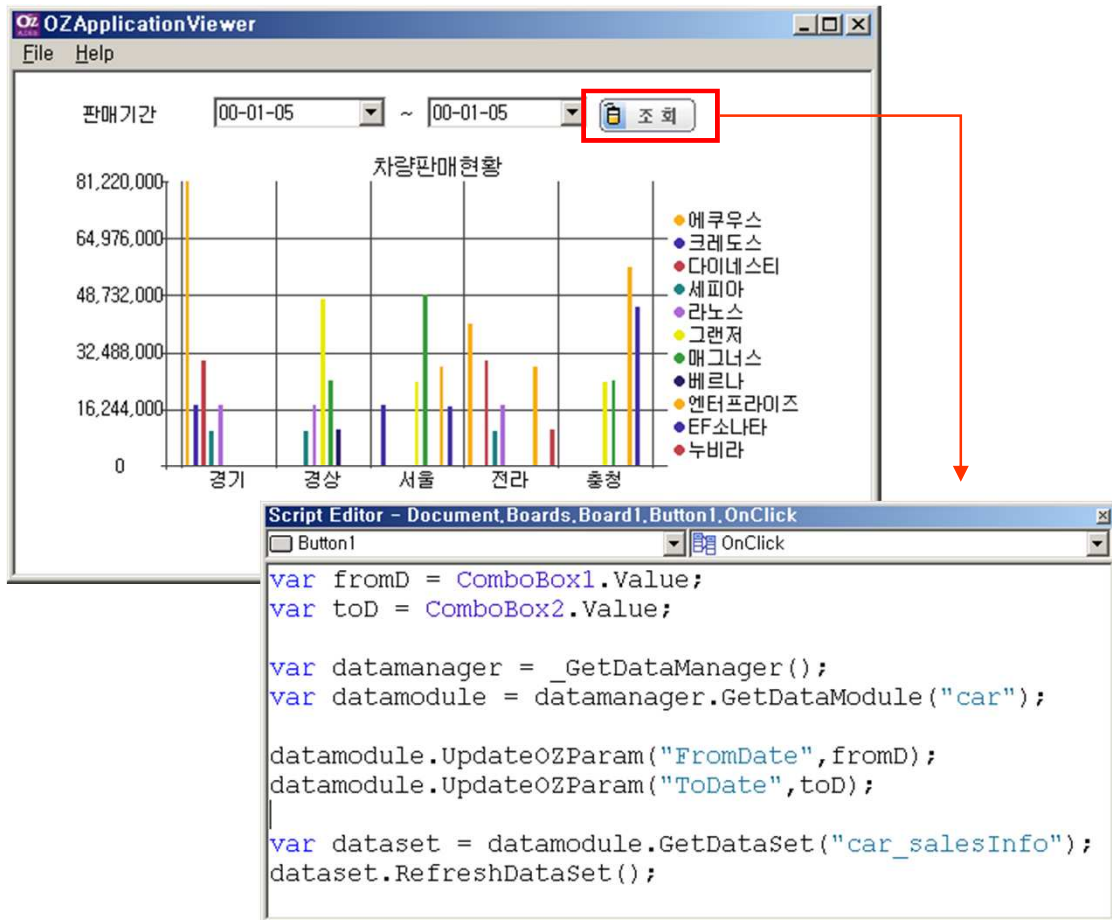
제공되는 DB와 SQL을 이용하여 “지역별 제조 회사별 차량 판매 현황”을 크로스탭과 차트를 이용하여 화면을 구성합니다.

CrossTab & Chart의 데이터셋

```
SELECT a.Region, a.BranchOffice,
       a.Quantity, a.TotalAmount,
       b.Maker, b.CarName
FROM Car_info a, Car_sales b
where a.CarID = b.CarID and
      a.OrderDate > '#OZParam.FromDate#'
and a.OrderDate < '#OZParam.ToDate#'
order by a.Region, a.BranchOffice
```

ComboBox의 데이터셋

```
SELECT OrderDate
FROM Car_info
ORDER BY OrderDate ASC
```



Lesson 4. 공통 모듈 사용하기

교육목표

- 전역변수와 Global Function 사용하기
- OZF,OZC,OZS 사용하기

Global Function/전역변수

❖ Global Function

- 방법1 : Document의 GlobalFunction 이벤트에 함수를 만듭니다.
- 방법2 : FunctionEditor를 이용하여 만들어진 ozf파일을 호출합니다.
 - 선언 `function TextBox_checkLength(Obj_Nm,len){     //자릿수 입력을 체크한다.`
 `var a = Obj_Nm.Text;`
 `if(Obj_Nm.Text.length != len){`
 `_MessageBox("입력값이 일치하지 않습니다.");`
 `Obj_Nm.focus();`
 `}`
 `}`
 - 사용 `TextBox_checkLength(Label1,5);`

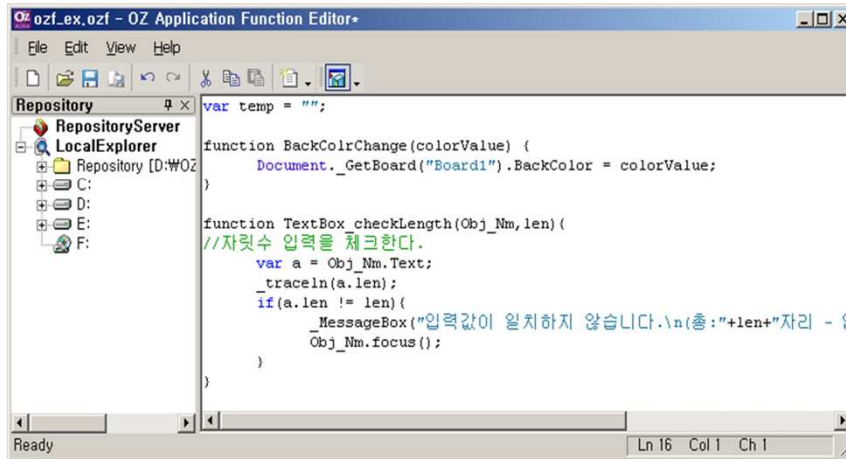
❖ 전역변수

- 방법1 : Document의 GlobalFunction 이벤트에 일반 변수와 같이 선언합니다.
- 방법2 : SetGlobal 와 GetGlobal 함수를 이용합니다.
 - 선언 `_SetGlobal("isNumber", "FALSE");`
 - 사용 `_GetGlobal("isNumber");`

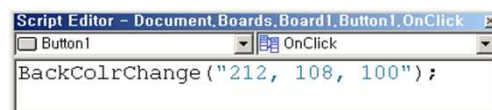
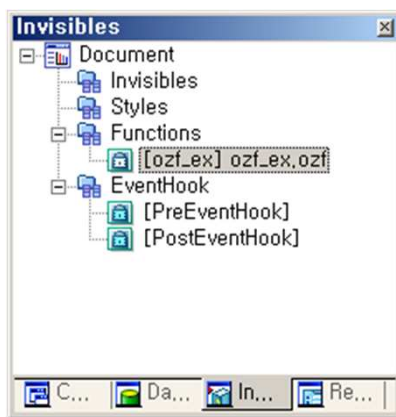
OZF(1)

OZF 파일 작성 예제

- ❖ Menu에서 Tools → External Tools → Function Editor → New Function을 선택 후 저장



- ❖ OZ Application Designer에 반영하기
 - Menu에서 File → Insert New Function을 선택하여 OZF 파일을 추가
 - Invisibles 탭에서 추가된 OZF 확인
 - 추가된 OZF 내의 함수는 Global Function과 동일하게 사용



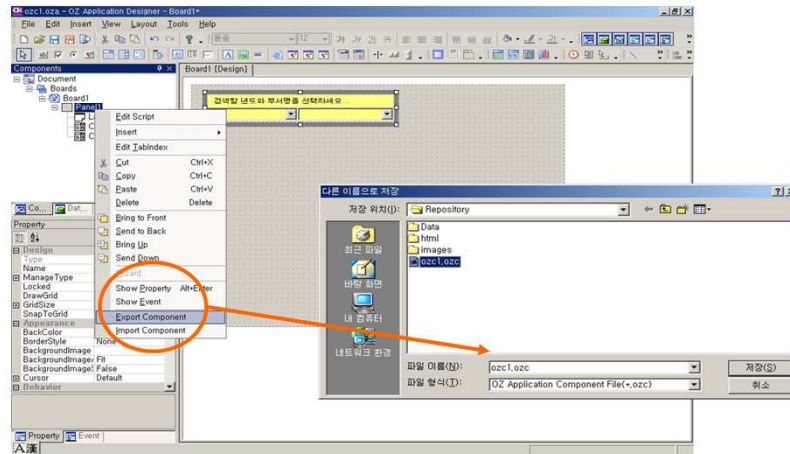
<버튼에서 호출한 예>

<호출된 ozf 파일은 Invisibles 창에서 확인>

OZC(1)

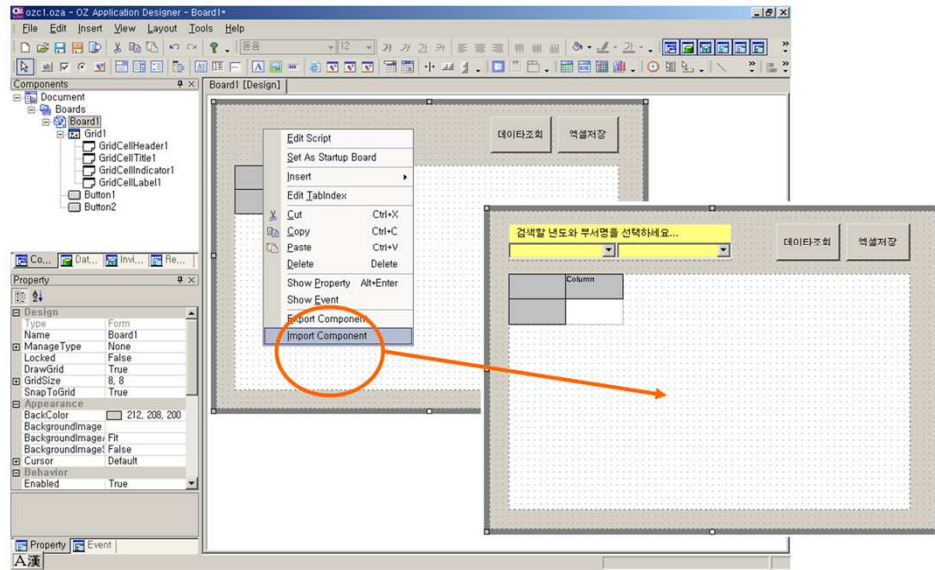
OZC 파일 작성

- ❖ 여러 컴포넌트를 Panel에 올립니다.
- ❖ Panel을 선택 마우스 오른쪽 버튼을 누르면 나타나는 PopUp Menu의 Export Component를 선택하여 OZC 파일로 저장합니다.



OZC(2)

Document에 OZC 파일 추가



❖ OZC의 속성

ManageType	Manual, /ozc1.ozc
Type	Manual
Category	/
Item	ozc1.ozc

• Type : OZC 파일과 연동 여부 결정

- None : OZC와 무관한 개별 또는 그룹 Component
- Manual : OZC와 연동, 변경된 내용을 저장하면 설정한 OZC도 변경됨
디자인시에는 보이거나 미리보기시에 ImportNewChild()를 호출하는 시점에 보임
- Auto : OZC와 연동, 변경된 내용을 저장하면 설정한 OZC도 변경됨
디자인시와 미리보기 시 모두 보임

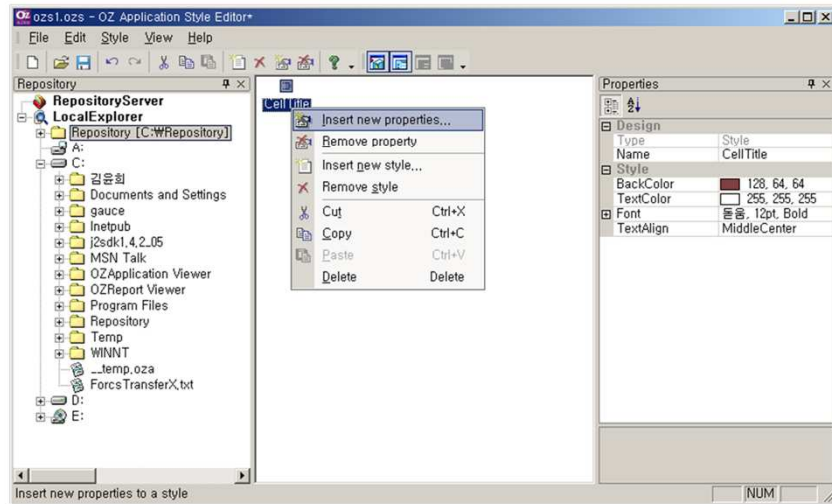
• Category : 연동할 OZC 파일의 경로

• Item : 연동할 OZC 파일 이름

OZS(1)

OZS 파일 작성

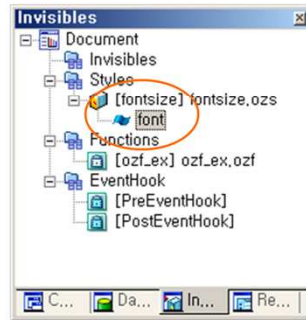
- ❖ Menu에서 Tools → External Tools → Style Editor → New Style을 선택
- ❖ 적용할 Property를 추가하고 해당 Value를 변경하여 OZS 파일 만들



OZS(2)

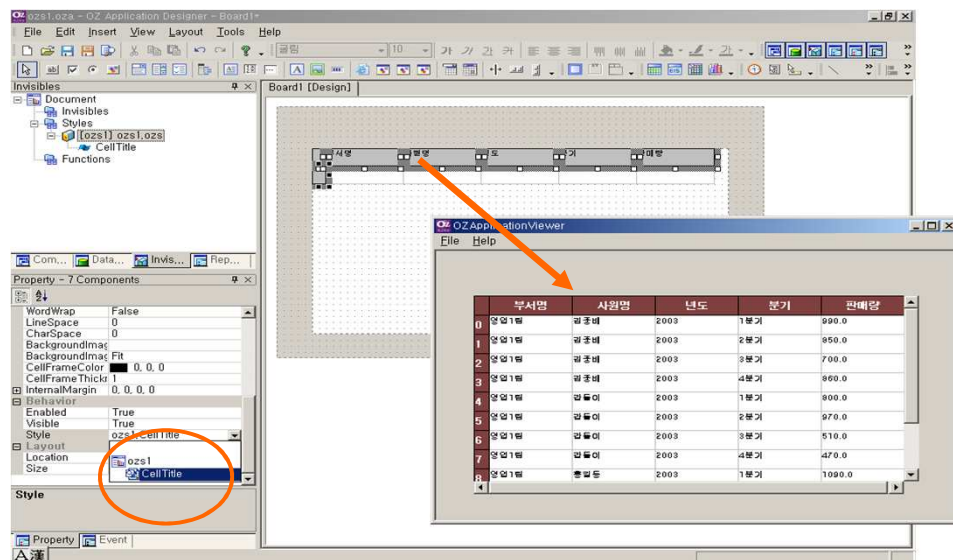
Document에 OZS 파일 추가

- ❖ Menu에서 File → Insert New Style을 선택하여 OZS 파일을 추가
- ❖ Invisibles 탭에서 추가된 OZS 확인



❖ 컴포넌트에 적용하기

- 특정 스타일을 Drag & Drop 하게 되면 속성창에서 호출된 ozs파일과 스타일을 적용할 수 있습니다.



- 미리보기를 하면 반영된 속성이 적용됩니다.

Lesson 5. Data Action

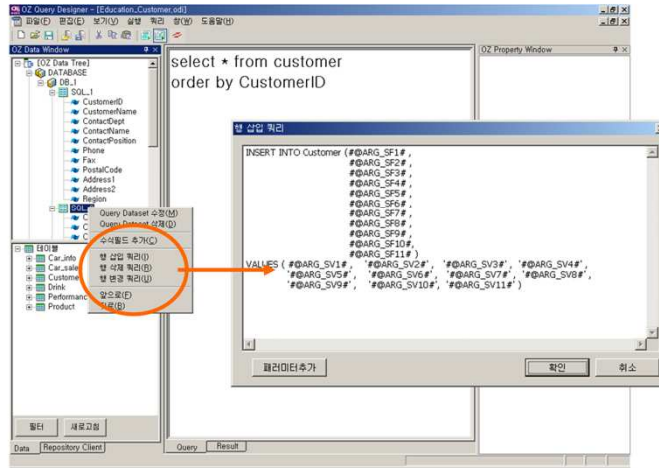
교육목표

- DB에 데이터를 삽입, 삭제, 변경 하기
- 오즈 파라미터를 이용하여 동적으로 Data Refresh하기

QueryDesigner에 DML입력

행 삽입/삭제/변경

- ❖ 데이터 셋을 만들고, 해당 데이터 셋의 오른쪽 마우스를 이용



❖ 행삽입 쿼리

```
INSERT INTO Customer
( #@ARG_SF1#, #@ARG_SF2#, #@ARG_SF3#, #@ARG_SF4# )
VALUES
( #@ARG_SV1#, #@ARG_SV2#, #@ARG_SV3#, #@ARG_SV4# )
```

❖ 행삭제 쿼리

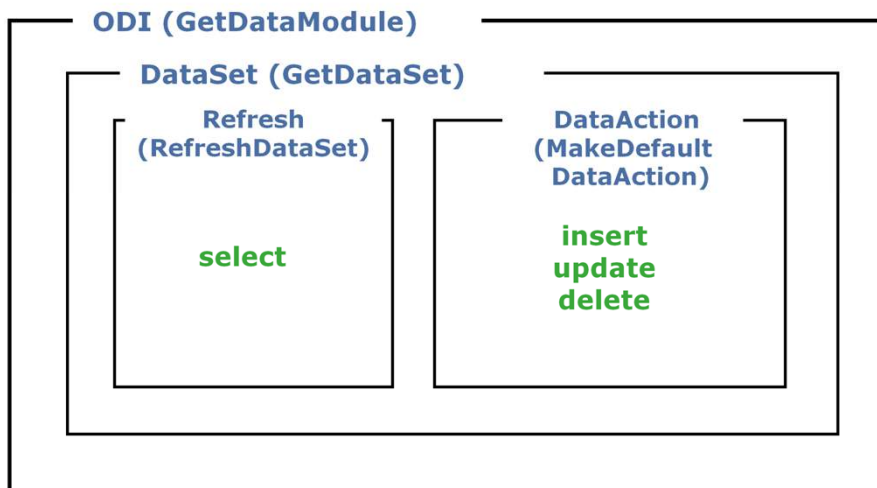
```
DELETE FROM CUSTOMER WHERE #@ARG_DF1# = #@ARG_DV1#
```

❖ 행변경 쿼리

```
UPDATE CUSTOMER
SET  [ #@ARG_SF1# ] = '#@ARG_SV1#',
     [ #@ARG_SF2# ] = '#@ARG_SV2#',
     [ #@ARG_SF3# ] = '#@ARG_SV3#',
WHERE [ #@ARG_DF1# ] = #@ARG_DV1#
```

DataAction Script API

Data 관련 객체의 구조



❖ Refresh (select)

```
var datamanager = _GetDataManager();
var datamodule = datamanager.GetDataModule("ODI1"); // ODI 이름
var dataset = datamodule.GetDataSet("Dataset1"); // DataSet 이름
datamodule.UpdateOZParam("FromDate", "00- 01- 01"); // Parameter 변경
dataset.RefreshDataSet(); // 조회 실행
```

❖ DataAction (insert, update, delete)

```
var datamanager = _GetDataManager();
var datamodule = datamanager.GetDataModule("ODI1"); // ODI 이름
var dataset = datamodule.GetDataSet("Dataset1"); // DataSet 이름
var dataactions = new Array();
dataactions[ 0 ] = dataset.MakeDefaultDataAction();
dataactions[ 0 ].DataActionType = DataActionTypeConst.Delete; // 삭제 선택
dataactions[ 1 ] = dataset.MakeDefaultDataAction();
dataactions[ 1 ].DataActionType = DataActionTypeConst.Insert; // 삽입 선택
dataset.QueueActionObjects(dataactions); // Queue에 쌓음
var result = "";
result = datamodule.CommitQueuedActions(); // Commit
```

Table Component 이용한 Insert, Update, Delete

Table의 CommitQueuedActions 함수

❖ 작성 방법

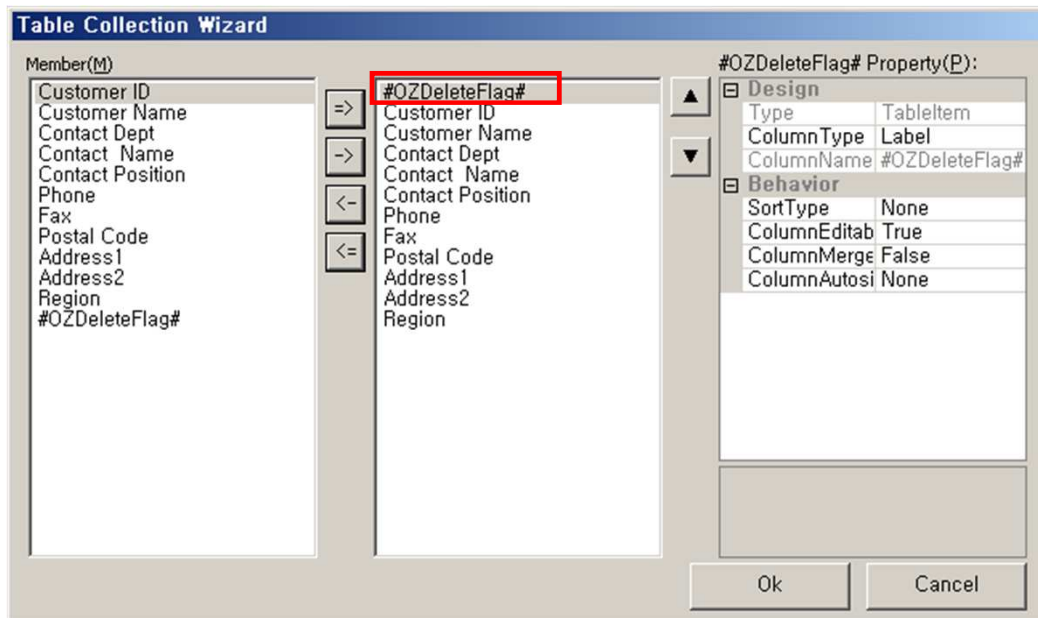
- 테이블 마법사를 이용하여 디자인 합니다.
- 테이블의 속성 중 AllowInsert AllowDelete AllowUpdate 을 모두 True 로 지정합니다.
- 버튼 등에 Table.CommitQueuedActions() 를 이용하여 처리 합니다.

```
var result = Table1.CommitQueuedActions();
```

❖ 특징

- 컴포넌트에 보여지는 데이터의 값들을 스크립트로 읽어 오지 않아도 Table컴포넌트에 나타나 있는 데이터를 바로 insert/update/delete 할 수 있습니다.
- #OZDeleteFlag# 를 이용하여 만들어진 체크박스의 칼럼에 체크를 하면 선택된 row가 삭제됩니다.

❖ 테이블 마법사에서 #OZDeleteFlag# 사용 예



(※ 참고 : 위의 Method는 추가된 기능입니다. 이전 버전에서는 지원되지 않습니다.)

입력 버튼

Data Insert

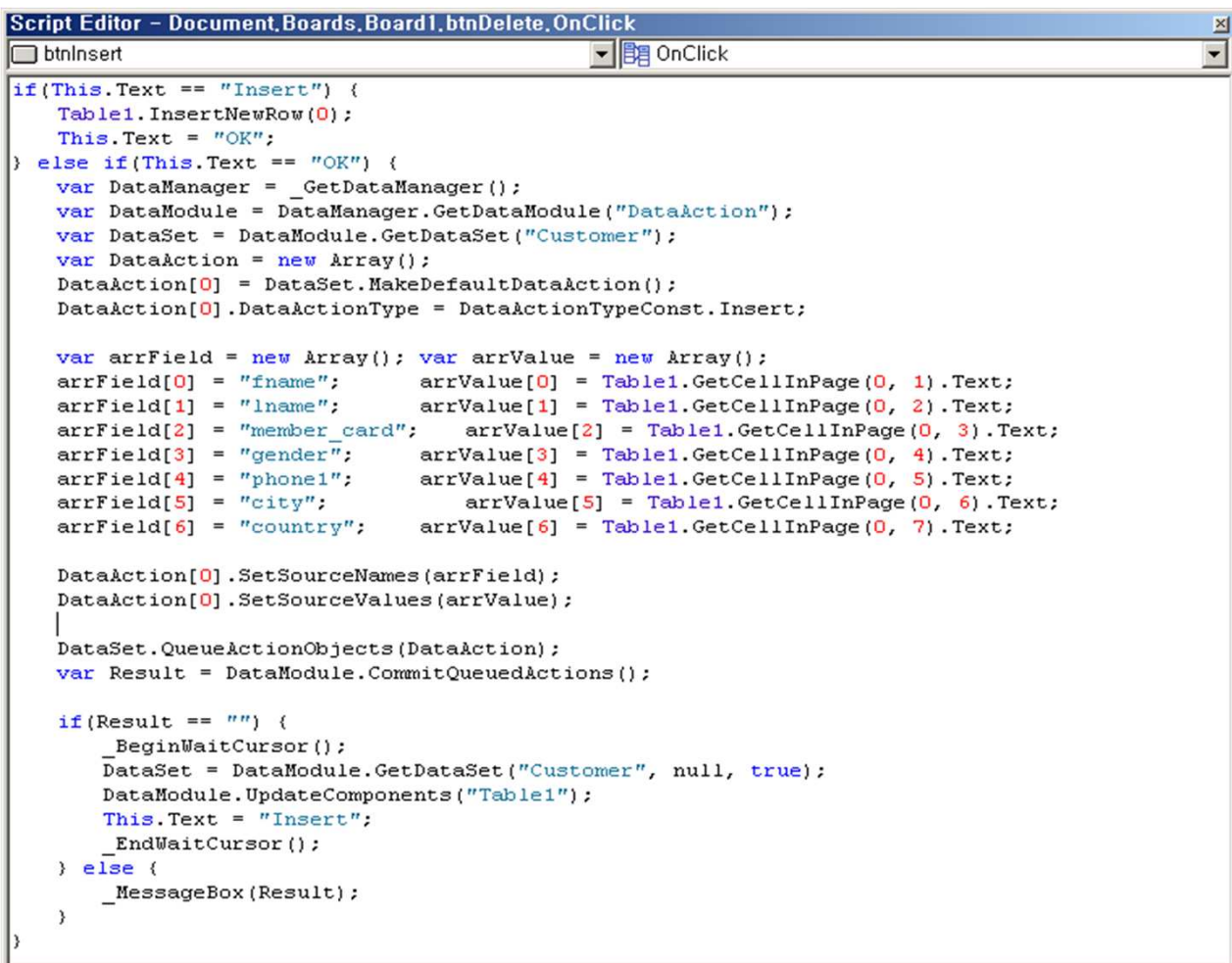
```
var DataManager = _GetDataManager();
var DataModule = DataManager.GetDataModule("DataAction");
var DataSet = DataModule.GetDataSet("Customer");
var DataAction = new Array();
DataAction[0] = DataSet.MakeDefaultDataAction();
DataAction[0].DataActionType = DataActionTypeConst.Insert;

var arrField = new Array();
var arrValue = new Array();

// 배열 arrField 와 arrValue 에 데이터 대입

DataAction[0].SetSourceNames(arrField);
DataAction[0].SetSourceValues(arrValue);
DataSet.QueueActionObjects(DataAction);
var Result = DataModule.CommitQueuedActions();
```

❖ btnInsert 의 OnClick 이벤트 실행 시에 Data Insert의 예



```
Script Editor - Document.Boards.Board1.btnDelete.OnClick
btnInsert OnClick

if(This.Text == "Insert") {
    Table1.InsertNewRow(0);
    This.Text = "OK";
} else if(This.Text == "OK") {
    var DataManager = _GetDataManager();
    var DataModule = DataManager.GetDataModule("DataAction");
    var DataSet = DataModule.GetDataSet("Customer");
    var DataAction = new Array();
    DataAction[0] = DataSet.MakeDefaultDataAction();
    DataAction[0].DataActionType = DataActionTypeConst.Insert;

    var arrField = new Array(); var arrValue = new Array();
    arrField[0] = "fname";      arrValue[0] = Table1.GetCellInPage(0, 1).Text;
    arrField[1] = "lname";      arrValue[1] = Table1.GetCellInPage(0, 2).Text;
    arrField[2] = "member_card"; arrValue[2] = Table1.GetCellInPage(0, 3).Text;
    arrField[3] = "gender";      arrValue[3] = Table1.GetCellInPage(0, 4).Text;
    arrField[4] = "phone1";      arrValue[4] = Table1.GetCellInPage(0, 5).Text;
    arrField[5] = "city";        arrValue[5] = Table1.GetCellInPage(0, 6).Text;
    arrField[6] = "country";     arrValue[6] = Table1.GetCellInPage(0, 7).Text;

    DataAction[0].SetSourceNames(arrField);
    DataAction[0].SetSourceValues(arrValue);
    DataSet.QueueActionObjects(DataAction);
    var Result = DataModule.CommitQueuedActions();

    if(Result == "") {
        _BeginWaitCursor();
        DataSet = DataModule.GetDataSet("Customer", null, true);
        DataModule.UpdateComponents("Table1");
        This.Text = "Insert";
        _EndWaitCursor();
    } else {
        _MessageBox(Result);
    }
}
```

변경 버튼

Data Update

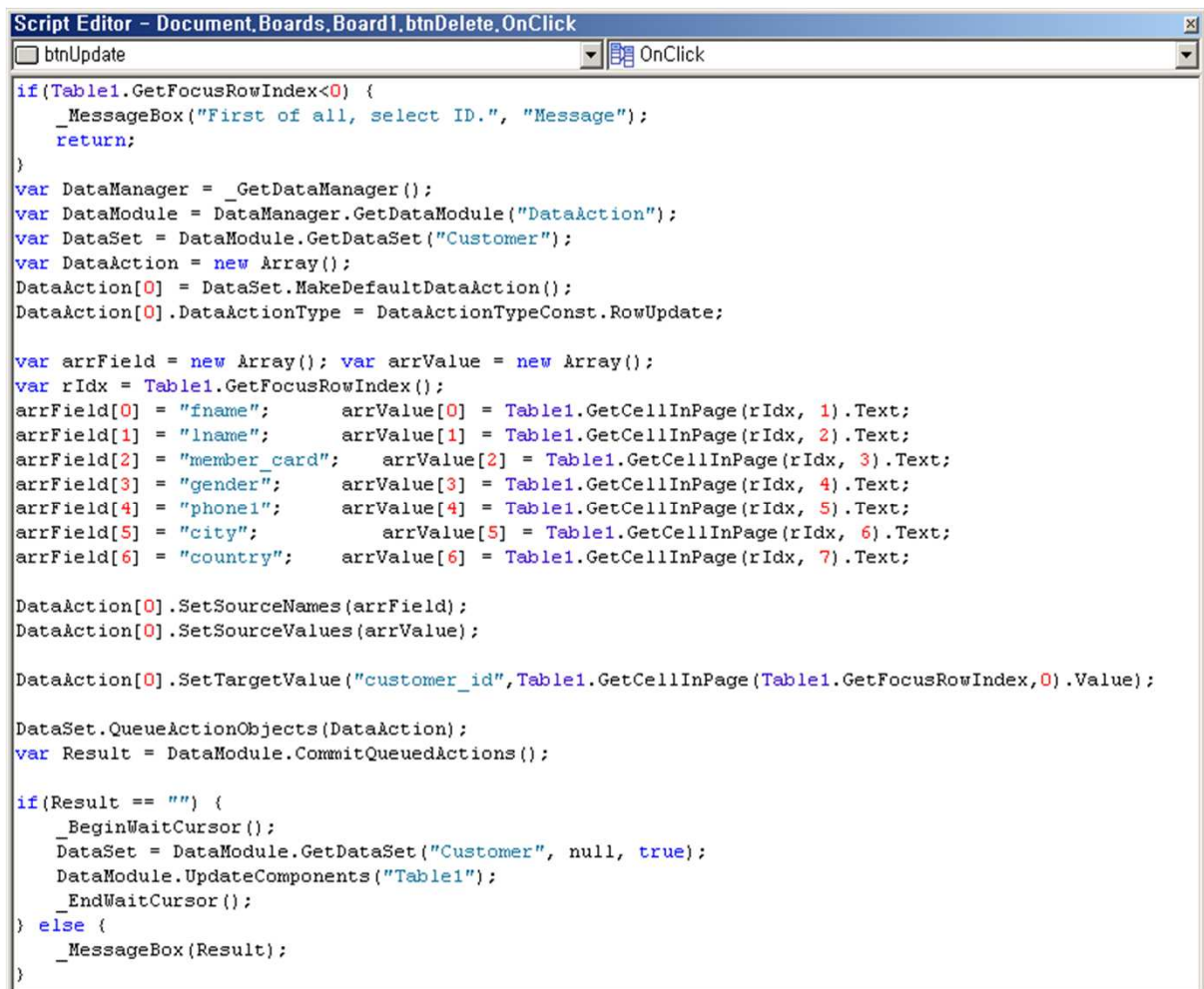
```
var DataManager = _GetDataManager();
var DataModule = DataManager.GetDataModule("DataAction");
var DataSet = DataModule.GetDataSet("Customer");
var DataAction = new Array();
DataAction[0] = DataSet.MakeDefaultDataAction();
DataAction[0].DataActionType = DataActionTypeConst.RowUpdate;

var arrField = new Array();    var arrValue = new Array();
// 배열 arrField 와 arrValue 에 데이터 대입
DataAction[0].SetSourceNames(arrField);
DataAction[0].SetSourceValues(arrValue);

DataAction[0].SetTargetValue("customer_id","100"); // 변경할 Target 의 값 설정

DataSet.QueueActionObjects(DataAction);
var Result = DataModule.CommitQueuedActions();
```

❖ btnUpdate 의 OnClick 이벤트 실행 시에 Data Update의 예



```
Script Editor - Document.Boards.Board1.btnDelete.OnClick
btnUpdate OnClick

if(Table1.GetFocusRowIndex<0) {
    _MessageBox("First of all, select ID.", "Message");
    return;
}

var DataManager = _GetDataManager();
var DataModule = DataManager.GetDataModule("DataAction");
var DataSet = DataModule.GetDataSet("Customer");
var DataAction = new Array();
DataAction[0] = DataSet.MakeDefaultDataAction();
DataAction[0].DataActionType = DataActionTypeConst.RowUpdate;

var arrField = new Array(); var arrValue = new Array();
var rIdx = Table1.GetFocusRowIndex();
arrField[0] = "fname";      arrValue[0] = Table1.GetCellInPage(rIdx, 1).Text;
arrField[1] = "lname";     arrValue[1] = Table1.GetCellInPage(rIdx, 2).Text;
arrField[2] = "member_card"; arrValue[2] = Table1.GetCellInPage(rIdx, 3).Text;
arrField[3] = "gender";    arrValue[3] = Table1.GetCellInPage(rIdx, 4).Text;
arrField[4] = "phone1";    arrValue[4] = Table1.GetCellInPage(rIdx, 5).Text;
arrField[5] = "city";      arrValue[5] = Table1.GetCellInPage(rIdx, 6).Text;
arrField[6] = "country";   arrValue[6] = Table1.GetCellInPage(rIdx, 7).Text;

DataAction[0].SetSourceNames(arrField);
DataAction[0].SetSourceValues(arrValue);

DataAction[0].SetTargetValue("customer_id",Table1.GetCellInPage(Table1.GetFocusRowIndex,0).Value);

DataSet.QueueActionObjects(DataAction);
var Result = DataModule.CommitQueuedActions();

if(Result == "") {
    _BeginWaitCursor();
    DataSet = DataModule.GetDataSet("Customer", null, true);
    DataModule.UpdateComponents("Table1");
    _EndWaitCursor();
} else {
    _MessageBox(Result);
}
```

삭제 버튼

Data Delete

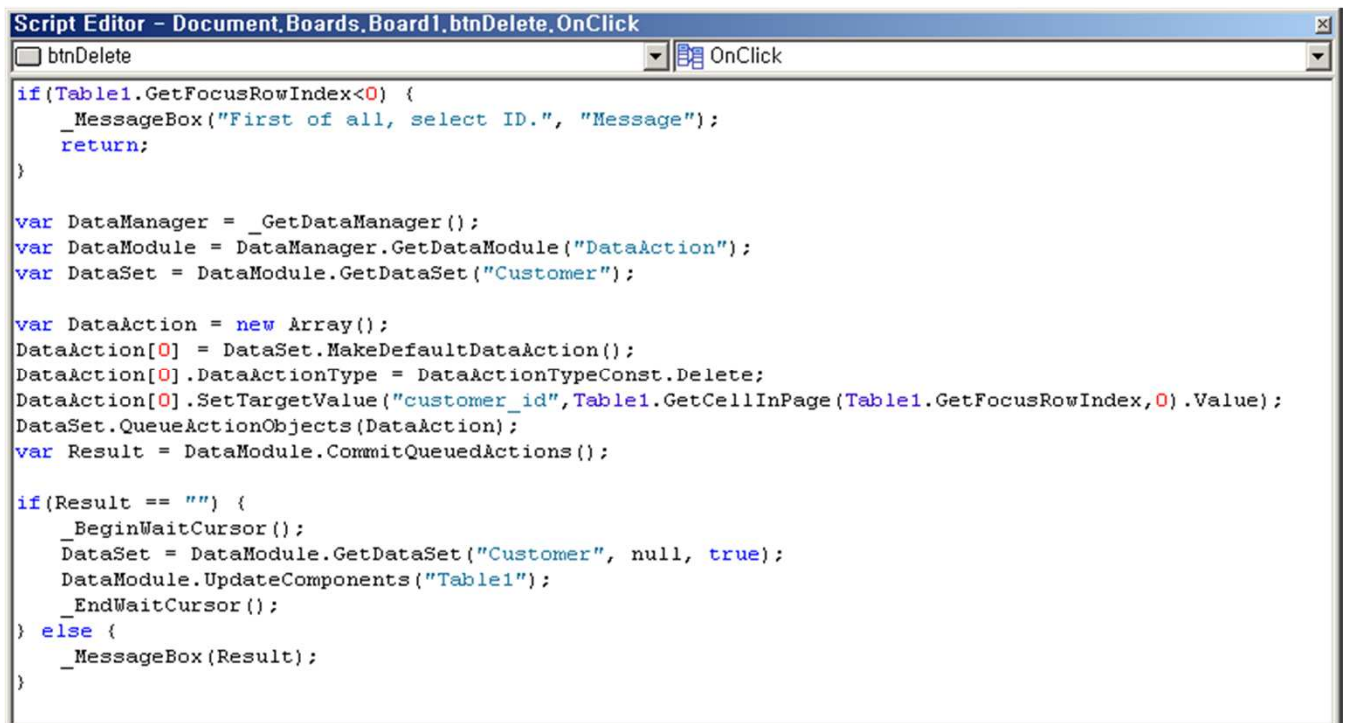
```
var DataManager = _GetDataManager();
var DataModule = DataManager.GetDataModule("DataAction");
var DataSet = DataModule.GetDataSet("Customer");

var DataAction = new Array();
DataAction[0] = DataSet.MakeDefaultDataAction();
DataAction[0].DataActionType = DataActionTypeConst.Delete;

DataAction[0].SetTargetValue("customer_id", "100"); // 삭제할 Target 값 설정

DataSet.QueueActionObjects(DataAction);
var Result = DataModule.CommitQueuedActions();
```

❖ btnDelete 의 OnClick 이벤트 실행 시에 Data Delete의 예



```
Script Editor - Document.Boards.Board1.btnDelete.OnClick
btnDelete OnClick

if (Table1.GetFocusRowIndex < 0) {
    _MessageBox("First of all, select ID.", "Message");
    return;
}

var DataManager = _GetDataManager();
var DataModule = DataManager.GetDataModule("DataAction");
var DataSet = DataModule.GetDataSet("Customer");

var DataAction = new Array();
DataAction[0] = DataSet.MakeDefaultDataAction();
DataAction[0].DataActionType = DataActionTypeConst.Delete;
DataAction[0].SetTargetValue("customer_id", Table1.GetCellInPage(Table1.GetFocusRowIndex, 0).Value);
DataSet.QueueActionObjects(DataAction);
var Result = DataModule.CommitQueuedActions();

if (Result == "") {
    _BeginWaitCursor();
    DataSet = DataModule.GetDataSet("Customer", null, true);
    DataModule.UpdateComponents("Table1");
    _EndWaitCursor();
} else {
    _MessageBox(Result);
}
```

연습문제

고객 정보를 추가, 삭제, 수정하는 화면을 만듭니다.

조건

- ❖ CUSTOMER 테이블을 이용하여 쿼리를 만듭니다.
- ❖ 테이블 컴포넌트를 이용하여 화면을 만듭니다.
- ❖ 버튼에 다음과 같이 스크립트를 적용합니다.

```
var result = Table1.CommitQueuedActions();
if(result.length > 0){
    _MessageBox(result);
}
```

스크립트 적용

※ 고객 정보

- 삭제 : 삭제하고자 하는 데이터를 체크 박스로 선택하여 [Submit] 버튼을 클릭합니다.
- 입력 : 제일 아래의 빈 줄에 추가하고자 하는 데이터를 입력하고 [Submit] 버튼을 클릭합니다.
- 수정 : 수정을 원하는 라벨을 선택하여 수정한 후에 [Submit] 버튼을 클릭합니다.

순번	고객명	담당부서	담당자명	직위	전화번호	팩스번호	우편번호	
☐	10	(주)쌍용컴퓨터	매핑시스템실	전민진	부장	02-2704-351	02-2704-597	100-71
☐	11	(주)아이.디.씨.텍	정보통신사업부	민영환	차장	02-424-1911	02-424-1267	138-11
☐	12	(주)에드베네트	기술지원팀	박성창	팀장	02-597-0234	02-598-0286	137-01
☐	13	(주)영신	기획관리과	김길수	계장	0345-492-9754	0345-491-5805	425-11
☐	14	(주)운영	관리과	최기철	주임	0417-558-9601	0417-552-9600	330-81
☐	15	(주)인토스	소프트웨어사업부	김은수	차장	02-3701-8124	02-725-0246	110-71
☐	16	(주)정림시스템컨설팅	정림과학경영연구소	남기철	대표컨설턴트	02-720-7053	02-723-9433	110-11
☐	17	(주)정보라인	영업부	이성창	과장	02-786-9841	02-780-5587	150-01
☐	18	(주)제나시스코리아		김남정	연구원	02-782-9350	02-782-9963	150-01
☐	19	(주)제이엠 하이테크		문지영	사원	0551-67-5090	0551-67-7000	641-01
☐	20	(주)주흥정보통신	ED 사업부	백남길	팀장	02-5678-619	02-5678-677	135-21

테이블 마법사에서 #OZDeleteFlag# 사용

Lesson 6. 오즈 리포트 연동

교육목표

- OZApplication뷰어에서 ReportControl을 이용하여 오즈리포트를 띄우기
- ReportButton을 이용하여 프린트하기

ReportControl

오즈 리포트 뷰어 띄우기

- ❖ 일반 버튼을 이용하여 리포트 호출
- ❖ ReportControl의 OnPreApplyReport 이벤트에 리포트 관련 파라미터 추가

```
Script Editor - Document.Boards.Board1.ReportControl1.OnPreApplyReport
Button1 OnClick
//표시할 리포트 파일 지정
ReportControl1.ReportViewerURL = "ozp://cat1/ozcar.ozr";
ReportControl1.ApplyReport();

Script Editor - Document.Boards.Board1.ReportControl1.OnPreApplyReport
ReportControl1 OnPreApplyReport
//Parameter 초기화
This.InitUserParameter();
//리포트 파라미터
This.AddReportParameter("connection.pcount", "1");
This.AddReportParameter("connection.args1", "fontsize=18");
//odi 파라미터
This.AddReportParameter("odi.odinames", "ozcar");
This.AddReportParameter("odi.ozcar.pcount", "2");
This.AddReportParameter("odi.ozcar.args1", "FromDate="+TextBox1.Value);
This.AddReportParameter("odi.ozcar.args2", "ToDate="+TextBox2.Value);
//뷰어 관련 파라미터
This.AddReportParameter("viewer.bgcolor", "ffffff");
```

- ❖ 스크립트로 적용하지 않고 Property창에서 제어하기

The Property window for ReportControl1 shows the following properties:

- Type: ReportControl
- Name: ReportControl1
- ClassID: {64DA633F-E73B-4344-83BF-...}
- Manage Type: None
- Locked: False
- ReportViewerOption: viewer.useoutborder=false
- ReportViewerURL:
- Behavior:
 - Enabled: True
 - Visible: True
 - Style:
 - AllowDrop: False
- Layout:
 - Location: 16, 64
 - Size: 760, 712
 - Anchor: Top, Left
 - Dock: None
- ActiveX Control

The ReportViewerOption dialog box is open to the 'Connection' tab, showing a list of parameters and their values:

Parameter Name	Parameter Value
connection.server	127.0.0.1
connection.port	8003
connection.servlet	
connection.reportname	
connection.refreshperiod	
connection.refreshcache	No
connection.compressedF...	No
connection.usescheduled...	
connection.displayName	
connection.openfile	
connection.presavedata	No
connection.usesecurity	No
connection.id	

ReportButton

오즈 리포트 출력

- ❖ 리포트 뷰어를 띄우지 않고 바로 출력을 해야 할 경우 사용합니다.
- ❖ ReportButton의 OnPreApplyReport 이벤트에 리포트 관련 파라미터를 추가합니다.



```
Script Editor - Document.Boards.Board1.ReportControl1.OnPreApplyReport
ReportButton1 OnPreApplyReport
//Parameter 초기화
This.InitUserParameter();

//호출할 리포트
This.ReportViewerURL = "ozp:///cat1/ozcar.ozr";

//리포트 파라미터
This.AddReportParameter("connection.pcount", "1");
This.AddReportParameter("connection.args1", "fontsize=18");

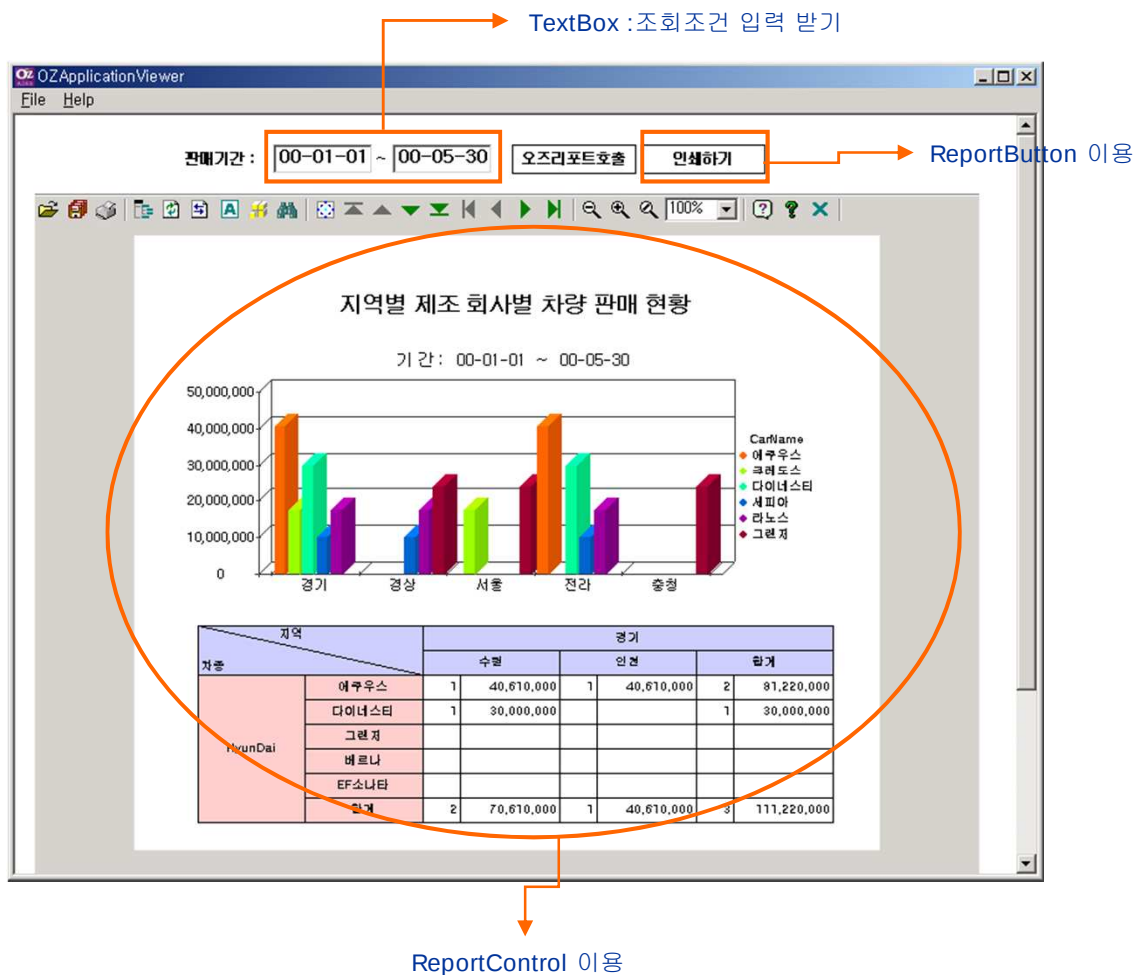
//ODI 파라미터
This.AddReportParameter("odi.odinames", "ozcar");
This.AddReportParameter("odi.ozcar.pcount", "2");
This.AddReportParameter("odi.ozcar.args1", TextBox1.Value);
This.AddReportParameter("odi.ozcar.args2", TextBox2.Value);
```

연습문제

ReportControl과 ReportButton을 이용하여 리포트를 띄우고, 출력합니다.

조건

- ❖ 텍스트 박스로 조회 조건을 입력받습니다.
- ❖ 일반 버튼을 이용하여 ReportControl과 연동합니다.
- ❖ ReportControl에 리포트 호출 관련 스크립트를 적용합니다.
- ❖ ReportButton을 이용하여 리포트를 출력합니다.



Chapter 5

Using Viewer

Lesson 1. OZ ApplicationViewer 호출

Lesson 1. Viewer 호출하기

교육목표

- 🔴 브라우저에서 오즈 애플리케이션 뷰어 호출 방법 숙지

OZ Application Viewer 설치



Client PC에 어플리케이션 뷰어가 설치 되도록 하는 설치 태그

```
<OBJECT id=ZTransferX style="LEFT: 0px; TOP: 0px"
  codeBase=http://127.0.0.1/ozapplication/ZTransferX.cab#version=1,0,2,4
    height=0 width=0 classid="CLSID:C7C7225A-9476-47AC-B0B0-FF3B79D55E67"
  VIEWASTEXT>

  <PARAM NAME="download.Server" VALUE="http://127.0.0.1/ozapplication">
  <PARAM NAME="download.Port" VALUE="80">
  <PARAM NAME="download.Instruction" VALUE="OZAViewerOCX.idf">
  <PARAM NAME="install.Base" VALUE="<PROGRAMS>/Forcs">
  <PARAM NAME="install.Namespace" VALUE="companyName">
</OBJECT>
```

OZ Application Viewer 호출

OZA 호출 태그

❖ 오즈 서버 타입이 Daemon 인 경우 연결부 Tag 생성

```
<param name="repository_agent.ozserver.server" value="127.0.0.7">
<param name="repository_agent.ozserver.port" value="8003">
```

❖ 오즈 서버 타입이 Servlet 인 경우 연결부 Tag 생성

```
<param name="repository_agent.ozserver.servlet" value="http://127.0.0.1:8080/oz/server">
```

```
<OBJECT bgcolor="#FFFFFF" width = "800" height = "600" ID="OZAppViewer"
  CLASSID="CLSID:907A00F3-7390-4EF2-931B-360546587804" VIEWASTEXT>
  <param name="repository_agent.type" value="FROM_OZSERVER">
  <param name="repository_agent.ozserver.server" value="127.0.0.1">
  <param name="repository_agent.ozserver.port" value="8003">
  <param name="repository_agent.ozp.file_path" value="">
  <param name="ozadoc.path" value="Grid2.oz">
</OBJECT>
```

❖ OBJECT 태그의 기본 구조

```
.....
<param name ="패러미터 이름" Value="패러미터 값">
<param name ="패러미터 이름" Value="패러미터 값">
<param name ="패러미터 이름" Value="패러미터 값">
.....
```

❖ odi 패러미터 사용 예

```
<param name="odi.ozcar.pcount" value="2">
<param name="odi.ozcar.args1" value="FromDate=00-01-01">
<param name="odi.ozcar.args2" value="ToDate=00-05-31">
```

OZ Viewer 호출시 서버 정보

오즈 서버 정보

- ❖ 오즈 서버 타입이 Daemon 인 경우 연결부 Tag 생성

```
<param name="repository_agent.ozserver.server" value="127.0.0.7">  
<param name="repository_agent.ozserver.port" value="8003">
```

- ❖ 오즈 서버 타입이 Servlet 인 경우 연결부 Tag 생성

```
<param name="repository_agent.ozserver.servlet" value="http://127.0.0.1:8080/oz/server">
```

Appendix

OZEdu.mdb ■ ODBC로 이용하기

OZEdu.mdb를 ODBC로 이용하기

❖ [제어판] – [관리도구] – [데이터 원본 ODBC] – [System DSN]에서 [Add] 선택

