목 차

Ⅰ. 시작하기 전에 5
본 매뉴얼의 목적 6
본 매뉴얼 구성 6
매뉴얼 로드맵 8

Ⅱ. OZ Application Designer 실행 9
시스템 요구사항 10
OZ Application Designer 파일 구성 11
OZ Application Designer 실행 16

Ⅲ. OZ Application Designer 화면 구성 17
OZ Application Designer 화면 구성 18
타이틀바 20
메뉴바 21
툴바 59
트리창 65
속성창, 이벤트창 79
Board창, 스크립트창 82

Ⅳ. OZ Application Designer 사용 84
Document, Board 개념 85

작업 폴더 설정 86
시작 Board 설정 92
컴포넌트 추가 93
컴포넌트 선택 97
속성 값 변경 100
레이아웃 편집 110
스크립트 코딩 118
스크립트 디버깅 131
스크립트 비교 141
전역 변수, 전역 함수 147

Ⅴ. 컴포넌트 기본 사용 152
Controls 컴포넌트 153
Invisibles 컴포넌트 205
Container Controls 컴포넌트 218
Complex Controls 컴포넌트 224
Drawings 컴포넌트 225
OZC (OZ Component) 236
EventHook 247

Ⅵ. 데이터 연동 251
데이터 연동 가능 컴포넌트 252
ODI 만들기, 추가 254
마법사로 Table에 데이터 표시 260
스크립트로 Grid에 데이터 표시 268
DataAction:Insert, Delete, Update 277
수식 필드 사용 287
FXData 모듈 사용 293

Ⅶ. 마법사 사용법 302
Chart 마법사 303
CheckListBox 마법사 340
ComboBox 마법사 342
ContextMenu/MainMenu/TrayContextMenu 마법사 344
CrossTab 마법사 347
ListBox 마법사 352
ListView 마법사 353
RichTextBox 마법사 358
TabControl 마법사 359
Table 마법사 360
TitleBar 마법사 366
TreeView 마법사 368

Ⅷ. OZ Function Editor 370
OZF(OZ Function) 개념 371
OZ Function Editor 실행 372
OZ Function Editor 화면 구성 373
메뉴바 375
툴바 385
OZF 만들기 386
OZS 사용하기 387

Ⅸ. OZ Style Editor 391
OZS(OZ Style) 개념 392
OZ Style Editor 실행 393
OZ Style Editor 화면 구성 394
메뉴바 396
툴바 406

OZS 만들기 407
OZS 사용하기 413

X. 서버 리파지토리 사용법 419
서버 리파지토리 개요 420
서버 리파지토리 연결 420
서버로부터 파일 열기 426
파일 업로드, 다운로드 429
체크인, 체크아웃 431
히스토리 보기 434
아이템 찾기 436

Ⅰ. 시작하기 전에

- 본 매뉴얼의 목적
- 본 매뉴얼 구성
- 매뉴얼 로드맵

본 매뉴얼의 목적

본 매뉴얼은 웹 기반의 기업용 Application 개발 솔루션인 'OZ Application Designer' 개발자를 대상으로, 'OZ Application Designer'의 화면 구성과 사용 방법 및 'OZ Application Designer'를 이용하여 데이터와 연동하는 방법, OZC, OZS, OZF 개념과 사용 방법 등을 소개하기 위해 제작되었습니다.

본 매뉴얼 구성

본 매뉴얼 구성과 각 장에서 설명하는 내용은 다음과 같습니다.

OZ Application Designer 실행

OZ Application Designer의 파일 구성 및 실행 방법에 대해 설명합니다.

OZ Application Designer 화면 구성

메뉴바, 툴바, 트리창, 속성창 등 OZ Application Designer의 사용자 인터페이스에 대해 설명합니다.

OZ Application Designer 사용

Document, Board의 개념과 OZ Application Designer를 사용하기 위해 알아두어야할 기본적인 사용 방법에 대해 설명합니다.

컴포넌트 기본 사용

OZ Application Designer에서 제공되는 컴포넌트의 개념과 컴포넌트의 기본적인 속성, 메소드, 사용 방법과 OZC, EventHook 등에 대해 설명합니다.

데이터 연동

데이터 셋을 연결하고 데이터를 조회, 추가, 삭제, 변경하는 방법과 수식 필드를 사용하는 방법을 예제를 통해 설명합니다.

마법사 사용법

OZ Application Designer에서 제공하는 마법사를 사용하는 방법에 대해 설명합니다.

OZ Function Editor

OZF의 개념, OZ Function Editor의 사용 방법, OZF를 프로젝트에 추가하고 사용하는 방법 대해 설명합니다.

OZ Style Editor

OZS의 개념, OZ Style Editor의 사용 방법, OZS를 프로젝트에 추가하고 사용하는 방법 대해 설명합니다.

서버 리파지토리 사용법

서버 리파지토리의 개념과 OZ Application Designer에서 서버 리파지토리를 이용하는 방법에 대해 설명합니다.

매뉴얼 로드맵

OZ Application Designer를 이용하여 프로젝트를 개발 시 도움을 주기 위해 제공되는 매뉴얼은 아래와 같습니다.

Manual	Description
오즈 애플리케이션 Getting Started	오즈 애플리케이션에 대한 이해와 오즈 애플리케이션 디자이너를 이용하여 프로젝트를 만들고 웹에 적용하는 기본적인 과정을 소개합니다.
오즈 애플리케이션 뷰어 사용자 가이드	오즈 애플리케이션 뷰어를 설치하고 실행하는 방법과 뷰어 패러미터와 활용 방법 등에 대해 설명합니다.
오즈 리포트 뷰어 사용자 가이드	보고서를 웹을 통해 조회하고 사용자 애플리케이션과 연동할 수 있도록 뷰어 패러미터와 활용 방법 등에 대해 설명합니다.
오즈 쿼리 디자이너 사용자 가이드	오즈 쿼리 디자이너를 이용하여 데이터셋 정보를 작성하는 방법을 설명합니다.
오즈 리파지토리 매니저 사용자 가이드	오즈 리파지토리 매니저를 사용하여 오즈 폼 및 데이터 파일 등 각종 아이템을 포괄적으로 관리하는 방법을 설명합니다.
오즈 엔터프라이즈 매니저 사용자 가이드	서버 관리자가 오즈 서버를 효율적으로 관리할 수 있도록 오즈 서버의 설치, 관리, 운영 방법 등을 설명합니다.
에러 메시지 레퍼런스	오즈 제품에서 보여지는 에러 메시지에 대해 설명합니다.

Ⅱ. OZ Application Designer 실행

- 시스템 요구 사항
- OZ Application Designer 파일 구성
- OZ Application Designer 실행

본 장에서는 OZ Application Designer의 파일 구성 및 실행 방법에 대해 설명합니다.

시스템 요구 사항

OS

Windows NT 4.0 이상, Windows XP, Windows 2000, Windows 2003

JRE 1.3 이상

※ 참고사항

Windows 95/98/ME에서는 Unicode를 지원하지 않으므로 사용할 수 없습니다.

CPU

Pentium Ⅲ 500MHz 이상

Main Memory

최소 256MB (512MB 이상 권장)

HDD

최소 100MB 이상의 빈 공간 필요

모니터

컬러 XGA (1024 x 768 이상의 해상도 권장)

OZ Application Designer 파일 구성

다음은 OZ Application Designer를 설치한 다음 생성되는 디렉토리와 파일의 종류 및 기능에 대해 설명합니다.

Folder	File Name	Description
%OZA_HOME%	Console.txt	OZ Application Designer 로그 파일
	dbconfig.xml	DB 벤더 연결 정보 설정 파일
	dbconfig.xsd	
%OZA_HOME%bin	msado15.dll	ADORecordSet 변환 라이브러리
	OZApplicationDesigner.exe	OZ Application Designer 실행 파일
	OZApplicationFunctionEditor.exe	OZ Function Edtior 실행 파일
	OZApplicationStyleEditor.exe	OZ Style Editor 실행 파일
	ozatransparent.dll	다이얼로그로 띄운 Board를 투명하게 설정할 때 사용되는 라이브러리
	OZAViewer.exe	OZ Application Viewer 실행 파일
	OZAViewerDll.dll	OZ Application Viewer 라이브러리
	OZAViewerOCX.ocx	OZ Application Viewer ActiveX United Control 파일
	ozxlswriter.dll	Excel Export 라이브러리
	OZForVista.dll	Windows Vista 관련 라이브러리
	OZLabelPtnWiz_vc71.dll	패턴 속성 라이브러리
	OZOptWiz_vc71.dll	OZ Report Viewer 속성 라이브러리
	OZRepositoryManager.ini	리파지토리 정보 설정 파일
	OZUSLClientDes128.dll	USL 관련 라이브러리
	OZUSLClientSession.dll	
	ScriptDebugger.dll	스크립트 디버그용 라이브러리
	unicows.dll	Microsoft Layer of Unicode 라이브러리
	user.cif	사용자 정보 파일

%OZA_HOME%bin/res	ad_언어명_국가명.dll	OZ Application Designer 리소스 예) ad_ko_kr.dll
	ad_error_msg_언어명_국가명.xml	OZ Application Designer 에러 발생시 코드별 메시지 리소스 예) ad_error_msg_ko_kr.xml
	af_언어명_국가명.dll	Function Editor 리소스 예) af_ko_kr.dll
	af_error_msg_언어명_국가명.xml	Function Editor 에러 발생시 코드별 메시지 리소스 예) af_error_msg_ko_kr.xml
	as_언어명_국가명.dll	Style Editor 리소스 예) as_ko_kr.dll
	as_error_msg_언어명_국가명.xml	Style Editor 에러 발생시 코드별 메시지 리소스 예) as_error_msg_ko_kr.xml
	av_error_msg_언어명_국가명.xml	OZ Application Viewer 에러 발생시 코드별 메시지 리소스 예) av_error_msg_ko_kr.xml
	chartres_언어명_국가명.xml	차트 종류별 속성 리소스
	label_pattern_언어명_국가명.xml	Label 패턴 속성 정의 리소스 예) label_pattern_ko_kr.xml
	MultiLanguageList.xml	언어 리소스
	rm_언어명_국가명.dll	리파지토리 리소스 예) rm_ko_kr.dll
	rm_error_msg_언어명_국가명.xml	리파지토리 에러 발생시 코드별 메시지 리소스 예) rm_error_msg_ko_kr.xml
	sd_언어명_국가명.dll	스크립트 디버깅 리소스 ex)sd_ko_kr.dll
	server_error_msg_언어명_국가명.xml	서버 에러 발생시 코드별 메시지 리소스 예) server_error_msg_ko_kr.xml
	viewer_parameter_언어명_국가명.xml	OZ Report Viewer 패러미터 리소스 예) viewer_parameter_ko_kr.xml
%OZA_HOME%config	assembly.properties	어셈블리 정보 설정 파일
	launch.cfg	OZ Application Designer 설정 파일
	OZApplicationDesigner.ini	OZ Application Designer 정보 파일

	OZApplicationDesignerLayout.xml	OZ Application Designer Layout 정보 파일
	OZApplicationFucntionEditor.ini	OZ Application Fucntion Editor 정보 파일
	OZApplicationFunctionEditorLayout.xml	OZ Application Fucntion Editor Layout 정보 파일
	OZApplicationStyleEditor.ini	OZ Application Style Editor 정보 파일
	OZApplicationStyleEditorLayout.xml	OZ Application Style Editor Layout 정보 파일
	OZOpenHistory.xml	폴더 히스토리 정보 저장 파일
	OZServerList.xml	OZ Repository Server 리스트 정보 파일
	xml.properties	XML 파서 설정 파일
%OZA_HOME%doc	CFFunc_5.0.xml	수식 필드 도움말 리소스
	codehelp_5.0.xml	스트립트 팝업 헬프창 리소스
	comphelp_5.0_언어명_국가명.xml	속성창에 표시되는 속성 도움말 리소스
	OZAPPAPI_언어명_국가명.chm	OZ Application Designer 스크립트 API 도움말 파일
	OZAPPCHILD_언어명_국가명.chm	OZ Application Designer 스크립트 API 파일(도움말을 디자이너에 도킹할 경우 사용되는 도움말 파일)
%OZA_HOME%images	OZ Application Designer에서 사용하는 이미지 파일	
%OZA_HOME%lib	activation.jar	메일 관련 라이브러리
	mail.jar	
	dtdparser120.jar	DTD 파일 파싱 클래스 라이브러리
	jcert.jar	J2EE 라이브러리(HTTPS/SSL 지원을 위해 필요한 라이브러리) JRE 1.3.X의 경우에만 필요함
	jnet.jar	
	jsee.jar	
	jsr173_api.jar	파일/HTTP 스토어의 XML 데이터셋 관련 라이브러리
	stax-1.2.0.jar	
	log4j.jar	Log 사용에 대한 라이브러리
	ozsfw50.jar	오즈 서버 라이브러리
	parser.jar	XML 파서 관련 클래스 라이브러리
	xerces.jar	

	dom4j-1.6.1.jar	
	jaxen-1.1.6.jar	
	xalan.jar	XML XPath 관련 클래스 라이브러리
	xercesImpl.jar	
	xml-apis.jar	
	soap2_3_1.jar	SOAP 관련 라이브러리
	jazzlib.jar	파일 압축 라이브러리
	js.jar	Java Script 구동 라이브러리
%OZA_HOME%license	ozlicense.xml	라이센스 파일
%OZA_HOME%UniversalDriver	classes12.jar	Oracle 8i, 9i용 JDBC Driver
	classes12.zip	
	db2java.jar	IBM DB2용 JDBC 드라이버
	db2java.zip	
	db2jcc.jar	IBM DB2용 JDBC 드라이버 Type 8 (DB2 버전 8 이상)
	db2jcc_license_cisuz.jar	
	db2jcc_license_cu.jar	
	ifxjdbc.jar	Infomix용 JDBC 드라이버
	ijava.jar	Oracle 7i 관련 JDBC 드라이버
	jconn2.jar	Sybase ASE 용 JDBC 드라이버
	jkjdbc.jar	MSSQL 97 JDBC 드라이버
	jt400.jar	IBM JT400 용 JDBC 드라이버
	msbase.jar	MSSQL 2000 JDBC 드라이버
	mssqlserver.jar	
	msutil.jar	
	mysql5.1.5.jar	Mysql용 JDBC 라이브러리
	ozsapjco2.jar	SAPJCO2 용 라이브러리
	ozsapjco3.jar	SAPJCO3 용 라이브러리 ※ 주의사항 : SAPJCO3 버전은 JRE 1.5 이상에서만 동작하므로, launch.cfg에 설정된 JRE 버전을 변경한 후 사용하시기 바랍니다.
	sqljdbc.jar	MSSQL 2008 JDBC 드라이버

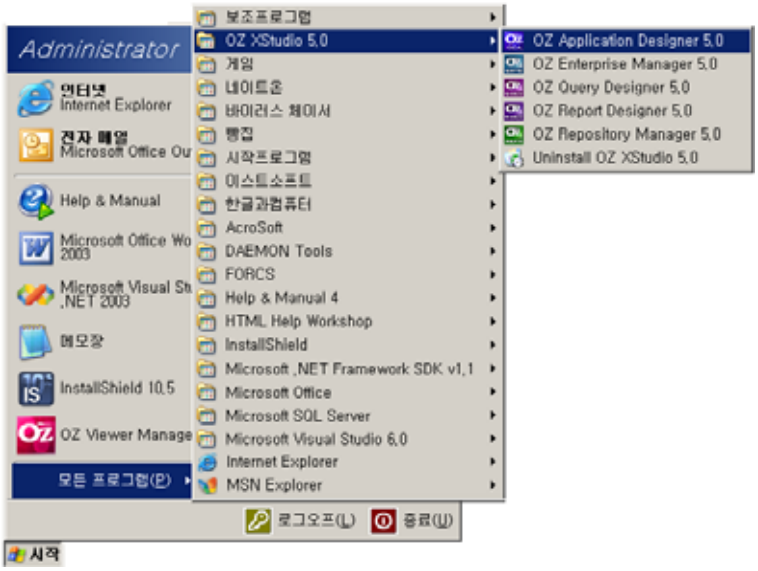
	unisqljdbc10.jar	UniSQL용 JDBC 드라이버
	unsqljdbc205.zip	

※ 주의사항 : 'UniversalDriver' 폴더의 파일은 기본적으로 제공하는 JDBC 드라이버로, 이미 사용중인 RDB의 JDBC 드라이버와 호환되지 않거나 충돌이 발생할 수 있습니다. 그러므로 이미 사용중인 RDB의 JDBC 드라이버를 사용할 것을 권장합니다.

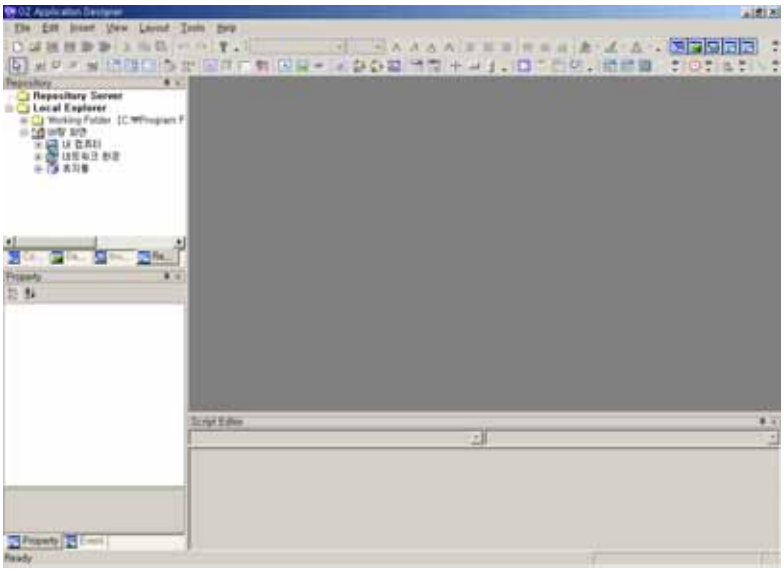
OZ Application Designer 실행

다음은 OZ Application Designer를 실행하는 방법에 대해 설명합니다.

윈도우 시작 프로그램의 [프로그램] → [OZ XStudio 5.0] → [OZ Application Designer 5.0]을 클릭하거나 윈도우 바탕 화면의 'OZ Application Designer 5.0' 아이콘을 클릭합니다.



OZ Application Designer가 실행되면 아래과 같은 초기 화면이 나타납니다.



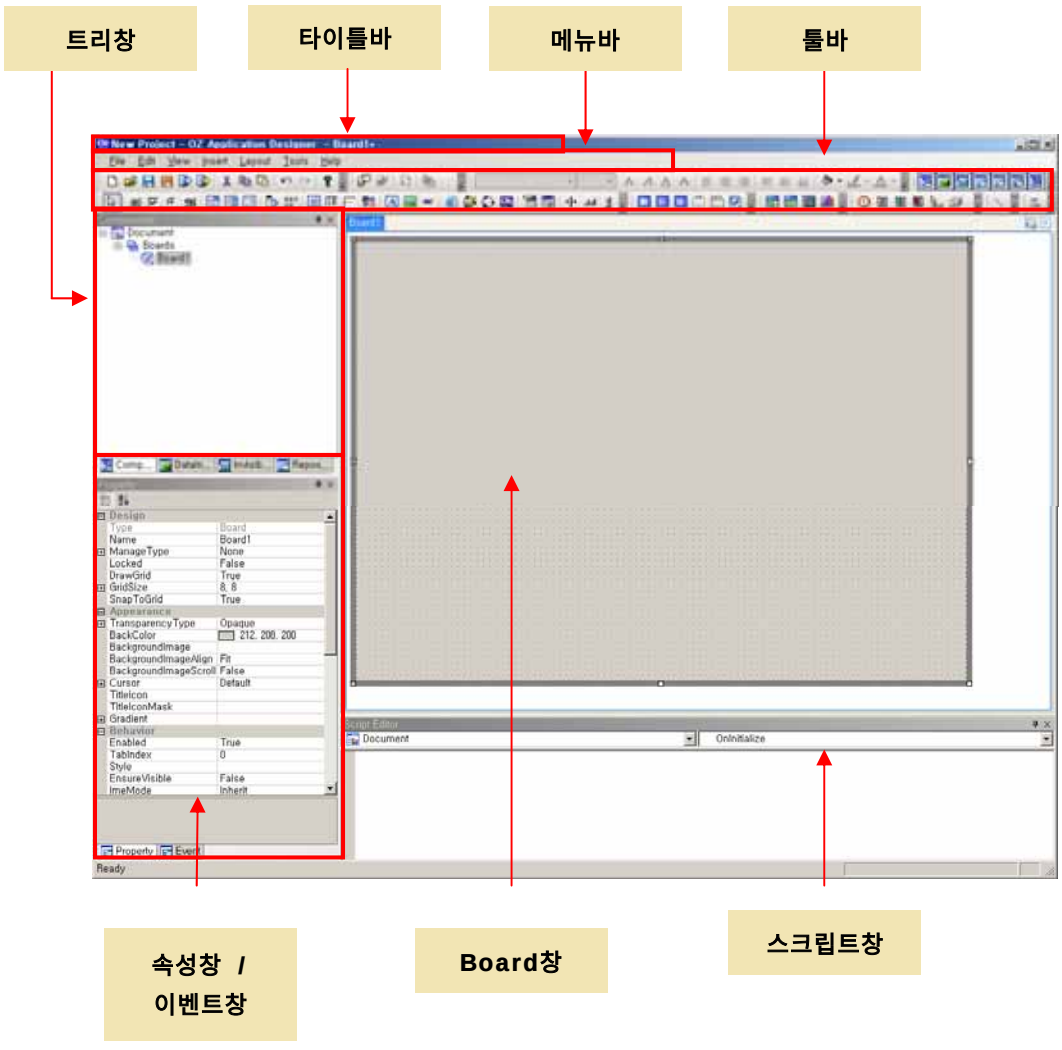
Ⅲ. OZ Application Designer 화면 구성

- OZ Application Designer 화면 구성
- 타이틀바
- 메뉴바
- 툴바
- 트리창
- 속성창, 이벤트창
- Board창, 스크립트창

본 장에서는 타이틀바, 메뉴바, 툴바, 트리창, 속성창 등 OZ Application Designer의 사용자 인터페이스에 대해 설명합니다.

OZ Application Designer 화면 구성

다음은 OZ Application Designer의 사용자 인터페이스 화면이며, 각 구성 요소의 이름과 기능은 아래와 같습니다.



Window	Description
타이틀바	현재 OZ Application Designer에 열려있는 프로젝트의 카테고리와 파일명을 표시하고, 작업 중인 보드명을 표시합니다.
메뉴바	메뉴바는 OZ Application Designer의 상단에 풀다운(pull down)형식으로 제공되며, 파일, 편집, 추가, 뷰, 레이아웃, 도구, 도움말로 구성됩니다.
툴바	OZ Application Designer의 일반적인 기능, 글꼴 지정, 컴포넌트 추가, 레이아웃, 윈도우 표시 등을 쉽고 빠르게 설정할 수 있도록 아이콘 형태로 제공됩니다. 파일 관련 기능과 잘라내기, 복사 등 일반적인 기능을 실행하는 '표준 툴바', 글꼴, 정렬방식, 색 등 텍스트 포맷을 설정하는 '포맷 툴바', Button, Label, Table 등 컴포넌트를 선택하는 '컴포넌트 툴바', 컴포넌트의 레이아웃을 설정하는 '레이아웃 툴바', 트리창, 속성창 등 윈도우의 표시 여부를 설정하는 '뷰 툴바', 서버 리파지토리와 관련된 기능을 실행하는 '리파지토리 툴바', ActiveX 컨트롤을 선택하는 'ActiveX 컨트롤 툴바'로 구성됩니다.
트리창	OZ Application Designer의 리파지토리 정보와 현재 프로젝트에 추가된 컴포넌트, ODI, OZF, OZS 등을 표시하는 트리창이 탭 형태로 제공됩니다. Document, Board, Board에 추가된 컴포넌트, 컴포넌트의 순서, 컴포넌트간 부모-자식 관계 등을 트리 형태로 표시하는 '컴포넌트 트리창', 추가한 ODI 파일의 데이터 소스, 데이터 셋, 필드 정보, 패러미터 등을 트리 형태로 표시하는 '데이터 트리창', 프로젝트에 추가한 컴포넌트 중 Invisibles 컴포넌트, OZF, OZS, EventHook 등을 트리 형태로 표시하는 'Invisibles 트리창', 로컬 및 서버의 리파지토리의 폴더 및 파일을 탐색기 형태로 표시하는 '리파지토리창'으로 구성됩니다.
속성창 / 이벤트창	프로젝트에 추가된 컴포넌트의 속성과 이벤트를 설정하고 표시하는 창이 탭 형태로 제공됩니다. 컴포넌트의 속성 값을 표시하고 설정하는 '속성창', 컴포넌트의 이벤트와 해당 이벤트의 스크립트 코딩 여부를 표시하는 '이벤트창'으로 구성됩니다.
Board창	프로젝트에 추가된 Board를 탭 형태로 표시합니다. 특정 Board를 선택하면 해당 Board에 추가된 자식 컴포넌트를 표시하고 컴포넌트의 위치, 크기 조절 등 레이아웃을 디자인할 수 있습니다.
스트립트창	Document, Board, 컴포넌트 등 프로젝트에 추가된 구성 요소의 스크립트를 코딩하거나 코딩된 내용을 표시합니다.

타이틀바

현재 OZ Application Designer에 열려있는 프로젝트의 카테고리와 파일명을 표시하고, 작업 중인 보드명을 표시합니다.

타이틀바에 표시되는 형식은 아래와 같습니다.
/카테고리/파일명 – 오즈 어플리케이션 디자이너 – *보드명*

- /카테고리/파일명
  작업 폴더를 기준으로 현재 작업 중인 프로젝트 파일의 경로와 파일명을 표시합니다.
  새 프로젝트에서 작업할 경우에는 /카테고리/파일명으로 표시되지 않고 New Project로 표시됩니다.

- 보드명
  현재 작업중인 Board의 이름을 표시합니다.

- 적용 예
  - /Project에 있는 TitleBar.oza 파일을 열었을 경우

    현재 작업중인 보드명이 "MainBoard"인 경우입니다.

  - 프로젝트를 새로 만들어 작업할 경우

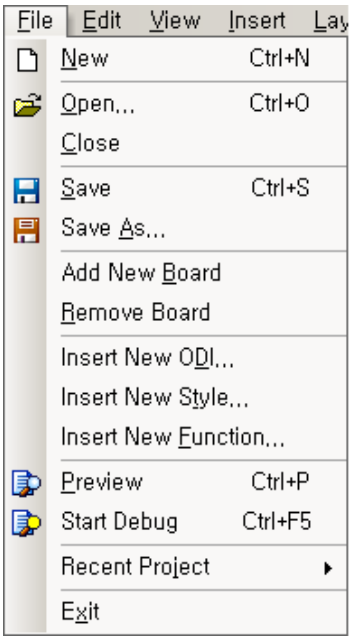
    현재 작업중인 보드명이 "Board1"인 경우입니다.

메뉴바

메뉴바는 OZ Application Designer의 상단에 풀다운(pull down)형식으로 제공되며, 파일, 편집, 추가, 뷰, 레이아웃, 도구, 도움말로 구성됩니다.

파일(File)

메뉴바의 [File] 메뉴를 선택하면 아래와 같은 파일 관련 메뉴가 나타납니다.



- New (Ctrl+N)

  새로운 프로젝트를 생성합니다. 프로젝트를 새로 생성하면 Board가 자동으로 추가됩니다.

- Open (Ctrl+O)

  저장된 프로젝트 파일을 읽어 옵니다.

- Close

  현재 작업 중인 프로젝트를 닫습니다.

- Save (Ctrl+S)
  현재 작업 중인 프로젝트를 저장합니다. OZ Application Designer의 프로젝트는 확장자가 oza인 파일로 저장됩니다.

- Save As
  현재 작업 중인 프로젝트를 다른 이름으로 저장합니다.

- Add New Board
  프로젝트에 새로운 Board를 추가합니다.

- Remove Board
  선택한 Board를 삭제합니다. Board를 삭제하면 Board에 추가된 모든 자식 컴포넌트도 함께 삭제됩니다.

- Insert New ODI
  프로젝트에 ODI 파일을 추가합니다. 추가된 ODI 파일의 정보는 데이터 트리창에서 확인할 수 있습니다.

- Insert New Style
  프로젝트에 OZ Style 파일(OZS)을 추가합니다. 추가된 OZS 파일의 정보는 Invisibles 트리창에서 확인할 수 있습니다.

- Insert New Function
  프로젝트에 OZ Function 파일(OZF)을 추가합니다. 추가된 OZF 파일의 정보는 Invisibles 트리창에서 확인할 수 있습니다.

- Preview (Ctrl+P)
  작업 중인 프로젝트를 미리보기합니다. 미리보기 된 상태에서 스크립트로 코딩된 내용을 실행할 수 있습니다.

  ※ 참고사항 : SDM으로 미리보기

  - 개요
    일반적인 미리보기는 매번 데이터를 바인딩하는 반면에 "SDM으로 미리보기"는 미리 데이터를 저장해 놓은 SDM 파일에 있는 데이터를 참조하여 바인딩합니다.
  - 사용방법

- 키보드의 <Shift> 또는 <Ctrl + Shift> 키를 누른 상태에서 메뉴바의 [Preview] 메뉴나 툴바의 미리보기 아이콘을 클릭하면 "SDM으로 미리보기"로 동작합니다.
- <Shift> 키를 누른 상태에서 미리보기 메뉴나 아이콘을 클릭하면 ODI 파일과 같은 위치에 있는 같은 이름의 SDM 파일을 찾아 미리보기합니다. 이 때 같은 이름의 SDM 파일을 찾지 못한 경우엔 각 ODI 파일별로 SDM 파일을 선택하는 파일 선택창이 표시됩니다.
- <Ctrl + Shift> 키를 누른 상태에서 미리보기 메뉴나 아이콘을 클릭하면 무조건 모든 ODI 파일별로 SDM 파일을 선택하는 파일 선택창이 표시됩니다.

※ 주의사항 : SDM으로 미리보기

- "SDM으로 미리보기" 기능은 ODI 파일이 추가된 프로젝트인 경우에만 실행할 수 있습니다.
- SDM으로 미리보기한 경우 항상 SDM 파일에서 데이터를 가져오기 때문에 데이터가 새로 갱신되었어도 새로 바뀐 내용은 반영되지 않습니다.
- SDM으로 미리보기한 경우 모든 ODI는 FetchUnit이 DM_PER_DATAMODULE으로 동작합니다.
- 여러 개의 SDM 파일을 이용한 프로젝트를 SDM으로 미리보기할 경우 SDM 파일은 추가된 ODI의 역순으로 선택하여야 제대로 동작됩니다. 예를 들어 데이터 트리창에 "A.odi → B.odi → C.odi" 순으로 추가되어있다면, SDM 파일을 선택할 경우에는 "C.sdm → B.sdm → A.sdm" 순으로 선택하여야 합니다.

■ Start Debug (Ctrl+F5)

작성한 스크립트를 디버깅합니다. [Start Debug] 메뉴를 클릭하면 작성한 스크립트를 디버깅하면서 실행 결과를 확인할 수 있도록 디버깅창과 미리보기창이 뜨며, 디버깅창에 대한 사용법은 본 매뉴얼의 "IV. OZ Application Designer 사용의 스크립트 디버깅" 부분을 참조하시기 바랍니다.

■ Recent Project

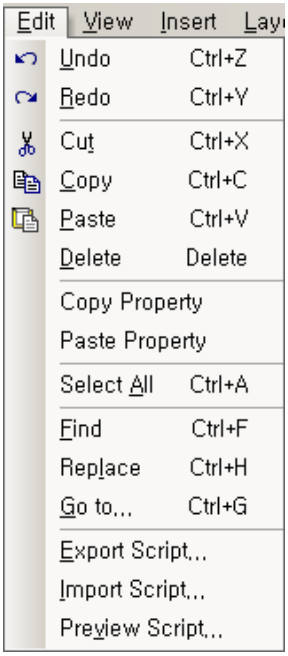
최신에 열어본 프로젝트 파일(OZA)을 선택합니다. 최근에 열어본 OZA 파일 중 최대 4개의 프로젝트 파일을 서브 메뉴로 표시합니다.

■ Exit

OZ Application Designer를 종료합니다.

편집(Edit)

메뉴바의 [Edit] 메뉴를 선택하면 아래와 같은 편집용 메뉴가 나타납니다.



- Undo (Ctrl+Z)

  바로 전에 실행했던 작업을 취소하고 이전 상태로 되돌립니다. Board창에 포커스를 두고 [Undo] 메뉴를 클릭하면 해당 Board에서 작업한 내용이 취소되고, 스크립트창에 포커스를 두고 [Undo] 메뉴를 클릭하면 해당 스크립트창에서 입력한 내용이 취소됩니다. 최대 256회까지 실행 취소가 가능합니다.

- Redo (Ctrl+Y)

  실행 취소했던 작업을 다시 복구하며, 실행 취소한 횟수만큼 복구 가능합니다.

- Cut (Ctrl+X)

  선택한 컴포넌트나 스크립트를 클립보드로 복사하고 현재 Board창이나 스크립트창에서는 삭제합니다. 컴포넌트 트리창에서 Board를 선택하고 잘라내기하면 Board 뿐만 아니라 Board의 모든 자식 컴포넌트까지 한꺼번에 클립보드로 복사하고, '컴포넌트 트리창'에서는 삭제합니다.

- Copy (Ctrl+C)

  선택한 컴포넌트나 스크립트를 클립보드로 복사합니다. 컴포넌트 트리창에서 Board를 선택

하고 복사하면 Board 뿐만 아니라 Board의 모든 자식 컴포넌트까지 한꺼번에 클립보드로 복사합니다.

- Paste (Ctrl+V)
  복사된 컴포넌트 또는 스크립트를 붙여넣기합니다.

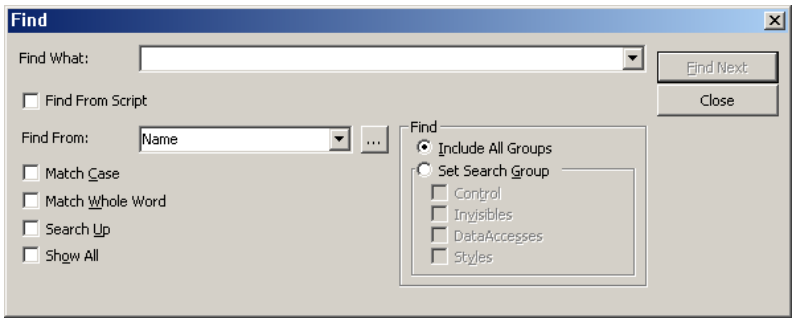
- Delete (Delete)
  선택한 컴포넌트나 스크립트를 삭제합니다. [Undo] 메뉴를 클릭하여 삭제한 작업을 취소할 수 있습니다.

- Copy Property
  선택된 컴포넌트의 속성을 복사합니다.

- Paste Property
  복사된 속성을 선택된 컴포넌트에 붙여넣기합니다. 동일한 컴포넌트 간에는 모든 속성에 대하여 붙여넣기가 가능하고, 서로 다른 컴포넌트 간에는 붙여넣을 컴포넌트의 속성 중 복사한 컴포넌트의 속성과 동일한 항목에 대해서만 복사가 가능합니다.

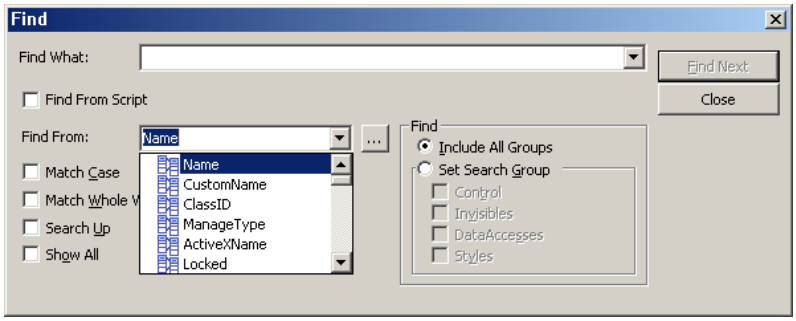
- Select All (Ctrl+A)
  Board창에 포커스가 있을 경우에는 현재 Board의 모든 컴포넌트를 선택하고, 스크립트창에 포커스가 있을 경우에는 현재 스크립트의 전체 내용을 선택합니다.

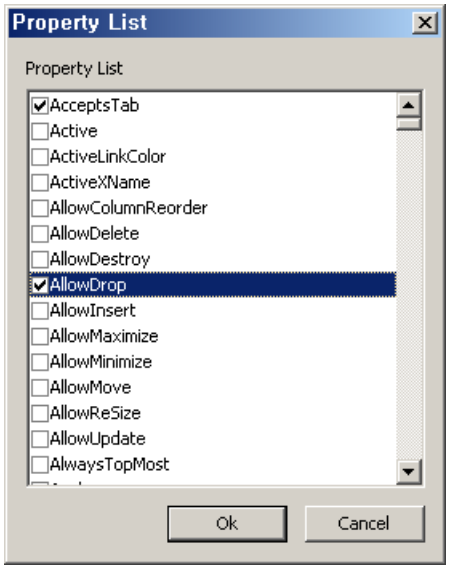
- Find (Ctrl+F)
  속성 값을 찾거나 스크립트 문자열을 찾습니다. Board창에 포커스가 있을 경우에는 프로젝트에 추가된 컴포넌트, ODI, OZS 중 지정한 속성 값에 해당하는 컴포넌트, ODI, OZS를 찾는 '속성 값 찾기 다이얼로그'가 표시되고, 스크립트창에 포커스가 있을 경우에는 입력한 문자열을 찾는 '스크립트 문자열 찾기 다이얼로그'가 표시됩니다.
  - 속성 값 찾기



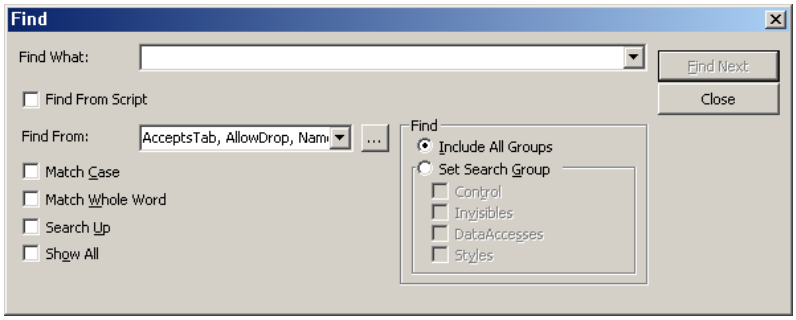
검색할 속성 값을 'Find What' 콤보 박스에 입력하고, 검색할 속성명을 'Find Form' 콤보 박스에서 선택한 후 검색 옵션을 설정합니다. [Find Next] 버튼을 클릭하여 속성 값을 찾습니다.

검색할 속성명은 여러 개를 선택할 수 있습니다. 검색할 속성명을 여러 개 설정하면 선택한 속성에 설정된 값이 하나라도 일치하는 모든 컴포넌트를 찾습니다.
'Find From' 콤보 박스를 클릭하면 아래 그림과 같이 속성명 리스트가 표시되며, 'Find From' 콤보 박스에서 검색할 속성명을 콤마(,)로 구분하여 직접 입력하거나 Property List 다이얼로그 표시 버튼()을 클릭하여 나타나는 Properyt List 다이얼로그에서 속성명 앞의 체크 박스를 체크하면 여러 개의 속성명을 선택할 수 있습니다.

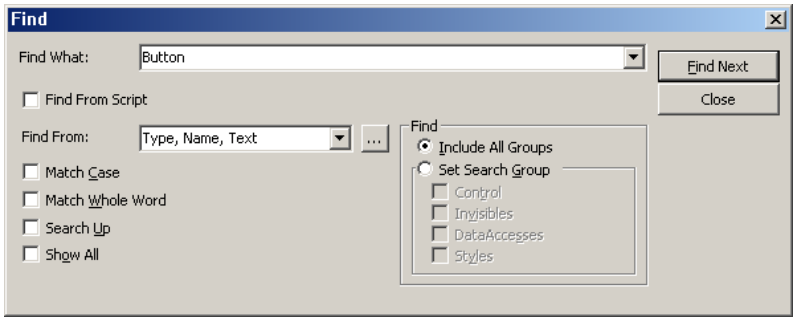




선택된 다수의 속성명은 콤마(,)로 구분되어 표시됩니다.



예를 들어, 컴포넌트의 속성 중 "Type" 또는 "Name" 또는 "Text"에 설정된 속성 값이 'Button'인 모든 컴포넌트를 찾고자 할 경우에는 아래 그림처럼 설정한 후 [Find Next] 버튼을 클릭하여 검색합니다.

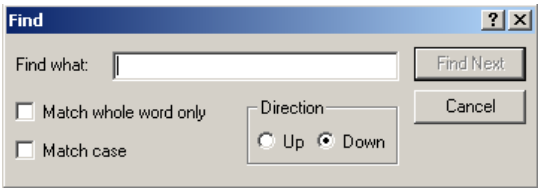


각 옵션과 기능은 아래와 같습니다.

Option	Description	
Find From Script	스크립트에서 속성 값을 찾습니다.	
Match Case	대/소문자 구분 여부를 설정합니다.	
Match Whole Word	전체 단어 단위로 찾을지를 설정합니다.	
Search Up	찾을 방향을 위쪽으로 설정합니다.	
Show All	조건과 일치하는 모든 컴포넌트를 표시합니다.	
Include All Groups	프로젝트 내의 모든 컴포넌트, ODI, OZS를 대상으로 하여 찾습니다.	
Set Search Group	검색할 그룹을 설정합니다.	
	Control	Controls, Container Controls, Complex Controls, Drawings 컴포넌트에서 검색

	Invisibles	Invisibles 컴포넌트에서 검색
	DataAccesses	ODI 파일에서 검색
	Styles	OZS 파일에서 검색

- 스크립트 문자열 찾기



텍스트 박스에 문자열을 입력하고 찾기 옵션을 설정한 후 [Find Next] 버튼을 클릭합니다.

각 옵션과 기능은 아래와 같습니다.

Option	Description
Match Whole World Only	전체 단어 단위로 찾을지를 설정합니다.
Match Case	대소문자를 구분하여 찾을지를 설정합니다.
Direction	찾을 방향을 위, 아래로 설정합니다.

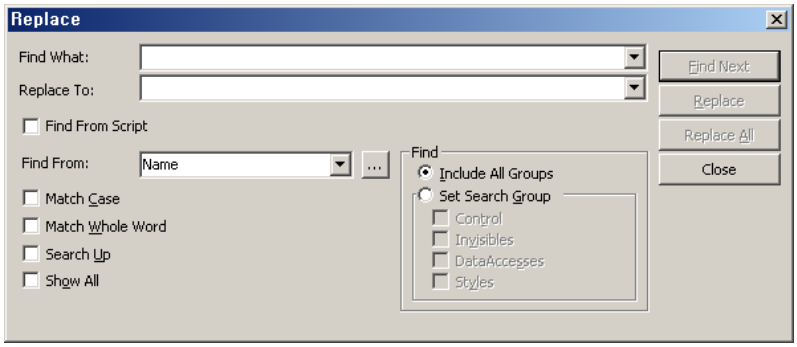
■ Relpace (Ctrl+H)

속성 값을 변경하거나 스크립트 문자열을 변경합니다.

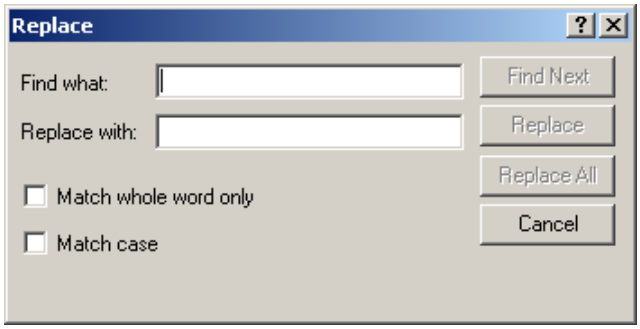
Board창에 포커스가 있을 경우에는 프로젝트에 추가된 컴포넌트, ODI, OZS의 속성 값을 변경하는 '속성 값 변경 다이얼로그'가 표시되고, 스크립트창에 포커스가 있을 경우에는 입력한 문자열을 변경하는 '스크립트 문자열 변경 다이얼로그'가 표시됩니다.

- 속성 값 변경

검색할 속성 값을 'Find What' 콤보 박스에서 입력하고 변경할 속성 값을 'Replace To' 콤보 박스에 입력하고, 해당 속성명을 'Find Form' 콤보 박스에서 선택한 후 검색 옵션을 지정합니다. [Find Next] 버튼을 클릭하여 다시 찾고, [Replace] 또는 [Replace All] 버튼을 이용하여 변경합니다.



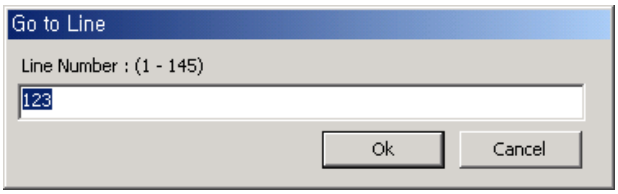
- 스크립트 문자열 변경



찾을 내용과 바꿀 내용을 텍스트 박스에 입력하고 옵션을 설정한 후 [Find Next] 버튼을 클릭하여 다시 찾고, [Replace] 또는 [Replace All] 버튼을 이용하여 변경합니다.

■ Go To (Ctrl+G)

스크립트창의 특정 줄 번호로 편집 커서를 이동합니다. 코딩한 스크립트의 전체 줄수가 다이얼로그에 표시되며 현재 편집 커서가 위치한 줄 번호가 자동으로 표시됩니다. 텍스트 박스에 이동할 줄 번호를 입력한 후 [Ok] 버튼을 클릭하여 편집 커서를 이동시킵니다.



■ Export Script

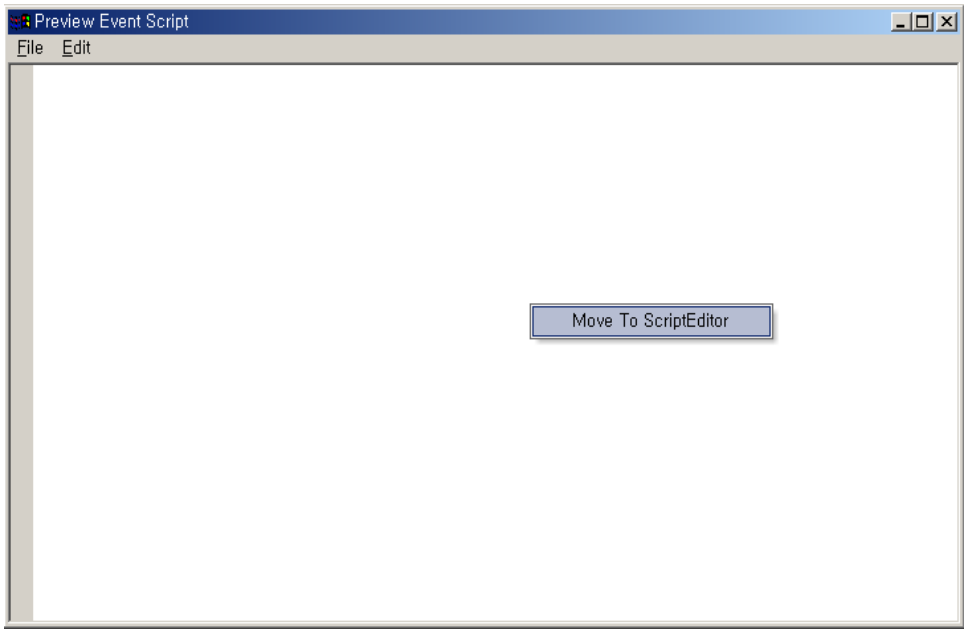
프로젝트에 코딩된 전체 스크립트 내용을 텍스트 파일로 저장합니다.

■ Import Script

텍스트 파일로 저장된 스크립트를 현재 프로젝트에 추가합니다.

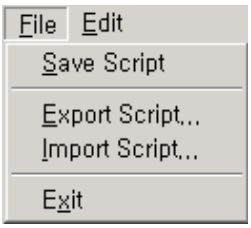
■ Preview Script

프로젝트에 코딩된 전체 스크립트 내용을 미리보기합니다. [Preview Script] 메뉴를 클릭하면 아래 그림과 같이 '스크립트 미리보기창'이 나타나며, '스크립트 미리보기창'에서도 스크립트를 추가, 삭제, 변경할 수 있습니다.



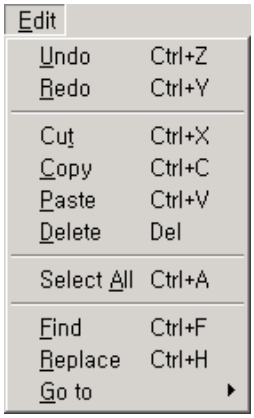
'스크립트 미리보기창'에는 [File] 메뉴와 [Edit] 메뉴, [Move To ScriptEditor] 팝업 메뉴가 있으며 아래와 같이 구성되어 있습니다.

- File



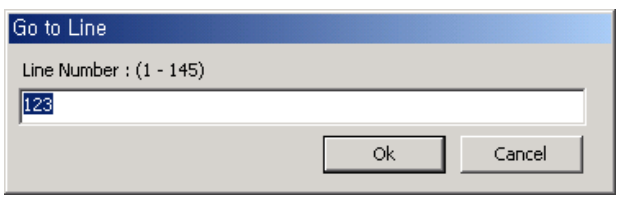
Menu	Description
Save Script	스크립트 미리보기창에서 변경한 내용을 저장합니다.
Export Script	스크립트 내용을 텍스트 파일로 저장합니다.
Import Script	텍스트 파일로 저장된 스크립트를 추가합니다.
Exit	스크립트 미리보기창을 종료합니다.

- Edit



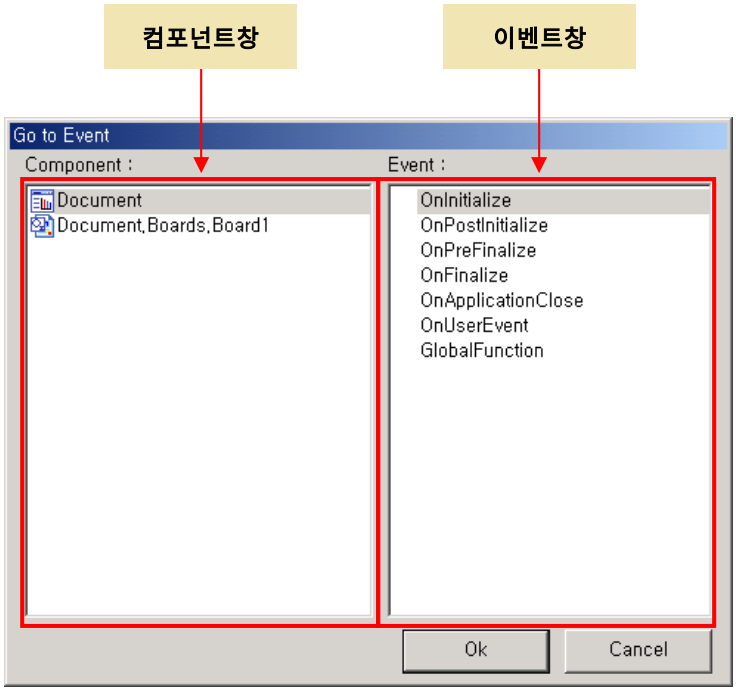
Menu	Description
Undo (Ctrl+Z)	바로 전에 실행했던 작업을 취소하고 이전 상태로 되돌립니다.
Redo (Ctrl+Y)	실행 취소했던 작업을 다시 복구합니다.
Cut (Ctrl+X)	선택한 스크립트를 클립보드에 복사하고 스크립트창에서는 삭제합니다.
Copy (Ctrl+C)	선택한 스크립트를 클립보드에 복사합니다.
Paste (Ctrl+V)	클립보드의 내용을 삽입합니다
Delete (Del)	선택한 스크립트를 삭제합니다.
Select All (Ctrl+A)	스크립트 전체 내용을 선택합니다.
Find (Ctrl+F)	스크립트 문자열을 찾습니다.
Replace (Ctrl+H)	스크립트 문자열을 변경합니다.
Go to	특정 라인으로 편집 커서를 이동하는 [Go to Line] 메뉴와 프로젝트에 포함된 컴포넌트의 이벤트별로 스크립트를 찾아 편집 커서를 이동시키는 [Go to Event] 메뉴로 구성되어 있습니다.

[Go to] 메뉴의 [Go to Line] 메뉴를 클릭하면 아래와 같이 특정 줄 번호로 편집 커서를 이동시키는 다이얼로그가 표시됩니다.



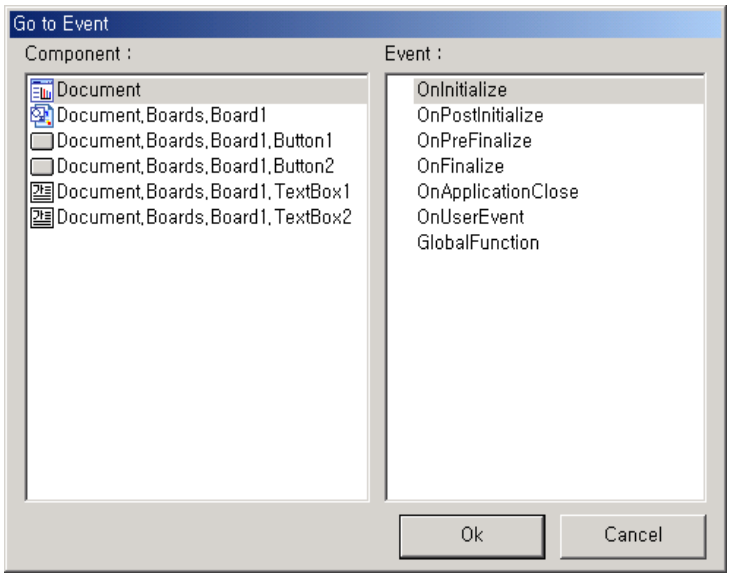
코딩한 스크립트의 전체 줄수가 다이얼로그에 표시되며 현재 편집 커서가 위치한 줄 번호가 자동으로 표시됩니다. 텍스트 박스에 이동하고자 하는 줄 번호를 입력한 후 [Ok] 버튼을 클릭하여 편집 커서를 이동합니다.

[Go to] 메뉴의 [Go to Event] 메뉴를 클릭하면 아래와 같이 '이벤트 이동 다이얼로그'가 표시됩니다. 아래 그림은 프로젝트에 'Board1'만 추가된 상태에서 [Go to Event] 메뉴를 클릭했을 때 모습입니다.

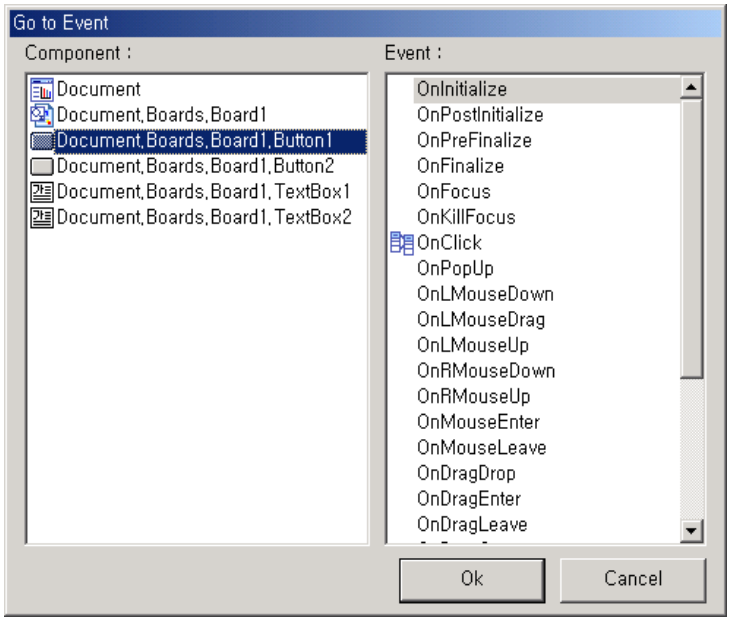


'이벤트 이동 다이얼로그'는 프로젝트에 추가된 모든 컴포넌트가 부모 컴포넌트의 정보를 포함하여 표시되는 컴포넌트창과 선택된 컴포넌트의 모든 이벤트가 표시되는 이벤트창으로 구성됩니다.

예를 들어, 프로젝트에 이름이 'Board1'인 하나의 Board가 있고, Board에 이름이 'Button1', 'Button2'인 두 개의 Button과 이름이 'TextBox1', 'TextBox2'인 두 개의 TextBox가 추가되어 있다고 가정할 경우 [Go to Event] 메뉴를 클릭하면 '이벤트 이동 다이얼로그'는 아래와 같이 표시됩니다.

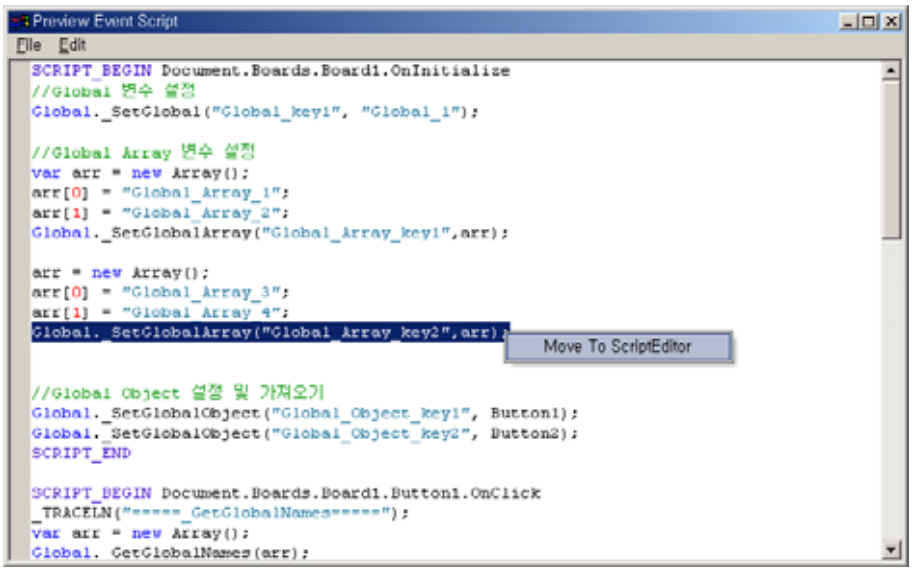


만일 'Button1'의 'OnClick' 이벤트에 스크립트가 코딩되어 있다고 가정할 때, 컴포넌트창에서 'Button1'을 클릭하면 '이벤트 이동 다이얼로그'는 아래와 같이 표시됩니다.



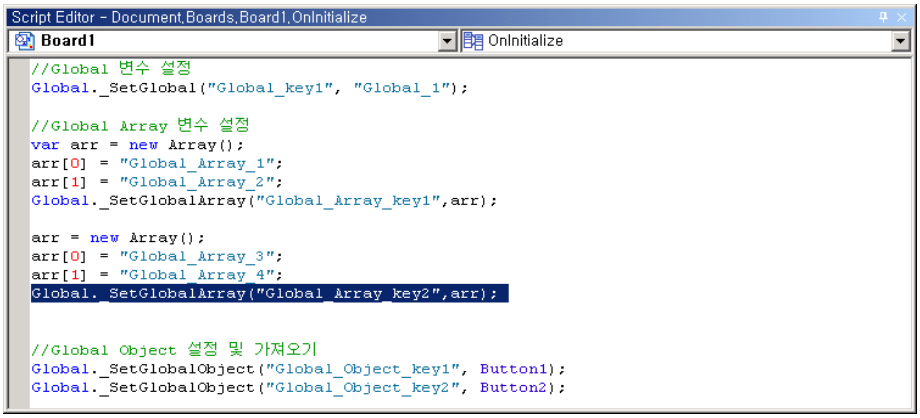
스크립트가 코딩되어 있는 'OnClick' 이벤트의 경우 이벤트 앞에 스크립트 아이콘( )이 표시되며, 어떤 이벤트에 스크립트가 코딩되어 있는지 쉽게 알 수 있습니다. 이 때 'OnClick'을 선택한 후 [OK] 버튼을 클릭하면 'Button1'의 'OnClick' 이벤트에 스크립트를 코딩한 줄 번호로 편집 커서를 이동시킵니다.

- 팝업 메뉴

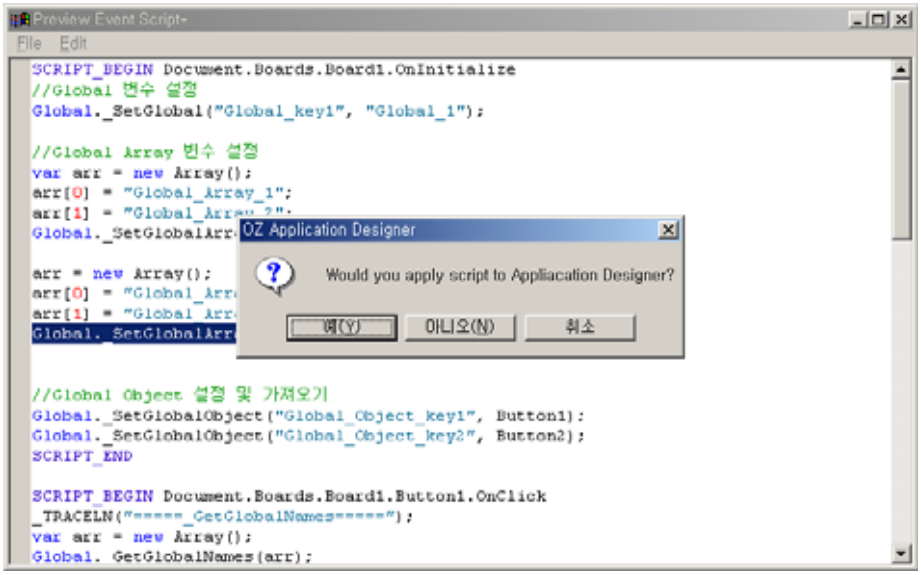


Menu	Description
Move To ScriptEditor	선택한 스크립트가 포함된 스크립트 편집창으로 이동합니다.

특정 스크립트를 선택한 후 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Move To ScriptEditor] 메뉴를 클릭하면 아래와 같이 특정 스크립트가 포함된 스크립트 편집창으로 이동합니다. 아래 그림은 특정 스크립트를 선택한 후 [Move To ScriptEditor] 메뉴를 클릭했을 때 '스크립트 편집창'으로 이동한 모습입니다.



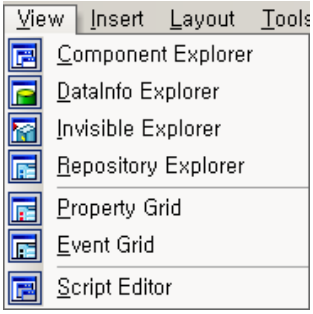
'스크립트 미리보기창'을 닫을 때 스크립트에 변경된 내용이 있을 경우 '저장 여부를 확인하는 다이얼로그'가 아래와 같이 표시됩니다.



이때 [예] 버튼을 클릭하면 수정한 스크립트가 적용된 후 해당 스크립트가 포함된 '스크립트 편집창'으로 이동합니다.

뷰(View)

메뉴바의 [View] 메뉴를 선택하면 트리창, 속성창, 이벤트창, 스크립트창의 표시 여부를 설정할 수 있는 아래와 같은 메뉴가 나타납니다.



- Component Explorer

  Board, Controls, Container Controls, Complex Controls, Drawings 등 프로젝트에 추가된 모든 컴포넌트, 컴포넌트의 순서, 컴포넌트간의 부모-자식 관계를 나타내는 컴포넌트 트리창을 OZ Application Designer 화면에 표시할지 여부를 설정합니다.

- Invisible Explorer

  프로젝트에 추가된 Invisibles 컴포넌트와 OZFunction 파일, OZStyle 파일, EventHook 등을 나타내는 Invisibles 트리창을 OZ Application Designer 화면에 표시할지 여부를 설정합니다.

- DataInfo Explorer

  프로젝트에 추가된 ODI 파일의 데이터 소스, 데이터 셋, 필드 정보, 패러미터등을 나타내는 데이터 트리창을 OZ Application Designer 화면에 표시할지 여부를 설정합니다.

- Repository Explorer

  로컬 및 서버의 리파지토리 내의 폴더 및 파일을 탐색기 형태로 나타내는 리파지토리창을 OZ Application Designer 화면에 표시할지 여부를 설정합니다.

- Property Grid

  컴포넌트의 속성을 설정하고 설정된 속성 값을 표시하는 속성창을 OZ Application Designer 화면에 표시할지 여부를 설정합니다.

- Event Grid

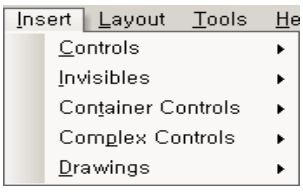
  컴포넌트의 이벤트 종류와 해당 이벤트 내의 스크립트 여부를 확인할 수 있는 이벤트창을 OZ Application Designer 화면에 표시할지 여부를 설정합니다.

- Script Editor

  프로젝트에 추가된 구성 요소에 대한 스크립트를 코딩하거나 코딩된 내용을 표시하는 스크립트창을 OZ Application Designer 화면에 표시할지 여부를 설정합니다.

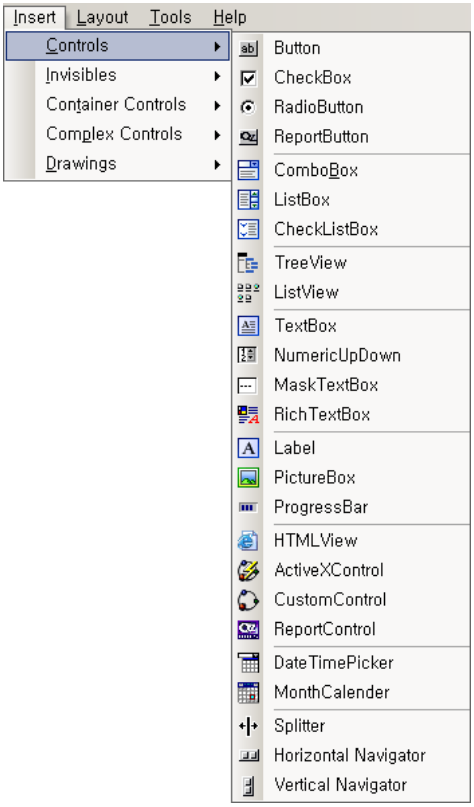
추가(Insert)

메뉴바의 [Insert] 메뉴를 선택하면 추가할 컴포넌트를 선택하는 아래와 같은 메뉴가 나타납니다.



■ Controls

컴포넌트 중 Controls 컴포넌트는 Button 등을 포함한 25개의 컴포넌트로 구성되어 있습니다.



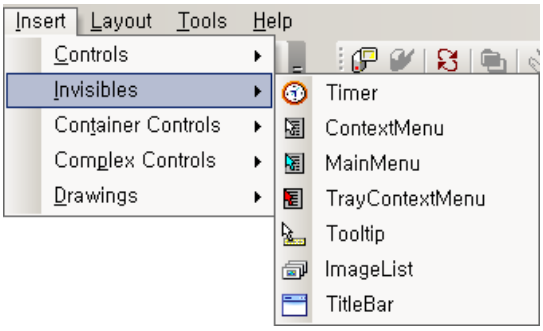
Controls 컴포넌트에 대한 간략한 설명은 다음과 같습니다.

Component	Description
Button	확인, 취소 등 사용자의 명령을 입력받는 컴포넌트
CheckBox	옵션의 선택 여부를 선택하는 컴포넌트로, 부모와 상관없이 다수의 옵션을 선택할 수 있는 컴포넌트
RadioButton	옵션의 선택 여부를 설정하는 컴포넌트로, 한 부모 내에서는 하나의 옵션만을 선택할 수 있는 컴포넌트
ReportButton	OZ Report Designer로 작성된 보고서 파일을 출력하거나 다른 파일로 Export하는 컴포넌트
ComboBox	다수의 아이템을 리스트 형태로 가지고 있으며, [아래 화살표]를 클릭하면 DropDown 형태로 나타내는 컴포넌트

ListBox	다수의 아이템을 리스트 형태로 나타내는 컴포넌트
CheckListBox	체크마크를 가진 아이템을 리스트 형태로 나타내는 컴포넌트
TreeView	다수의 아이템을 다단계의 트리 형태로 나타내는 컴포넌트
ListView	다수의 아이템을 컬럼과 서브 아이템 형태로 나타내는 컴포넌트
TextBox	숫자나 문자열을 입력받는 컴포넌트
NumericUpDown	TextBox와 Up/Down 버튼이 조합된 컴포넌트로, 숫자를 직접 입력하거나 Up/Down 버튼을 클릭하여 숫자값을 증감시키는 컴포넌트
MaskTextBox	특정한 형식을 지정하여 문자열을 입력받는 컴포넌트
RichTextBox	글자별로 텍스트 형식을 달리 설정하여 문자열을 표시할 수 있는 컴포넌트
Label	숫자나 문자열을 표시하는 컴포넌트
PictureBox	이미지를 표시하는 컴포넌트
ProgressBar	진행 상태를 Bar 형태로 나타내는 컴포넌트
HTML View	웹페이지를 표시하는 컴포넌트
ActiveX Control	ActiveX 컨트롤을 연결하는 컴포넌트
Custom Control	Custom 컨트롤을 연결하는 컴포넌트
Report Control	OZ Report Designer로 작성된 보고서 파일을 리포트뷰어와 연동하여 표시하는 컴포넌트
DateTimePicker	날짜 또는 시간을 입력하거나 달력을 표시하는 컴포넌트
MonthCalendar	달력을 표시하는 컴포넌트
Splitter	영역을 분리하는 컴포넌트
Horizontal Navigator	Table 컴포넌트와 연동하여 Table에 표시될 Row수를 나누어 페이징 시키는 컴포넌트로, 맨처음/이전/다음/맨뒤로 페이지를 이동시키는 Button과 페이지 번호를 입력할 수 있는 TextBox를 수평으로 배치한 컴포넌트
Vertical Navigator	Table 컴포넌트와 연동하여 Table에 표시될 Row수를 나누어 페이징 시키는 컴포넌트로, 맨처음/이전/다음/맨뒤로 페이지를 이동시키는 Button과 이전/다음으로 페이지를 이동시키는 VScrollBar를 수직으로 배치한 컴포넌트

- Invisibles

  컴포넌트 중 Invisibles 컴포넌트는 Timer 등을 포함한 6개의 컴포넌트로 구성되어 있습니다.

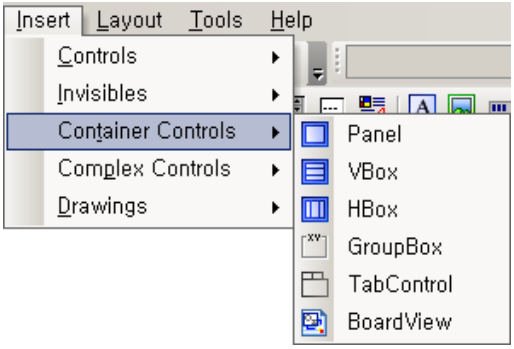


Invisibles 컴포넌트에 대한 간략한 설명은 다음과 같습니다.

Component	Description
Timer	규칙적인 시간 간격으로 특정한 동작을 실행시키고자 할 때 사용하는 컴포넌트
ContextMenu	팝업 메뉴를 나타내는 컴포넌트
MainMenu	화면 상단에 메뉴를 표시하는 컴포넌트
TrayContextMenu	뷰어 종료 시 뷰어를 종료하지 않고 트레이에 숨길 수 있게 하는 컴포넌트
Tooltip	툴팁을 나타내는 컴포넌트
Image List	Context Menu, MainMenu, TreeView 등의 각 아이템 앞에 이미지를 나타낼 수 있게 하는 컴포넌트
TitleBar	타이틀 바에서 최소화, 최대화 기능 외에 아이콘 등을 표시하여 다른 기능을 구현할 수 있게하는 컴포넌트

- Container Controls

  컴포넌트 중 Container Controls 컴포넌트는 Panel 등을 포함한 6개의 컴포넌트로 구성되어 있습니다.

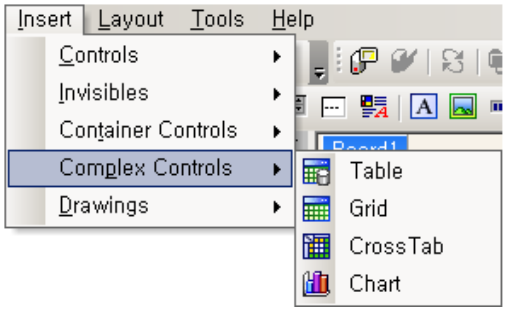


Container Controls 컴포넌트에 대한 간략한 설명은 다음과 같습니다.

Component	Description
Panel	다수의 컴포넌트를 자식 컴포넌트로 가질 수 있는 컴포넌트
VBox	다수의 컴포넌트를 자식 컴포넌트로 가질 수 있으며 수직으로 일정 간격을 유지하며 컴포넌트들을 정렬할 수 있는 컴포넌트
HBox	다수의 컴포넌트를 자식 컴포넌트로 가질 수 있으며 수평으로 일정 간격을 유지하며 컴포넌트들을 정렬할 수 있는 컴포넌트
GroupBox	타이틀과 프레임이 있으며, 다수의 컴포넌트를 자식 컴포넌트로 가질 수 있는 컴포넌트
TabControl	다수의 TabPanel로 구성되며, 다수의 컴포넌트를 자식 컴포넌트로 가질 수 있는 컴포넌트
BoardView	프로젝트에 추가되어있는 Board를 표시하는 컴포넌트

- Complex Controls

  컴포넌트 중 Complex Controls 컴포넌트는 Table 등을 포함한 4개의 컴포넌트로 구성되어 있습니다.



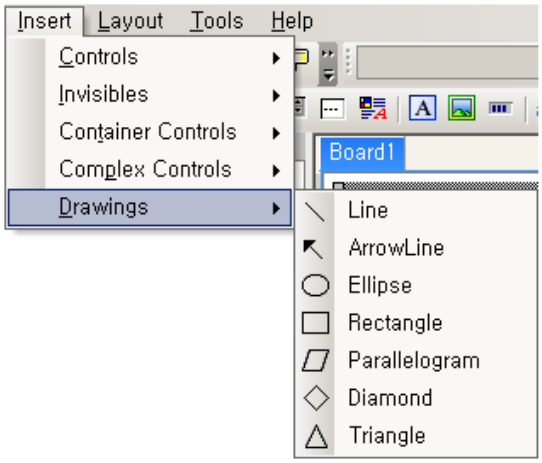
Complex Controls 컴포넌트에 대한 간략한 설명은 다음과 같습니다.

Component	Description
Table	데이터를 테이블 형태로 표시하는 컴포넌트
Grid	헤더와 타이틀이 있고 행과 열로 구성되는 그리드를 나타내는 컴포넌트
CrossTab	데이터를 크로스탭 형태로 표시하고 그룹핑하여 합계 등을 표시하는 컴포넌트

Chart	데이터를 2차원 또는 3차원의 차트 형태로 나타내는 컴포넌트

■ Drawings

컴포넌트 중 Drawings 컴포넌트는 Line 등을 포함한 7개의 컴포넌트로 구성되어 있습니다.

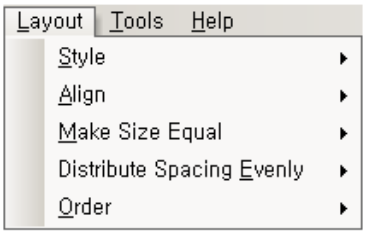


Drawings 컴포넌트에 대한 간략한 설명은 다음과 같습니다.

Component	Description
Line	직선, 점선 등 선을 표시하는 컴포넌트
ArrowLine	화살표가 있는 선을 표시하는 컴포넌트
Ellipse	원형, 타원형 등 원을 표시하는 컴포넌트
Rectangle	사각형을 표시하는 컴포넌트
Parallelogram	평행 사변형을 표시하는 컴포넌트
Diamond	마름모를 표시하는 컴포넌트
Triangle	삼각형을 표시하는 컴포넌트

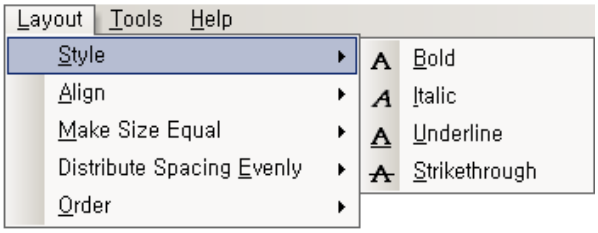
레이아웃(Layout)

메뉴바의 [Layout]를 선택하면 컴포넌트의 레이아웃을 설정할 수 있는 아래와 같은 메뉴가 나타납니다.



- Style

  글자 스타일을 설정할 수 있는 [Style] 메뉴는 4개의 서브 메뉴로 구성되어 있습니다.

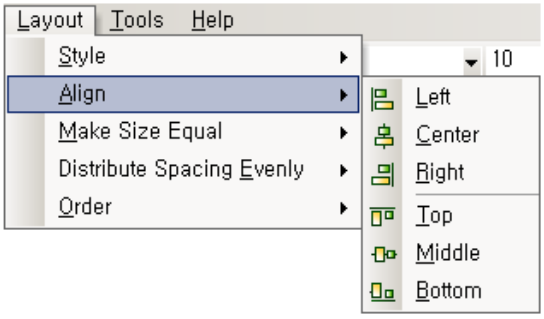


[Style] 메뉴에 대한 간략한 설명은 다음과 같습니다.

Menu	Description
Bold	글자스타일을 굵게 설정합니다.
Italic	글자스타일을 기울림꼴로 설정합니다.
Underline	글자스타일을 밑줄로 설정합니다.
StrikeOut	글자스타일을 취소선으로 설정합니다.

- Align

  다수의 컴포넌트 간의 위치를 정렬할 수 있는 [Align] 메뉴는 4개의 서브 메뉴로 구성되어 있으며, 2개 이상의 컴포넌트를 선택하여야만 활성화됩니다.



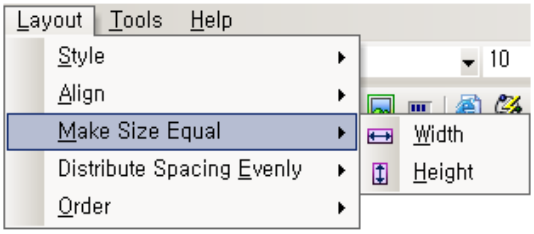
[Align] 메뉴에 대한 간략한 설명은 다음과 같습니다.

Menu	Description

Left	맨 마지막에 선택된 컴포넌트를 기준으로 하여 컴포넌트들을 왼쪽으로 위치 정렬합니다.
Center	맨 마지막에 선택된 컴포넌트를 기준으로 하여 컴포넌트들을 중앙으로 위치 정렬합니다.
Right	맨 마지막에 선택된 컴포넌트를 기준으로 하여 컴포넌트들을 오른쪽으로 위치 정렬합니다.
Top	맨 마지막에 선택된 컴포넌트를 기준으로 하여 컴포넌트들을 위쪽으로 위치 정렬합니다.
Middle	맨 마지막에 선택된 컴포넌트를 기준으로 하여 컴포넌트들을 가운데로 위치 정렬합니다.
Bottom	맨 마지막에 선택된 컴포넌트를 기준으로 하여 컴포넌트들을 아래쪽으로 위치 정렬합니다.

- Make Size Equal

  다수의 컴포넌트의 크기를 동일하게 조절하는 [Make Size Equal] 메뉴는 2개의 서브 메뉴로 구성되어 있으며, 2개 이상의 컴포넌트를 선택하여야만 활성화됩니다.

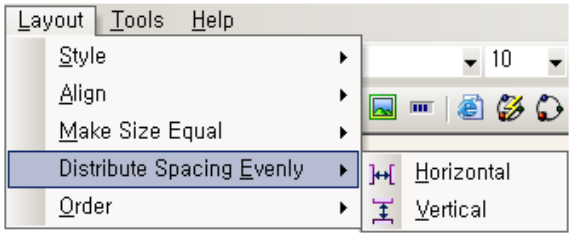


  [Make Size Equal] 메뉴에 대한 간략한 설명은 다음과 같습니다.

Menu	Description
Width	맨 마지막에 선택된 컴포넌트를 기준으로하여 선택된 다수의 컴포넌트들의 너비를 같게 설정합니다.
Height	맨 마지막에 선택된 컴포넌트를 기준으로하여 선택된 다수의 컴포넌트들의 높이를 같게 설정합니다.

- Distribute Spacing Evenly

  다수의 컴포넌트 간의 간격을 동일하게 설정하는 [Distribute Spacing Evenly] 메뉴는 2개의 서브 메뉴로 구성되어 있으며, 3개 이상의 컴포넌트를 선택하여야만 활성화됩니다.

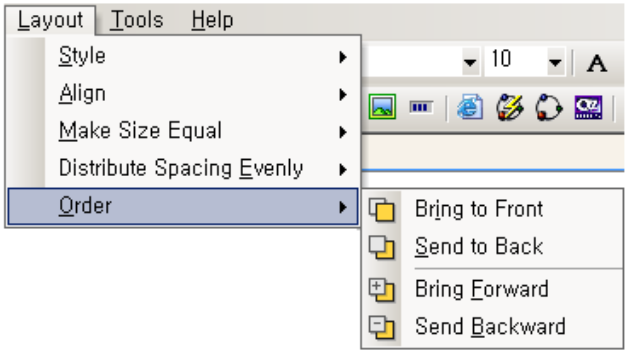


[Distribute Spacing Evenly] 메뉴에 대한 간략한 설명은 다음과 같습니다.

Menu	Description
Horizontal	컴포넌트간 수평 간격을 동일하게 설정합니다.
Vertical	컴포넌트간 수직 간격을 동일하게 설정합니다.

■ Order

컴포넌트 고유한 순서 즉 컴포넌트의 인덱스를 설정하는 [Order] 메뉴는 4개의 서브 메뉴로 구성되어 있습니다.

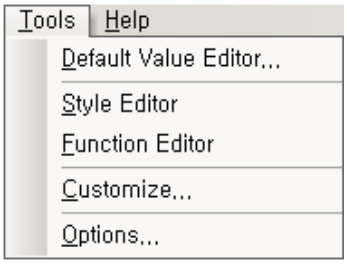


[Order] 메뉴에 대한 간략한 설명은 다음과 같습니다.

Menu	Description
Bring to Front	컴포넌트의 인덱스를 맨 위 단계 즉, 가장 큰값으로 변경합니다.
Send to Back	컴포넌트의 인덱스를 맨 아래 단계 즉, 가장 작은값으로 변경합니다.
Bring Up	컴포넌트의 인덱스를 한 단계 위 즉, 현재 인덱스보다 '1' 큰 값으로 변경합니다.
Send Down	컴포넌트의 인덱스를 한 단계 아래 즉, 현재 인덱스보다 '1' 작은 값으로 변경합니다.

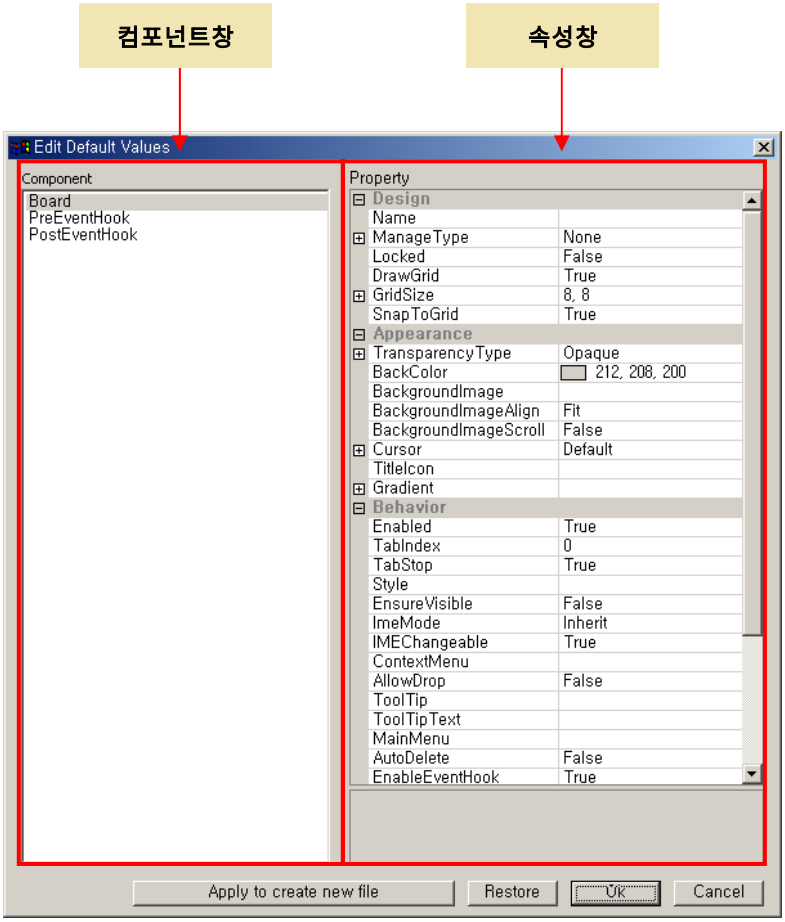
도구(Tools)

메뉴바의 [Tools] 메뉴를 선택하면 여러 가지 옵션을 설정할 수 있는 아래와 같은 메뉴가 나타납니다.



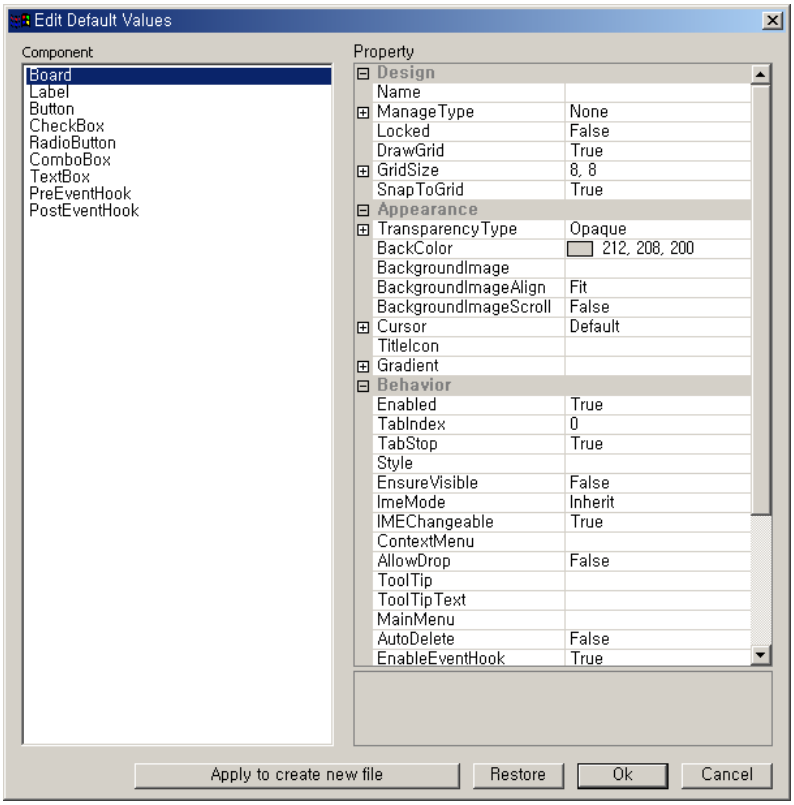
- Default Value Editor

  각각의 컴포넌트에 대해 속성의 기본 값을 설정할 수 있는 '기본 값 편집 다이얼로그'를 실행합니다. 아래 그림은 프로젝트에 Board만 추가된 상태에서 [Default Value Editor] 메뉴를 클릭했을 때 모습입니다.



'기본 값 편집 다이얼로그'는 프로젝트에 추가된 컴포넌트 타입을 나타내는 컴포넌트창과 컴포넌트 타입의 모든 속성이 표시되는 속성창으로 구성됩니다.

예를 들어, 프로젝트에 Board, Button, Table, TextBox가 추가되어 있다고 가정할 경우 [Default Value Editor] 메뉴를 클릭하면 '기본 값 편집 다이얼로그'는 아래와 같이 표시됩니다.



컴포넌트창에서 컴포넌트 타입을 선택하고 속성창에서 기본 값을 변경한 후 [Ok] 버튼을 클릭하여 해당 컴포넌트 속성의 기본 값을 설정합니다. 기본 값을 변경한 후 새로 추가되는 컴포넌트는 변경된 기본 값이 적용되어 추가됩니다.

[Ok] 버튼은 현재 프로젝트에서만 적용하도록 설정하는 버튼이며, [Apply to create new file] 버튼은 현재 프로젝트 뿐만 아니라 새로운 프로젝트에서도 적용하도록 설정하는 버튼입니다.

OZ Application Designer 초기에 설정된 기본 값으로 변경하려면 [Restore] 버튼을 클릭한 후 적용할 상황에 따라 [Ok] 또는 [Apply to create new file] 버튼을 클릭합니다.

- Style Editor

  OZS를 만드는 툴인 OZ Style Editor를 실행합니다.

  OZS와 OZ Style Editor의 사용법은 Ⅸ.장을 참조하십시오.

■ Function Editor

OZF를 만드는 툴인 OZ Function Editor를 실행합니다.

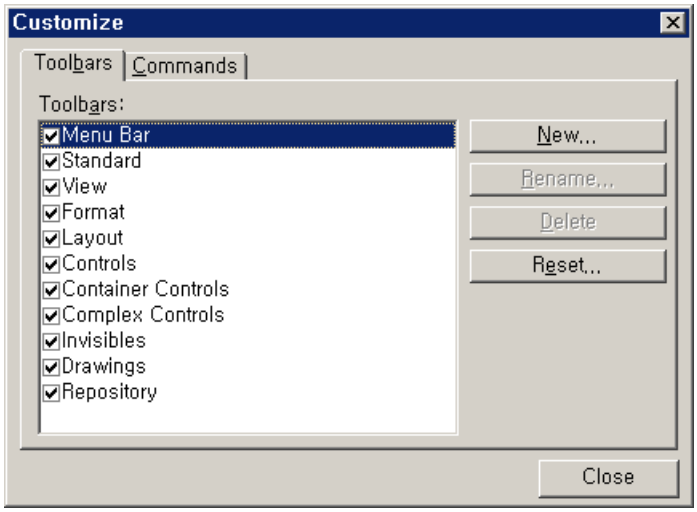
OZF와 OZ Function Editor의 사용법은 Ⅷ.장을 참조하십시오.

■ Customize

툴바의 표시 여부를 사용자가 임의로 설정할 수 있도록 합니다.

- Toolbars

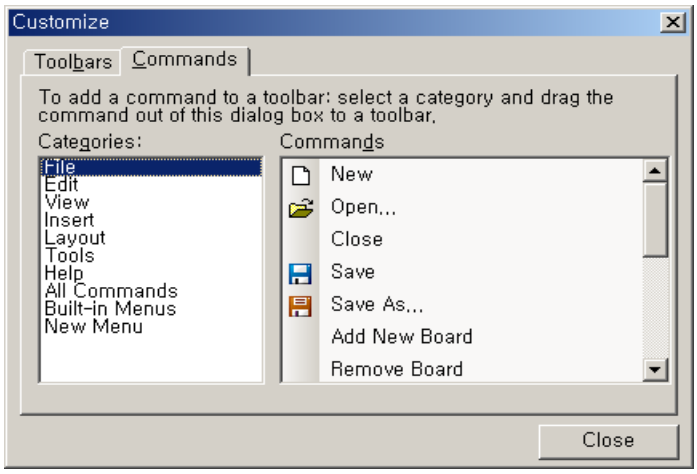
툴바 그룹의 표시 여부를 설정하거나 사용자가 임의로 툴바 그룹을 만들수 있습니다.



※ 참고사항 : 메뉴바는 OZ Application Designer에 항상 표시되어야 하는 것으로, 사용자가 표시 여부를 설정할 수 없습니다.

- Commands

원하는 기능을 툴바 그룹과 상관없이 별도로 추가하거나 삭제합니다.

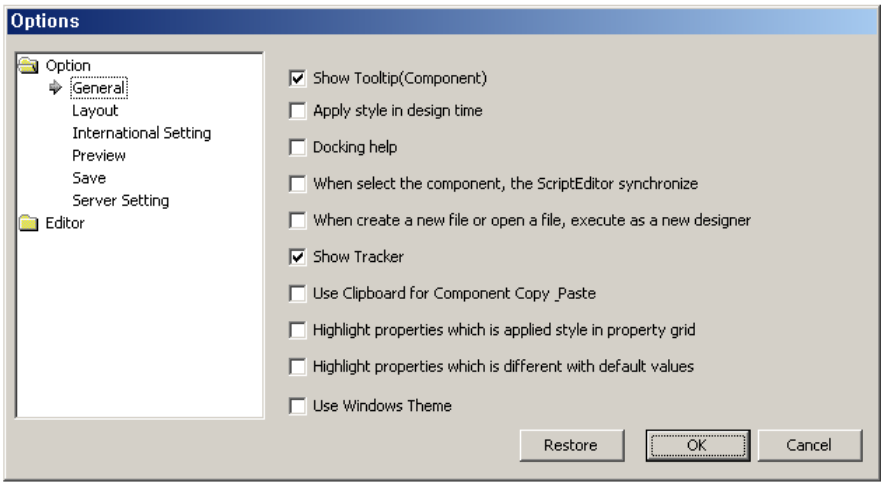


원하는 기능을 추가하려면 "Categories"에서 그룹을 선택한 후 "Commands" 리스트에서 원하는 명령을 마우스로 선택한 후 툴바 또는 메뉴바로 드래그 & 드롭하면 추가됩니다.

특정 기능을 삭제할 경우에는 툴바 또는 메뉴바에서 아이콘이나 메뉴를 선택한 후 보고서창으로 드래그 & 드롭하면 삭제됩니다.

■ Options

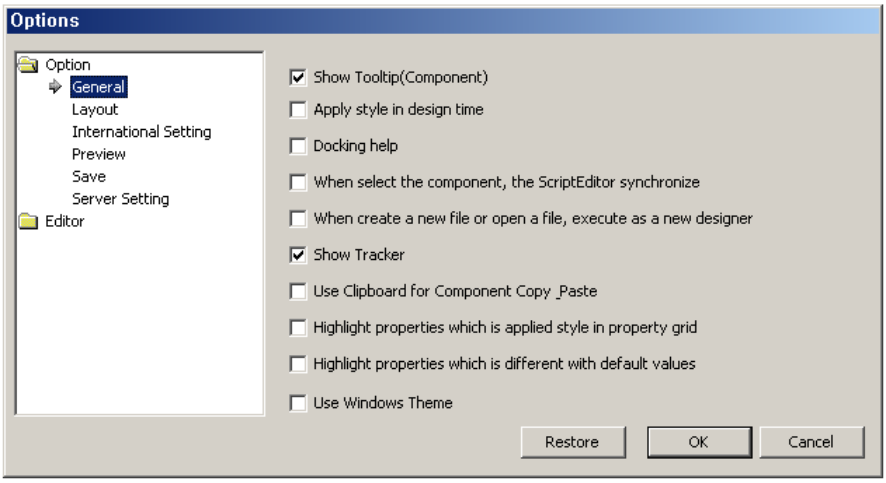
툴바, 트리창, 속성창, 이벤트창, 스크립트창 등 OZ Application Designer의 레이아웃과 메뉴바의 언어 설정, 편집기의 글꼴 등의 옵션을 설정합니다.



변경한 옵션을 저장 및 적용하려면 [Ok] 버튼을 OZ Application Designer 초기에 설정된 기본 옵션으로 적용하려면 [Restore] 버튼을 클릭합니다.

- Option
  - ➢ General

    디자인 시 컴포넌트 정보의 툴팁 표시 여부, 스타일 적용 여부, 도움말창 고정 여부 등을 설정합니다.



- Show Tooltip(Component)

  OZ Application Designer의 디자인 화면에서 Board에 추가된 컴포넌트의 타입과 이름을 툴팁으로 표시할지 여부를 설정합니다.

- Apply style in design time

  프로젝트에 OZ Style 파일을 추가하였을 경우 컴포넌트에 설정된 OZS 파일의 스타일에 설정된 속성 값을 디자인시에 해당 컴포넌트의 속성 값에 적용할지 여부를 설정합니다.

- Docking help

  도움말창을 OZ Application Designer 화면에 고정시켜 표시할지 여부를 설정합니다.

- When select the component, the ScriptEditor synchronize

  컴포넌트를 선택하면 스크립트 편집창에도 해당 컴포넌트가 선택되게하여, 컴포넌트 트리창에 선택되어 있는 컴포넌트와 스크립트 편집창에 선택되어 있는 컴포넌트를 동기화시킬지 여부를 설정합니다.

- When create a new file or open a file, execute as a new designer

  새로 파일을 만들거나 리파지토리에서 파일을 열었을 때 OZ Application Designer를 새로 실행하여 해당 파일을 열지 여부를 설정합니다.

- Show Tracker

  선택된 컴포넌트의 테두리에 트랙커를 표시할지 여부를 설정합니다.

- Use Clipboard for Component Copy_Paste
  실행 중인 OZ Application Designer 간에 컴포넌트 또는 스크립트를 복사하여 붙여넣기하게 할지 여부를 설정합니다.

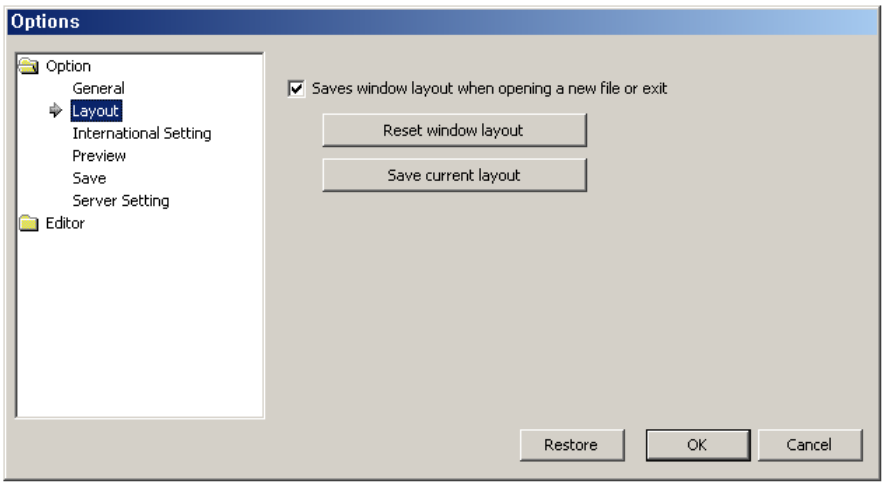
- Highlight properties which is applied style in property grid
  오즈 스타일(OZS 파일)이 적용된 속성일 경우 속성명의 글꼴을 굵게 표시할지 여부를 설정합니다. 해당 옵션을 체크하면 컴포넌트에 연결된 OZS 파일에 지정된 속성이 있을 경우 해당 속성명을 굵게 표시합니다.

- Highlight properties which is different with default values
  속성 값이 기본 값과 다를 경우 속성 값의 글꼴을 굵게 표시할지 여부를 설정합니다.

- Use Window Theme
  CheckBox, CheckListBox, ComboBox, DateTimePicker, ListBox, ListView, MonthCalendar, NumericUpDown, ProgressBar, RadioButton, TabControl, TextBox, TreeView 컴포넌트에 윈도우 XP 테마를 사용할 지 여부를 설정합니다.

  ※ 참고사항
  - 해당 기능을 사용하려면 사용자 PC의 OS가 윈도우 XP 이상이어야 합니다.
  - TabControl의 경우 TabAligment 속성 값이 "Top"인 경우에만 윈도우 XP 테마 기능을 사용할 수 있으며, TabPanel의 TabTitleBackColor와 윈도우 테마가 동시 사용될 경우 TabTitleBackColor는 적용되지 않습니다.

➢ Layout
  OZ Application Designer의 레이아웃을 설정합니다.

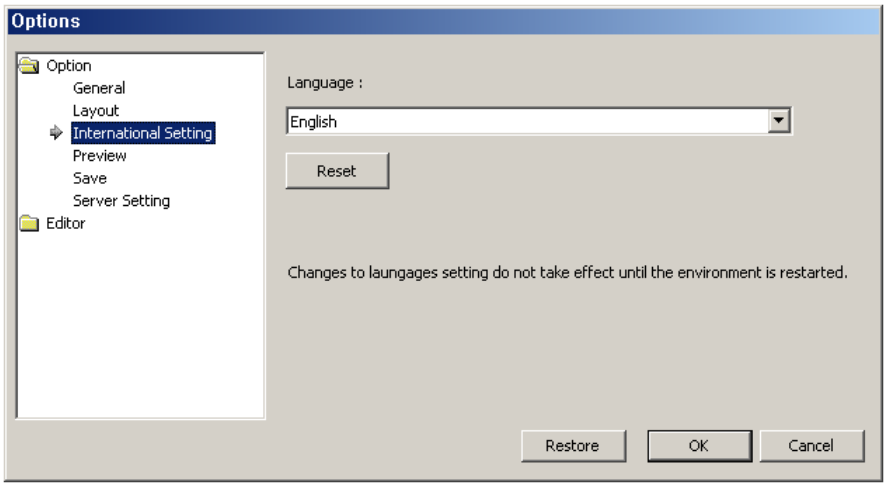


- Save layout when exit

  OZ Application Designer가 종료될 때 툴바, 트리창, 속성창, 이벤트창, 스크립트창의 표시 여부, 크기, 위치 등 OZ Application Designer 레이아웃의 저장 여부를 설정합니다. OZ Application Designer의 레이아웃을 기본 값으로 초기화 할 경우에는 [Reset window layout] 버튼을 클릭하고 현재 레이아웃을 저장할 경우에는 [Save current layout] 버튼을 클릭합니다.

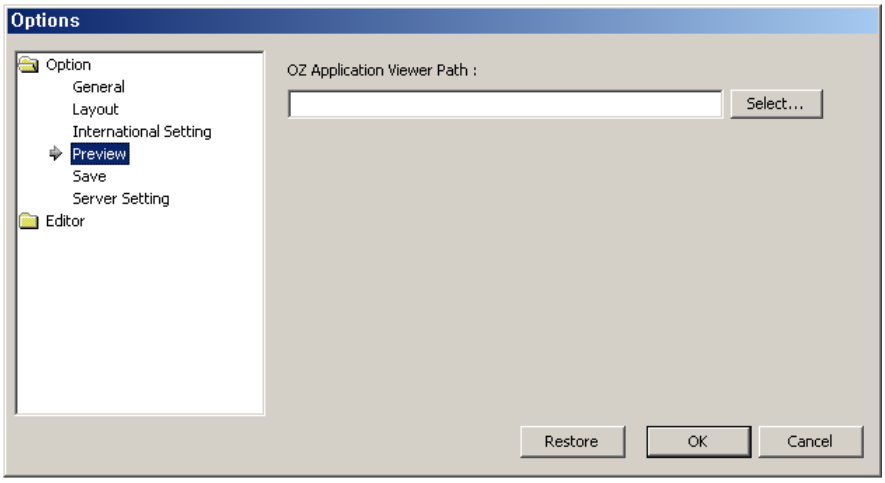
➢ International Setting

메뉴바나 툴팁에 표시할 언어를 설정합니다.



➢ Preview

사용자가 미리보기 할 뷰어를 설정합니다.



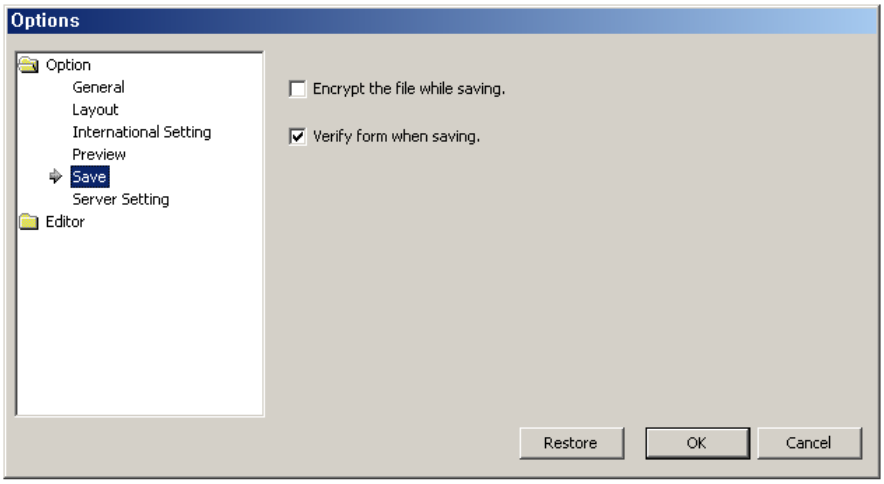
- OZ Application Viewer Path

  오즈 애플리케이션 뷰어의 실행 파일을 설정합니다.

  미리보기 시에 사용되는 오즈 애플리케이션 뷰어를 오즈 애플리케이션 디자이너와 함께 제공되는 뷰어가 아니라 다른 버전의 뷰어를 사용할 경우에는 [선택] 버튼을 클릭하여 뷰어 실행 파일 경로를 설정합니다.

➢ Save

프로젝트 파일을 저장 옵션을 설정합니다.



- Encrypt the file while saving

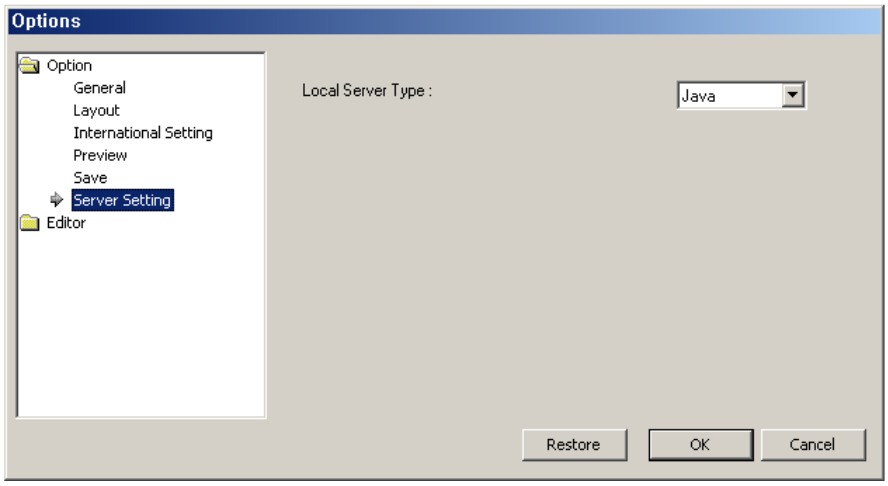
  프로젝트 파일을 암호화하여 저장할지 여부를 설정합니다.

  오즈 애플리케이션의 경우 해당 옵션을 체크하고 파일을 저장하면 저장된 OZA 파일을 일반 편집 프로그램으로 열어 파일의 내용을 볼 수 없도록 암호화하는 기능과 특정 사이트에서만 파일을 열 수 있도록 파일을 암호화하고

복호화할 수 있는 기능을 제공합니다.
단, 특정 사이트에서만 파일을 열 수 있도록 암호화하는 기능에 대한 자세한 사항은 오즈 기술 지원 센터로 문의하시기 바랍니다.

- Verify form when saving
  프로젝트 파일을 저장할 때 파일의 검증 여부를 설정합니다. 이 옵션을 체크하면 프로젝트 파일을 저장할 때 다음과 같은 경우에는 경고 메시지를 표시합니다.
  ① Document에 등록된 ODI로 ODIKey, DataSet, Field, ValueField 등 데이터 관련 속성을 설정했지만 프로젝트 파일을 검증하는 시점에 해당 ODI가 Document에 추가되지 않았을 경우
  ② 컴포넌트의 속성 중 Style에 설정된 OZS 파일이 프로젝트 파일을 검증하는 시점에 Document에 추가되지 않았을 경우
  ③ 컴포넌트의 속성 중 ContextMenu, MainMenu, ToolTip, ImageList에 설정된 ContextMenu, MainMenu, ToolTip, ImageList 컴포넌트가 프로젝트 파일을 검증하는 시점에 Document에 추가되지 않았을 경우
  ④ Navigator의 속성 중 LinkComponentName에 연결된 컴포넌트가 프로젝트 파일을 검증하는 시점에 Document에 추가되지 않았을 경우

➢ Server Setting
OZ Application Designer를 이용하여 폼 디자인 시 구동할 서버의 타입을 설정합니다.



- Local Server Type
  로컬 서버 타입을 자바와 닷넷 중 하나로 설정합니다.

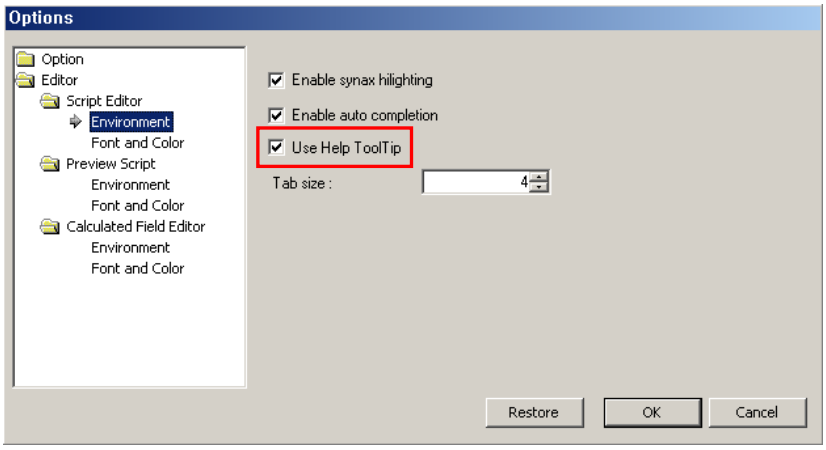
※ 참고사항

만약 선택한 로컬 서버 타입이 설치되어 있지 않은 경우에는 설치되어 있는 서버 타입으로 서버가 기동됩니다.

- Editor

➢ Environment : Script Editor / Preview Script / Calculated Field Editor

스크립트창, 스크립트 미리보기창, 수식 필드 편집창에서 스크립트를 표시하는 텍스트에 대해 구문 강조 여부, 구문 자동 완성 여부, 함수 또는 이벤트의 프로토 타입을 툴팁으로 표시할 지 여부, 탭 크기를 설정합니다.



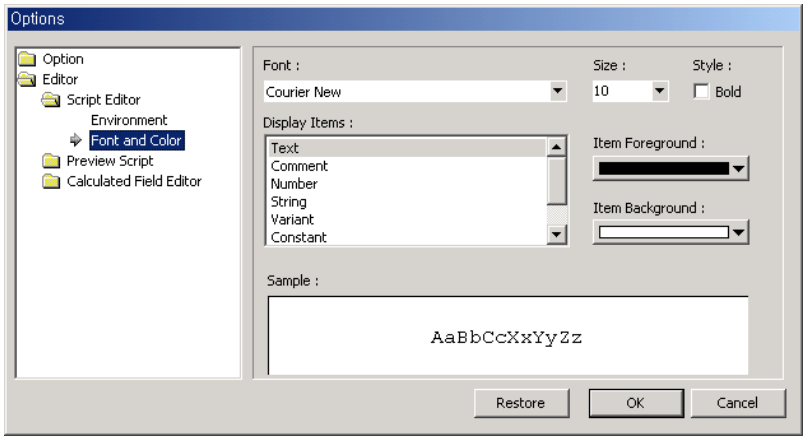
※ 참고사항

- 스크립트창 : 스크립트창은 OZ Application Designer의 화면 기본적으로 표시되는 창으로 각 컴포넌트의 각 이벤트에 대한 스크립트를 코딩하는 창입니다. 자세한 설명은 본 매뉴얼의 Ⅲ.장 스크립트창 부분을 참조하십시오.
- 스크립트 미리보기창 : 프로젝트에 코딩된 전체 스크립트 내용을 표시하는 창으로 메뉴바의 [Edit] 메뉴의 [Preview Script] 메뉴를 클릭하면 표시되며, 자세한 설명은 본 매뉴얼의 Ⅲ.장 메뉴바의 [Edit] 메뉴 부분을 참조하십시오.
- 수식 필드 편집창 : 데이터셋에 수식 필드를 추가하고 수식 필드에 적용할 함수를 코딩할 때 표시되는 편집창으로 추가된 데이터셋을 마우스 오른쪽 버튼으로 선택하여 나타나는 팝업 메뉴에서 [Insert New CalculatedField] 메뉴를 클릭하면 표시되는 창입니다. 자세한 설명은 본 매뉴얼의 Ⅵ.장 수식 필드 사용 부분을 참조하십시오.

➢ Font and Color : Script Editor / Preview Script / Calculated Field Editor

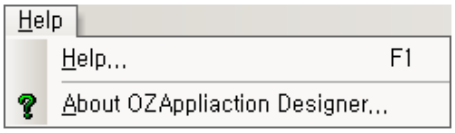
스크립트창, 스크립트 미리보기창, 수식 필드 편집창에서 스크립트를 표시하는 텍

스트의 글꼴, 글꼴 크기, 전경색, 배경색을 설정합니다.



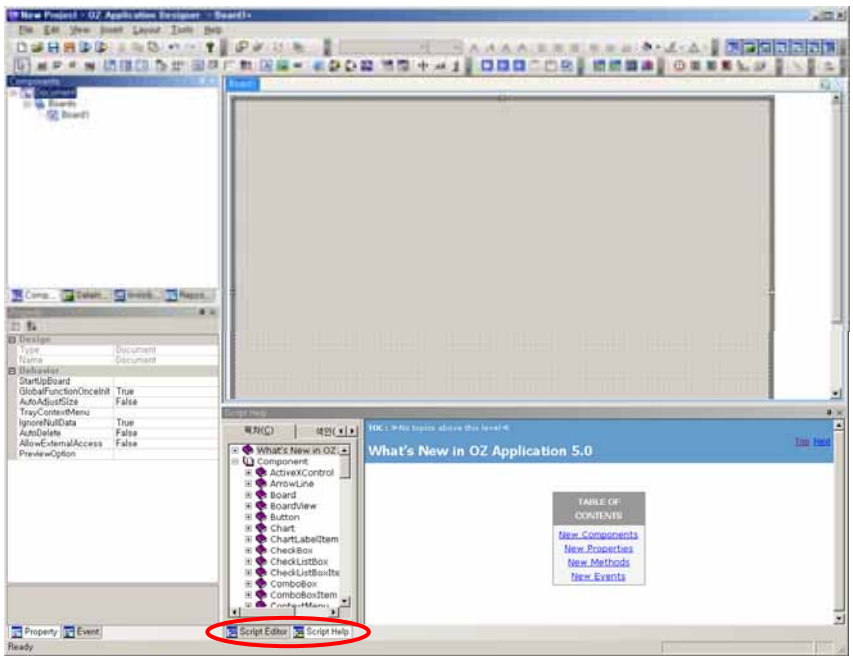
도움말(Help)

메뉴바의 [Help] 메뉴를 선택하면 아래와 같은 메뉴가 나타납니다.



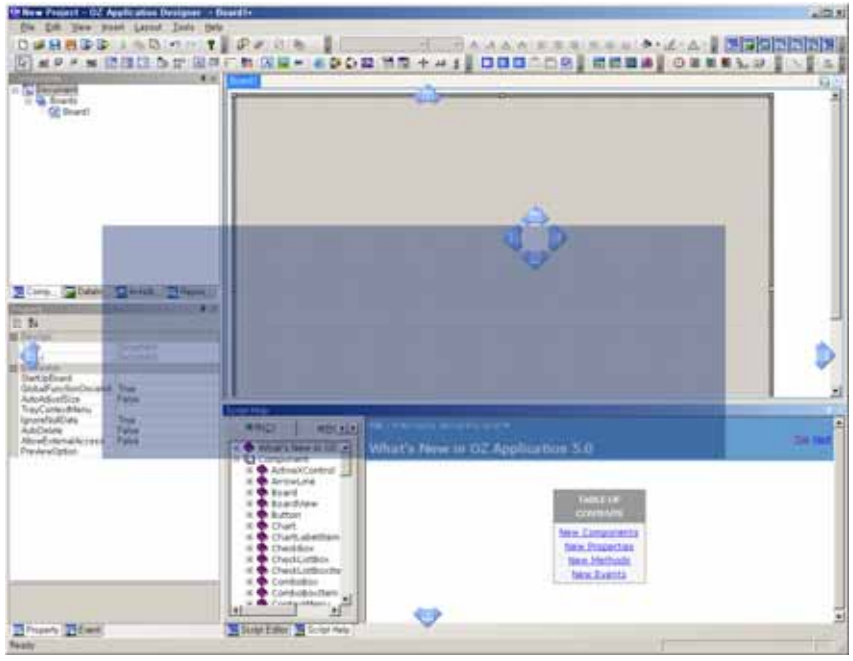
- Help

  OZ Application Designer의 컴포넌트의 속성, 메소드, General Script, 상수 등에 대한 설명과 예제를 보여주는 도움말창이 스크립트창 표시 영역에 탭 형태로 표시됩니다.

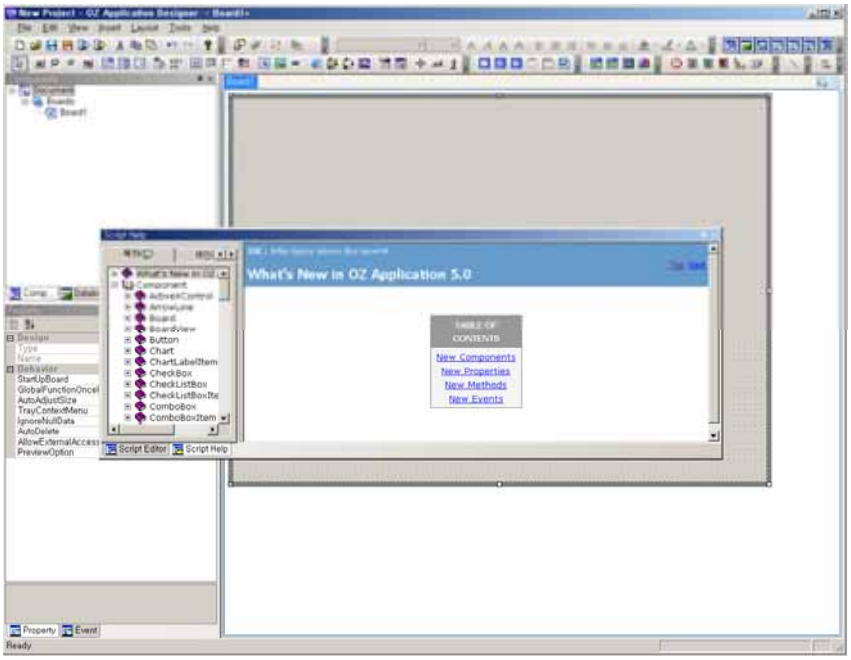


도움말창은 스크립트창의 표시 영역에 고정시킬 수도 있지만 따로 분리할 수도 있습니다. 도움말창을 나타내는 탭을 마우스로 클릭하여 화면 가운데로 Drag & Drop하면 도움말창이 분리되는 것을 확인할 수 있습니다.

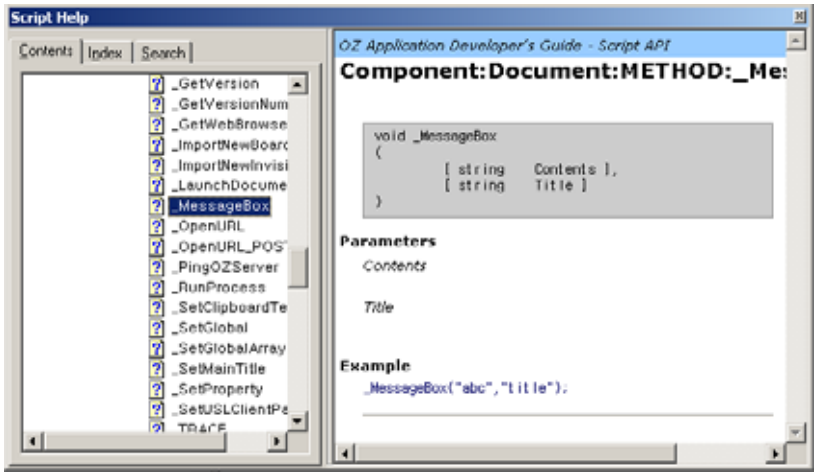
- 아래 그림은 도움말창을 나타내는 탭(Script Help라고 써진 부분)을 마우스로 클릭하여 화면 가운데로 Drag & Drop하고 있는 상태의 모습입니다.



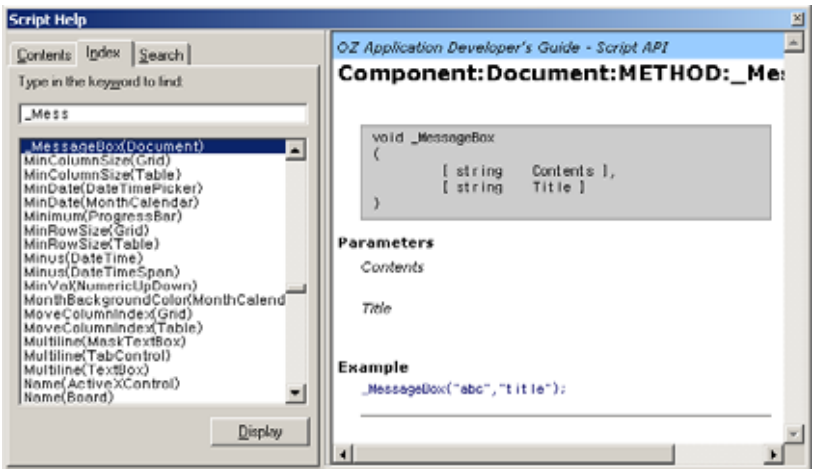
- 아래 그림은 도움말창이 분리된 모습입니다.



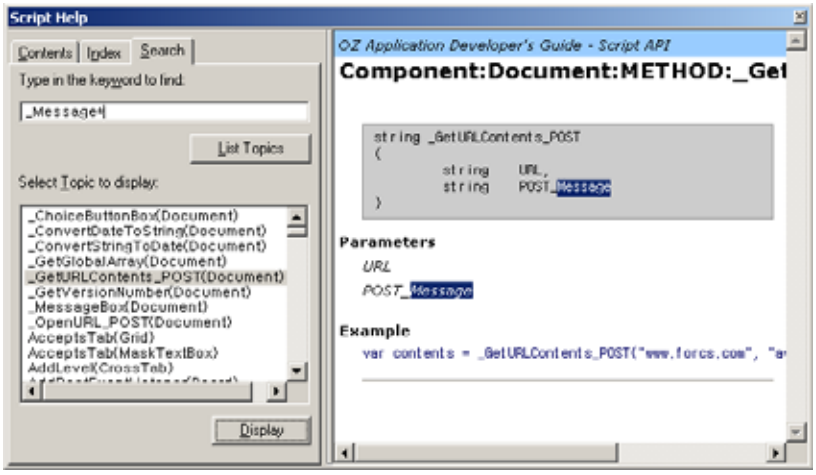
원하는 아이템을 좌측 윈도우에서 마우스로 클릭하거나 더블 클릭하면 우측 윈도우에 해당 아이템의 도움말이 표시됩니다.



[Index] 탭을 클릭하여 알파벳순으로 정렬된 목록에서 선택하여 볼 수 있습니다. 텍스트 박스에 검색할 텍스트를 입력하면 해당 텍스트에 일치하는 아이템이 선택되어 표시됩니다.



[Search] 탭을 클릭하여 검색한 문자열이 포함되어 있는 모든 내용을 검색할 수 있습니다. 검색할 문자열을 텍스트 박스에 입력한 후 [List Topics] 버튼을 클릭하거나 키보드의 [Enter] 키를 입력하면 검색한 문자열이 포함되어 있는 항목이 리스트 형태로 표시됩니다. 검색된 항목을 더블 클릭하거나 [Display] 버튼을 클릭하면 해당 내용이 표시됩니다.



- About OZ Application

  OZ Application Designer의 버전 정보와 사용자 PC 정보를 표시합니다.

  OZ Application Designer의 버전 정보 창에서 [Information] 버튼을 클릭하면 'PC Information 다이얼로그'가 표시되고 사용자 PC의 IP Address, Mac Address, HDD Serial을 확인할 수 있습니다. [Copy] 버튼을 클릭하여 사용자 PC 정보를 복사한 후 일반 편집 프로그램에 붙여넣기 할 수 있습니다.

툴바

툴바는 OZ Application Designer의 메뉴바의 기능을 보다 쉽고 빠르게 사용할 수 있도록 메뉴바 바로 아래에 아이콘 형태로 제공되며, 표준 툴바, 포맷 툴바, 컴포넌트 툴바, 레이아웃 툴바, 뷰 툴바, 리파지토리 툴바, ActiveX 컨트롤 툴바로 구성됩니다.

표준(Standard) 툴바

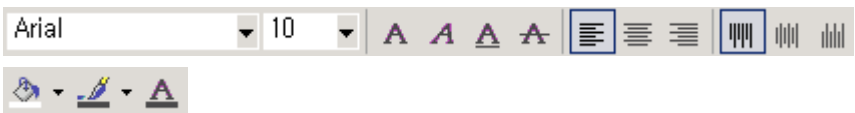
파일 관련 기능과 잘라내기, 복사 등 일반적인 기능을 실행합니다.



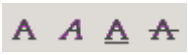
Icon	Description
	프로젝트 새로 만들기, 프로젝트 열기
	저장하기, 다른 이름으로 저장하기
	미리보기, 디버깅하기
	잘라내기, 복사하기, 붙여넣기
	실행 취소, 다시 실행
	OZ Application Designer 버전 보기

포맷(Format) 툴바

글꼴, 글자 크기, 글자 속성, 정렬 방식, 색 등 텍스트 포맷을 설정합니다.



Icon	Description
	글자 글꼴 선택

	글자 크기 선택
	굵게, 기울임꼴, 밑줄, 취소선으로 글자 속성 설정
	좌측, 중앙, 우측으로 글자를 수평 정렬
	상단, 가운데, 하단으로 글자를 수직 정렬
	배경색, 선색, 글자색 선택

컴포넌트(Component) 툴바

Controls, Invisibles, Container Controls, Complex Controls, Drawings 등 컴포넌트를 선택합니다.



Icon	Description
	Button, CheckBox, RadioButton, ReportButton 선택
	ComboBox, ListBox, CheckListBox 선택
	TreeView, ListView 선택
	TextBox, NumericUpDown, MaskTextBox, RichText Box 선택
	Label, Picture, ProgressBar 선택
	HTMLView, ActiveXControl, CustomControl, Report Control 선택
	DateTimePicker, MonthCalendar 선택
	Splitter, HorizontalNavigator, VerticalNavigator 선택

	Timer, ContextMenu, MainMenu, TrayContextMenu, Tooltip, ImageList, TitleBar 선택
	Panel, VBox, HBox, GroupBox, TabControl, BoardView 선택
	Table, Grid, CrossTab, Chart 선택
	Line, ArrowLine, Ellipse, Rectangle, Parallelogram, Diamond, Triangle 선택

레이아웃(Layout) 툴바

컴포넌트의 위치 정렬, 크기, 사이즈, 인덱스 등 컴포넌트간 레이아웃을 설정합니다.



Icon	Description
	컴포넌트들을 왼쪽, 오른쪽, 위쪽, 아래쪽 방향으로 위치 정렬
	컴포넌트들의 너비, 높이를 동일하게 설정
	컴포넌트간 수평, 수직 간격을 동일하게 설정
	컴포넌트의 인덱스를 맨 위, 맨 아래, 한 단계 위, 한 단계 아래로 변경

뷰(View) 툴바

트리창, 속성창 등 윈도우의 표시 여부를 설정합니다.



Icon	Description
	컴포넌트 트리창 표시 여부 설정

	Invisibles 트리창 표시 여부 설정
	데이터 트리창 표시 여부 설정
	리파지토리창 표시 여부 설정
	속성창 표시 여부 설정
	이벤트창 표시 여부 설정
	스크립트창 표시 여부 설정

리파지토리(Repository) 툴바

서버 리파지토리와 관련된 기능을 실행합니다.



Icon	Description
	리파지토리 열기, 리파지토리 닫기
	새로고침
	작업 폴더 설정
	카테고리 생성, 삭제
	파일 업로드, 파일 다운로드, 체크아웃, 체크인, 체크아웃 취소

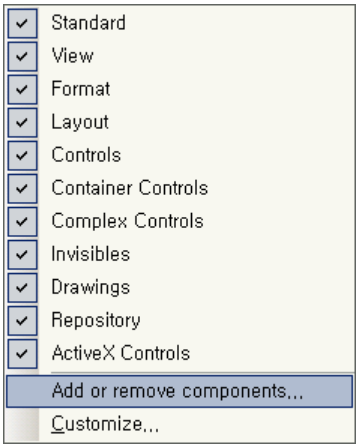
ActiveX 컨트롤(ActiveX Controls) 툴바

사용자가 추가한 ActiveX 컨트롤을 선택합니다.

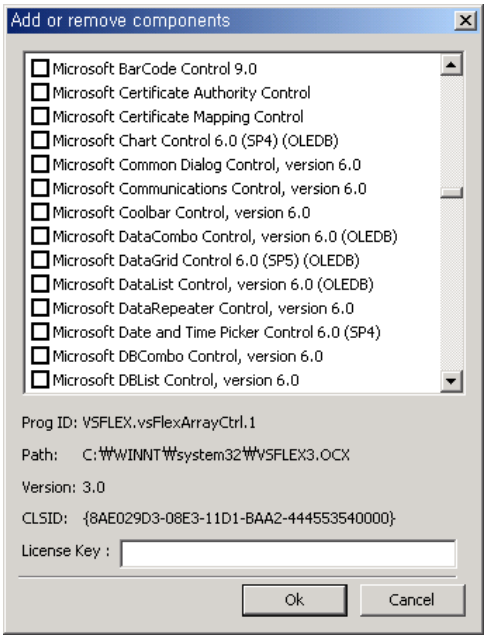
ActiveX 컨트롤 툴바는 OZ Application Designer를 처음 실행할 시에는 표시되지 않으며, 사용자가 ActiveX 컨트롤을 추가하여야만 표시됩니다.

ActiveX 컨트롤 툴바를 표시하는 방법은 다음과 같습니다.

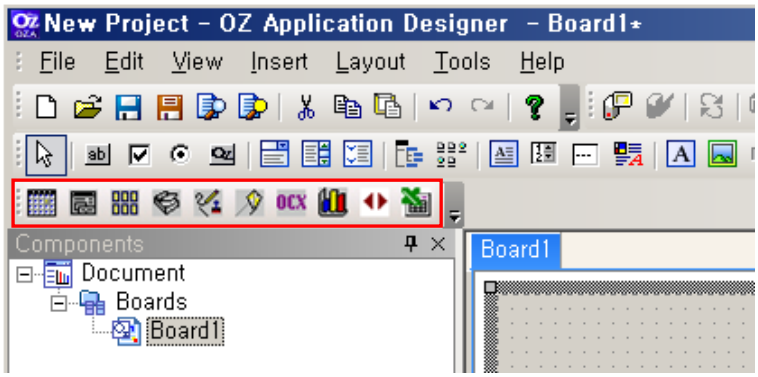
툴바를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴 중 ActiveX 컨트롤을 툴바에 추가하는 메뉴인 [Add or remove components] 메뉴를 클릭합니다.



[Add or remove components] 메뉴를 클릭하면 아래 그림과 같이 사용자 PC에 설치되어있는 ActiveX 컨트롤을 나타낸 다이얼로그가 표시됩니다.



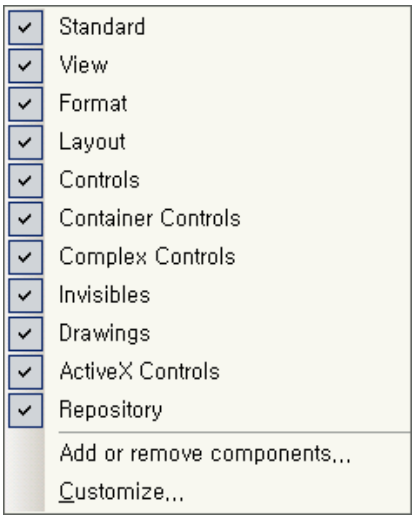
툴바에 추가할 ActiveX 컨트롤을 체크한 후 [Ok] 버튼을 클릭하면 ActiveX 컨트롤 툴바가 OZ Applicaton Designer에 추가되고, 체크한 ActiveX 컨트롤이 ActiveX 툴바에 표시됨을 확인할 수 있습니다.



ActiveX 컨트롤 툴바는 사용자가 추가한 ActiveX 컨트롤에 따라 다르게 표시되며, 최대 10개까지의 ActiveX 컨트롤을 추가할 수 있습니다.

※ 참고사항 : 툴바 표시 설정 및 사용자 지정

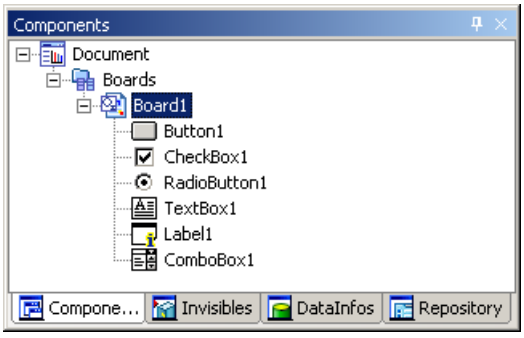
툴바를 마우스 오른쪽 버튼으로 클릭하면 아래 그림과 같이 툴바의 표시 여부를 나타내는 팝업 메뉴가 나타납니다.



표시하지 않을 툴바는 체크를 해제하고, 표시할 툴바는 체크합니다. [Customize] 메뉴를 클릭하여 사용자가 임의로 설정할 수 있습니다.

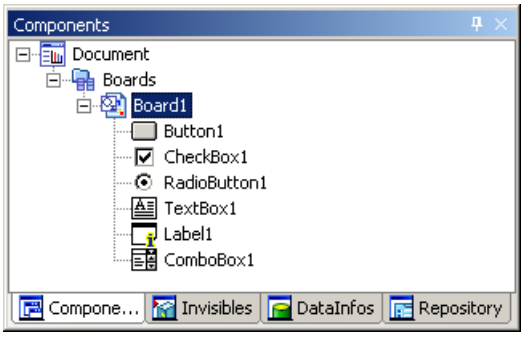
트리창

트리창은 컴포넌트, ODI, Invisibles, 리파지토리 등을 표시하는 각각의 창이 탭 형태로 제공되며, 컴포넌트 트리창, 데이터 트리창, Invisibles 트리창, 리파지토리창으로 구성됩니다.



컴포넌트 트리창

프로젝트에 추가된 모든 컴포넌트, 컴포넌트의 인덱스, 컴포넌트간의 부모-자식 관계를 나타냅니다. 스크립트가 입력되어 있는 컴포넌트는 컴포넌트 이름이 굵은 글꼴로 표시되어 스크립트 작성에 도움을 줍니다.



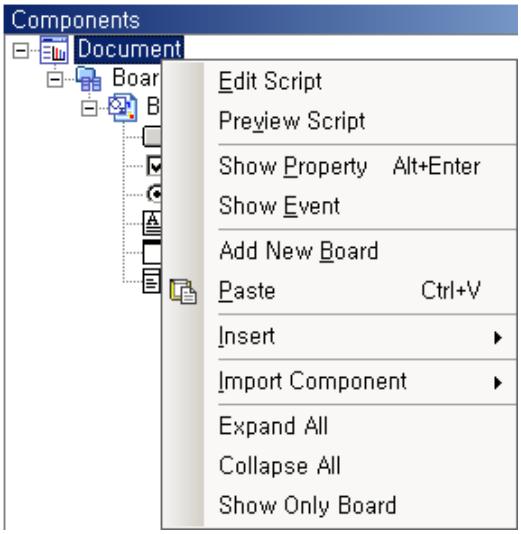
위 그림은 한 개의 Board가 있고, Board의 자식 컴포넌트로 Button, CheckBox, RadioButton, TextBox, Label, ComboBox가 한 개씩 있는 프로젝트를 나타낸 것입니다. 또한 컴포넌트의 인덱스는 Button1, CheckBox1, RadioButton1, TextBox1, Label1, ComboBox1 순서이고, Button1과 TextBox1에 스크립트가 작성되어 있으며 Lable1의 Manage Type이 Auto나 Manual 타입임을 나타냅니다.

※ 참고사항 : 컴포넌트의 Manage Type None이 아닌 Auto나 Manual일 경우 컴포넌트 트리창에서 컴포넌트 그림 옆에 영문 "i"()가 표시됩니다.

컴포넌트 트리창에서는 각각의 아이템 별로 팝업 메뉴가 제공됩니다.

※ 참고사항 : 중복되는 메뉴에 대해서는 한번만 설명하였사오니, 선택한 컴포넌트에 맞게 적절히 적용하시기 바랍니다.

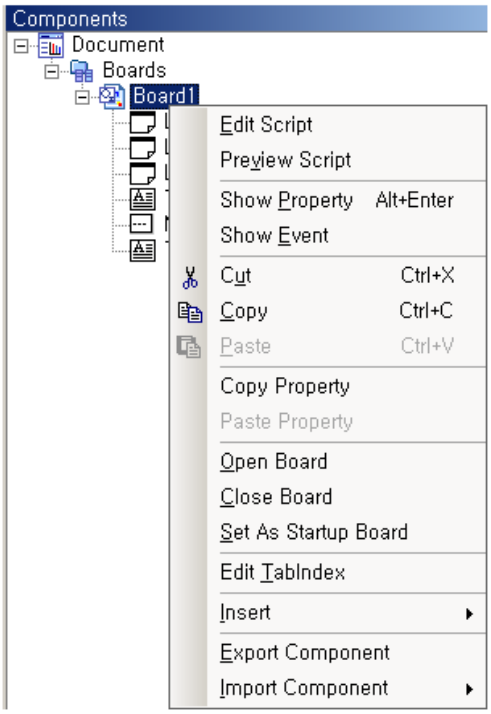
■ Document 팝업 메뉴



Edit Script	Document의 스크립트 편집창이 활성화되고, 마우스 커서가 스크립트 편집창에 자동으로 위치하여, 스크립트를 입력할 수 있습니다.
Preview Script	Document와 Document에 추가된 자식 컴포넌트에 입력되어 있는 스크립트를 나타내는 스크립트 미리보기창이 화면에 나타납니다
Show Property	Document의 속성창이 활성화되고, Document의 속성을 표시합니다.
Show Event	Document의 이벤트창이 활성화되고, Document의 이벤트를 표시합니다.
Add New Board	Document에 새 Board를가 추가합니다.
Paste	복사된 Board를 붙여넣기합니다.
Insert	프로젝트에 Invisible 컴포넌트, ODI, OZS, OZF를 추가합니다.
Import Component	프로젝트에 OZC 파일을 추가합니다.

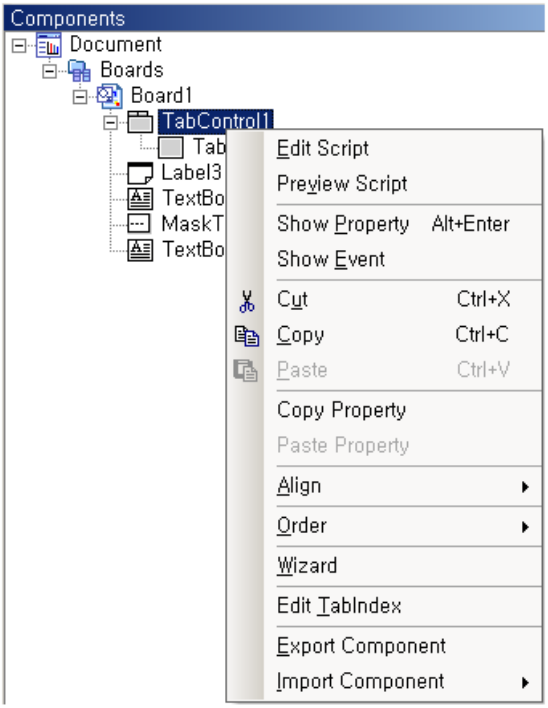
Expand All	컴포넌트 트리창에 추가되어있는 모든 컴포넌트를 펼쳐서 표시합니다.
Collapse All	컴포넌트 트리창에 추가되어있는 모든 컴포넌트를 접어서 표시합니다.
Show Only Board	컴포넌트 트리창에 추가되어있는 컴포넌트를 보드 단위로 표시하고 다른 컴포넌트는 접어서 표시합니다.

■ Board 팝업 메뉴



Copy Property	선택된 컴포넌트의 속성을 복사합니다.
Paste Property	복사된 속성을 선택된 컴포넌트에 붙여넣기합니다. 동일한 컴포넌트 간에는 모든 속성에 대하여 붙여넣기가 가능하고, 서로 다른 컴포넌트 간에는 붙여넣을 컴포넌트의 속성 중 복사한 컴포넌트의 속성과 동일한 항목에 대해서만 복사 가능합니다.
Open Board	디자인 창에서 해당 Board를 보여줍니다.
Close Board	디자인 창에서 해당 Board를 숨깁니다.
Set As Startup Board	선택한 Board를 시작 Board로 설정합니다.
Edit TabIndex	해당 Board의 자식 컴포넌트들의 탭순서를 편집합니다.

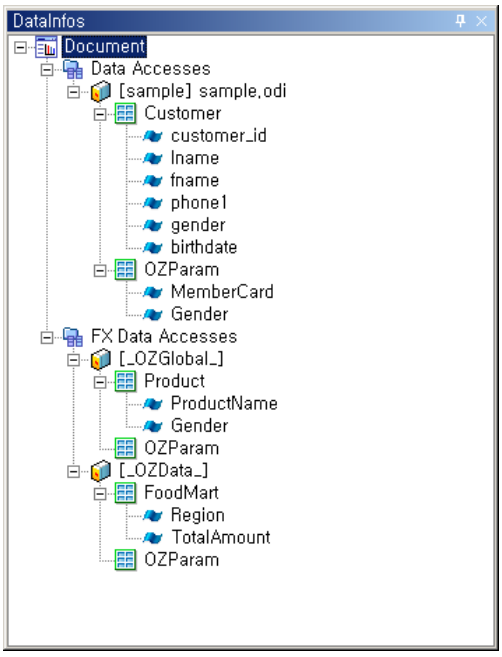
■ 그외 컴포넌트 팝업 메뉴



Align	다수의 컴포넌트 간의 위치를 정렬할 수 있으며, 2개 이상의 컴포넌트를 선택하여야만 활성화됩니다. 자세한 설명은 본 매뉴얼의 Ⅲ. OZ Application Designer 화면 구성의 [메뉴바] – [레이아웃] 부분을 참조하십시오.
Order	컴포넌트 고유한 순서 즉 컴포넌트의 인덱스를 설정합니다. 자세한 설명은 본 매뉴얼의 Ⅲ. OZ Application Designer 화면 구성의 [메뉴바] – [레이아웃] 부분을 참조하십시오.
Wizard	컴포넌트의 속성을 설정할 수 있는 마법사 다이얼로그가 나타납니다. 마법사가 존재하는 컴포넌트인 경우에만 해당 메뉴가 활성화됩니다.

데이터 트리창

프로젝트에 추가된 ODI 파일 또는 FXData 모듈에 추가되어있는 데이터 소스, 데이터 셋, 필드 정보, 패러미터 등을 나타냅니다.



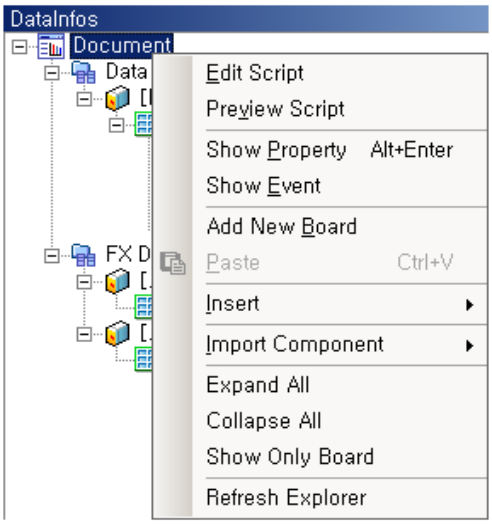
위 그림은 프로젝트에 sample.odi 파일과 FXData 모듈이 추가되어있는 모습입니다.
추가된 ODI 파일의 데이터 소스는 "sample"이고, 데이터 셋 이름은 "Customer"이고, 그 필드는 "customer_id", "lname", ..., "birthdate"이며, 패러미터는 "MemberCard", "Gender"라는 것을 알 수 있습니다.
추가된 FXData 모듈 중 "_OZGlabl_" 데이터 모듈의 데이터 셋 이름은 "Product"이고, 그 필드는 "ProductName", "Price"이며, "_OZData_" 데이터 모듈의 데이터 셋 이름은 "Foodmart"이고 그 필드는 "Region","TotalAmout"라는 것을 알 수 있습니다.

※ 참고사항 : FXData 모듈은 오즈 서버를 사용하지 않고 클라이언트에서 직접 데이터를 바인딩할 수 있는 데이터 모듈로 서로 다른 OZA간에 공유할 수 있는 데이터 모듈(_OZGlobal_)과 해당 OZA에서만 사용 가능한 데이터 모듈(_OZData_)로 구성되어 있습니다. FXData 모듈을 추가하고 사용하는 방법은 본 매뉴얼 "Ⅵ. 데이터 연동 - FXData 모듈 사용" 부분을 참조하시기 바랍니다.

데이터 트리창에서는 각각의 아이템 별로 팝업 메뉴가 제공됩니다.

※ 참고사항 : 중복되는 메뉴에 대해서는 한번만 설명하였사오니, 선택한 컴포넌트에 맞게 적절히 적용하시기 바랍니다.

■ Document 팝업 메뉴



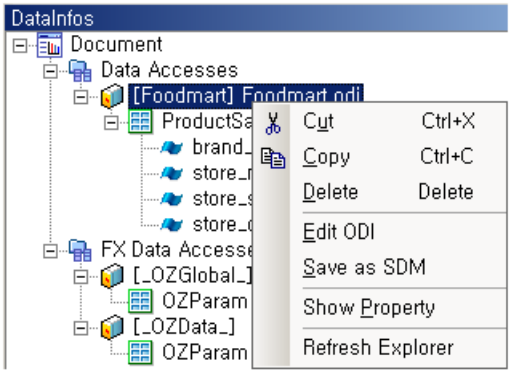
Expand All	데이터 트리창에 추가되어있는 모든 아이템을 펼쳐서 표시합니다.
Collapse All	데이터 트리창에 추가되어있는 모든 아이템을 접어서 표시합니다.

■ Data Accesses 팝업 메뉴



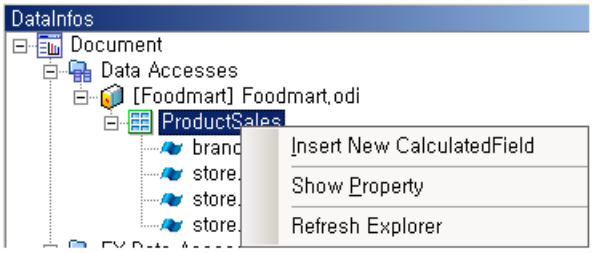
Insert New ODI	프로젝트에 ODI 파일을 추가합니다.

■ ODI 팝업 메뉴



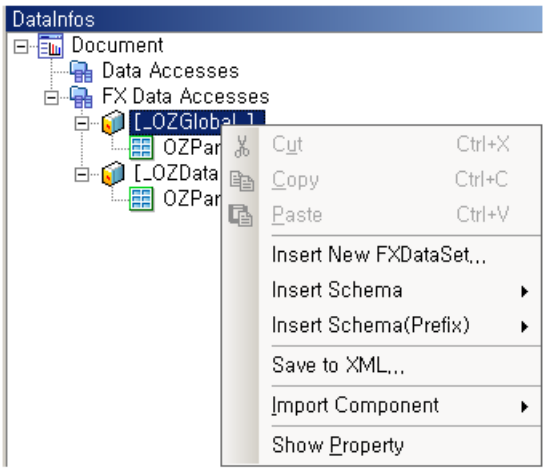
Edit ODI	선택한 ODI 파일을 편집합니다. [Edit ODI] 메뉴를 클릭하면 오즈 쿼리 디자이너가 자동으로 실행되어 선택한 ODI 파일을 편집할 수 있도록 해줍니다. ※ 참고사항 : [Edit ODI] 메뉴 실행시 오즈 쿼리 디자이너의 실행 파일이 설정한 경로에 없거나 ODI 편집 경로가 지정되어 있지 않은 경우 오즈 쿼리 디자이너 실행 파일의 경로를 선택할 수 있는 다이얼로그가 나타납니다.
Save As SDM	선택한 ODI 파일의 데이터를 SDM 파일로 저장합니다.

■ 데이터 셋 팝업 메뉴



Insert New CalculatedField	해당 데이터 셋에 수식 필드를 추가합니다.

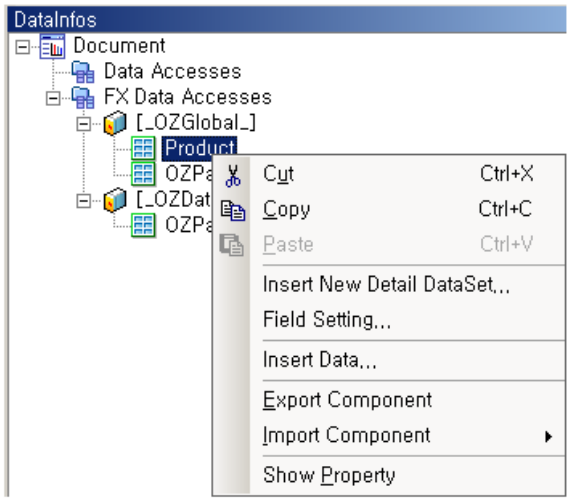
■ _OZGlobal_



Insert New FXDataSet	_OZGlobal_에 데이터셋을 추가합니다.

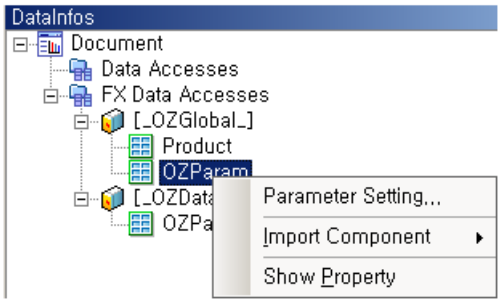
Insert Schema	_OZGlobal_에서 ODI, SDM, XML, Framework에 정의되어 있는 스키마를 추가합니다. 해당 메뉴를 이용하면 데이터셋이나 패러미터 등을 일일이 추가하지 않고, 해당 ODI, SDM, XML, Framework에 추가되어있는 데이터셋이나 패러미터 등의 정보를 그대로 사용할 수 있습니다.
Insert Schema(Prefix)	_OZGlobal_에서 ODI, SDM, XML, Framework에 정의되어있는 스키마 추가 시 데이터셋 이름에 접두어를 지정하여 추가합니다. 접두어를 지정하면 중복된 데이터셋 이름이 사용되어 충돌되는 문제를 방지할 수 있습니다.
Save to XML	_OZGlobal_의 데이터 정보 및 데이터를 XML 포맷으로 저장합니다.

■ FX DataSet



Insert New Detail DataSet	선택한 데이터셋의 디테일 셋을 추가합니다.
Field Setting	선택한 데이터셋의 필드를 추가하거나 이미 추가되어있는 필드의 정보를 설정합니다.
Insert Data	선택한 데이터셋의 필드에 데이터를 추가합니다.

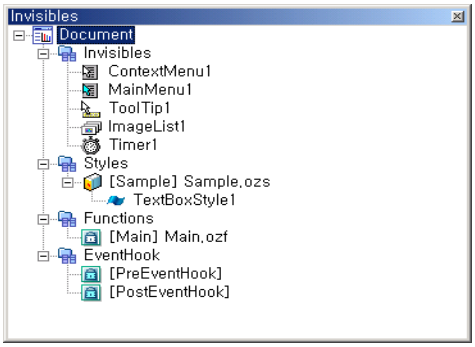
■ FX ParameterSet



Parameter Setting	_OZGlobal_에 패러미터를 추가하거나 이미 추가되어있는 패러미터의 정보를 설정합니다.

Invisibles 트리창

프로젝트에 추가된 컴포넌트 중 Invisibles 컴포넌트나 OZF, OZS, EventHook 등을 나타냅니다.



위 그림은 Invisibles 컴포넌트로는 ContextMenu1, MainMenu1, ToolTip1, ImageList1, Timer1가 추가되었고, OZ Style 파일로는 Sample.ozs 파일이 추가되었고, OZ Function 파일로는 Main.ozf 파일이 추가되었음을 나타낸 것입니다.

EventHook은 사용자가 추가하는 것이 아니라 프로젝트를 만들면 Document에 자동으로 추가되는 컴포넌트로, 자세한 설명은 본 매뉴얼의 Ⅴ.장 컴포넌트 기본 사용의 EventHook 부분을 참조하십시오.

Invisbles 트리창에서는 각각의 아이템 별로 팝업 메뉴가 제공됩니다.

※ 참고사항 : 중복되는 메뉴에 대해서는 한번만 설명하였사오니, 선택한 컴포넌트에 맞게 적절

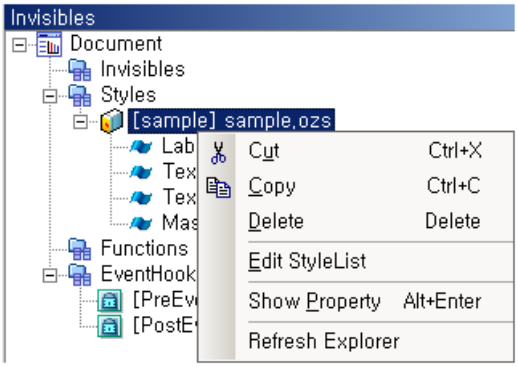
히 적용하시기 바랍니다.

■ Styles 팝업 메뉴



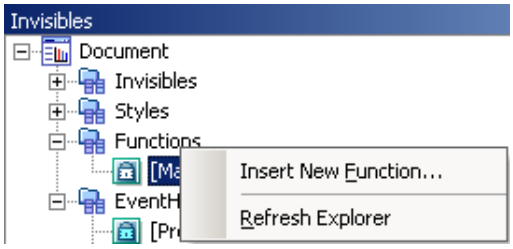
Insert New Style	Document에 트리에 새로운 스타일을 추가합니다.

■ OZS 팝업 메뉴



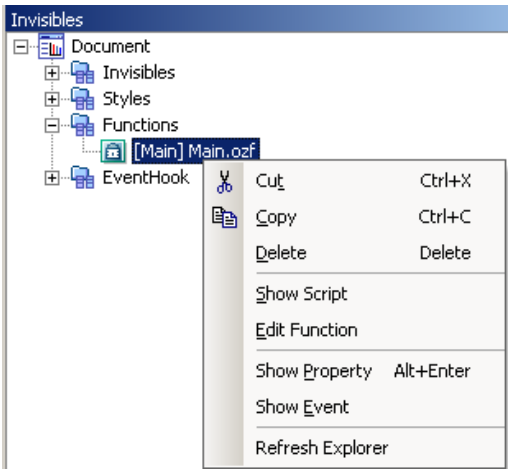
Edit StyleList	선택한 OZS 파일을 편집합니다. [Edit StyleList] 메뉴를 클릭하면 오즈 애플리케이션 스타일 편집기가 자동으로 실행되어 선택한 OZS 파일을 편집할 수 있도록 해줍니다.

■ Functions 팝업 메뉴



Insert New Function	Document에 새로운 함수를 추가합니다.

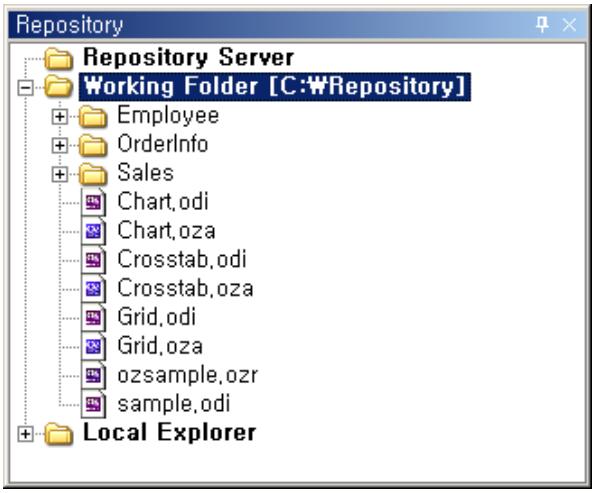
■ OZF 팝업 메뉴



Edit Function	선택한 OZF 파일을 편집합니다. [Edit Function] 메뉴를 클릭하면 오즈 애플리케이션 함수 편집기가 자동으로 실행되어 선택한 OZF 파일을 편집할 수 있도록 해줍니다.

리파지토리창

로컬 및 서버의 리파지토리 내의 폴더 및 파일을 탐색기 형태로 나타냅니다.



리파지토리창에서는 기본적으로 윈도우에서 제공하는 팝업 메뉴 외에 각각의 아이템 별로 팝업 메뉴가 제공됩니다.

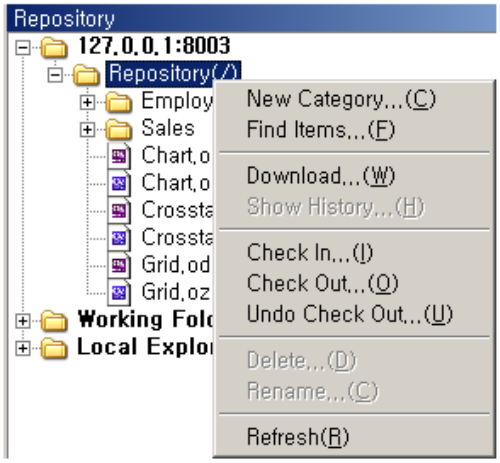
※ 참고사항 : 중복되는 메뉴에 대해서는 한번만 설명하였사오니, 선택한 컴포넌트에 맞게 적절히 적용하시기 바랍니다.

■ RepositoryServer 팝업 메뉴



Open Repository	ServerList 다이얼로그에서 접속할 서버 리파지토리 정보를 설정한 후 서버 리파지토리에 접속합니다. 자세한 설명은 OZ Repository Manager User's Guide 의 "Ⅱ. 오즈 리파지토리 매니저 – 오즈 리파지토리 매니저 메뉴" 부분을 참조하시기 바랍니다.
Close Repository	서버 리파지토리와의 접속을 끊습니다.

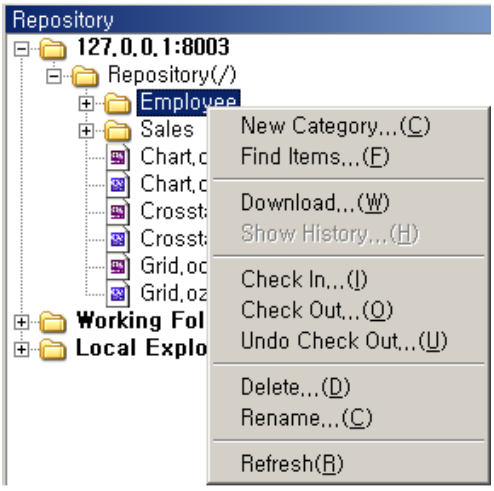
■ 서버의 Repository 팝업 메뉴



New Category	서버 리파지토리에 새로운 카테고리를 생성합니다.
Download	선택된 아이템을 로컬 시스템의 작업 폴더로 다운로드 합니다.
Check In	선택된 파일을 체크인하여 서버 리파지토리에 올립니다.
Check Out	서버 리파지토리에 있는 파일을 편집하기 위해 선택된 파일을 체크 아웃합니다.
Undo Check Out	선택된 파일을 체크 아웃 취소합니다.

Refresh	연결된 오즈 서버 리파지토리의 상태를 새로고침합니다.

■ 서버의 리파지토리 아이템 팝업 메뉴



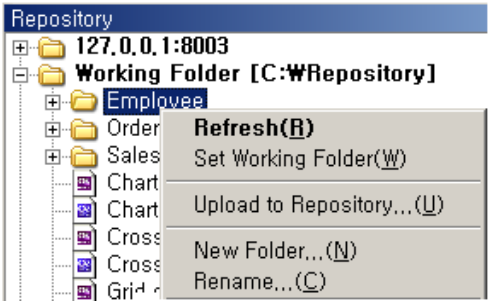
Delete	선택된 파일/카테고리를 삭제합니다.
Rename	선택된 파일/카테고리의 이름을 변경합니다.

■ Working Folder 팝업 메뉴



Upload to Repository	서버 리파지토리로 아이템을 업로드합니다. 서버 리파지토리에 접속했을 경우에만 활성화됩니다.
New Folder	새로운 폴더를 만듭니다.

■ 작업 폴더 아이템 팝업 메뉴



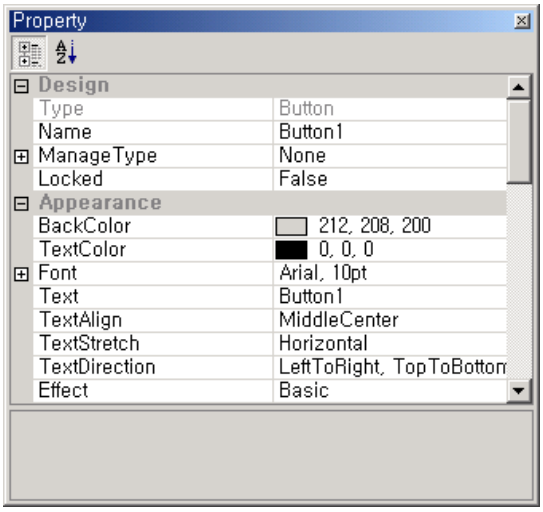
Set Working Folder	선택된 폴더를 작업 폴더로 설정합니다.

속성창, 이벤트창

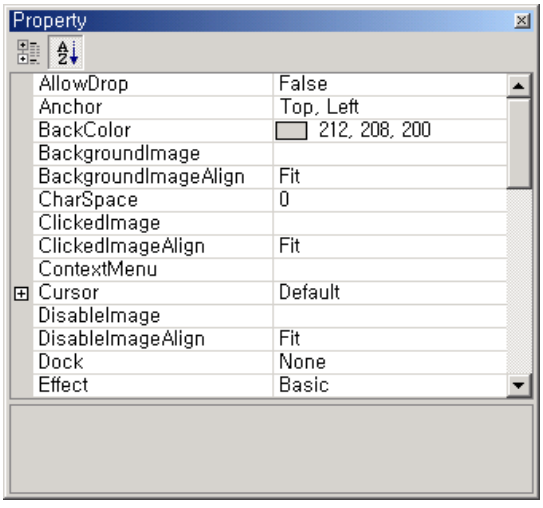
속성창은 컴포넌트의 속성 값을 표시하고 설정하는 창이며, 이벤트창은 선택한 컴포넌트의 이벤트와 해당 이벤트의 스크립트 코딩 여부를 표시하는 창입니다.

속성창

컴포넌트 트리창, Invisibles 트리창에서 선택한 컴포넌트 또는 데이터 트리창에서 선택한 데이터에 대한 속성과 해당 속성에 설정된 값을 표시합니다.

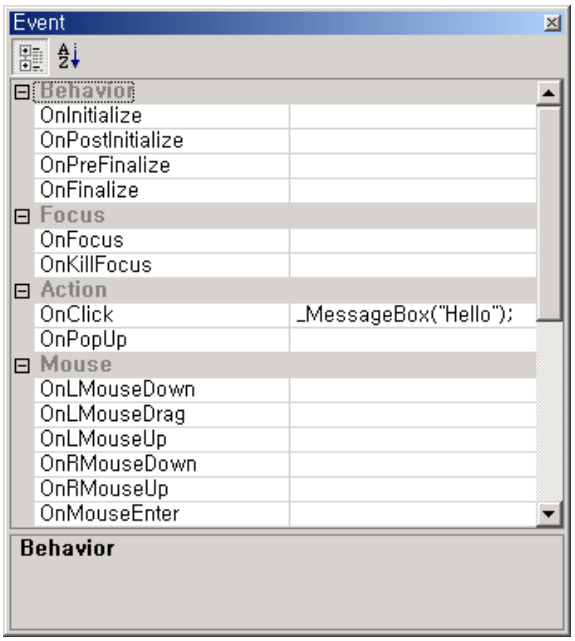


알파벳순 정렬 아이콘( )을 클릭하면 속성을 알파벳순으로 정렬하여 볼 수 있습니다.

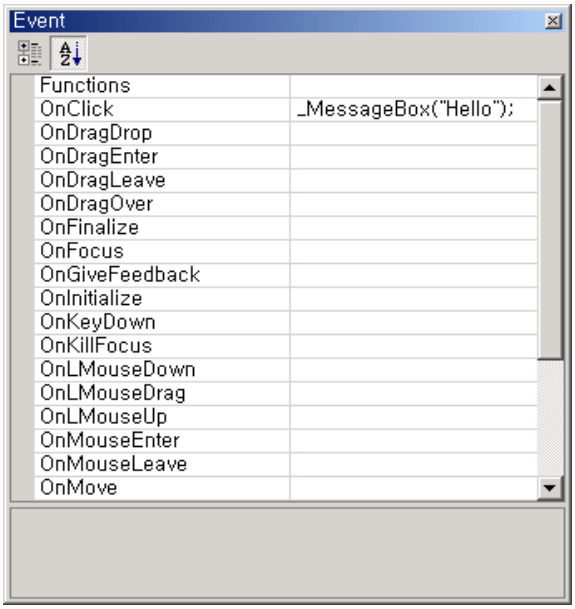


이벤트창

컴포넌트 트리창, Invisibles 트리창에서 선택한 컴포넌트에 대한 이벤트를 표시하며 아래 그림에서는 OnClick 이벤트에 스크립트가 코딩되어 있다는 것을 알 수 있습니다.



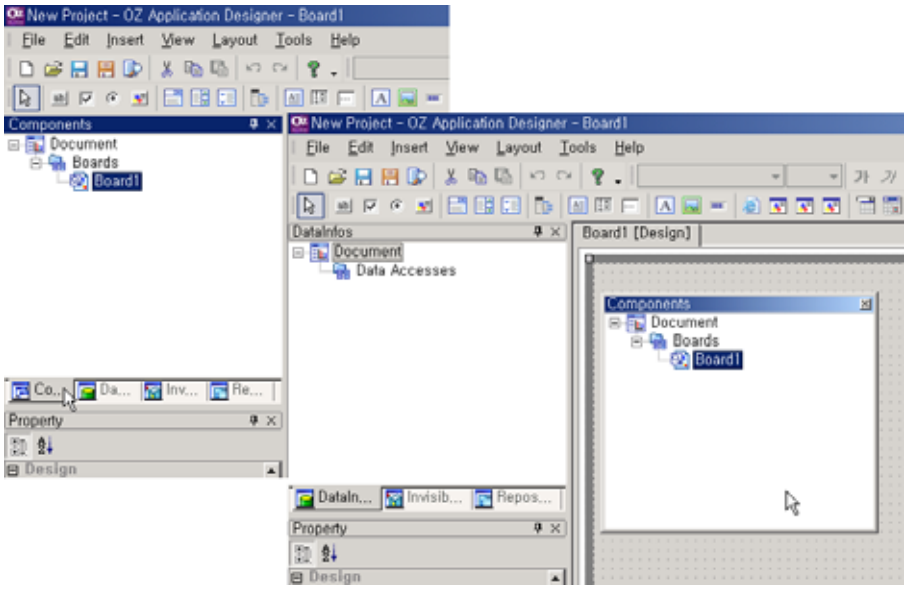
알파벳순 정렬 아이콘( )을 클릭하면 속성을 알파벳순으로 정렬하여 볼 수 있습니다.



※ 참고사항

트리창, 속성창, 이벤트창은 OZ Application Designer 화면에 고정시킬 수도 있지만 따로 분리할 수도 있습니다.

예를 들어 컴포넌트 트리창만 분리할 경우 컴포넌트 트리창을 나타내는 탭을 마우스로 클릭하여 Board창으로 Drag & Drop하면 컴포넌트 트리창이 분리되는 것을 확인할 수 있습니다.

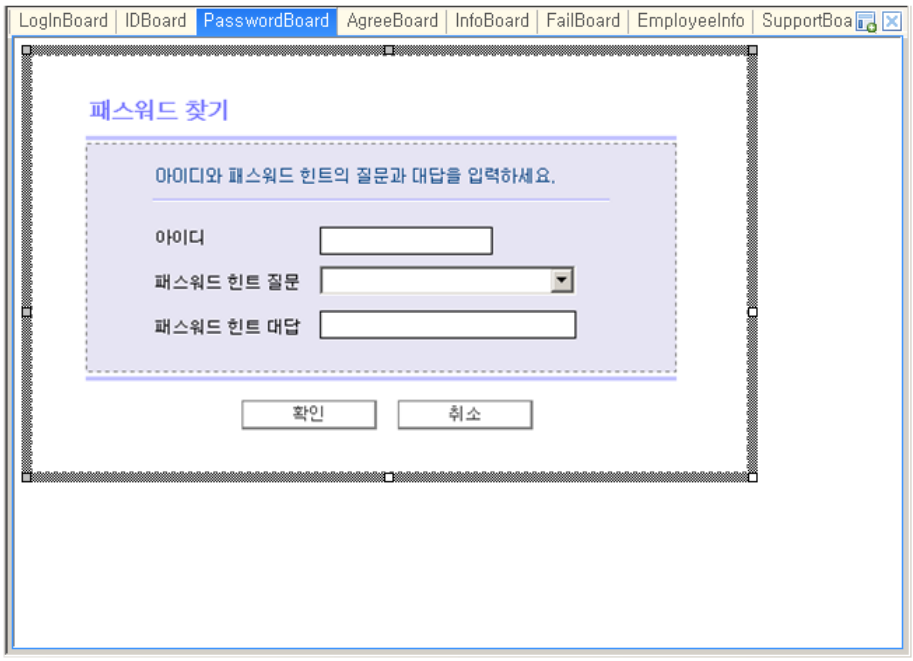


분리된 창을 다시 고정시킬 경우에는 창의 타이틀(창의 상단 파란색 부분)을 더블 클릭하거나 메뉴바의 [Tools] → [Options] 메뉴의 [Reset Window Layout] 버튼을 클릭합니다.

Board창, 스크립트창

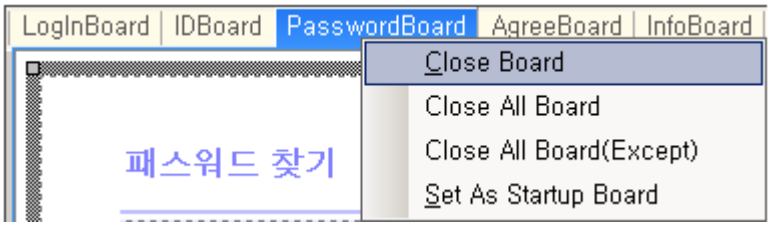
Board창

Board창은 프로젝트에 추가된 Board를 탭 형태로 표시하고, Board에 컴포넌트를 추가하고 추가된 컴포넌트의 위치, 크기 등의 레이아웃을 디자인하는 창입니다.

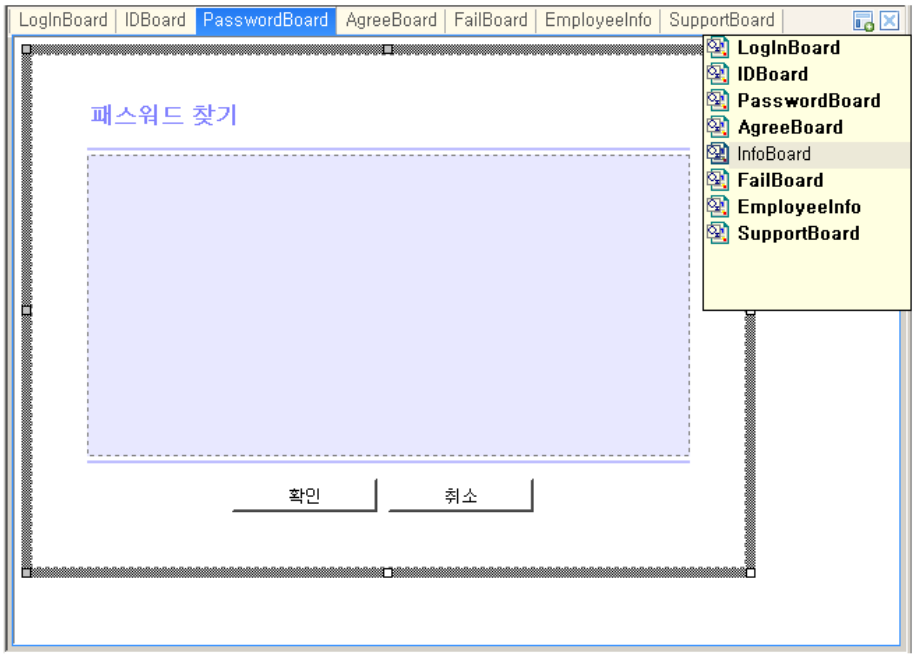


프로젝트에 여러 개의 Board가 추가되어 있어 디자인 시에 Board 선택이 불편할 경우에는 작업하지 않는 Board 탭을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Close Board] 메뉴를 클릭하거나 Board창 상단의 [ ] 버튼을 클릭하면 해당 Board가 Board창에 표시되지 않습니다.

또한 프로젝트에 추가된 모든 Board를 닫을 경우에는 [Close All Board] 메뉴를, 현재 선택된 Board를 제외한 모든 Board를 닫을 경우에는 [Close All Board(Except Current Board)] 메뉴를 클릭하여 Board를 닫습니다.



Board가 닫혀진 상태라고해서 프로젝트에서 삭제되는 것은 아니며, 컴포넌트 트리창에서 해당 Board를 더블 클릭하거나 해당 Board를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Open Board] 메뉴를 클릭하면 Board창에 다시 표시됩니다. 또한 보드창의 우측 상단에 표시된 [ ] 버튼을 클릭하여 나타나는 보드 리스트창에서 해당 Board를 클릭하여도 Board창에 닫혀진 Board가 다시 표시됩니다.



보드 리스트창에는 프로젝트에 추가된 모든 Board가 표시되며, 열려진 보드의 이름은 글꼴 스타일이 "굵게", 닫혀진 보드의 이름은 글꼴 스타일이 "보통"으로 표시됩니다.
보드 리스트창에서 보드 이름을 클릭하면 해당 보드가 보드 창에 표시됩니다.

스크립트창

Document, Board, 컴포넌트 등 프로젝트에 추가된 모든 구성 요소의 각 이벤트에 대한 스크립트를 코딩하거나 코딩된 내용을 표시하는 창입니다. 스크립트창에 대한 자세한 사용 방법은 Ⅳ. 장을 참조하십시오.

Ⅳ. OZ Application Designer 사용

- Document, Board 개념
- 작업 폴더 설정
- 시작 Board 설정
- 컴포넌트 추가
- 속성 값 변경
- 레이아웃 편집
- 스크립트 코딩
- 스크립트 디버깅
- 스크립트 비교
- 전역 변수, 전역 함수

본 장에서는 Document, Board의 개념과 OZ Application Designer를 사용하기 위해 알아두어야 할 기본적인 사용 방법에 대해 설명합니다.

Document, Board 개념

Document

Document는 프로젝트에 추가된 모든 컴포넌트, ODI, OZF, OZS 정보 등을 포함하는 OZ Application Designer의 최상위 객체로, 한 프로젝트당 하나의 Document로 구성됩니다.

OZ Application Designer의 컴포넌트 중 Timer, Context Menu, Main Menu, Tooptip, Image List 등 Invisibles 컴포넌트와 Board는 Document의 자식 컴포넌트로 추가됩니다.

Board

Board는 프로젝트의 User Interface를 담당하는 폼으로, Button, TextBox, Table 등 OZ Application Designer의 컴포넌트 뿐만 아니라 OZC에 포함된 컴포넌트와 OZC 파일 정보를 포함하는 Document의 자식 컴포넌트입니다.

OZ Application Designer의 컴포넌트 중 Controls 컴포넌트, Container Controls 컴포넌트, Complex Controls 컴포넌트, Drawings 컴포넌트는 Board의 자식 컴포넌트로 추가됩니다.

작업 폴더 설정

작업 폴더는 리파지토리 클라이언트가 리파지토리 서버로부터 파일을 업로드하거나 다운로드 하는 폴더로 프로젝트에 추가되는 ODI, OZC, OZF, OZS 파일의 기준이 되는 폴더입니다. 프로젝트에 ODI, OZC, OZF, OZS 파일 등을 추가하고자 할 때 해당 파일의 경로는 작업 폴더를 기준으로하여 상대 경로로 인식됩니다. 또한 리파지토리를 이용한 파일의 공유 과정에서도 작업 폴더는 매우 중요한 역할을 담당합니다.

작업 폴더 기본 값

OZ Application Designer가 처음 실행되면 OZ Application Designer가 설치된 디렉토리의 Repository라는 폴더가 작업 폴더 기본 값으로 자동으로 설정됩니다. 만일 해당 디렉토리에 Repository라는 폴더가 없으면 OZ Application Designer가 실행될 때 자동으로 생성됩니다.
예를 들어, OZ Application Designer가 설치된 폴더가 다음과 같다면,

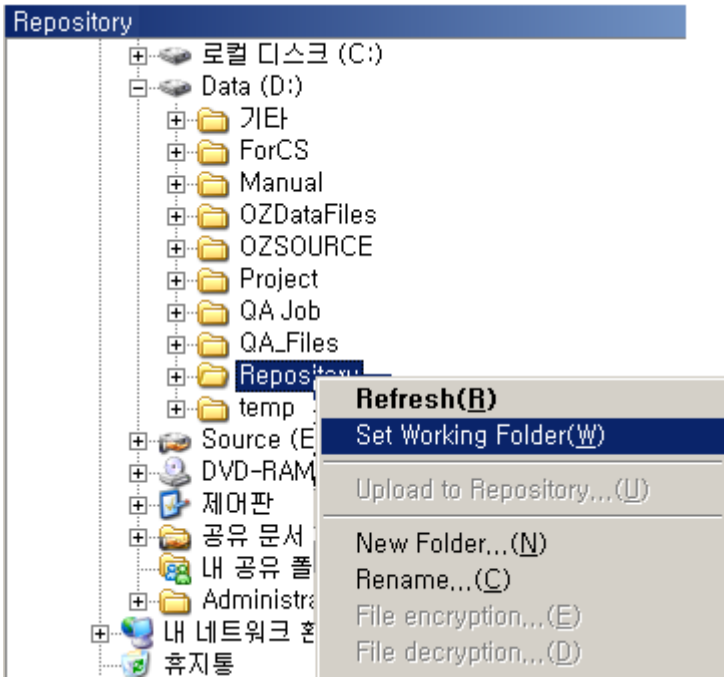
C:\Program Files\FORCS\OZ Xstudio 5.0\OZ Application Designer 5.0

작업 폴더 기본 값은 아래와 같습니다.

C:\Program Files\FORCS\OZ Xstudio 5.0\Repository

작업 폴더 설정

작업 폴더는 리파지토리창에서 쉽게 변경할 수 있습니다.
본 매뉴얼에서는 'C:\Repository'를 작업 폴더로 설정하겠습니다. 리파지토리창에서 새 작업 폴더로 설정할 폴더를 선택하고 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [Set Working Folder] 메뉴를 클릭하면 해당 폴더가 작업 폴더로 설정됩니다.

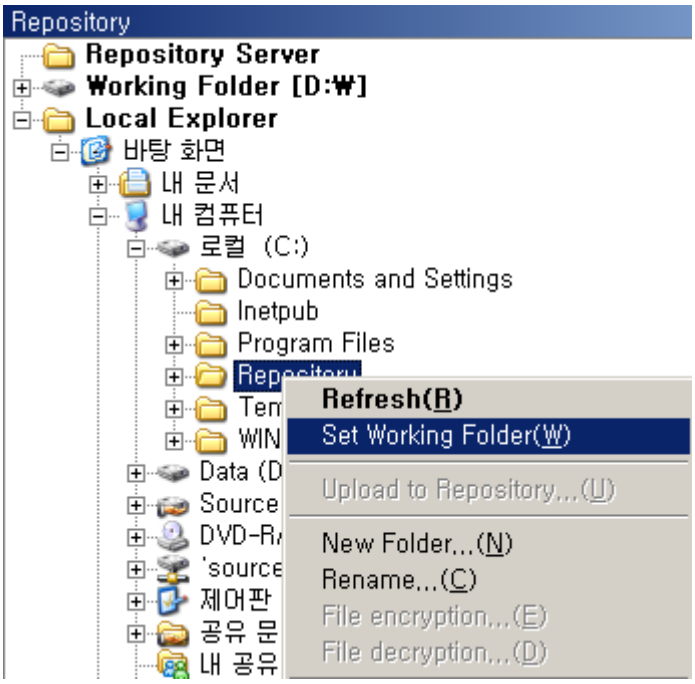


설정된 작업 폴더 경로는 리파지토리창에서 확인할 수 있으며 로컬 탐색기에서 작업 폴더를 설정할 경우 작업 폴더는 아래 그림과 같이 "Working Folder [작업 폴더 위치]"의 형태로 표시됩니다.



만일 선택한 경로가 작업 폴더 매니저에 등록되어 있는 경로일 경우 작업 폴더 이름과 작업 폴더 경로가 함께 표시됩니다.

예를 들어 "C:/Repository" 경로가 "Working folder1"로 작업 폴더 매니저에 등록되어 있을 경우 아래와 같이 리파지토리 창에서 작업 폴더를 설정하면

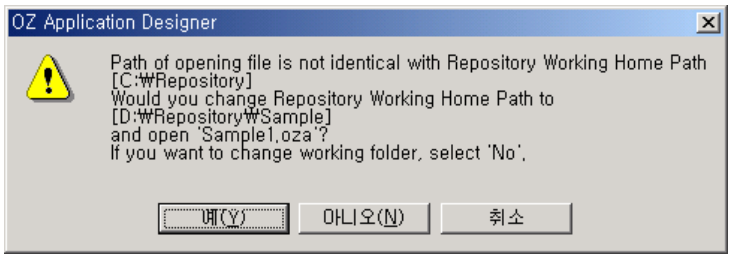


작업 폴더 이름과 경로가 함께 표시됩니다.



※ 참고사항

- 작업 폴더에서 OZA 파일을 더블 클릭하면 해당 파일이 열리며, ODI 또는 OZF 파일을 더블 클릭하면 현재 열려있는 프로젝트 파일에 선택한 ODI 또는 OZF 파일이 추가됩니다.
- 작업 폴더 또는 작업 폴더의 하위 폴더가 아닌 다른 폴더에 있는 OZA 파일을 열기하면 작업 폴더를 다시 설정할 것 인지 물어보는 메시지 박스가 나타납니다.



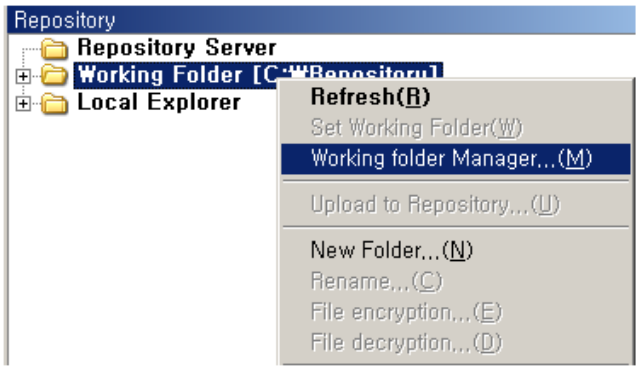
Button	Description

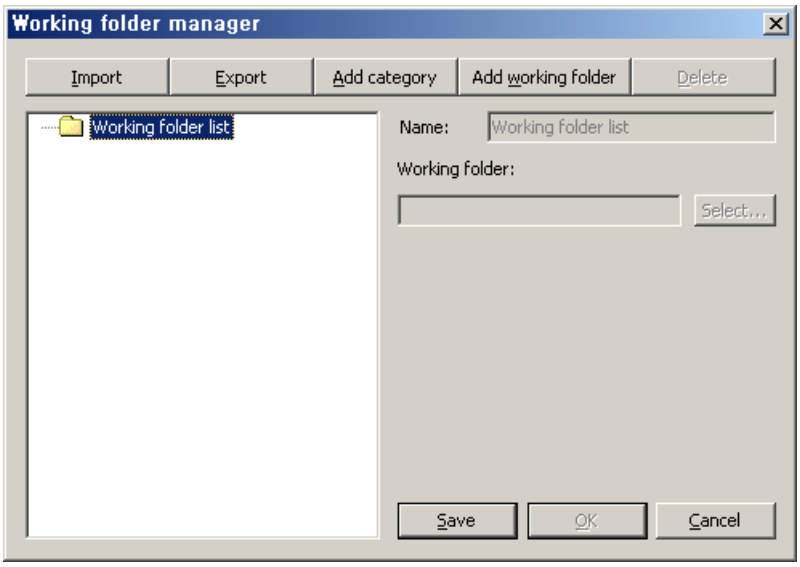
예	선택된 파일이 있는 디렉토리가 작업 폴더로 변경되고 해당 파일이 열립니다.
아니오	작업 폴더를 설정할 수 있는 "폴더 찾아보기" 다이얼로그가 나타나며 선택한 디렉토리로 작업 폴더의 경로가 변경된 후 해당 파일이 열립니다. 폴더 찾아보기 다이얼로그에서 작업 폴더로 설정할 수 있는 디렉토리는 선택한 파일이 있는 폴더 또는 그 상위 폴더만 가능합니다.
취소	작업 폴더를 변경하지 않으며 해당 파일도 열리지 않습니다.

작업 폴더 매니저 사용

작업 폴더 매니저는 로컬 탐색기에서 여러 개의 작업 폴더를 추가하여 관리할 수 있는 관리 도구입니다.

리파지토리 창에서 작업 폴더를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [작업 폴더 매니저] 메뉴를 클릭하여 실행합니다.





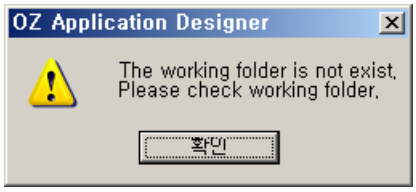
'작업 폴더 매니저'의 버튼 및 옵션에 대한 설명은 아래와 같습니다.

- 버튼

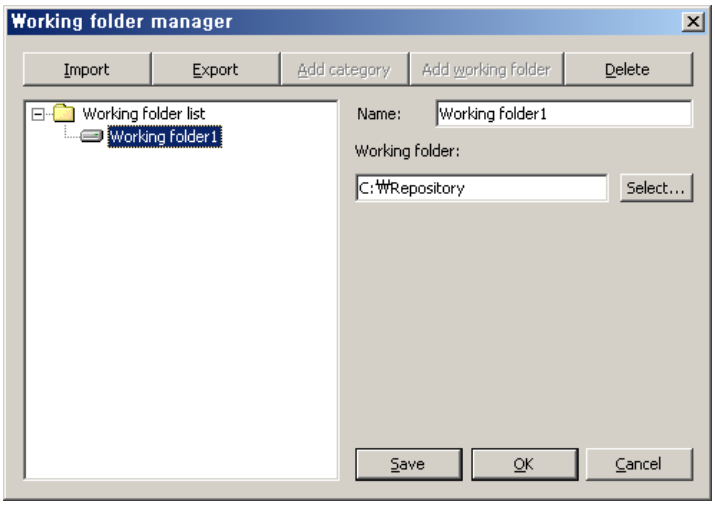
Button	Description
Import	작업 폴더 리스트 정보를 저장하고 있는 XML 파일을 가져와서 서버 리스트를 표시합니다.
Export	현재 다이얼로그에 추가되어 있는 서버 리스트 정보를 XML 파일로 저장합니니다.
Add Category	작업 폴더 리스트에 카테고리를 추가합니다.
Add Working Folder	새로운 작업 폴더를 추가합니다. [작업 폴더 추가] 버튼을 클릭하면 새로운 작업 폴더를 추가할 수 있도록 옵션 설정이 활성화됩니다.
Delete	선택된 작업 폴더 또는 카테고리를 삭제합니다.
Save	현재 작업 폴더 정보를 저장합니다.
OK	작업 폴더 매니저에서 설정한 옵션 정보를 저장 및 적용한 후 다이얼로그를 닫습니다.
Cancel	작업 폴더 매니저에서 설정한 옵션 정보를 적용하지 않고 다이얼로그를 닫습니다.

- 옵션

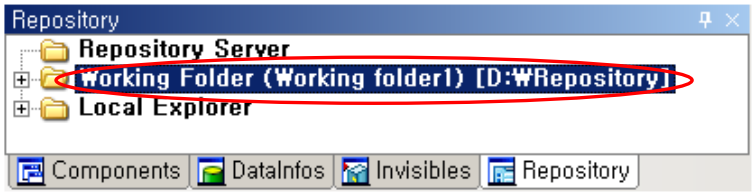
Option	Description

Name	설정할 작업 폴더의 이름을 입력합니다.
Working Folder	작업 폴더 다이얼로그에서 작업 폴더의 경로를 설정합니다. ※ 주의사항 : 마침표(.)가 포함되어 있는 경로는 작업 폴더로 설정할 수 없습니다. ※ 참고사항 : 작업 폴더 위치가 존재하지 않거나 작업 폴더로 설정할 수 없는 경로일 경우 [확인] 버튼을 클릭하면 아래와 같은 에러 메시지가 표시됩니다. 
Select	경로 선택 다이얼로그를 표시합니다.

작업 폴더 정보를 추가한 후 작업 폴더를 설정합니다. 본 매뉴얼에서는 아래와 같이 작업 폴더 정보를 입력한 후 [OK] 버튼을 클릭하여 설정하였습니다



설정된 작업 폴더 경로는 리파지토리창에서 확인할 수 있으며 작업 폴더 매니저에서 작업 폴더를 설정할 경우 작업 폴더는 아래 그림과 같이 "작업 폴더 (작업 폴더 이름) [작업 폴더 위치]"의 형태로 표시됩니다.

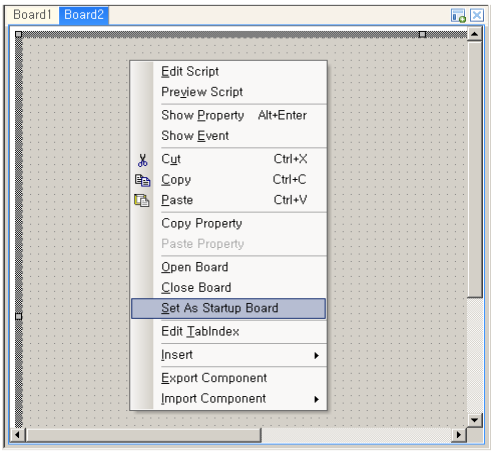


시작 Board 설정

다음은 프로젝트에 추가된 여러 개의 Board 중 미리보기를 했을 때 표시되는 Board 즉, 시작 Board를 설정하는 방법에 대해 설명합니다.

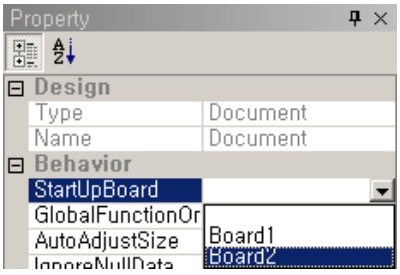
팝업 메뉴를 이용하여 시작 Board 설정

시작 Board로 설정할 Board를 선택한 후 마우스 오른쪽 버튼을 눌러 나타나는 팝업 메뉴에서 [Set As Startup Board] 메뉴를 클릭하여 화면에 보이는 Board를 시작 Board로 설정합니다.



Document의 속성을 이용하여 시작 Board 설정

컴포넌트 트리창에서 Document를 선택한 후 속성창에서 StartupBoard 속성 값을 시작 Board로 설정할 Board의 이름으로 변경합니다.



컴포넌트 추가

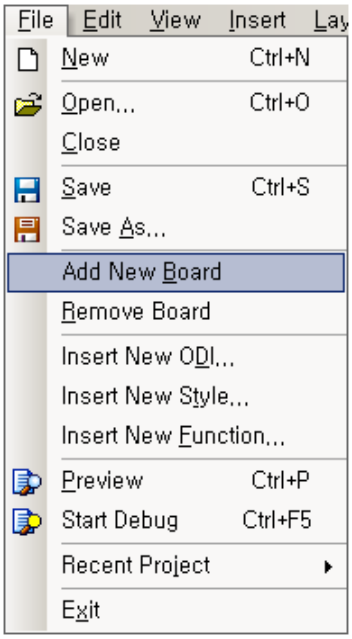
다음은 메뉴나 툴바를 이용하여 프로젝트에 컴포넌트를 추가하는 방법을 Document에 추가되는 컴포넌트와 Board에 추가되는 컴포넌트를 예로 들어 설명합니다.

Document에 추가되는 컴포넌트

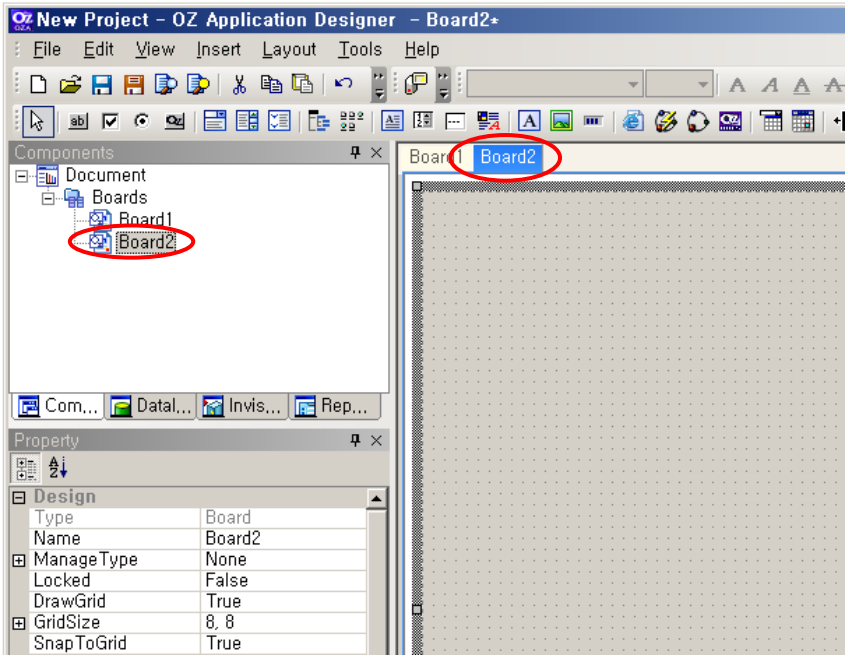
Board와 Invisibles 컴포넌트는 Document의 자식 컴포넌트로 추가됩니다.

- Board 추가

  [File] 메뉴의 [Add New Board] 메뉴를 클릭하면 Document에 Board가 추가됩니다.

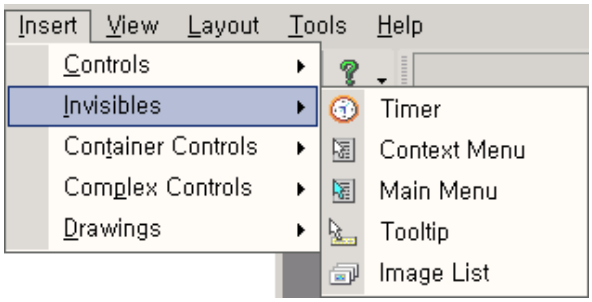


추가된 Board는 컴포넌트 트리창이나 Board 탭에서 확인할 수 있습니다.

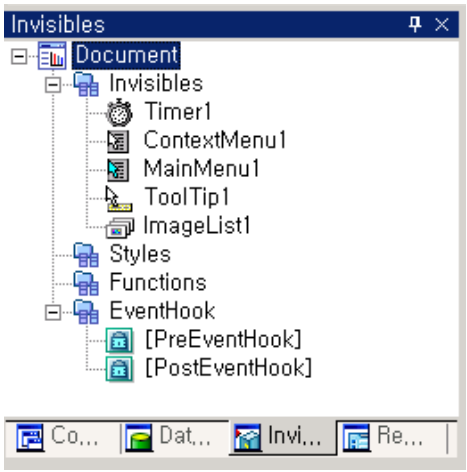


■ Invisibles 컴포넌트 추가

[Insert] 메뉴의 [Invisibles] 메뉴에서 추가할 컴포넌트를 클릭하거나 툴바에서 추가할 컴포넌트의 아이콘을 클릭하면 Document에 Invisibles 컴포넌트가 추가됩니다.



'Invisibles 트리창'의 'Invisibles' 하위 항목에서 추가된 Invisibles 컴포넌트를 확인할 수 있습니다.

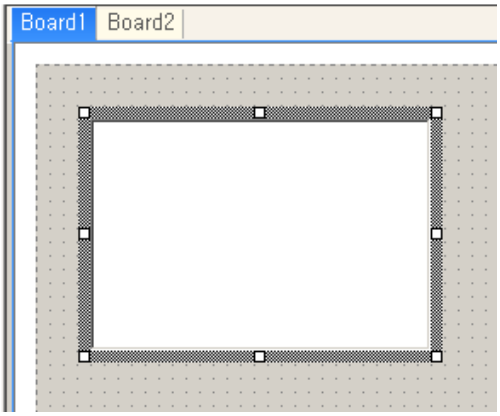


Board에 추가되는 컴포넌트

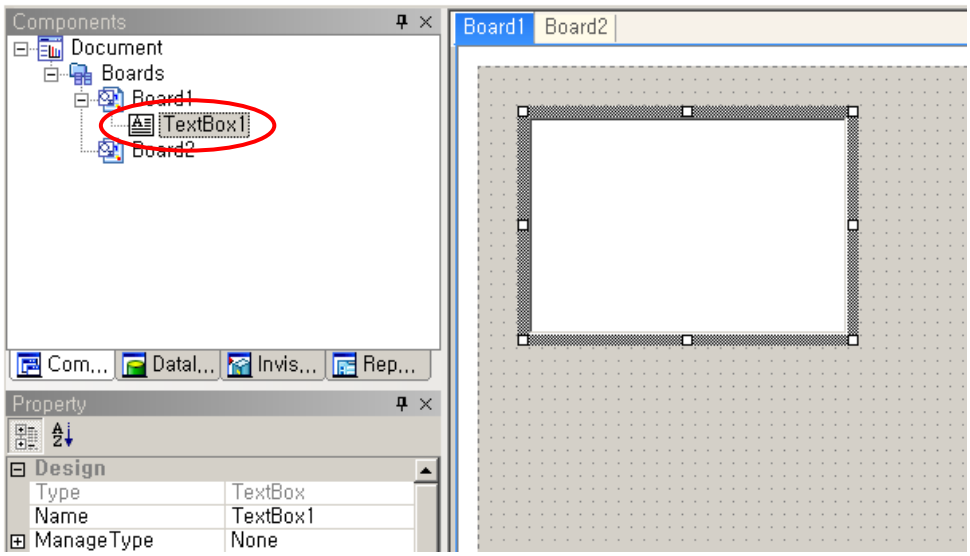
Controls 컴포넌트, Container Controls 컴포넌트, Complex Controls 컴포넌트, Drawings 컴포넌트는 Board의 자식 컴포넌트로 추가됩니다. 각각의 컴포넌트는 [Insert] 메뉴나 툴바에서 추가할 컴포넌트를 선택한 후 Board에 표시할 위치와 크기만큼 마우스로 Drag & Drop하여 추가합니다. 본 매뉴얼에서는 TextBox를 추가하는 방법을 예로 들어 설명하겠습니다.

■ TextBox 추가

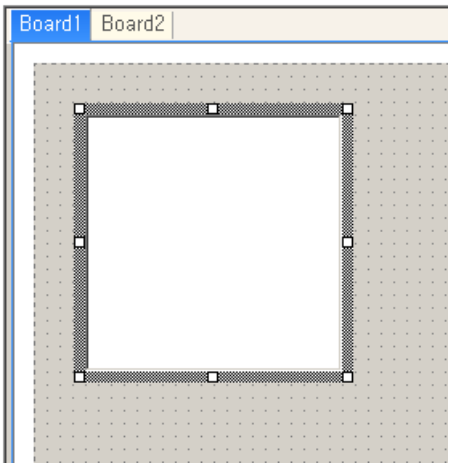
먼저 TextBox를 추가할 Board를 선택한 후 OZ Application Designer의 메뉴바나 Board의 팝업 메뉴의 [Insert] -> [Controls] -> [TextBox]를 클릭하거나 툴바에서 TextBox 아이콘()을 클릭하여 TextBox를 선택합니다. Board에 표시할 위치와 크기만큼 마우스로 Drag & Drop하여 추가합니다.



'컴포넌트 트리창'에서 'Board1'에 추가된 'TextBox1'을 확인할 수 있습니다.



컴포넌트를 추가할 때 <Shift> 키를 누른 상태에서 Drag & Drop하면 정사각형 모양으로 추가할 수 있습니다.



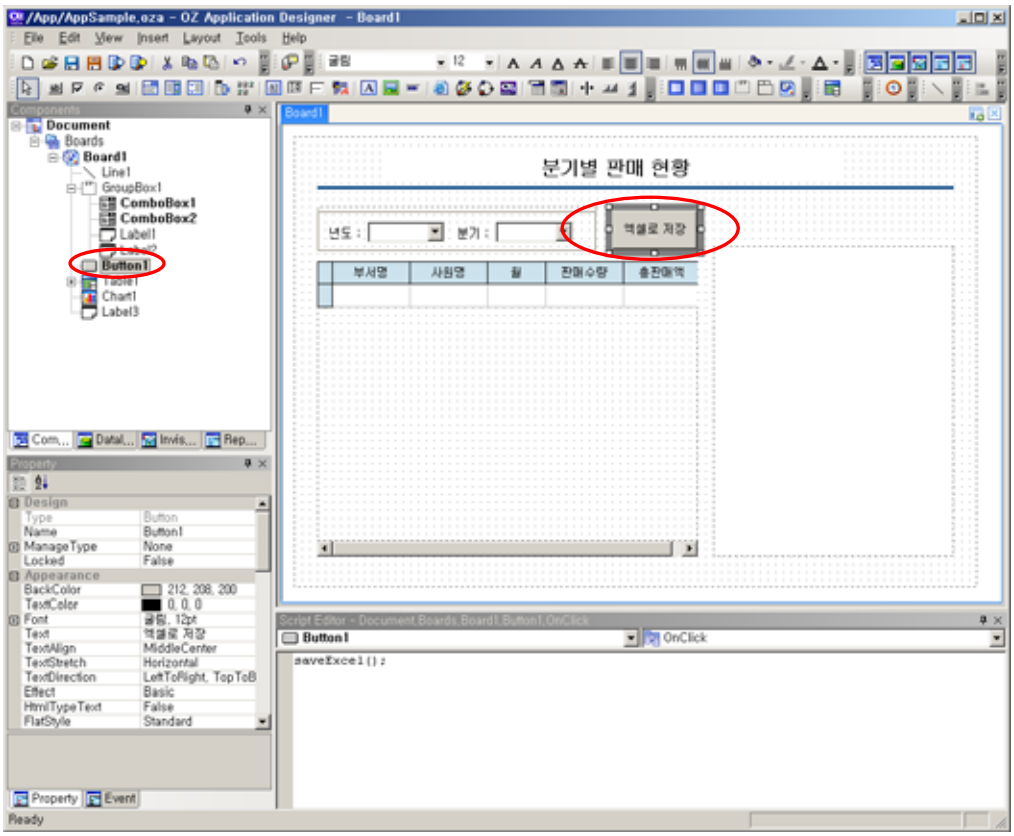
컴포넌트 선택

컴포넌트를 선택하는 방법과 기준 컴포넌트에 대해 설명합니다.

컴포넌트 선택

컴포넌트를 선택하는 방법은 두 가지로 보드창에서 컴포넌트를 마우스로 클릭하거나 컴포넌트 트리창에서 컴포넌트를 마우스로 클릭하는 방법이 있습니다.

선택된 컴포넌트는 보드창에서는 컴포넌트의 테두리에 선택 모양이 나타나고 컴포넌트 트리창에서는 선택된 컴포넌트의 이름에 회색 음영으로 채워져서 표시됩니다.



다음은 보드창에서 컴포넌트를 선택하는 방법에 대해 자세히 살펴 보겠습니다.

■ 단일 컴포넌트 선택

마우스로 해당 컴포넌트를 클릭합니다.

컴포넌트를 클릭하면 선택된 컴포넌트의 테두리로 선택 모양 즉 트랙커가 나타납니다. 트랙커에는 사각형 모양의 기준점이 있으며, 밴드는 한 개의 기준점이 표시되고, 라벨, 테이블 등의 컴포넌트는 8개의 기준점이 표시됩니다.

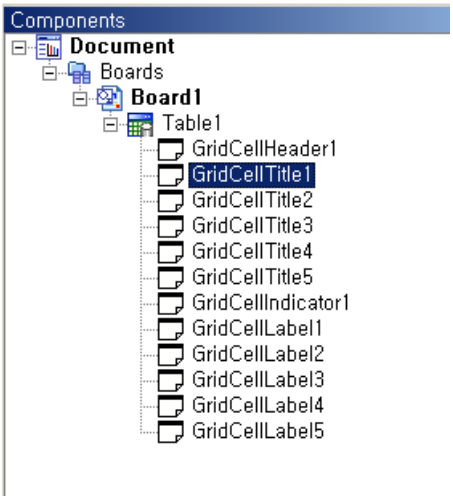
■ 다중 컴포넌트 선택

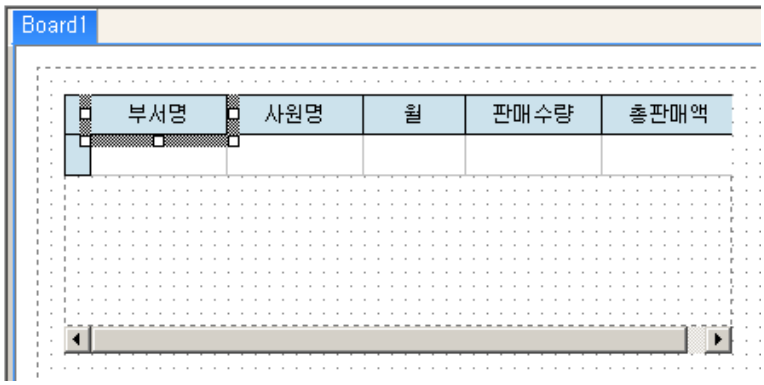
선택할 컴포넌트를 포함하는 영역만큼 마우스로 드래그하여 선택하거나 키보드의 <Shift> 또는 <Ctrl> 키를 누른 상태에서 마우스로 여러 개의 컴포넌트를 클릭하여 선택합니다.

■ 상위 컴포넌트 선택

특정 컴포넌트의 상위 컴포넌트를 선택하려면 해당 컴포넌트를 선택한 상태에서 키보드의 <Esc> 키를 누릅니다.

예를 들어, 아래 그림과 같이 Table1의 "GridCellTitle1" 선택되어 있다고 가정합니다.



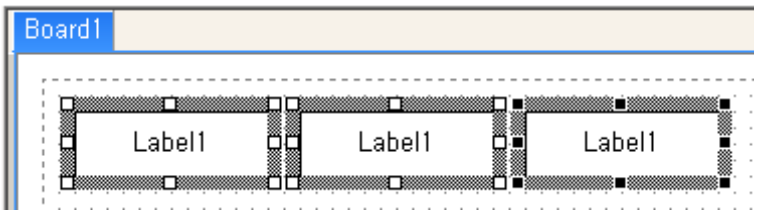


위 그림과 같은 상태에서 키보드의 <Esc> 키를 계속해서 누르면 "Table1 → Baord1" 순서로 선택됩니다.

기준 컴포넌트

기준 컴포넌트란 여러 개의 컴포넌트가 선택된 상태에서 위치 정렬, 크기 조절 등 컴포넌트의 레이아웃을 설정할 때 기준이 되는 컴포넌트를 말합니다.

기준 컴포넌트는 보고서창에서 확인할 수 있습니다. 아래 그림처럼 선택된 경우에는 트랙커의 기준점 사각형이 검정색으로 채워진 "Label1"이 기준 컴포넌트입니다.



컴포넌트를 선택하는 방법에 따라 기준 컴포넌트는 다르게 설정됩니다.

선택 방법	기준 컴포넌트
마우스로 드래그	컴포넌트 트리에서 맨 위에 위치한 컴포넌트 즉, 맨 처음에 바인딩되는 컴포넌트
<Shift> 키 + 마우스로 드래그&드롭	트리 창의 맨 위에 있는 컴포넌트
<Ctrl> 키 + 마우스로 드래그&드롭	트리 창의 맨 아래에 있는 컴포넌트
<Shift>키 + 마우스 클릭	맨 처음에 선택한 컴포넌트
<Ctrl>키 + 마우스 클릭	맨 마지막에 선택한 컴포넌트

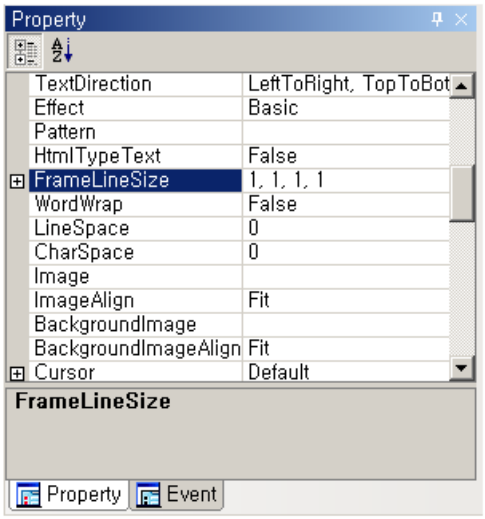
속성 값 변경

컴포넌트의 속성 값을 변경하는 방법은 두 가지로, 디자인시에 속성창을 이용하여 변경하는 방법과 스크립트로 코딩하여 실행시에 변경하는 방법이 있습니다. 본 장에서는 속성창을 이용하여 컴포넌트의 속성 값을 변경하는 방법에 대해 설명합니다.

'OZ Application Designer'는 속성창을 이용하여 속성 값을 변경하는 방법으로 속성 값을 직접 입력하는 방법, 속성 값 리스트에서 값을 선택하는 방법, 속성 값 설정 다이얼로그에서 값을 입력하거나 선택하는 방법 등 다양한 방법을 제공합니다.

직접 입력

컴포넌트를 선택한 후 속성창에서 속성 값을 직접 입력합니다. 아래 그림은 'FrameLineSize' 속성 값을 직접 입력하는 모습입니다.



이미지 파일 등의 경로를 직접 입력할 경우에는 OZURL 방식 또는 HTTP 경로로 입력할 수 있습니다.

※ 참고사항 : OZURL

OZURL이란, 이미지 파일 등의 경로를 설정할 때 OZ Application에서 인식하는 파일 경로로 다음과 같이 설정할 수 있습니다.

- 작업 폴더 또는 리파지토리의 파일 경로 설정 시

  ozp:///파일명 ex) ozp:///main.jpg

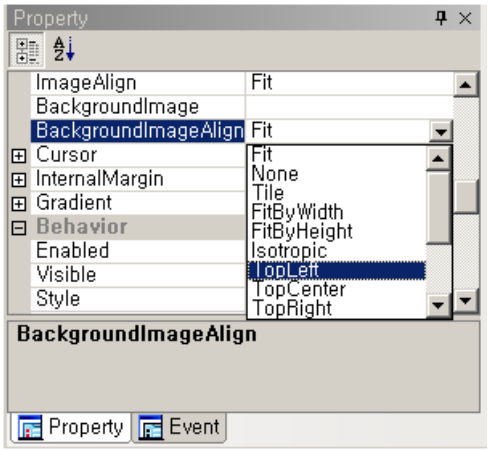
  ozp://카테고리명/파일명 ex) ozp://images/main.jpg

- 파일 경로를 직접 설정 시

  file://디렉토리명/파일명 ex) file://C:/title.gif

속성 값 리스트에서 선택

컴포넌트를 선택한 후 속성창에서 해당 속성이 가질 수 있는 값을 리스트 형태로 나타낸 콤보 박스에서 속성 값을 선택합니다. 아래 그림은 'BackgroundImageAlign' 속성 값을 콤보 박스에서 선택하는 모습입니다.

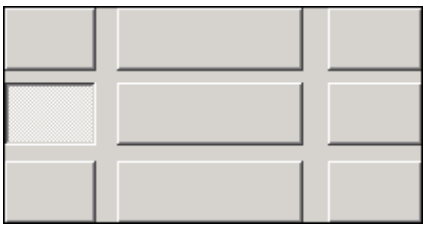


속성 값 설정 다이얼로그 이용

속성 값 옆의 콤보 박스 또는 다이얼로그 표시 버튼(...)을 클릭하여 나타나는 속성 값 설정 다이얼로그를 이용하여 속성 값을 변경합니다.

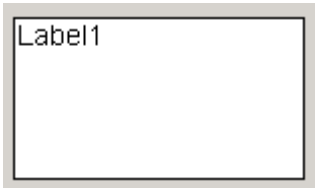
■ Alignment 다이얼로그

텍스트나 체크 박스의 표시 위치 즉, 'TextAlign', 'CheckAlign' 속성 값을 변경할 때 사용하는 다이얼로그입니다.



텍스트나 체크 박스의 위치를 마우스로 클릭하여 설정하며, 선택 버튼 중 눌러진 모양이 선택된 설정값입니다.

- Label의 TextAlign 속성 값을 'MiddleLeft'로 변경한 경우



(변경전)

TextAlign = TopLeft



(변경후)

TextAlign = MiddleLeft

■ Anchor 다이얼로그

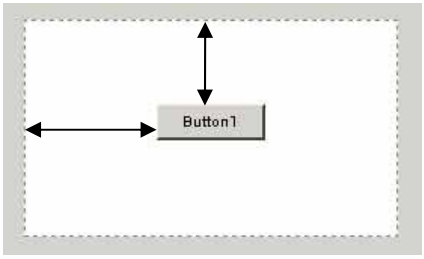
부모 컴포넌트와 자식 컴포넌트 사이의 위, 아래, 왼쪽, 오른쪽 사면의 간격 즉, 부모 컴포넌트의 테두리에 대한 자식 컴포넌트의 상대 위치를 고정시키는 방향값인 'Anchor' 속성 값을 변경할 때 사용하는 다이얼로그입니다.



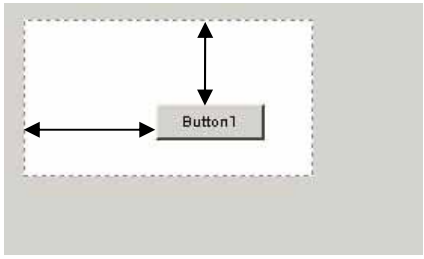
Top, Bottom, Left, Right 등 4가지의 값을 복수로 선택할 수 있으며, 선택 버튼 중 눌러진 모양이 선택된 설정값입니다.

'Anchor' 속성 값을 설정하여 자식 컴포넌트를 부모 컴포넌트의 테두리에 고정시키면 부모 컴포넌트의 크기가 변경되어도 설정한 Anchor 속성 값을 기준으로 부모 컴포넌트의 테두리와 자식 컴포넌트 간의 위치는 동일한 간격을 유지합니다.

- Panel의 자식 컴포넌트인 Button의 Anchor 속성 값이 'Top, Left'인 경우

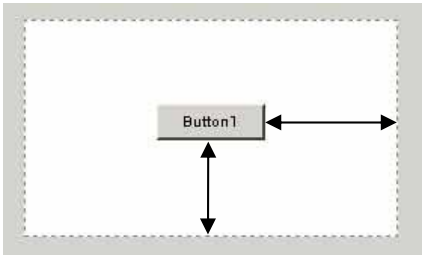


(Panel 크기 변경전)

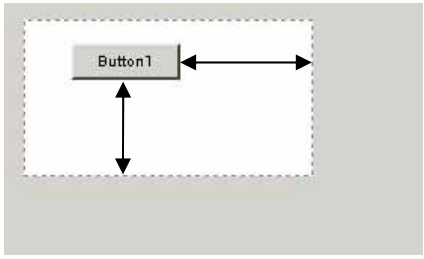


(Panel의 크기 변경후)

- Panel의 자식 컴포넌트인 Button의 Anchor 속성 값이 'Bottom, Right'인 경우



(Panel 크기 변경전)

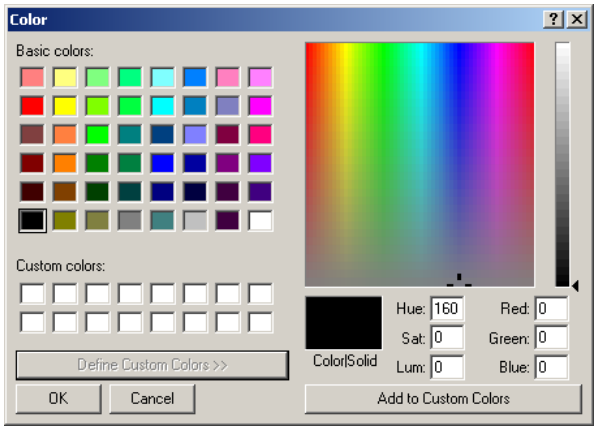


(Panel의 크기 변경후)

■ Color 다이얼로그

- 한 가지 색을 선택하는 경우

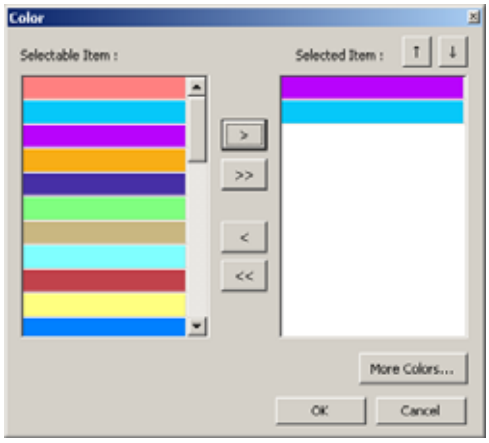
글자색, 배경색, 선색 즉, 'TextColor', 'BackColor', 'LineColor'등 색과 관련된 속성 값 속성 값을 변경할 때 사용하는 다이얼로그입니다.



원하는 색을 기본색에서 선택하거나 [Define Custom Colors] 버튼을 클릭하여 색 다이얼로그를 확장하여 사용자가 임의로 색을 선택할 수 있습니다. 자주 쓰이는 색은 [Add to Custom Colors] 버튼을 클릭하여 사용자 정의색에 저장할 수 있습니다.

- 두 가지 이상의 색을 선택하는 경우

차트의 "ChartColors" 속성과 "ChartGridLineColors" 속성, 테이블의 "RowColors" 속성 등 두 가지 이상의 색을 선택할 경우 사용하는 다이얼로그입니다.



원하는 색을 기본색에서 선택하거나 [More Colors] 버튼을 클릭하여 나타나는 "Colors" 다이얼로그에서 사용자가 임의로 색을 선택할 수 있습니다.





Standard 탭에서 정의된 색외에 Custom 탭에서 사용자가 임의로 색을 선택할 수 있습니다.

■ Date 다이얼로그

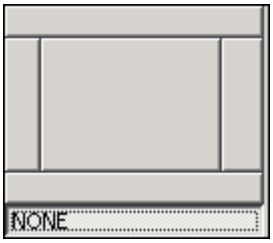
MonthCalendar, DateTimePicker의 'TodayDate', 'MinDate', 'MaxDate' 등 날짜와 관련된 속성 값을 변경할 때 사용하는 다이얼로그입니다.



해당 날짜를 클릭하여 날짜를 변경합니다.

■ Dock 다이얼로그

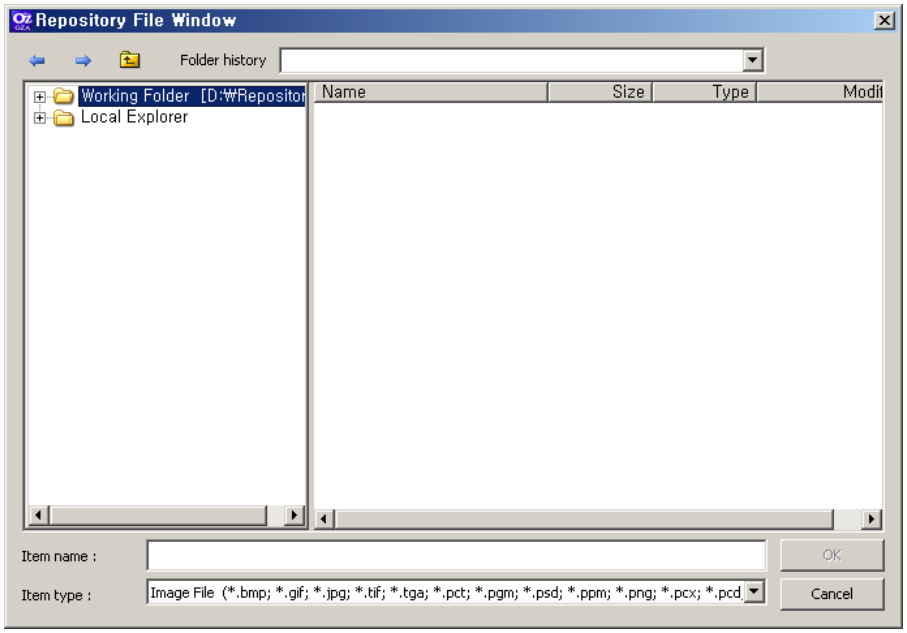
부모에 Docking하는 방향 즉, 'Dock' 속성 값을 변경할 때 사용하는 다이얼로그입니다.



자식 컴포넌트를 Docking하지 않을 경우에는 [NONE]을 클릭하고, Docking할 경우에는 위, 아래, 왼쪽, 오른쪽, 전체 중 하나를 클릭합니다.

■ File 다이얼로그

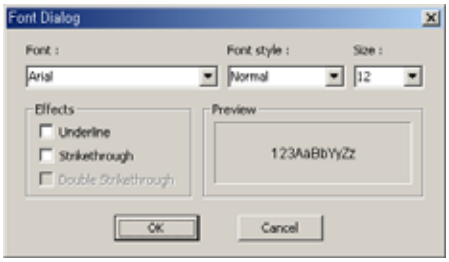
'BackgroundImage', 'ImageListUrl' 등 컴포넌트에 연결할 파일을 선택할 때 사용하는 다이얼로그입니다.



파일을 선택한 후 [OK] 버튼을 클릭하여 연결할 파일을 설정합니다.

- Font 다이얼로그

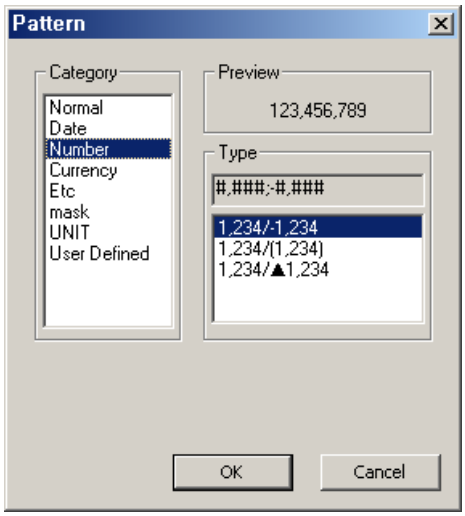
  글꼴, 글꼴 스타일, 글꼴 크기 등을 변경할 때 사용하는 다이얼로그입니다.



  글꼴, 글꼴 스타일, 글꼴 크기 등을 선택한 후 [OK] 버튼을 클릭하여 글꼴을 변경합니다.

- Pattern 다이얼로그

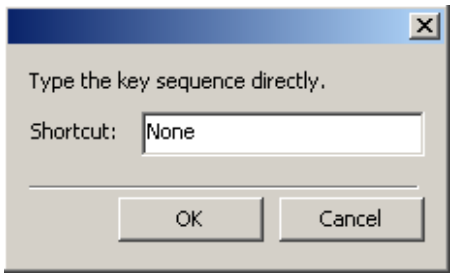
  화면에 보이는 텍스트의 패턴 즉, 'Pattern' 속성 값을 변경할 때 사용하는 다이얼로그입니다.



  'Category'에서 표시할 타입을 선택하고 세부 형태를 선택한 후 [OK] 버튼을 클릭하여 'Pattern' 속성 값을 변경합니다.

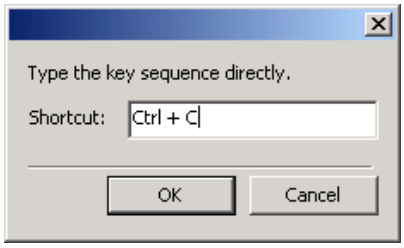
- Shortcut 다이얼로그

  단축키 즉, 'Shortcut' 속성 값을 설정하거나 해제할 때 사용하는 다이얼로그입니다.



- 단축키 설정

  '단축키 설정 다이얼로그'에서 단축키로 설정할 키를 입력하여 설정합니다. 예를 들어 'Ctrl + C' 키를 단축키로 설정할 경우 먼저 키보드의 [Ctrl] 키를 누릅니다. 이 때 텍스트 박스의 텍스트가 'None'에서 'Ctrl + '로 변경되고, 이 상태에서 알파벳 [C] 키를 누르면 텍스트가 'Ctrl + C'로 변경됩니다.



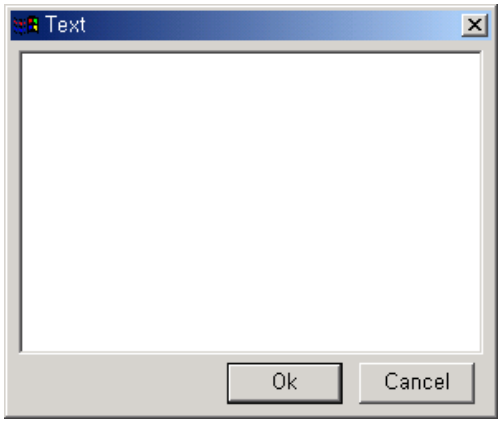
  [OK] 버튼을 클릭하여 단축키를 'Ctrl + C'로 변경합니다.

- 단축키 해제

  '단축키 설정 다이얼로그'를 띄운 후 키보드로 [Delete] 버튼을 클릭하여 설정된 단축키를 해제합니다.

■ Text 다이얼로그

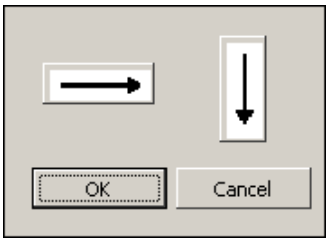
텍스트를 입력하거나 수정할 때 사용하는 다이얼로그입니다.



여러 줄로 입력한 내용이 컴포넌트에 적용되어 표시되려면 'Multiline' 속성 값을 'True'로 설정하여야 합니다.

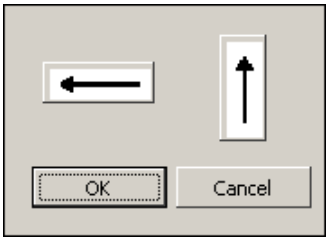
■ TextDirection 다이얼로그

텍스트의 출력 방향 즉, 'TextDirection' 속성 값을 변경할 때 사용하는 다이얼로그입니다.



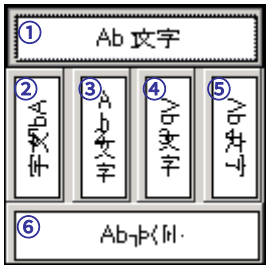
수평 방향에 대한 설정값(LeftToRight, RightToLeft)과 수직 방향에 대한 설정값(TopToBottom, BottomToTop)을 조합하여 설정합니다. 기본 값은 'LeftToRight, TopToBottom'이며, 수평 또는 수직 방향을 나타내는 버튼을 클릭하여 원하는 방향으로 변경한 후 [OK] 버튼을 클릭하면 설정값을 변경할 수 있습니다.

아래 그림은 'RightToLeft, BottomToTop' 값으로 변경한 모습입니다.



■ TextStretch 다이얼로그

텍스트의 레이아웃 및 회전 방향 즉, 'TextStretch' 속성 값을 변경할 때 사용하는 다이얼로그입니다.



설정할 값을 나타내는 버튼을 클릭하여 설정하며 각 옵션의 간략한 설명은 다음과 같습니다.

No	Option	Description

①	**Horizontal**	가로쓰기
②	**CountClockWise**	가로쓰기를 반시계방향으로 회전하여 표시
③	**Vertical**	세로쓰기
④	**Vertical2**	세로쓰기에서 1Byte 문자만 시계 방향으로 회전하여 표시
⑤	**ClockWise**	가로쓰기를 시계 방향으로 회전하여 표시
⑥	**Horizontal2**	가로쓰기에서 2Byte 문자만 반시계 방향으로 회전하여 표시

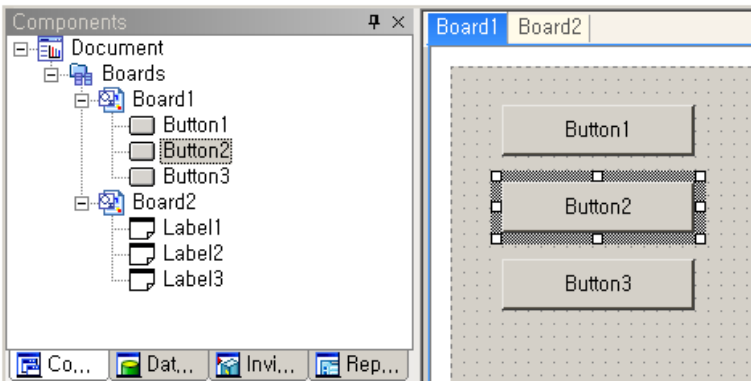
레이아웃 편집

다음은 컴포넌트의 선택, 위치, 크기, 편집, 정렬, 간격 등 컴포넌트의 레이아웃을 편집하는 방법에 대해 설명합니다.

선택

■ 단일 선택

Board창이나 컴포넌트 트리창에서 컴포넌트를 마우스로 클릭하여 선택합니다.

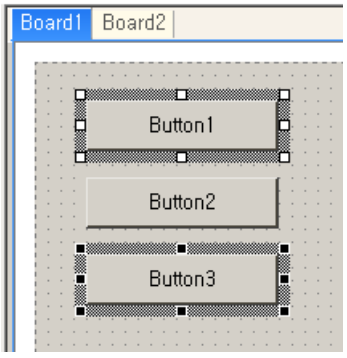


선택된 컴포넌트는 Board창에서는 컴포넌트의 테두리에 선택 모양이 표시되고, 컴포넌트 트리창에서는 컴포넌트명이 음영으로 표시됩니다

■ 다중 선택

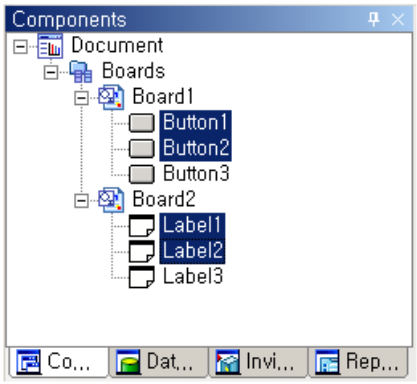
- Board창에서 다중 선택

  Board에 추가된 컴포넌트를 포함하는 범위만큼 마우스로 영역을 선택하거나 'Ctrl' 또는 'Shift' 키를 누른 상태에서 컴포넌트를 클릭하여 다중 선택합니다.



- 컴포넌트 트리창에서 다중 선택

  'Ctrl' 또는 'Shift' 키를 누른 상태에서 컴포넌트를 클릭하여 다중 선택합니다. Board창에서 다중 선택하는 것과 달리 컴포넌트 트리창에서는 다른 Board에 있는 컴포넌트도 선택할 수 있습니다.



위치 이동

■ 마우스 이용

컴포넌트를 하나 이상 선택한 상태에서 원하는 위치로 Drag & Drop합니다. 만약 컴포넌트가 속한 부모 컴포넌트의 속성 중 'SnapToGrid' 속성 값이 'True'로 설정되어 있으면, 컴포넌트는 부모 컴포넌트의 'GridSize' 속성 값에 설정된 눈금에 맞춰 이동됩니다.

■ 키보드 이용

컴포넌트를 하나 이상 선택한 상태에서 키보드의 방향키(← → ↑ ↓)를 클릭하여 위치를 이동합니다. 만약 컴포넌트가 속한 부모 컴포넌트의 속성 중 'SnapToGrid' 속성 값이 'True'로 설정되어 있으면, 컴포넌트는 부모 컴포넌트의 'GridSize' 속성 값에 설정된 눈금에 맞춰 이동됩니다.

크기 조정

■ 마우스 이용

컴포넌트를 하나 이상 선택한 상태에서 테두리선으로 마우스를 이동하여 마우스 커서가 크

기 조정 모양으로 변경될 때 마우스를 Drag하여 크기를 조정합니다. 만약 컴포넌트가 속한 부모 컴포넌트의 속성 중 'SnapToGrid' 속성 값이 'True'로 설정되어 있으면, 컴포넌트는 부모 컴포넌트의 'GridSize' 속성 값에 설정된 눈금에 맞춰 크기 조절됩니다.

■ 키보드 이용

컴포넌트를 하나 이상 선택한 상태에서 'Shift' 키를 누른 상태에서 방향키를 클릭하여 크기를 조정합니다. 만약 컴포넌트가 속한 부모 컴포넌트의 속성 중 'SnapToGrid' 속성 값이 'True'로 설정되어 있으면, 컴포넌트는 부모 컴포넌트의 'GridSize' 속성 값에 설정된 눈금에 맞춰 크기 조절됩니다.

Key	Description
Shift + ←	너비를 줄입니다.
Shift + →	너비를 늘립니다.
Shift + ↑	높이를 줄입니다.
Shift + ↓	높이를 늘립니다.

편집

■ 복사하기

컴포넌트를 하나 이상 선택한 상태에서 'Ctrl + C' 키를 클릭하거나 마우스 오른쪽 버튼을 클릭하여 복사하기를 선택합니다.

■ 잘라내기

컴포넌트를 하나 이상 선택한 상태에서 키보드로 'Ctrl + X' 키를 클릭하거나 마우스 오른쪽 버튼을 클릭하여 잘라내기를 선택합니다.

■ 붙여넣기

붙여넣기할 부모 컴포넌트를 선택한 상태에서 키보드로 'Ctrl + V' 키를 클릭하거나 마우스 오른쪽 버튼을 클릭하여 붙여넣기를 선택합니다.

■ 삭제하기

컴포넌트를 하나 이상 선택한 상태에서 키보드로 'Delete' 키를 클릭하거나 마우스 오른쪽 버튼을 클릭하여 삭제하기를 선택합니다.

■ 실행 취소

'Ctrl + Z' 키를 클릭하여 바로 전에 실행했던 작업을 취소합니다.

■ 다시 실행

'Ctrl + Y' 키를 클릭하여 실행 취소했던 작업을 다시 복구합니다.

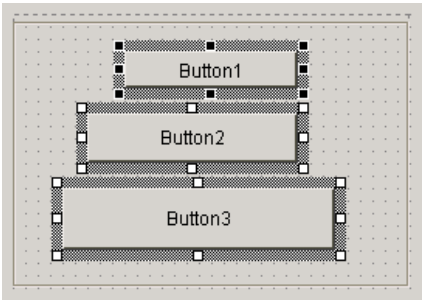
컴포넌트간 레이아웃

■ 위치 정렬

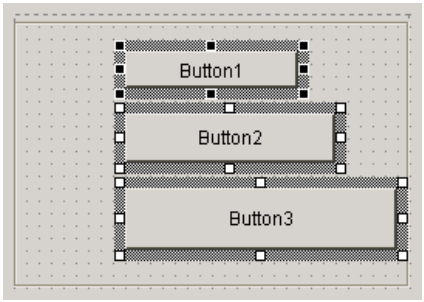
컴포넌트를 둘 이상 선택한 상태에서 툴바의 정렬 아이콘이나 [Layout] 메뉴의 [Align] 메뉴를 이용하여 왼쪽, 오른쪽, 위쪽, 아래쪽으로 컴포넌트들을 위치 정렬합니다.

- 왼쪽 정렬 ( )

맨 마지막에 선택된 컴포넌트의 왼쪽을 기준으로 하여 컴포넌트들을 왼쪽으로 위치 정렬합니다. 아래 그림은 'Button1'을 맨 마지막에 선택한 후 왼쪽 정렬한 모습입니다.



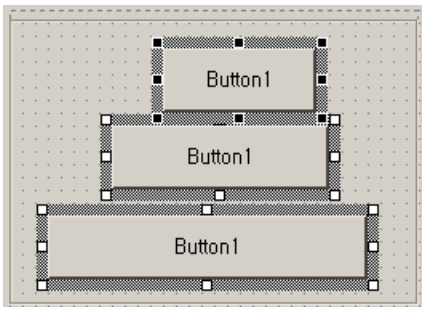
(위치 정렬 전)

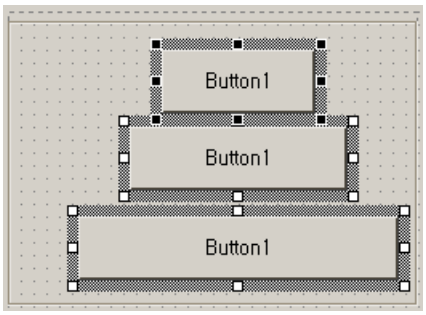


(위치 정렬 후)

- 가운데 정렬 ( )

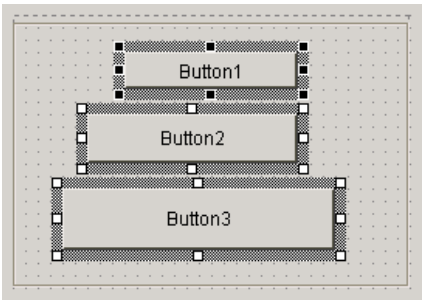
맨 마지막에 선택된 컴포넌트의 가운데를 기준으로 하여 컴포넌트들을 가운데로 위치 정렬합니다. 아래 그림은 'Button1'을 맨 마지막에 선택한 후 가운데 정렬한 모습입니다.

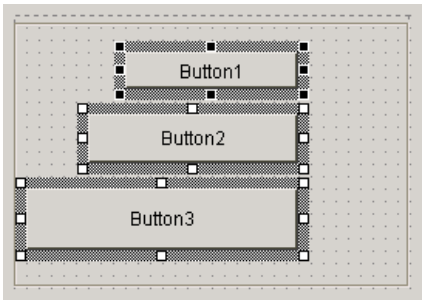




- 오른쪽 정렬 ( )

  맨 마지막에 선택된 컴포넌트의 오른쪽을 기준으로 하여 컴포넌트들을 오른쪽으로 위치 정렬합니다. 아래 그림은 'Button1'을 맨 마지막에 선택한 후 오른쪽 정렬한 모습입니다.

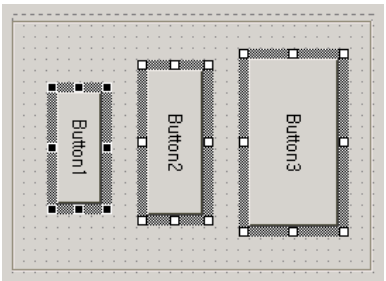


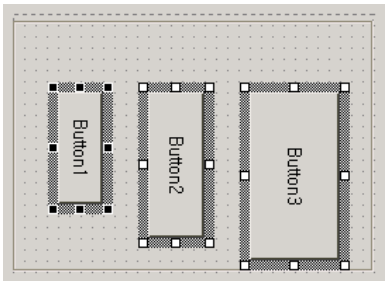


(위치 정렬 전) (위치 정렬 후)

- 위쪽 정렬 ( )

  맨 마지막에 선택된 컴포넌트의 위쪽을 기준으로 하여 컴포넌트들을 위쪽으로 위치 정렬합니다. 아래 그림은 'Button1'을 맨 마지막에 선택한 후 위쪽 정렬한 모습입니다.

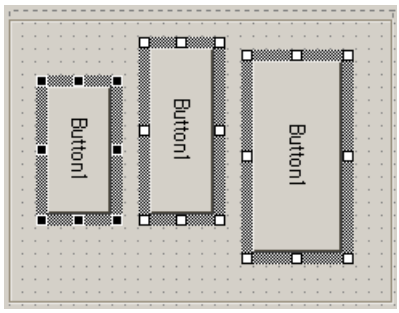




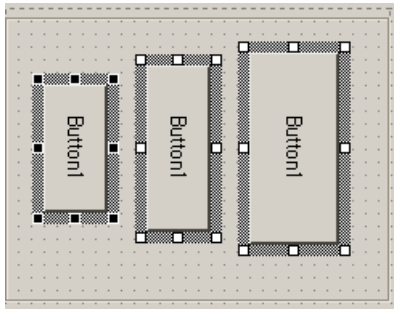
(위치 정렬 전) (위치 정렬 후)

- 중간 정렬 ( )

  맨 마지막에 선택된 컴포넌트의 중간을 기준으로 하여 컴포넌트들을 중간으로 위치 정렬합니다. 아래 그림은 'Button1'을 맨 마지막에 선택한 후 중간 정렬한 모습입니다.



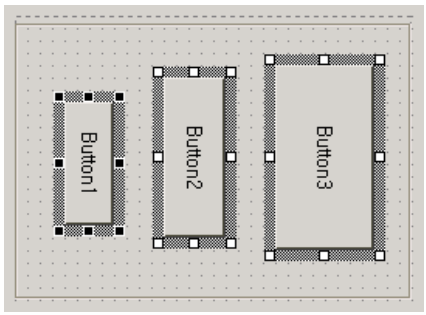
(위치 정렬 전)



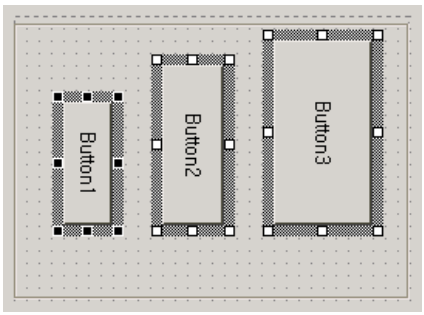
(위치 정렬 후)

- 아래쪽 정렬 ()

  맨 마지막에 선택된 컴포넌트의 아래쪽을 기준으로 하여 컴포넌트들을 아래쪽으로 위치 정렬합니다. 아래 그림은 'Button1'을 맨 마지막에 선택한 후 아래쪽 정렬한 모습입니다.



(위치 정렬 전)



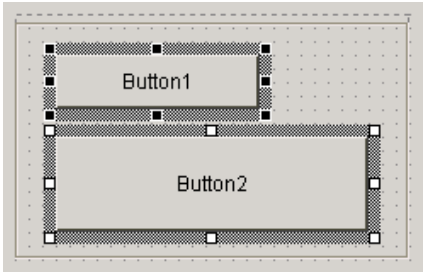
(위치 정렬 후)

■ 크기 동일화

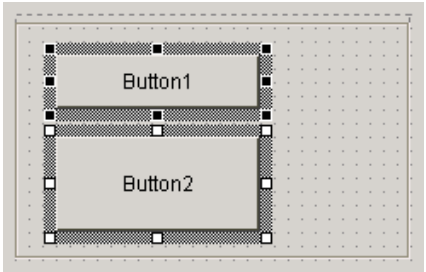
컴포넌트를 둘 이상 선택하여 컴포넌트의 크기를 동일하게 변경합니다.

- 너비 동일화 ()

  맨 마지막에 선택된 컴포넌트의 너비를 기준으로 하여 컴포넌트들을 너비를 동일하게 변경합니다. 아래 그림은 'Button1'을 맨 마지막에 선택한 후 너비를 동일화한 모습입니다.



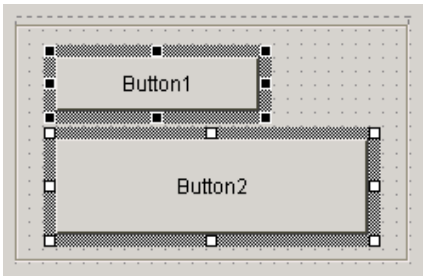
(동일화 전)



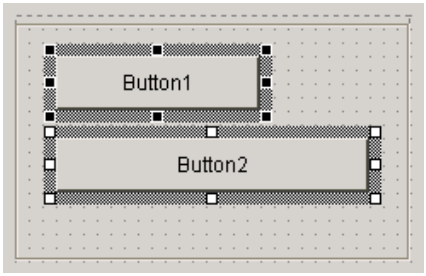
(동일화 후)

- 높이 동일화 ( )

  맨 마지막에 선택된 컴포넌트의 높이를 기준으로 하여 컴포넌트들을 높이를 동일하게 변경합니다. 아래 그림은 'Button1'을 맨 마지막에 선택한 후 높이를 동일화한 모습입니다.



(동일화 전)



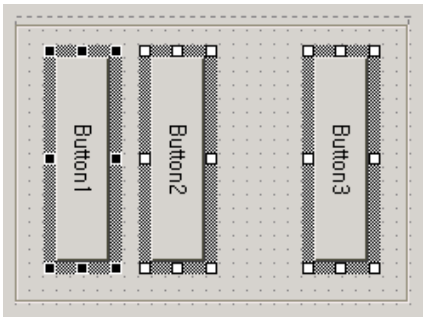
(동일화 후)

■ 간격 동일화

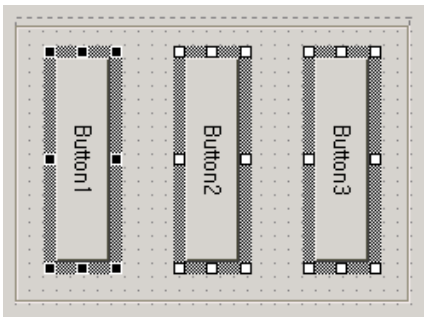
컴포넌트를 셋 이상 선택한 상태에서 툴바의 간격 조정 아이콘이나 [Layout] 메뉴의 [Distribute Spacing Evenly] 메뉴를 이용하여 컴포넌트간의 수평, 수직 간격을 동일하게 조정합니다.

- 수평 간격 동일화 ( )

  선택된 컴포넌트 사이의 수평 간격을 동일하게 조정합니다.



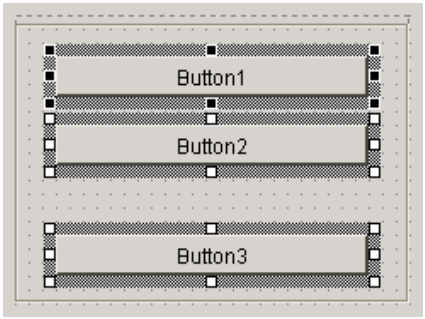
(동일화 전)



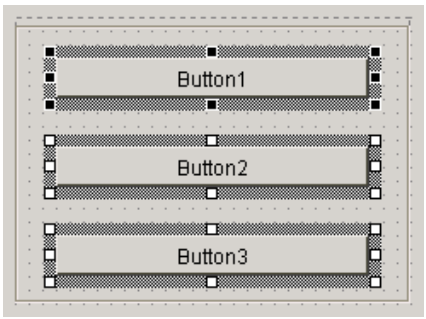
(동일화 후)

- 수직 간격 동일화 ( )

  선택된 컴포넌트 사이의 수직 간격을 동일하게 조정합니다.



(동일화 전)



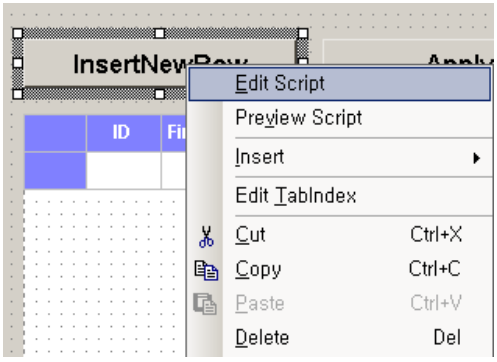
(동일화 후)

스크립트 코딩

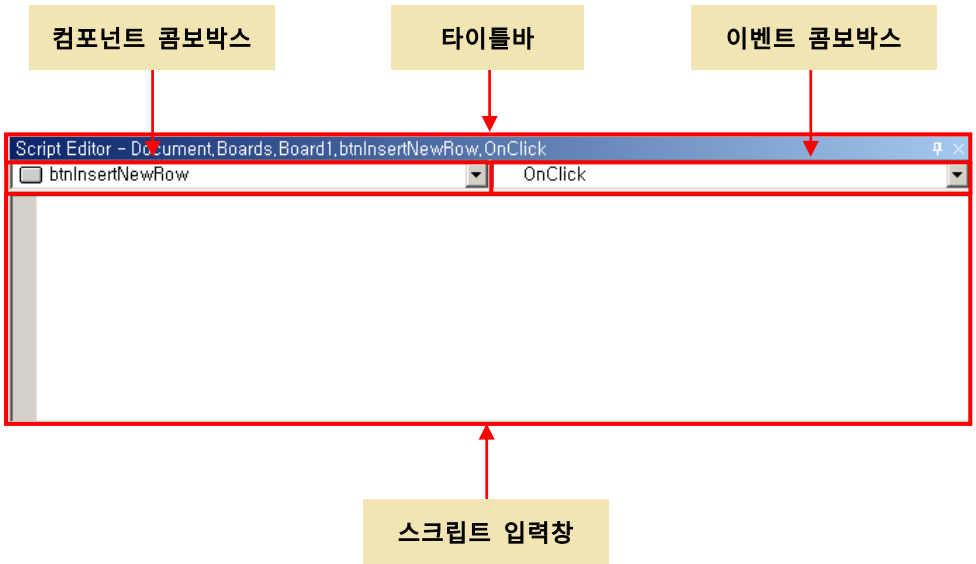
다음은 스크립트 편집창과 스크립트 미리보기창에서 스크립트를 코딩하고, 스크립트 팝업 헬프창을 사용하는 방법 등에 대해 설명합니다.

스크립트 편집창

Board창 또는 컴포넌트 트리창에서 컴포넌트를 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [Edit Script] 메뉴를 클릭하면 스크립트 편집창이 활성화되고, 마우스 커서가 스크립트 편집창에 자동으로 위치하여, 스크립트를 입력할 수 있습니다.



스크립트 편집창은 아래와 같은 화면으로 구성되어 있습니다.

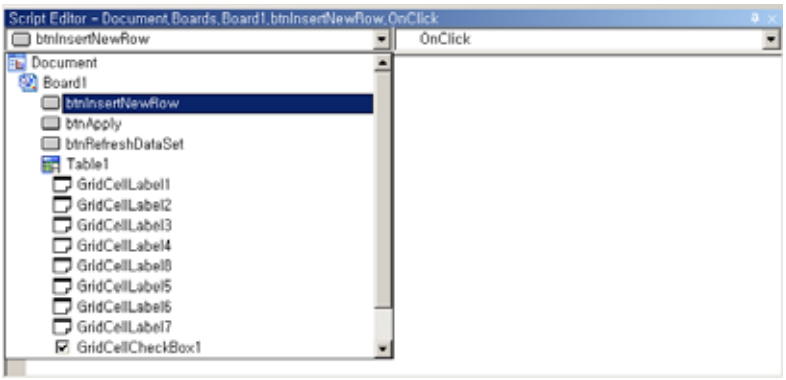


■ 타이틀바

현재 선택된 컴포넌트와 이벤트를 텍스트로 나타냅니다. 위 그림과 같은 경우에는 현재 선택된 컴포넌트는 "Board1"에 있는 "btnInsertNewRow"라는 이름의 컴포넌트이고 선택된 이벤트는 "OnClick"임을 나타냅니다.

■ 컴포넌트 콤보박스

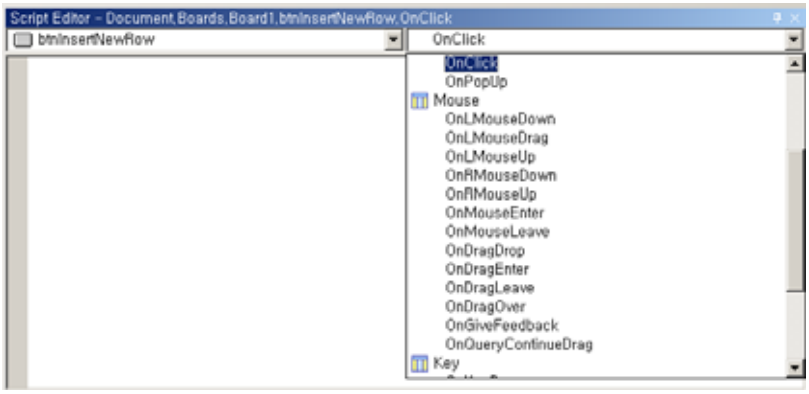
현재 선택된 컴포넌트의 이름을 나타냅니다. 콤보박스를 클릭하면 현재 프로젝트에 추가되어있는 모든 컴포넌트의 이름이 리스트 형태로 나타납니다.



컴포넌트 이름을 클릭하면 컴포넌트 리스트가 닫히면서, 해당 컴포넌트의 스크립트를 입력할 수 있습니다.

■ 이벤트 콤보박스

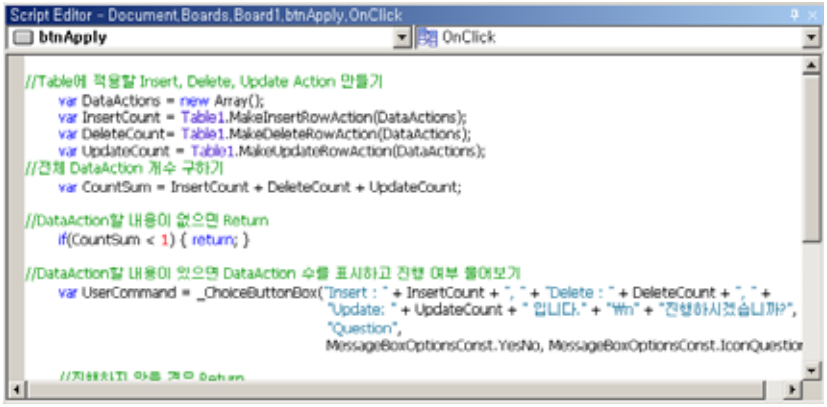
현재 선택된 이벤트를 나타냅니다. 콤보박스를 클릭하면 컴포넌트 콤보박스에 선택된 컴포넌트가 가질 수 있는 모든 이벤트가 리스트 형태로 나타납니다.



이벤트를 클릭하면 이벤트 리스트가 닫히면서, 해당 이벤트의 스크립트를 입력할 수 있습니다.

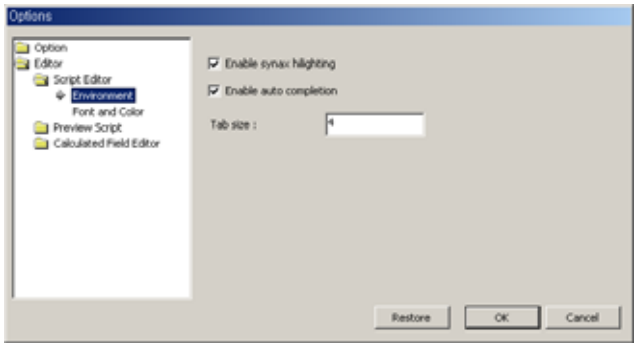
■ 스크립트 입력창

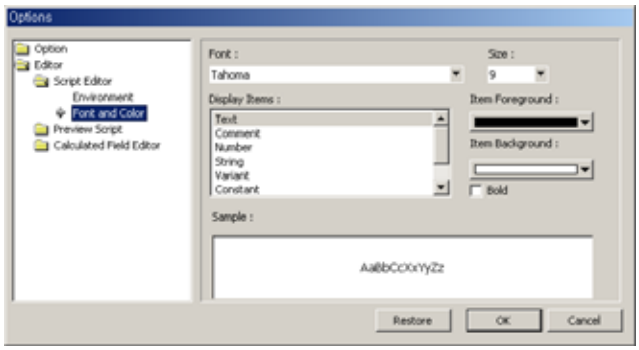
스크립트를 입력할 컴포넌트와 이벤트를 선택한 후 스크립트를 입력합니다.



※ 참고사항 : 스크립트 입력창 환경 설정

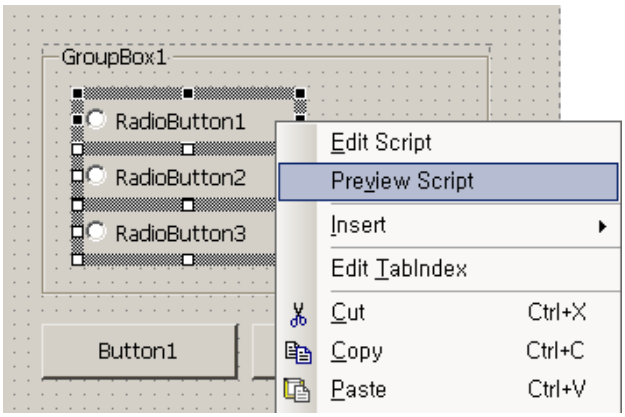
스크립트 입력창에서 구문 강조 기능, 배경색, 글꼴, 글꼴색 등은 [Tool] → [Option] 메뉴를 클릭하여 나타나는 옵션 다이얼로그에서 설정하시기 바랍니다.



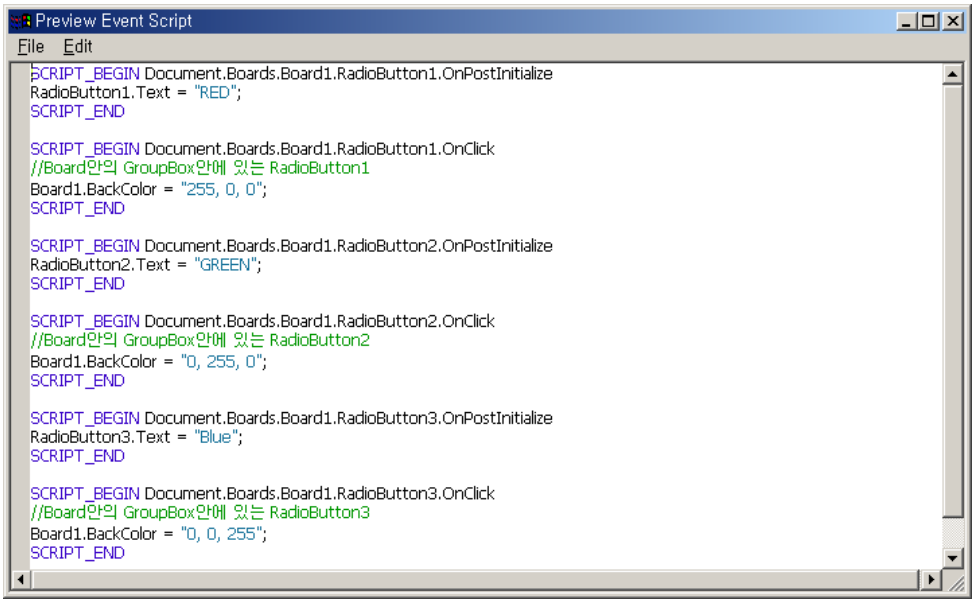


스크립트 미리보기창

Board창 또는 컴포넌트 트리창에서 컴포넌트를 한 개 또는 여러 개 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [Preview Script] 메뉴를 클릭하면 선택한 컴포넌트에 입력되어 있는 스크립트를 나타내는 스크립트 미리보기창이 화면에 나타납니다.

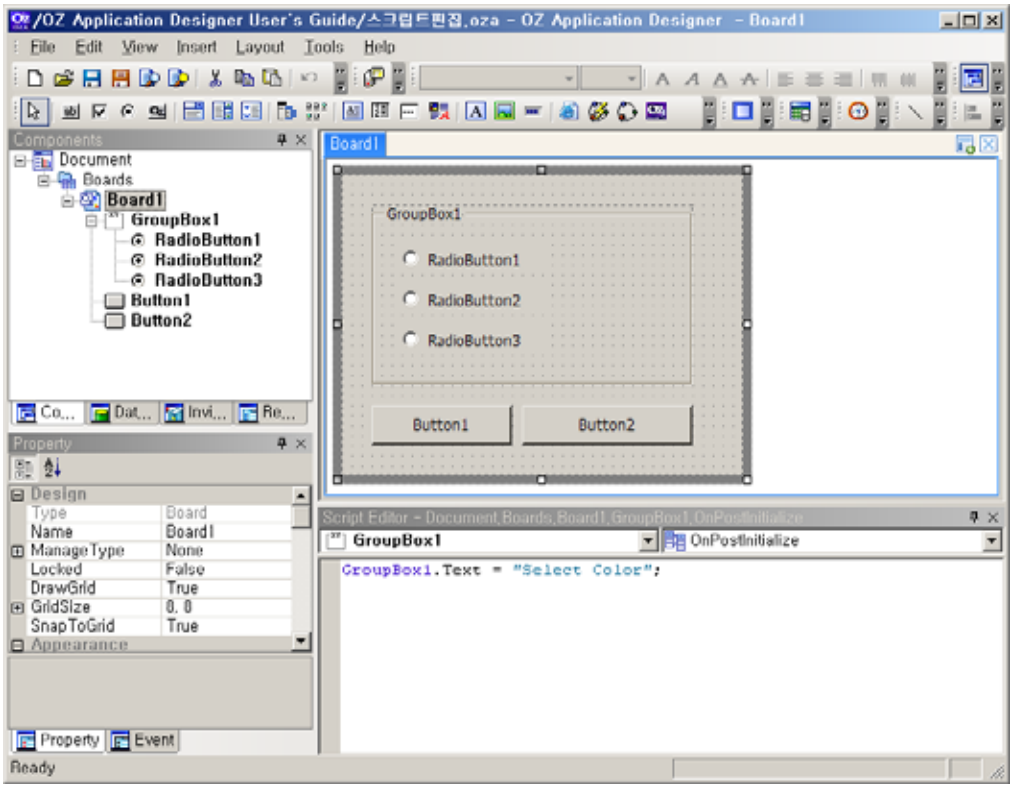


스크립트 미리보기창은 아래 그림과 같은 구조이며, 스크립트 미리보기창의 메뉴바와 사용법은 본 매뉴얼의 메뉴바 설명 부분의 [Edit] 메뉴의 [Preview Script] 부분을 참조하시기 바랍니다.

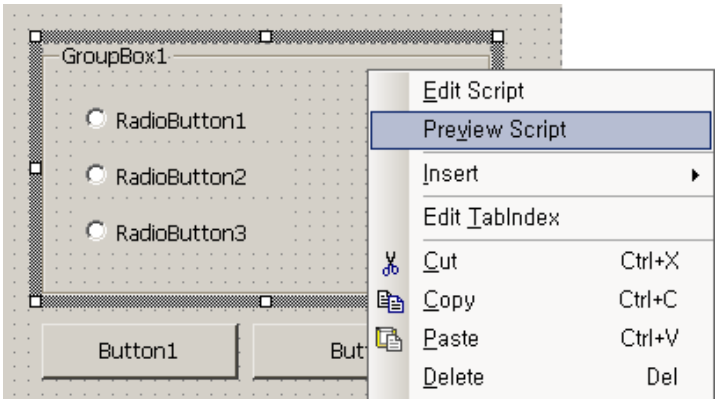


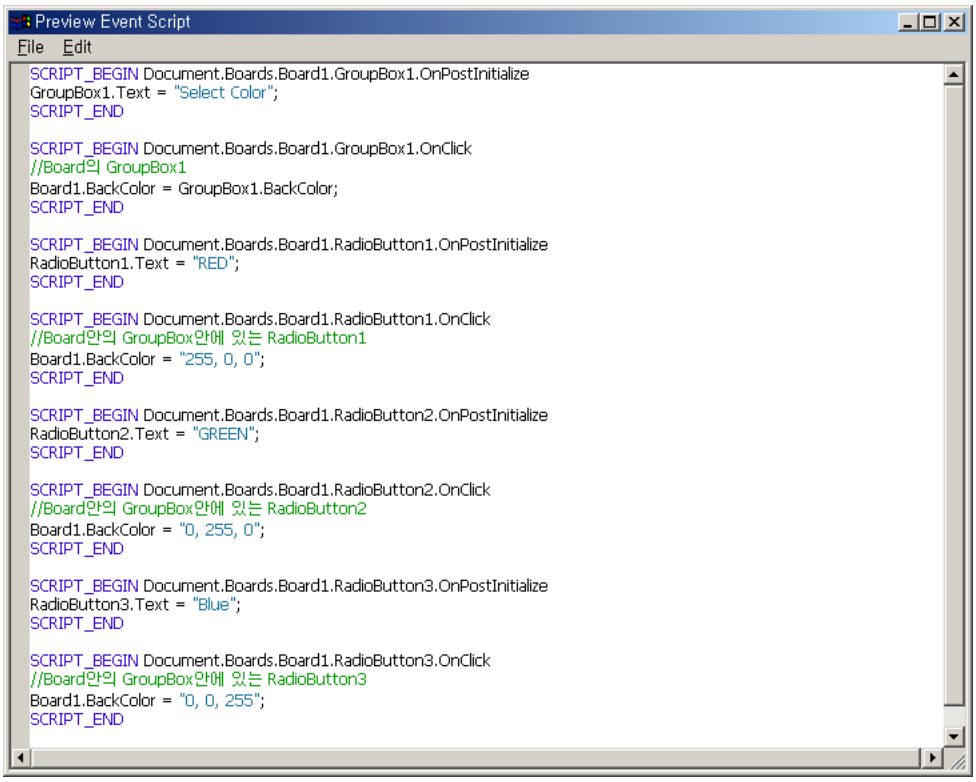
스크립트 미리보기창은 컴포넌트를 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 [Preview Script] 메뉴를 클릭하여 표시하거나 메뉴바에서 [Edit] 메뉴의 [Preview Script] 메뉴를 클릭하여 나타낼 수 있습니다. 팝업 메뉴를 클릭하여 나타나는 스크립트 미리보기창은 선택된 컴포넌트와 자식 컴포넌트에 입력되어있는 스크립트를 나타내고, 메뉴바를 클릭하여 나타나는 스

크립트 미리보기창은 프로젝트에 추가되어있는 모든 컴포넌트에 입력된 스크립트를 나타냅니다. 예를 들어 아래와 같은 프로젝트의 경우

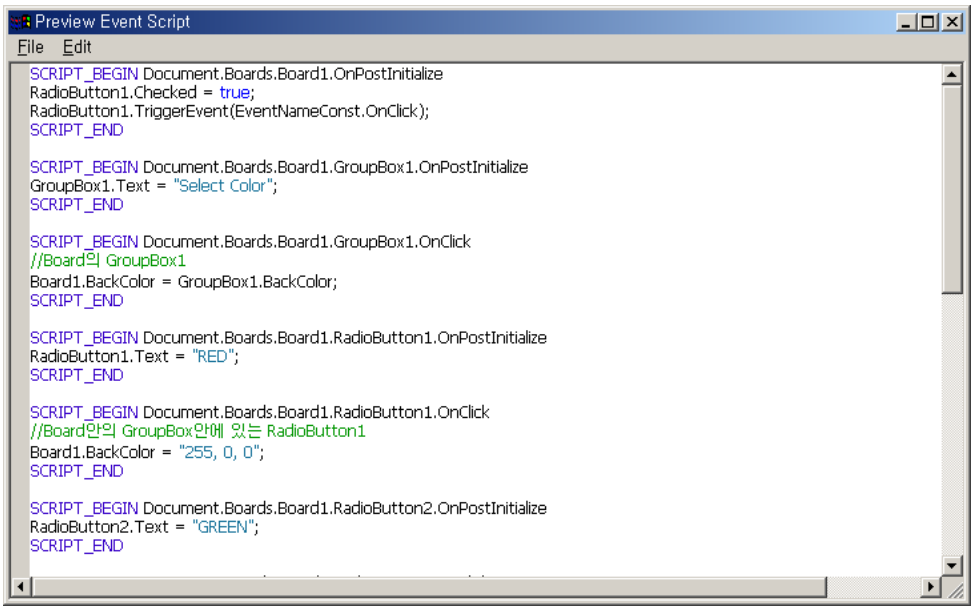


GroupBox1을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Preview Script] 메뉴를 클릭하여 나타나는 스크립트 미리보기창은 GroupBox1와 GroupBox1의 자식 컴포넌트인 RadioButton1, RadioButton2, RadioButton3에 입력되어있는 스크립트를 표시합니다.





메뉴바에서 [Edit] 메뉴의 [Preview Script] 메뉴를 클릭하여 나타나는 스크립트 미리보기창은 프로젝트에 추가되어있는 모든 컴포넌트에 입력된 스크립트를 표시합니다.

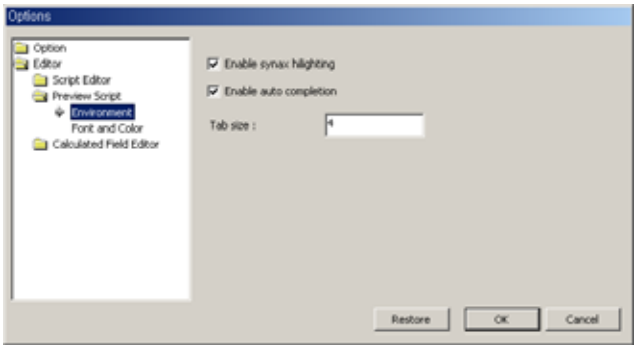


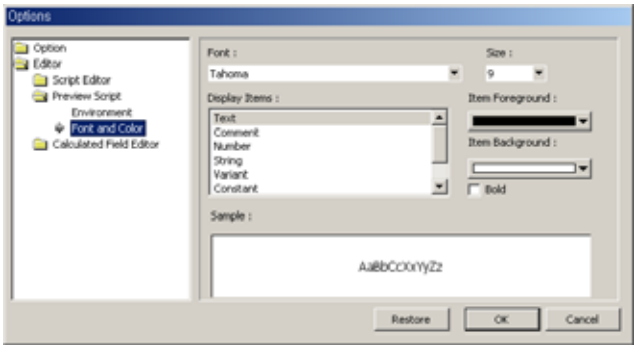
오즈 애플리케이션에서는 프로젝트의 최상위 개념이 Document이므로, Document를 마우스 오

른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Preview Script] 메뉴를 클릭하면 메뉴바에서 [Edit] 메뉴의 [Preview Script] 메뉴를 클릭한 것과 동일한 결과를 얻을 수 있습니다.

※ 참고사항 : 스크립트 미리보기창 환경 설정

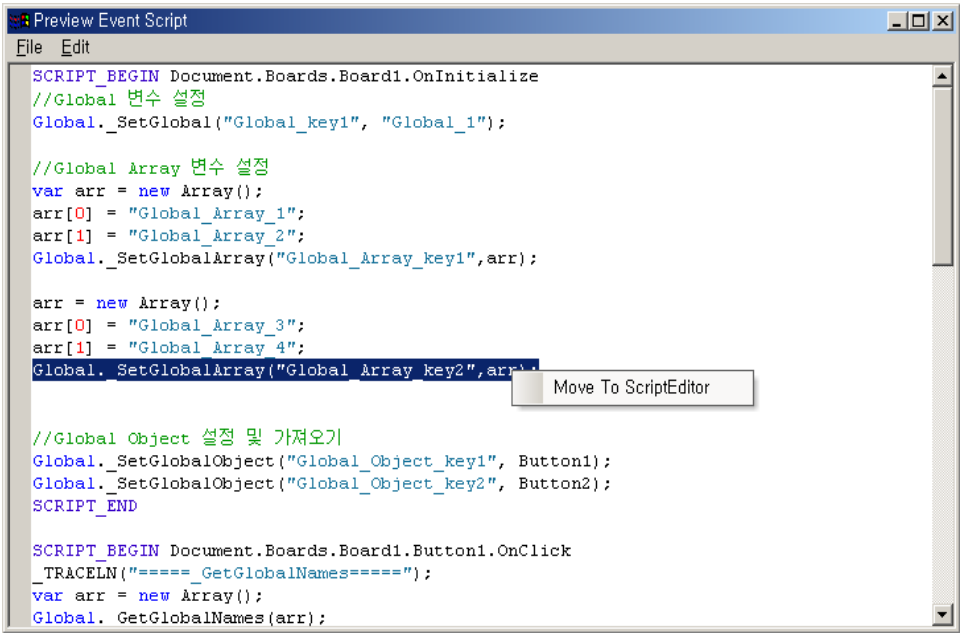
스크립트 미리보기창에서 구문 강조 기능, 배경색, 글꼴, 글꼴색 등은 [Tool] → [Option] 메뉴를 클릭하여 나타나는 옵션 다이얼로그에서 설정하시기 바랍니다.





스크립트 미리보기창 팝업 메뉴

스크립트 미리보기창에서 마우스 오른쪽 버튼을 누르면 스크립트 미리보기 팝업 메뉴가 표시됩니다.



팝업 메뉴에 대한 간략한 설명은 다음과 같습니다.

Menu	Description
Mone To ScriptEditor	선택한 스크립트가 포함된 스크립트 편집창으로 이동합니다.

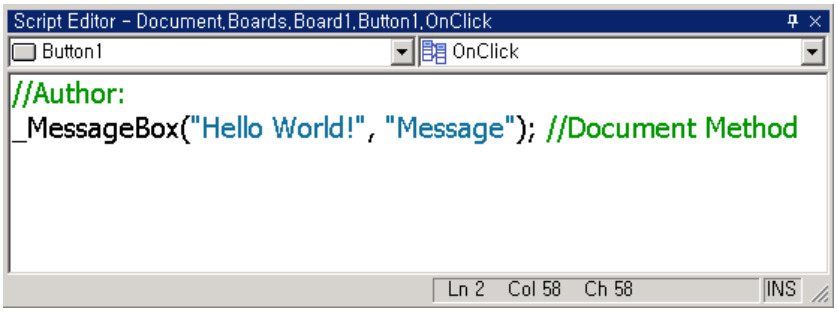
주석 설정

주석 문자를 직접 입력하거나 핫키를 이용하여 한 줄 또는 여러 줄을 주석으로 설정합니다.

■ 주석 문자 직접 입력

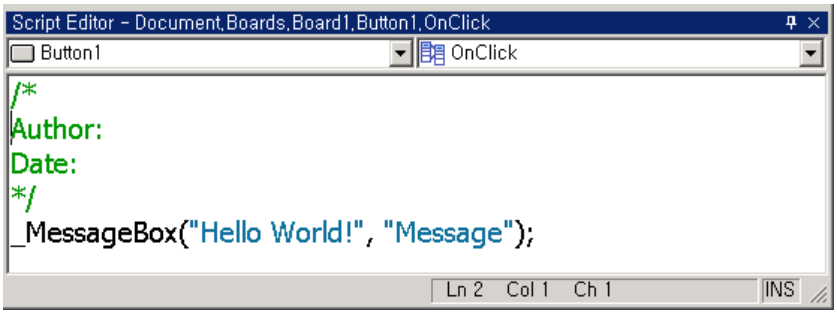
- 한 줄 주석

  주석으로 처리할 스크립트 앞에 '//'을 입력하며, '//' 뒤의 스크립트는 주석으로 처리되어 실행되지 않습니다.



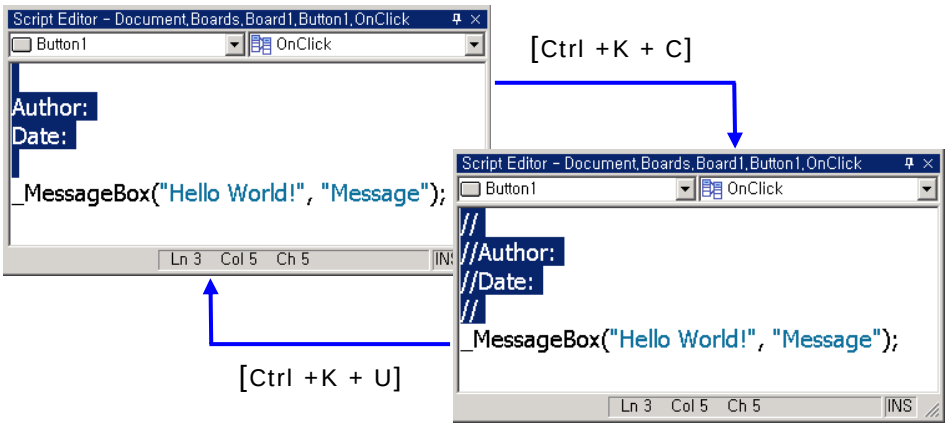
- 여러 줄 주석

  주석으로 처리할 스크립트를 '/*'와 '*/'로 묶어서 여러 줄을 주석으로 처리합니다.



■ 핫키 이용

주석으로 처리할 스크립트를 선택한 후 'Ctrl + K + C' 키를 눌러 주석으로 설정하고, 'Ctrl + K + U' 키를 눌러 주석을 해제합니다.



스크립트 팝업 헬프창

스크립트 팝업 헬프창은 OZ Application Designer의 모든 속성, 메소드, 상수, 프로젝트에 추가된 컴포넌트명을 리스트 형태로 나타낸 팝업창으로, 자동 완성 기능으로 스크립트 작성을 쉽게 할 수 있도록 도와 줍니다.

■ 전체 보기

스크립트창에서 빈 문자 다음에 'Ctrl + Space' 키를 클릭하면 스크립트 팝업 헬프창이 나타납니다. 이 때 나타난 스크립트 팝업 헬프창에는 OZ Application Designer의 모든 속성, 메소드, 상수, 프로젝트에 추가된 모든 컴포넌트명이 아이템으로 표시됩니다.



스크립트 팝업 헬프창이 뜬 상태에서 입력할 속성, 메소드, 상수, 컴포넌트명을 입력하면 입력한 문자열에 해당하는 아이템으로 자동으로 포커스가 이동하게 되며, 이 때 'Tab' 키를 누르거나 마우스로 더블 클릭하여 자동 입력합니다.



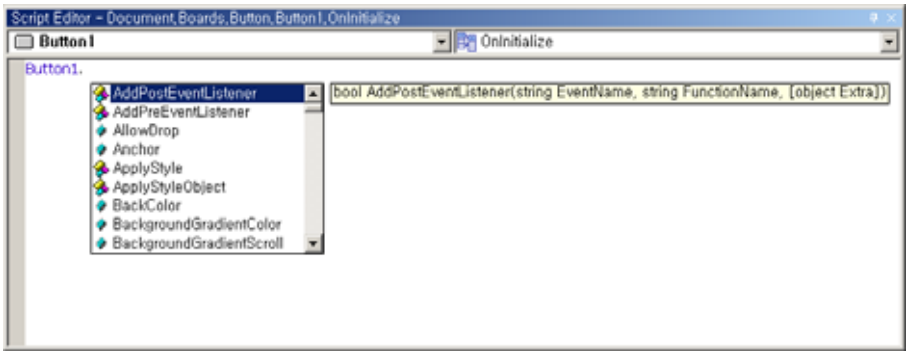
스크립트 팝업 헬프창 각각의 아이템 앞에는 해당 아이템이 어떤 내용인지 나타내는 아이콘이 있어서 해당 아이템이 속성인지 메소드인지 상수인지 컴포넌트인지를 쉽게 식별할 수 있습니다.

각각의 아이콘에 대한 간략한 설명은 다음과 같습니다.

Icon	Description
	메소드
	속성
	상수
	Document
	Board
컴포넌트 고유 아이콘	기타 컴포넌트

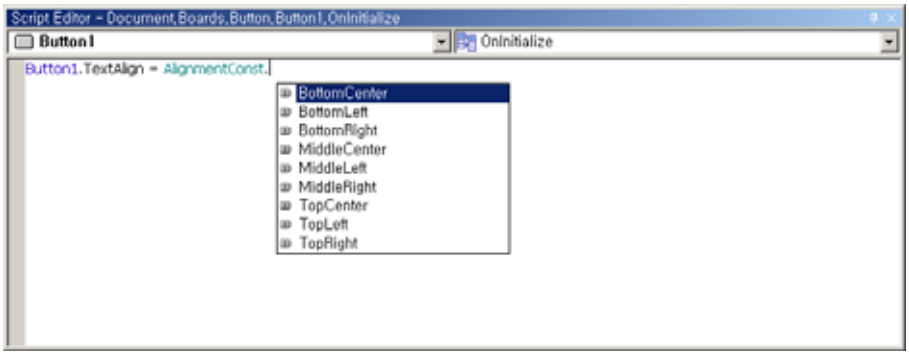
■ 컴포넌트별 속성, 메소드 보기

컴포넌트명을 입력하고 '.'를 입력하면 해당 컴포넌트가 가질 수 있는 속성, 메소드를 아이템으로 나타낸 스크립트 팝업 헬프창이 자동으로 표시됩니다.



■ 상수값 보기

상수명을 입력하고 '.'를 입력하면 해당 상수가 가질 수 있는 값을 아이템으로 나타낸 스크립트 팝업 헬프창이 자동으로 표시됩니다.



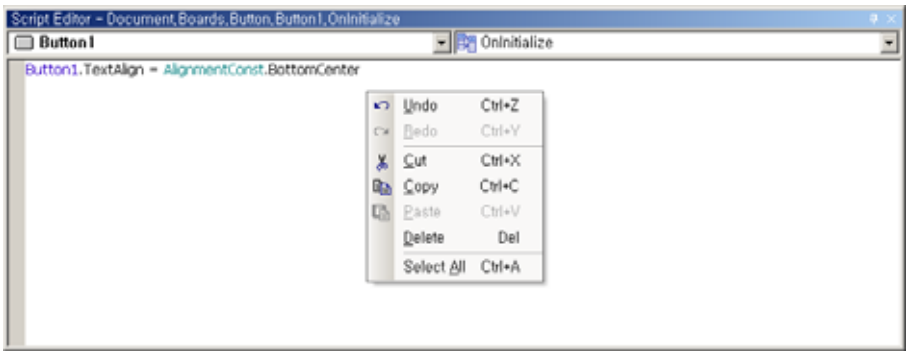
■ 스크립트 자동 완성 규칙

스크립트 창에 스크립트 입력 시 입력한 문자열에 따라 메소드, 속성, 상수가 자동 완성되어 표시됩니다. 메소드, 속성, 상수가 자동 완성되는 규칙은 다음 표와 같습니다.

동작	자동 완성 규칙
컴포넌트 이름 없이 <Ctrl + Space> 키를 눌렀을 경우	모든 메소드, 속성, 상수가 팝업 헬프 창에 표시됩니다.
컴포넌트 이름 없이 "*문자열*" 입력 후 <Ctrl + Space> 키를 눌렀을 경우	입력한 문자열로 시작되는 모든 메소드, 속성, 상수가 팝업 헬프 창에 자동으로 선택되어 표시됩니다.
"*컴포넌트명*." 입력 후 <Ctrl + Space> 키를 눌렀을 경우	해당 컴포넌트에서 사용할 수 있는 메소드 또는 속성이 팝업 헬프 창에 표시됩니다.
"*컴포넌트명.문자열*" 입력 후 <Ctrl + Space> 키를 눌렀을 경우	해당 컴포넌트에서 사용할 수 있는 메소드 또는 속성 중 입력한 문자열로 시작되는 메소드 또는 속성이 팝업 헬프 창에 자동으로 선택되어 표시됩니다. 만일, 입력한 문자열로 시작되는 메소드 또는 속성이 유일한 경우에는 팝업 헬프 창이 뜨지 않고 메소드 또는 속성의 전체 이름으로 입력됩니다.
"*상수명*." 입력 후 <Ctrl + Space> 키를 눌렀을 경우	해당 상수 값이 팝업 헬프 창에 표시됩니다.
"*상수명.문자열*" 입력 후 <Ctrl + Space> 키를 눌렀을 경우	해당 상수 값 중 입력한 문자열로 시작되는 상수 값이 팝업 헬프 창에 자동으로 선택되어 표시됩니다. 만일, 입력한 문자열로 시작되는 상수 값이 유일한 경우에는 팝업 헬프 창이 뜨지 않고 상수 값의 전체 이름으로 입력됩니다.

스크립트창 팝업 메뉴

스크립트창에서 마우스 오른쪽 버튼을 누르면 스크립트 팝업 메뉴가 표시됩니다.



팝업 메뉴에 대한 간략한 설명은 다음과 같습니다.

Menu	Description
Undo(Ctrl+Z)	바로 전에 실행했던 작업을 취소합니다.
Redo(Ctrl+Y)	실행 취소했던 작업을 복구합니다.
Cut(Ctrl+X)	선택한 스크립트를 클립보드에 복사하고 스크립트창에서는 삭제합니다.
Copy(Ctrl+C)	선택한 스크립트를 클립보드에 복사합니다.
Paste(Ctrl+V)	클립보드에 복사된 스크립트를 삽입합니다.
Delete(Delete)	선택한 스크립트를 삭제합니다.
Select All(Ctrl+A)	현재 스크립트 창의 전체 내용을 선택합니다.

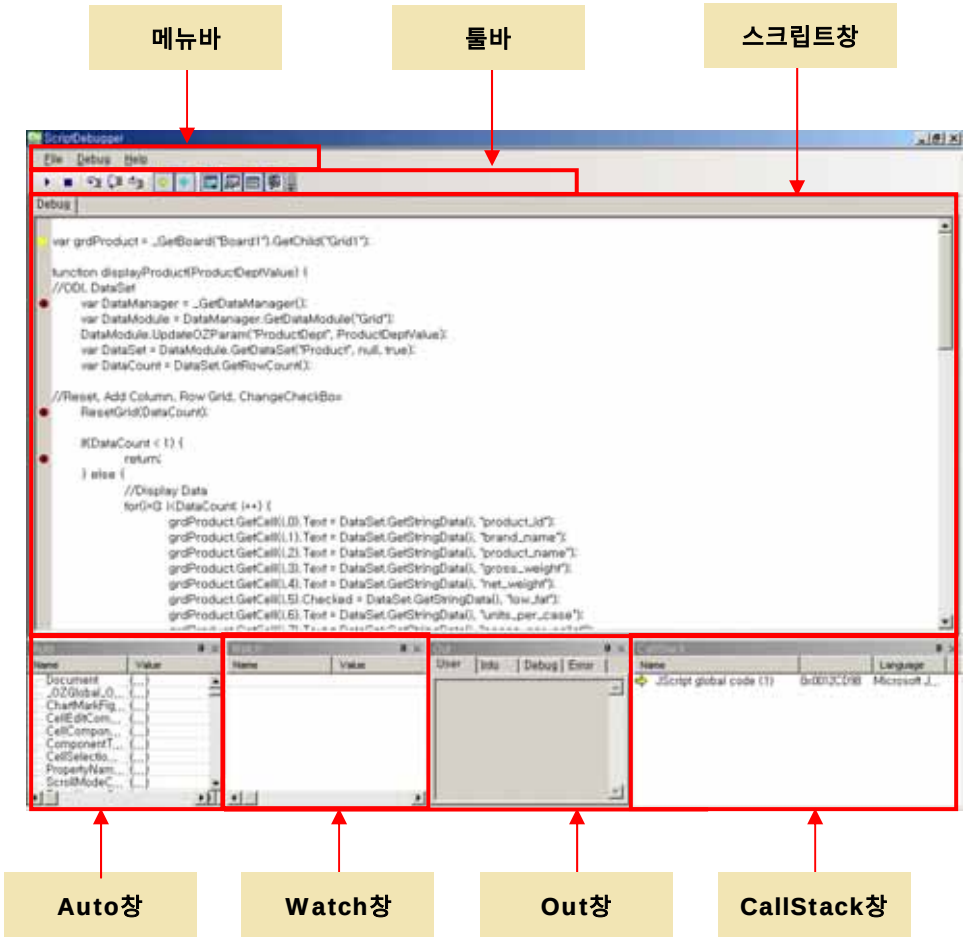
스크립트 디버깅

다음은 스크립트 디버깅창의 화면 구성과 BreakPoint 사용 방법 및 제약사항에 대해 설명합니다.

스크립트 디버깅창 화면 구성

오즈 애플리케이션 디자이너를 실행하여 스크립트가 입력되어있는 OZA 파일을 연 후 메뉴바에서 [File] → [Start Debug] 메뉴 또는 툴바에서 디버깅하기 아이콘( )을 클릭하여 스크립트 디버깅창을 띄웁니다.

OZ Application에서는 디버깅창이 별도의 실행 화면으로 화면에 나타나며, 스크립트 디버깅창의 화면 구성은 아래 그림과 같습니다.



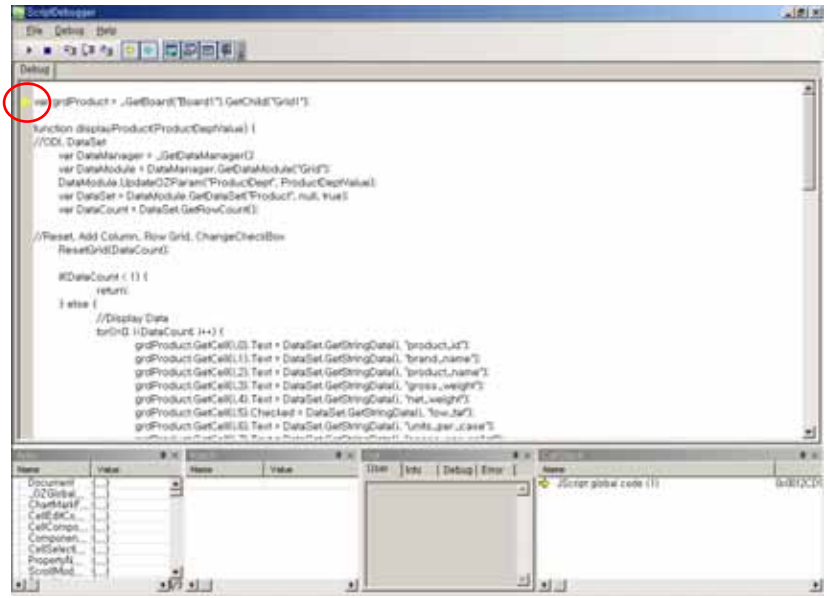
■ 메뉴 및 툴바

Icon	Menu	Description
	**Continue**	디버깅 상태를 마치고, 이후 스크립트를 모두 실행합니다.
	**Stop Debugging**	디버깅을 정지합니다. 이후 스크립트는 실행되지 않습니다.
	**Step Into**	한 단계씩 스크립트를 들어갑니다. 함수일 경우에는 해당 함수 내의 스크립트 코드로 들어갑니다. 함수가 아닌 경우에는 "Step Over"와 같은 동작입니다.
	**Step Over**	다음 단계의 스크립트로 들어갑니다.
	**Step Out**	현재 함수를 빠져 나옵니다.
	**Function Resume**	스크립트 각 라인마다 디버깅할지 여부를 설정합니다. 해당 아이콘이 선택된 상태에서는 특정 이벤트에 의해 함수가 호출될 때 스크립트의 각 라인마다 디버깅을 하고, 선택을 해제하면 BreakPoint가 있거나 에러가 발생할 경우에만 디버깅됩니다.
	**BreakPoint Resume**	BreakPoint가 설정된 라인마다 디버깅할지 여부를 설정합니다. 해당 아이콘이 선택된 상태에서는 BreakPoint가 설정된 각 라인마다 디버깅을 하고, 선택을 해제하면 BreakPoint가 설정된 라인에는 디버깅을 하지 않고, 에러가 있을 경우에만 디버깅됩니다. 단, BreakPoint가 설정된 라인에 에러가 발생한 경우에는 해당 라인이 디버깅됩니다.
	**Auto창**	Auto창 표시 여부를 설정합니다. Auto창은 현재 함수에 사용되는 변수의 값과 타입을 자동으로 표시하는 창입니다.
	**Watch창**	Watch창 표시 여부를 설정합니다. Watch창에 변수를 입력하는 행을 더블 클릭하여 사용자가 직접 입력한 변수의 값을 보거나 변수에 연산한 결과값을 확인할 때 이용합니다.
	**Out창**	Out창 표시 여부를 설정합니다. Out창은 디버깅시에 발생하는 에러 메시지 등을 표시하는 창입니다. 디버깅 상태에서 미리보기를 할 때 Trace창은 따로 뜨지 않고, 디버깅창의 Out창에 표시됩니다.

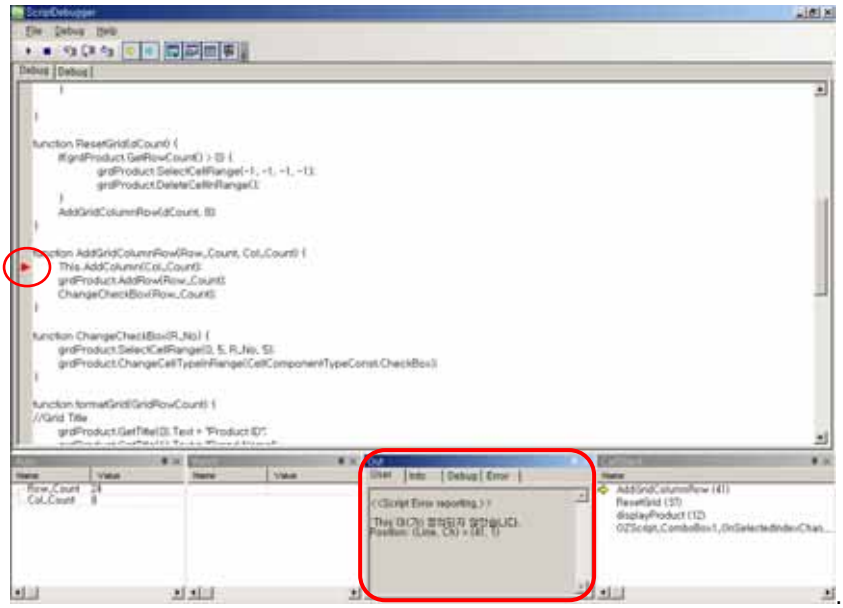
	**CallStack창**	CallStack창 표시 여부를 설정합니다. CallStack창은 이전 호출 스택 정보를 표시하는 창으로 실행중인 함수명과 실행중인 스크립트가 몇 번째 라인인지 등의 정보를 표시합니다.

■ 스크립트창

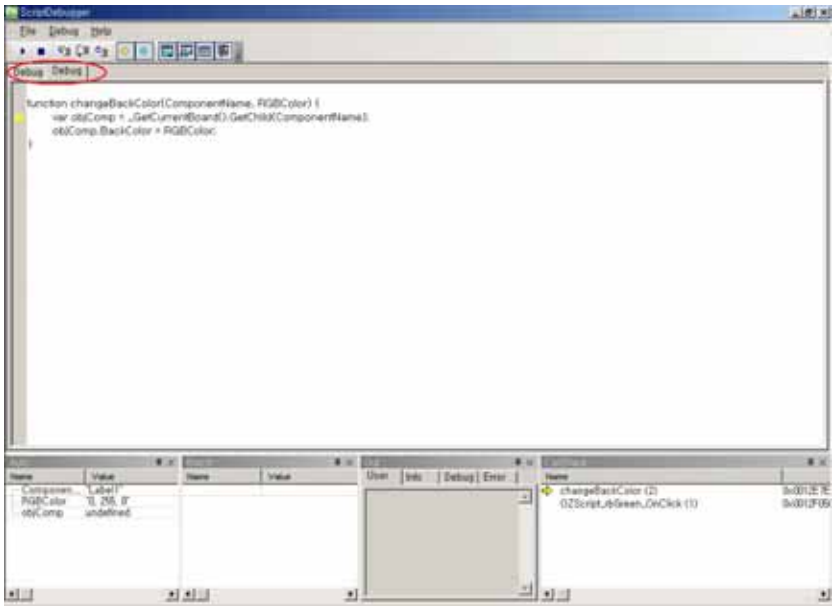
스크립트창에는 현재 실행 중인 함수의 스크립트가 화면에 표시되며, 실행 중인 라인 앞에 노란색 화살표()가 나타납니다.



스크립트 디버깅 중에 에러가 발생하면 스크립트 실행이 일시 정지되며, 에러가 발생한 라인 앞에 빨간색 화살표()가 표시되며, 발생한 에러 메시지는 Out창에 표시됩니다.



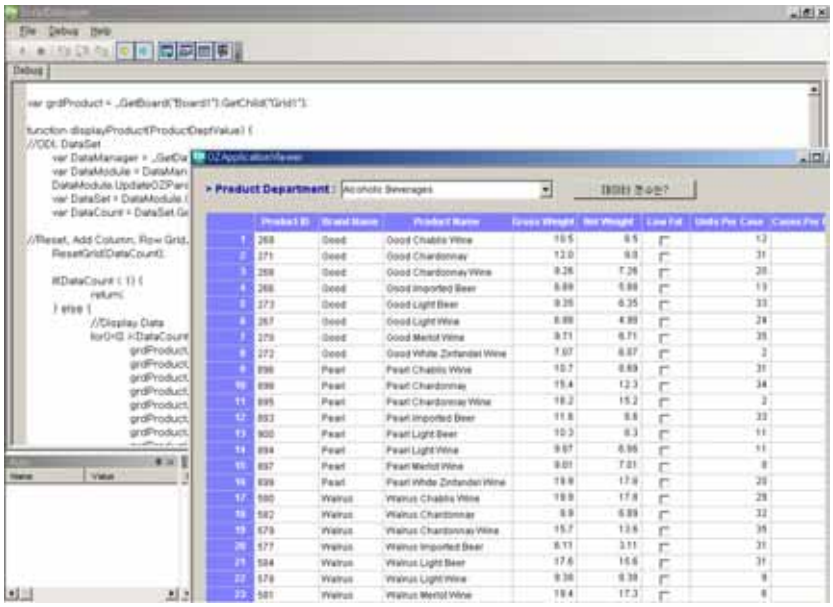
실행 중인 스크립트는 스크립트창에 탭 형태로 표시됩니다. 만일 Step Into 아이콘을 클릭하여 한 단계씩 디버깅할 경우에 디버깅되는 스크립트가 함수에서 또 다른 함수를 호출하는 경우에는 스크립트 탭이 하나 더 추가되어 호출된 함수의 스크립트가 화면에 표시됩니다.



스크립트 실행이 완료되면 스크립트 탭이 자동으로 닫히면서 실행되었던 스크립트가 스크립트창에서 사라집니다.

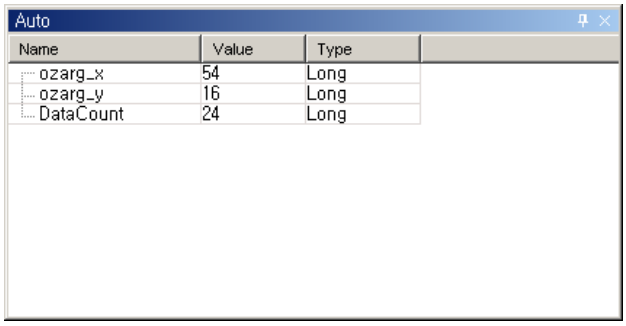
단, 전역 함수 또는 OZF에 입력되어있는 스크립트는 스크립트 실행이 완료된 후에도 스크립트탭이 닫히지 않고, 스크립트창에 그대로 표시됩니다. 아래 그림은 스크립트 실행이 모두

완료되어 뷰어가 활성화된 후에 전역 함수에 입력되어있는 스크립트가 닫히지 않고 그대로 표시되어 있는 모습입니다.



■ Auto창

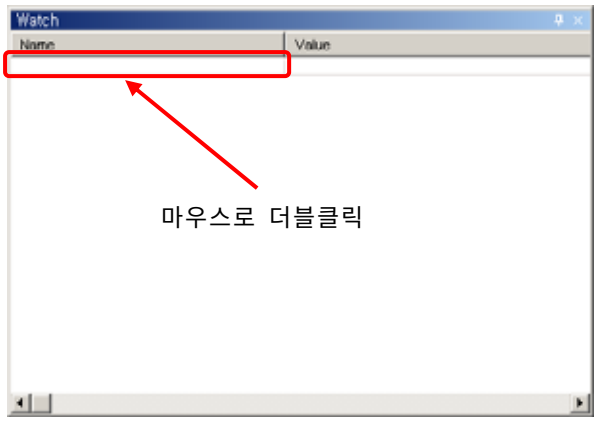
Auto창은 현재 함수에서 사용되는 변수의 값과 타입을 자동으로 표시하는 창입니다.

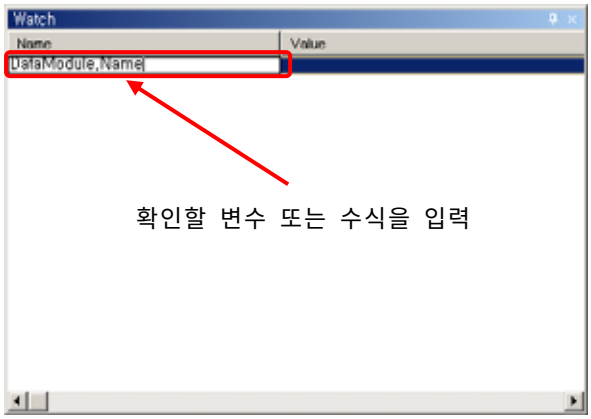


■ Watch창

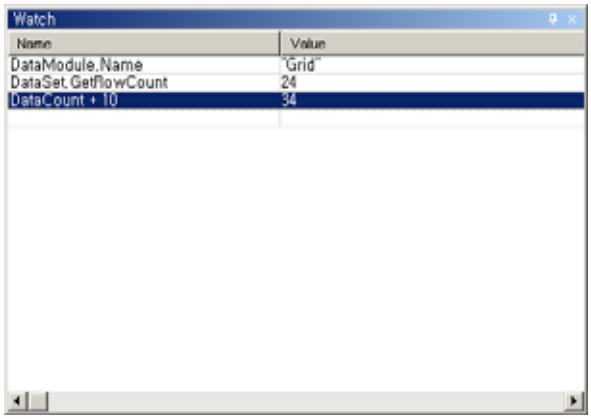
Watch창은 사용자가 직접 입력한 변수의 값을 보거나 변수에 연산한 결과값을 확인할 때 이용합니다.

Watch창의 Name을 나타내는 행을 더블 클릭하여 변수를 직접 입력하거나 수식등을 입력합니다.



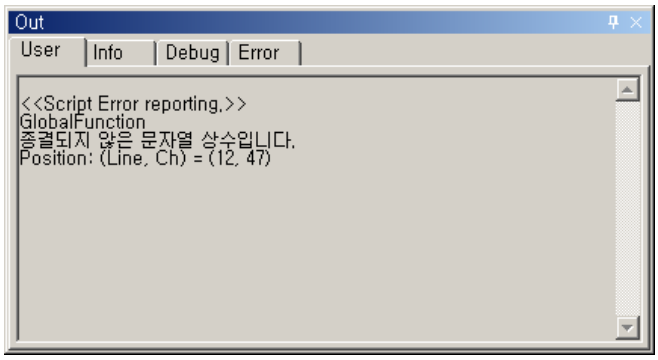


변수 또는 수식을 입력한 후 <Enter>키를 눌러 결과값을 확인할 수 있습니다.

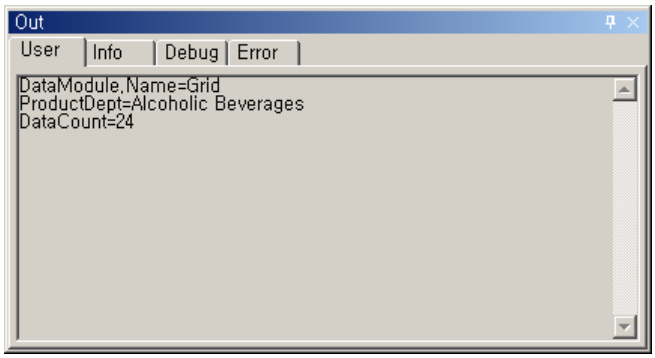


- Out창

  Out창은 디버깅시에 발생하는 에러 메시지를 표시하는 창입니다. 에러 메시지는 Out창의 User 탭에 표시됩니다.



또한 스크립트 입력시에 _Trace() 또는 _TraceLn() 메소드를 이용하여 Trace창에 메시지를 띄고자 할 때 디버깅 상태에서 미리보기를 할 때에는 Trace창이 따로 뜨지 않고, Out창의 User 탭에 표시됩니다.

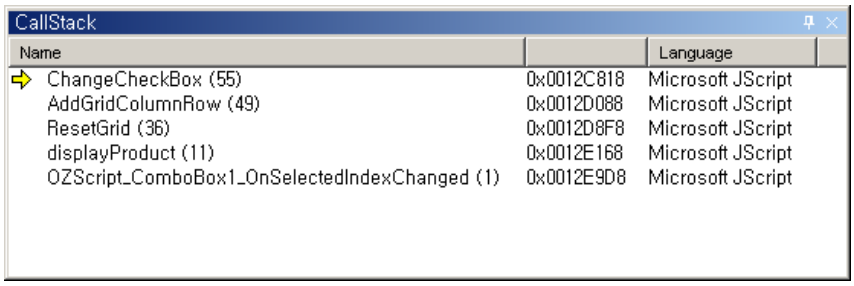


Out창의 Info, Debug, Error 탭은 사용자가 입력한 스크립트에 대한 에러가 아닌 OZ Application의 내부적인 에러 메시지를 표시하기 위한 탭으로, 실제 메시지가 표시되는 일은 거의 없습니다.

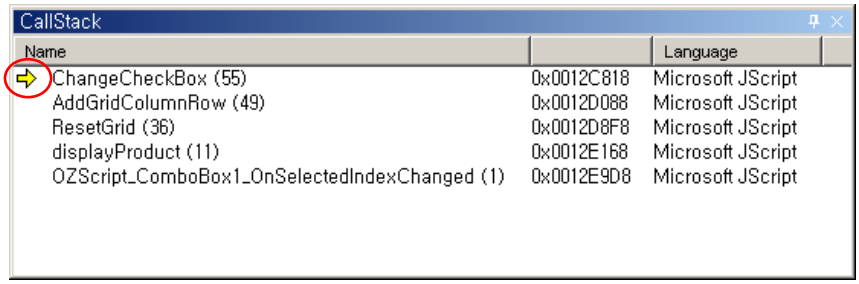
- CallStack창

  CallStack창은 이전 호출 스택 정보를 표시하는 창으로 실행중인 함수명과 실행중인 스크립트가 몇 번째 라인인지 등의 정보를 표시합니다.

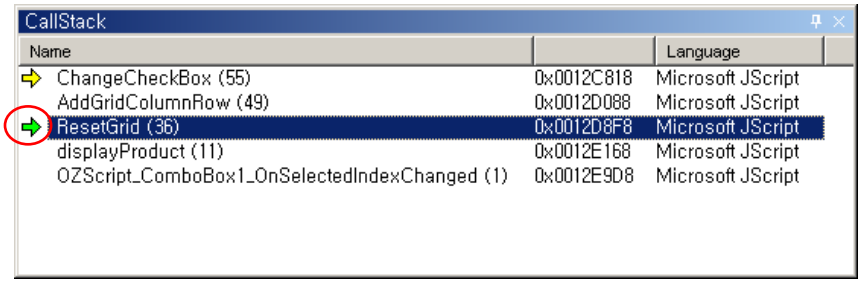
  CallStack창에는 실행중인 함수 리스트가 표시됩니다. 이때 함수명 뒤에 실행중인 스크립트의 라인 번호가 표시되는데, 표시되는 라인 번호는 스크립트창에 표시된 스크립트를 기준으로 한 라인 번호입니다.



CallStack창에 표시된 함수 리스트 중 현재 디버깅 상태인 함수명 앞에는 노란색 화살표 ()가 표시됩니다.



함수 리스트에 표시된 함수명을 마우스로 더블 클릭하면 해당 함수의 스크립트가 스크립트 창에 표시되고, CallStack창에는 해당 함수명 앞에 초록색 화살표가 표시됩니다.



BreakPoint 사용 방법

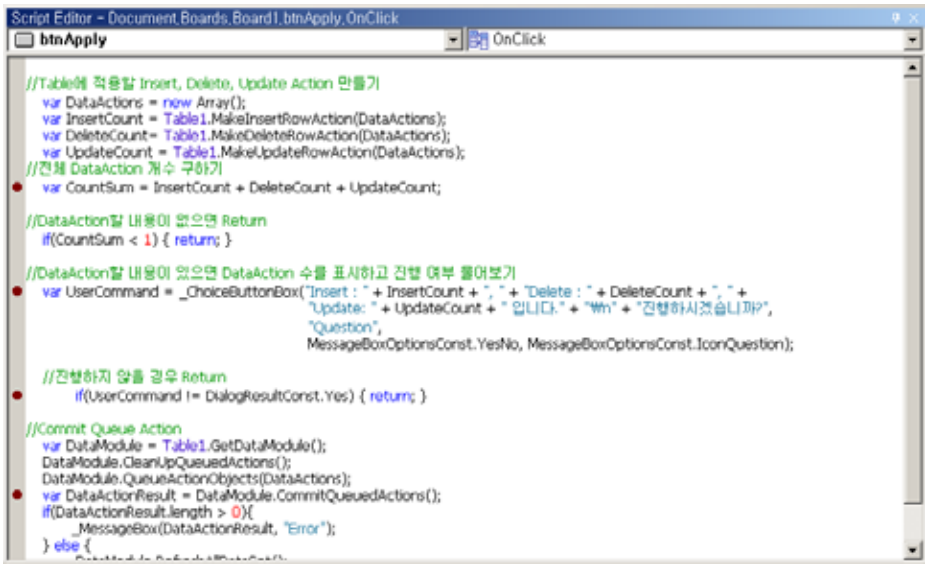
OZ Application에서는 BreakPoint를 사용하여 디버깅할 수 있습니다.

BreakPoint를 설정할 라인에서 키보드의 <F9>키를 누르면 BreakPoint가 설정되고, BrerakPoint가 설정된 라인에서 다시 <F9>키를 누르면 BreakPoint가 해제됩니다.

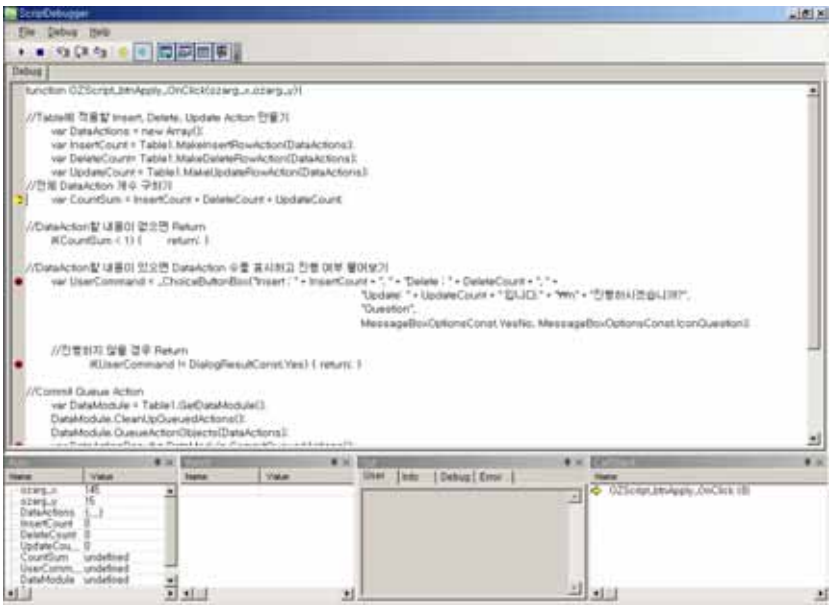
BreakPoint는 OZ Application Designer의 스크립트 편집창, 스크립트 미리보기창 또는 스크립

트 디버깅창에서 설정할 수 있습니다. 스크립트 편집창과 스크립트 미리보기창에서 설정한 BreakPoint는 프로젝트에 저장되지만, 디버깅창에서 설정한 BreakPoint는 디버깅창이 닫히면 자동으로 해제되고, 프로젝트에 저장되지 않습니다.

아래 그림은 스크립트 편집창에서 BreakPoint를 설정한 모습입니다.



스크립트를 디버깅을 실행하면 BreakPoint가 설정된 곳에 스크립트 실행이 멈추고, 디버깅할 수 있습니다.



※ 참고사항 : BreakPoint를 설정한 상태에서 BreakPoint를 찍은 스크립트 라인에서만 디

버깅을 할 경우에는 툴바에서 Function Resume 아이콘은 선택을 해제하고, BreakPoint Resume 아이콘은 선택된 상태()로 변경하여야 합니다.

스크립트 디버깅 제약사항

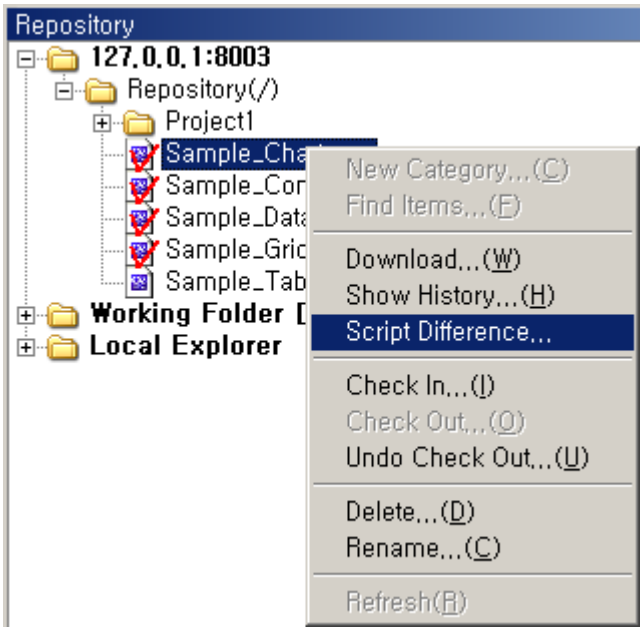
전역 함수에 스크립트가 입력되어 있는 경우 Document의 속성 중 "GlobalFunctionOnceInit" 속성 값이 "False"이면, 전역 함수에 입력된 스크립트는 디버깅할 수 없습니다. 이는 스크립트 디버깅의 제약사항이며, "GlobalFunctionOnceInit" 속성 값을 "True"로 변경하거나, 해당 스크립트를 OZF로 만들어 프로젝트에 추가하여 사용하시기 바랍니다.

스크립트 비교

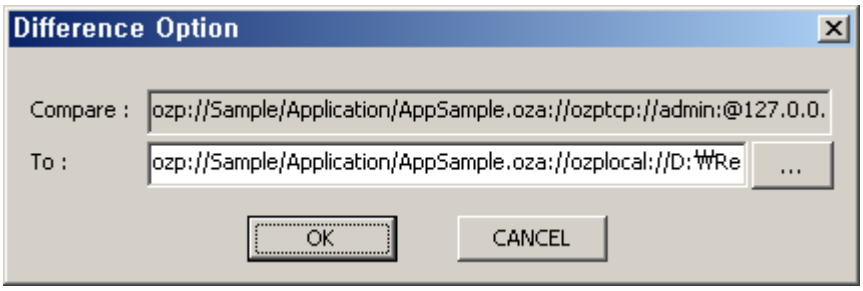
오즈 애플리케이션에서는 OZA, OZF, OZC 파일에 입력된 스크립트를 비교할 수 있는 기능을 제공합니다.

스크립트 비교 – 서버 파일과 로컬 파일 비교

리파지토리창에서 스크립트를 비교할 OZA, OZF, OZC 파일을 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [Script Difference] 메뉴를 클릭합니다.

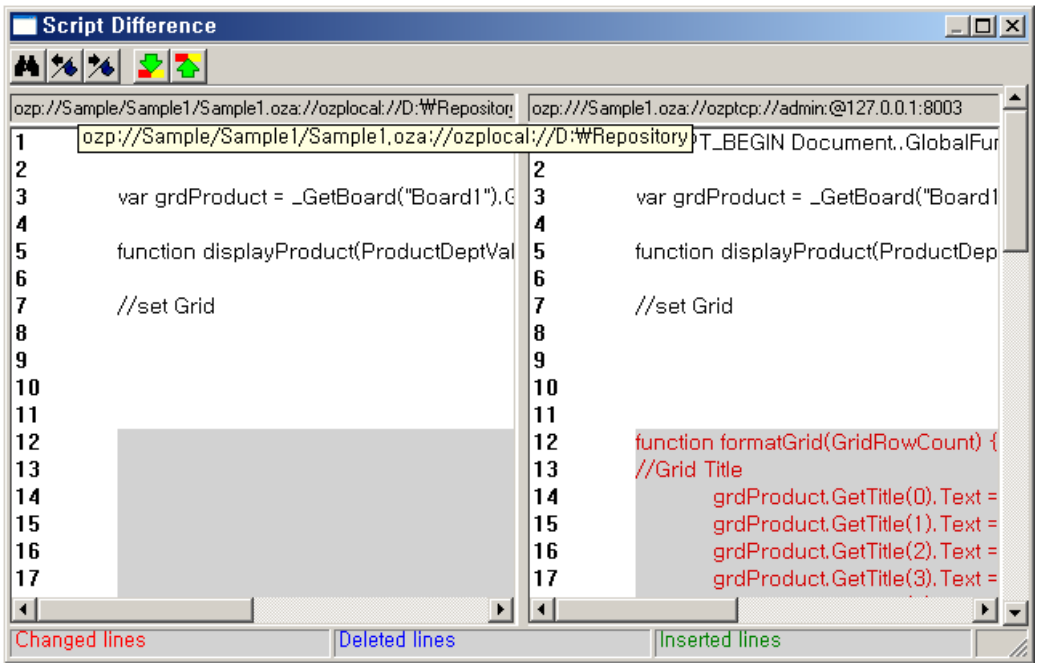


[Script Difference] 메뉴를 클릭하면 원본 파일(Compare)과 대상 파일(To)을 선택할 수 있는 'Difference Option 다이얼로그'가 나타납니다.



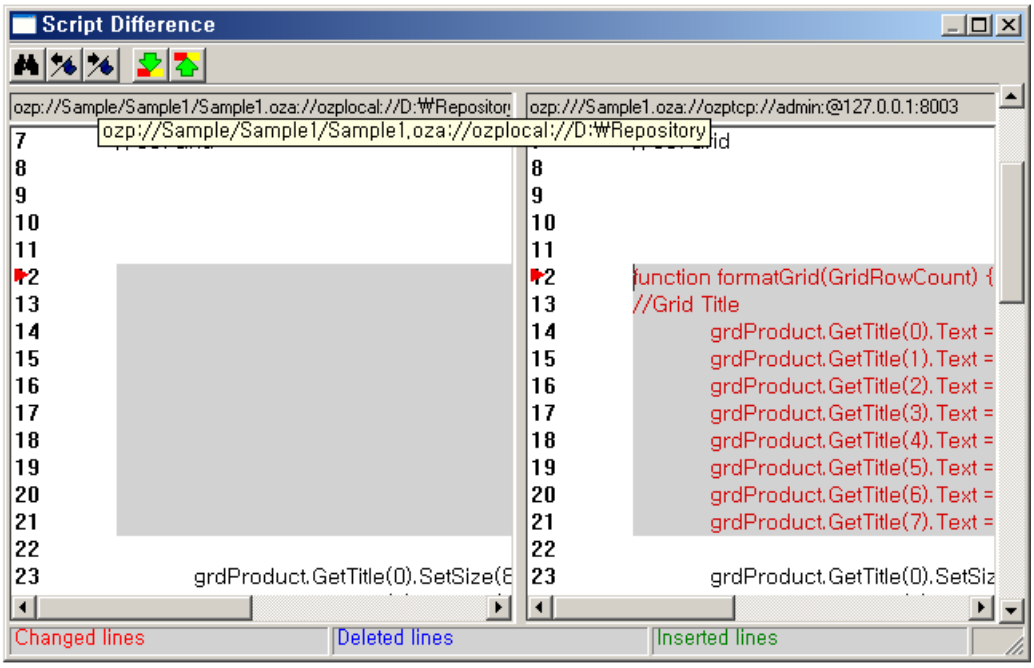
원본 파일(Compare)에는 비교할 원본 파일의 전체 경로를 리파지토리 위치를 포함하여 나타내고, 대상 파일(To)에는 비교할 대상 파일의 전체 경로를 리파지토리 위치를 포함하여 나타냅니다. 리파지토리 서버에 연결한 후 로컬 작업 폴더에 있는 파일을 선택하고 [Script Difference] 메뉴를 클릭하면 서버 리파지토리에서 파일을 찾아 자동으로 대상 파일에 표시해주고, 서버 리파지토리에서 [Script Difference] 메뉴를 클릭하면 로컬 작업 폴더에서 파일을 찾아 자동으로 대상 파일에 표시해줍니다. 파일을 찾지 못했을 경우 대상 파일은 빈 항목으로 표시됩니다.
대상 파일은 [...] 버튼을 클릭하여 변경할 수 있으며, 원본 파일과 같은 확장자의 파일만 선택할 수 있습니다.

스크립트 비교 옵션을 선택한 후 [확인] 버튼을 클릭하면 스크립트 비교 다이얼로그가 실행됩니다.



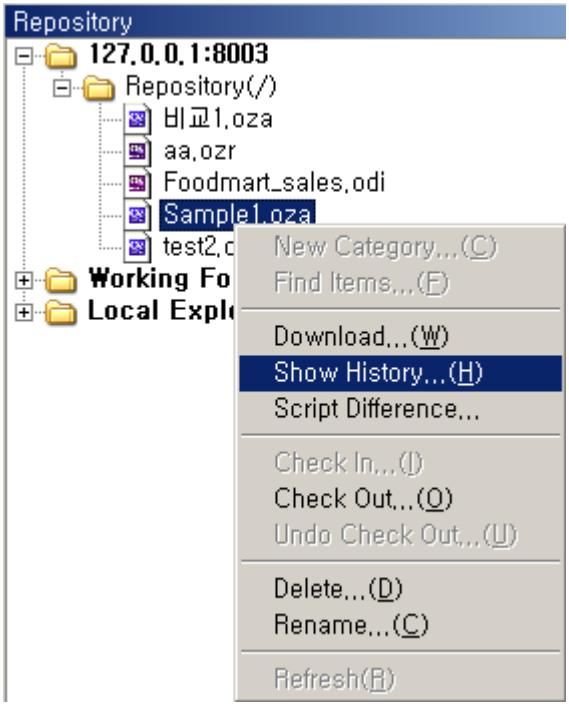
스크립트 비교 시 원본 파일에만 스크립트가 있을 경우 파란색으로, 대상 파일에만 스크립트가 있을 경우 초록색으로, 같은 라인이 다를 경우 빨간색 글씨로 표시됩니다.

찾기(), 이전 찾기(), 다음 찾기() 아이콘을 클릭하여 텍스트를 검색할 수 있으며, 다음 차이점 이동(), 이전 차이점 이동() 아이콘을 클릭하여 스크립트가 다른 라인으로 커서를 이동할 수 있습니다.
차이점을 이동하면 해당 라인에 빨간색 화살표가 표시됩니다.

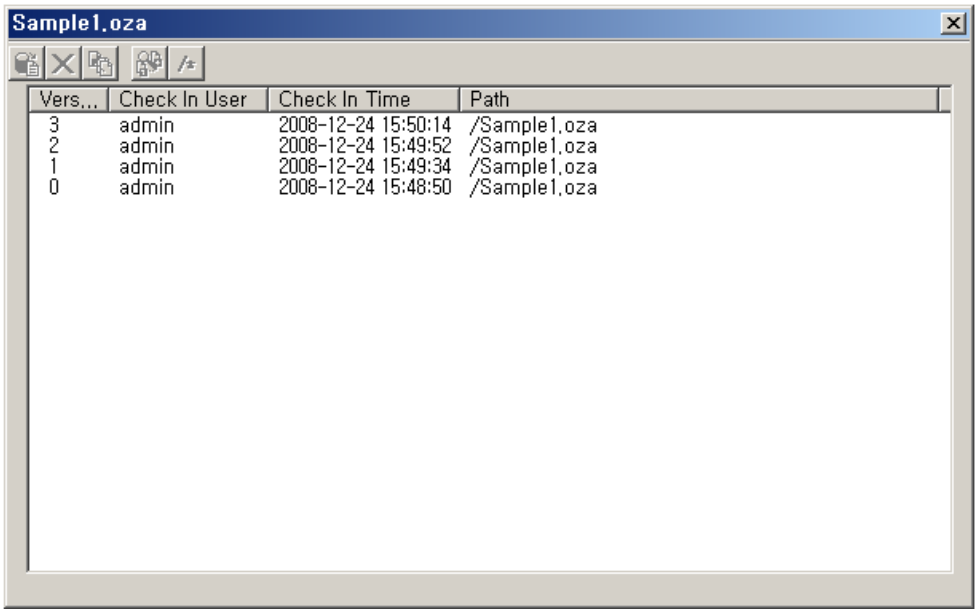


스크립트 비교 – 서버 파일의 버전 간 비교

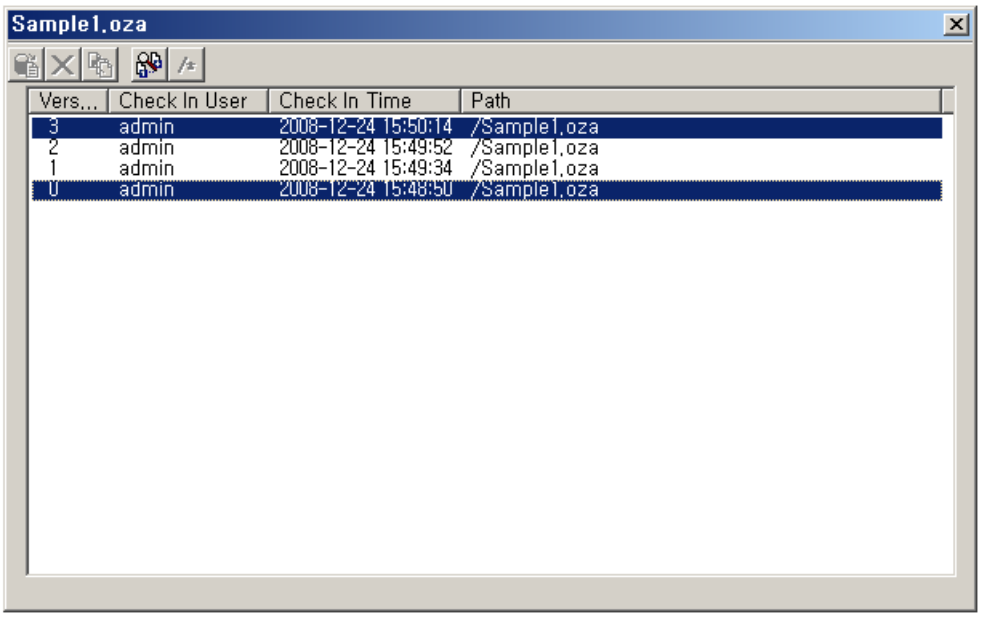
리파지토리창의 서버 리파지토리에서 버전 간 스크립트를 비교할 OZA, OZF, OZC 파일을 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [Show History] 메뉴를 클릭합니다.



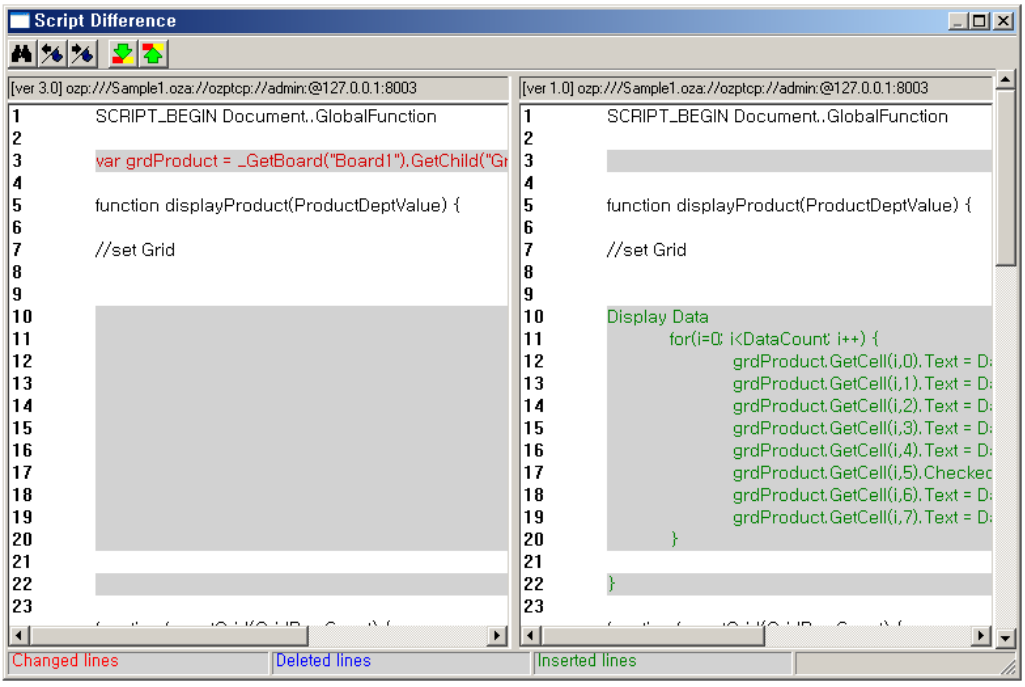
[Show History] 메뉴를 클릭하면 선택한 아이템의 히스토리를 나타낸 다이얼로그가 나타납니다.



스크립트를 비교할 버전을 선택한 후 스크립트 비교 아이콘()을 클릭합니다.



스크립트 비교 다이얼로그가 실행되면 버전별로 스크립트를 비교할 수 있습니다.



버전은 화변 상단 타이틀을 확인하시기 바랍니다.

※ 참고사항

- 특정 버전과 로컬에 있는 파일 또는 작업 중인 파일과 비교하려면 해당 버전을 선택한

후 스크립트 비교 아이콘()을 클릭하시기 바랍니다.

- 두 개의 버전을 선택하면 툴바의 다운로드, 삭제, 롤백, 주석 아이콘은 비활성화되고, 세 개 이상의 버전을 선택하면 모든 아이콘이 비활성화됩니다.

스크립트 비교 관련 참고사항

- 같은 확장자의 파일끼리만 스크립트 비교를 할 수 있습니다.
- OZA 파일의 스크립트 비교 시 Auto 또는 Manual Include 방식으로 OZA 파일에 포함된 OZC 파일의 스크립트는 비교할 수 없으므로, 반드시 OZC 파일로 비교하시기 바랍니다.
- 스크립트 비교 다이얼로그는 단순히 비교만 하는 다이얼로그로, 스크립트 편집은 할 수 없습니다.

전역 변수, 전역 함수

다음은 전역 변수와 전역 함수를 사용하는 방법에 대해 설명합니다.

전역 변수

특정 컴포넌트 내에서만 사용되는 지역 변수와 달리, 전역 변수는 프로젝트 어디에서든지 선언하고 값을 변경하고 사용할 수 있습니다.

- 전역 변수 선언, 값 변경

  OZ Application Designer에서는 전역 변수 선언과 동시에 전역 변수 값을 변경할 수 있습니다.

```
Document._SetGlobal("변수명", 변수값);

//Document의 메소드는 Document를 생략하고 아래와 같이 코딩할 수 있습니다.
_SetGlobal("변수명", 변수값);
```

- 전역 변수 사용

```
var myVar = _GetGlobal("변수명");
```

전역 함수

전역 함수는 프로젝트 내 어디에서든지 호출할 수 있는 함수로, 공통적이거나 자주 쓰이는 함수를 전역 함수로 선언하여 사용하면 불필요한 코딩을 줄이고 개발 기간을 단축하는 등의 이점이 있습니다.

프로젝트 내 어디에서든지 선언할 수 있는 전역 변수와 달리, 전역 함수는 'Document'의 'GlobalFunction'에 선언하여야만 전역 함수로 사용할 수 있습니다.

■ 전역 함수 선언

```
function 함수명 (Argument1, Argument2, ... ) {
  //Script 코딩
  ...
  ...
  //값을 리턴하는 경우
  return 리턴값;
}
```

■ 전역 함수 호출

전역 함수를 호출하고자 하는 곳에 함수명과 인자값을 입력하여 호출합니다.

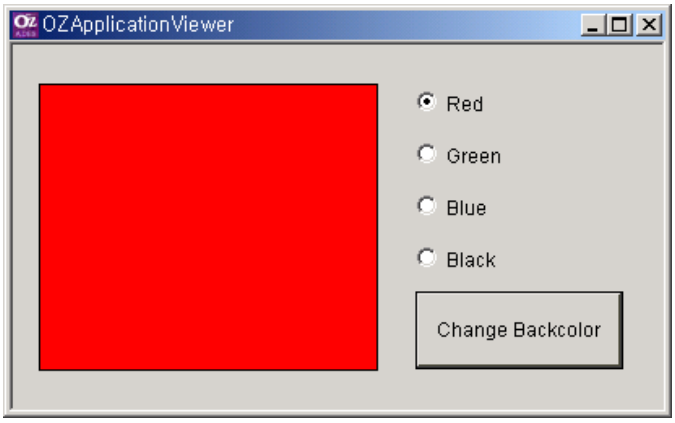
```
//리턴값을 받지 않는 경우
  함수명(Argument1, Argument2, ... );

//리턴값을 받을 경우
  var result = 함수명(Argument1, Argument2, ... );
```

Example

전역 변수와 전역 함수를 사용하는 방법을 다음과 같은 예제를 통해 설명합니다.

미리보기 화면

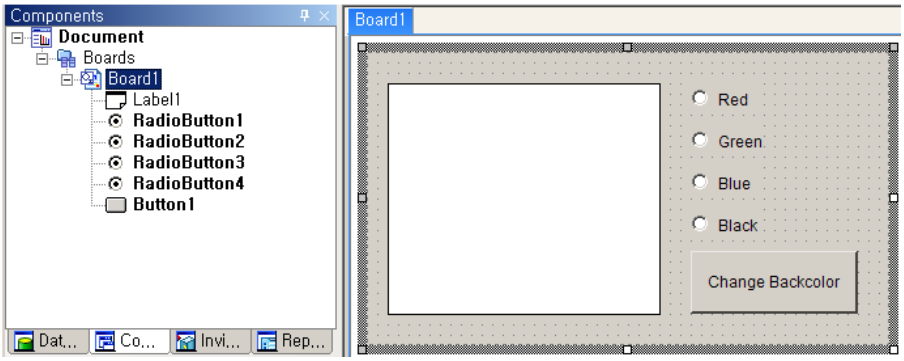


기능

- 색을 선택한 후 [Change Backcolor] 버튼을 클릭하여 Label의 배경색을 변경합니다.
- 색을 나타내는 RadioButton을 클릭하면 선택한 색을 전역 변수로 저장합니다.
- [Change Backcolor] 버튼을 클릭하면 배경색을 변경하는 전역 함수를 호출합니다.
- 전역 함수는 컴포넌트명을 인자값으로 받아 해당 컴포넌트의 배경색을 변경하는 함수로 코딩합니다.

Step 1 컴포넌트 추가, 디자인

프로젝트를 새로 만들어 Board에 Label, RadioButton, Button등을 추가하고 텍스트, 레이아웃 등을 아래 그림처럼 디자인합니다.



Step 2 스크립트 코딩

전역 변수와 전역 함수로 Label의 배경색을 변경하는 스크립트를 코딩합니다.

- 컴포넌트명을 인자값으로 받고, 전역 변수에 저장된 RGB색으로 컴포넌트의 배경색을 변경하는 전역 함수를 코딩합니다. 스크립트창에서 'Document'의 'GlobalFunction'를 선택한 후 아래와 같이 입력합니다.

```
function changeBackColor(CompName) {
    var Comp = _GetCurrentBoard().GetChild(CompName);
    Comp.BackColor = _GetGlobal("Color");
}
```

- 'RadioButton1'을 클릭하면, 선택한 색값을 저장하는 전역 변수값을 빨간색을 나타내는 RGB값인 '255, 0, 0'으로 변경하는 스크립트를 코딩합니다. 스크립트창에서 'RadioButton1'의 'OnClick' 이벤트를 선택한 후 아래와 같이 입력합니다.

```
_SetGlobal("Color", "255, 0, 0");
```

- 'RadioButton1'과 같이 'RadioButton2', ''RadioButton3', 'RadioButton4'의 'OnClick' 이벤트에도 선택한 색값을 저장하는 전역 변수값을 각각 초록색, 파란색, 검정색을 나타내는 RGB값으로 변경하는 스크립트를 코딩합니다.
  - RadioButton2의 OnClick 이벤트

```
_SetGlobal("Color", "0, 255, 0");
```

- RadioButton3의 OnClick 이벤트

```
_SetGlobal("Color", "0, 0, 255");
```

- RadioButton4의 OnClick 이벤트

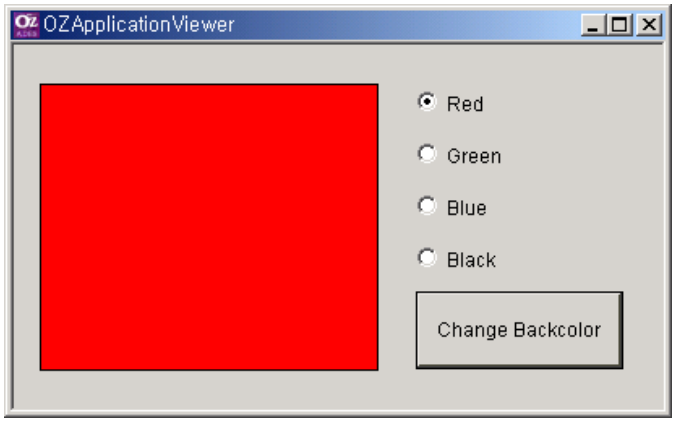
```
_SetGlobal("Color", "0, 0, 0");
```

➢ 'Button1'을 클릭하면, 배경색을 바꿀 컴포넌트인 Label의 이름을 인자값으로 넘겨 배경색을 변경하는 전역 함수를 호출하는 스크립트를 코딩합니다. 스크립트창에서 'Button1'의 'OnClick' 이벤트를 선택한 후 아래와 같이 입력합니다.

```
changeBackColor("Label1");
```

Step 3 미리보기

[File] 메뉴의 [Preview]를 클릭하거나 툴바의 미리보기 아이콘()을 클릭하여 미리보기합니다.



Ⅴ. 컴포넌트 기본 사용

- Controls 컴포넌트
- Invisibles 컴포넌트
- Container Controls 컴포넌트
- Complex Controls 컴포넌트
- Drawings 컴포넌트
- OZC (OZ Component)
- EventHook

본 장에서는 OZ Application Designer에서 제공되는 컴포넌트의 개념과 컴포넌트의 기본적인 속성, 메소드, 사용 방법과 OZC, EventHook 등에 대해 설명합니다.

※ 참고사항 : 컴포넌트 네이밍 룰

컴포넌트 이름은 한글, 영문 알파벳, 숫자, "_"만 가능하며, 첫 글자는 숫자 이외의 문자로 지정하여야 합니다. 만일 그 외의 문자로 설정하면 컴포넌트의 기본 이름 또는 이전 이름으로 자동으로 변경됩니다.

Controls 컴포넌트

Button

확인, 취소 등 사용자의 명령을 입력받는 버튼을 나타낼 때 사용하는 컴포넌트입니다.

■ Property

Button의 기본적인 속성은 아래와 같습니다.

Property	Description
Text	Button에 표시할 텍스트
FlatStyle	Button의 모양 ∗ Standard : 3차원 입체모양으로 표시 ∗ Flat : 평면으로 표시

■ Example

- 구현 기능

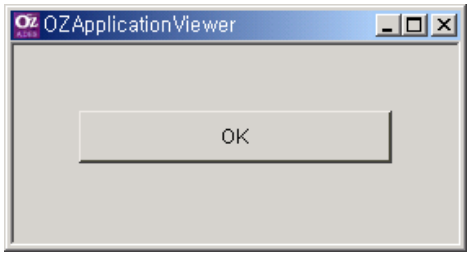
  3차원 입체 모양의 [OK] 버튼을 표시하는 예제입니다.

- 디자인

  ➢ Board에 Button을 추가한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
Text	OK	**FlatStyle**	Standard

- 미리보기



Label

화면에 숫자나 문자열 등을 표시할 때 사용하는 컴포넌트입니다.

■ Property

Label의 기본적인 속성은 아래와 같습니다.

Property	Description
Text	Label에 표시할 텍스트
TextAlign	텍스트 정렬 방식
Pattern	텍스트 표시 형식

■ Example

- 구현 기능

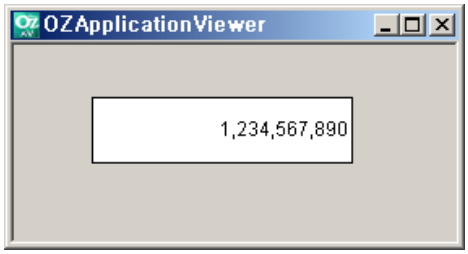
  숫자 값이 입력된 텍스트를 천단위 구분자를 주어 표시하는 예제입니다.

- 디자인

  ➢ Board에 Label을 추가한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
Text	1234567890	**TextAlign**	MiddleRight
Pattern	#,###;-#,###	**-**	-

- 미리보기



※ 참고사항 : Pattern

오즈 애플리케이션에서는 기본적으로 STRING, 날짜, 숫자, 통화, 백분율, 마스크, 단위 변환을 지원하며, 사용자가 패턴 문자를 임의로 조합하여 설정할 수 있습니다.

오즈 애플리케이션에서는 기본적으로 제공하는 표시 형식은 속성 창에서 Pattern 속성을 클릭하여 나타나는 다이얼로그를 참고하시기 바라며, 자주 사용하는 표시 형식의 예제는 아래 표를 참고하시기 바랍니다.

- 문자

Pattern	Data	Display Text
STRING	123.45	123.45
	Sample	Sample

STRING으로 설정하면 라벨에 설정된 값을 서식없이 그대로 표시합니다.

- 날짜

Pattern	Data	Display Text
date_yyyy/MM/dd	2008-07-25	2008/07/25
date_yyyy년 MM월 dd일	2008-07-25	2008년 07월 25일

날짜 포맷이 적용되려면 라벨에 연결된 데이터 필드의 타입이 날짜형이거나 라벨의 텍스트가 "yyyy-MM-dd" 형태이어야 합니다.

- 숫자

Pattern	Data	Display Text

###,###.#	1234	1,234
000,000.0	1234	001,234.0
#.0#	.567	.57
0.0#	.567	0.57
#,### 명	1234	1,234 명
#,### 감소	-1234	-1,234 감소
△#,###;▽#,###	1234	△1,234
	-1234	▽1,234

숫자는 #, 0, 쉼표(,), 마침표(.) 등의 패턴 문자를 사용할 수 있으며, 임의의 문자열과 조합하여 사용할 수 있습니다.

기호는 입력한 숫자의 자릿수가 소수점 왼쪽 또는 오른쪽에서 서식에 지정된 # 기호의 개수보다 적은 경우 추가로 0을 표시하지 않고, 0 기호는 추가로 0을 표시합니다.

#, 0 기호 사용 시 소수점 오른쪽의 자릿수가 서식에 지정된 기호 개수보다 많으면 지정한 개수만큼 소수점 자리수가 반올림되고, 소수점 왼쪽의 자릿수가 서식에 지정된 기호 개수보다 많으면 남은 자릿수도 그대로 표시됩니다.

TextBox

숫자나 문자열을 입력받을 때 사용하는 컴포넌트입니다.

■ Property

TextBox의 기본적인 속성은 아래와 같습니다.

Property	Description
BorderStyle	테두리 스타일 * Fixed3D : 3차원 형태의 안으로 들어간 모양의 테두리 * FixedSingle : 실선 모양의 테두리 * None : 테두리 없음

ScrollBars	스크롤바 표시 여부 ∗ Horizontal : 수평 스크롤바 표시, 'Multiline' 속성 값이 'True'이고 'WordWrap' 속성 값이 'False'인 경우 적용됨 ∗ Vertical : 수직 스크롤바 표시 'Multiline' 속성 값이 'True'이고 'WordWrap' 속성 값이 'False'인 경우 적용됨 ∗ Both : 수평, 수직 스크롤바 표시, 'Multiline' 속성 값이 'True'이고 'WordWrap' 속성 값이 'False'인 경우 적용됨 ∗ None : 스크롤바 없음
Multiline	여러 줄 입력 가능 여부
PasswordChar	입력되는 텍스트를 변경할 문자 예를 들어, '*'로 설정하면 TextBox에 'AAA'를 입력하면 '***'로 표시됨
WordWrap	자동 줄바꿈 여부

■ Example

- 구현 기능

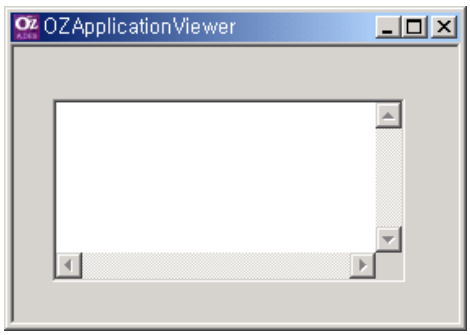
  수평, 수직 스크롤바가 있고 여러 줄을 입력하는 TextBox 예제입니다.

- 디자인

  ➢ Board에 TextBox를 추가한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
ScrollBars	Both	**Multiline**	True
WordWrap	False	**-**	-

- 미리보기



CheckBox

다수의 옵션을 선택하는 체크박스를 나타낼 때 사용하는 컴포넌트입니다.

■ Property

CheckBox의 기본적인 속성은 아래와 같습니다.

Property	Description
Checked	체크 여부 * True : 체크됨, CheckState 속성 값을 자동으로 'Checked'로 변경함 * False : 체크 안됨, CheckState 속성 값을 자동으로 'Unchecked'로 변경함
CheckState	CheckBox의 선택 상태 * Checked : Appearance 속성 값이 'Normal'이면 체크표시를 하고, 'Button'이면 들어간 버튼 모양으로 표시, Checked 속성 값을 자동으로 'True'로 변경함 * Unchecked : Appearance 속성 값이 'Normal'이면 체크표시를 하지 않고, 'Button'이면 튀어 나온 버튼 모양으로 표시, Checked 속성 값을 자동으로 'False'로 변경함 * Indeterminate : Appearance 속성 값이 'Normal'이면 체크표시를 하며 체크색이 회색으로 표시되고, 'Button'이면 들어간 버튼 모양에 버튼색이 회색으로 표시
Appearance	CheckBox의 모양 * Normal : 체크 표시가 있는 기본적인 모양으로 설정 * Button : 버튼 모양으로 설정
ThreeState	CheckState의 Check, Uncheck, Indeterminate 세가지 상태 허용 여부

■ Example

- 구현 기능

  CheckBox의 선택 상태가 Checked 상태이며, Check, Uncheck 두 가지 상태를 선택할 수 있는 CheckBox와 CheckBox의 선택 상태가 Indeterminate 상태이며, Check, Uncheck, Indeterminate 세 가지 상태를 선택할 수 있는 CheckBox 예제입니다.

- 디자인

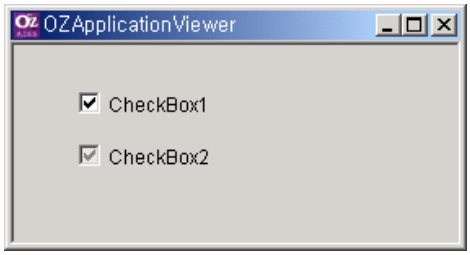
➢ Board에 CheckBox를 추가한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
Checked	True	**CheckState**	Checked
ThreeState	False	**-**	-

➢ Board에 CheckBox를 하나 더 추가한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
Checked	True	**CheckState**	Indeterminate
ThreeState	True	**-**	-

- 미리보기



RadioButton

한 부모 컴포넌트 내에서 하나의 옵션만을 선택하는 옵션 버튼을 나타낼 때 사용하는 컴포넌트입니다.

■ Property

RadioButton의 기본적인 속성은 아래와 같습니다.

Property	Description
Checked	체크 여부

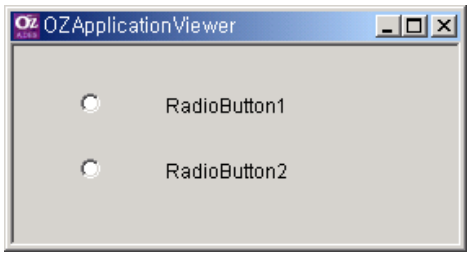
■ Example

- 구현 기능

Board에 RadioButton 2개가 있는 RadioButton 예제입니다.

- 디자인
  - ➢ Board에 RadioButton을 두 개 추가합니다.

- 미리보기



ComboBox

문자열을 입력할 수 있는 편집창과 아래 화살표로 구성되며, 다수의 아이템을 리스트 형태로 나타내는 콤보 박스를 나타낼 때 사용하는 컴포넌트입니다. ComboBox에 나타낼 아이템 즉, ComboBoxItem은 메소드 뿐만 아니라 마법사로도 추가할 수 있으며, 마법사 사용법은 본 매뉴얼의 Ⅶ.장 마법사 사용법을 참조하십시오.

■ Property

ComboBox의 기본적인 속성은 아래와 같습니다.

Property	Description
DropDownStyle	아이템이 DropDown 되는 형식 * DropDown : 아이템 중에서 선택할 수도 있고 TextBox에서 편집도 가능함 * DropDownList : 아이템 중에서 선택만 가능하고 편집할 수 없음
AutoSelect	ComboBox의 편집창에서 문자를 입력하면 가장 유사한 아이템을 자동으로 선택할지 여부. DropDownStyle 속성 값이 'DropDown' 이었을 경우에만 적용됨
SelectedIndex	선택된 아이템의 인덱스. 인덱스는 0부터 시작됨

SelectedText	편집창에서 편집중인 텍스트. DropDownStyle 속성 값이 DropDownList인 경우는 항상 공백 문자임

■ Method

ComboBox의 기본적인 메소드는 아래와 같습니다.

- AddItem

ComboBox 아이템의 마지막 인덱스에 설정한 텍스트의 아이템을 추가하고, 아이템 객체를 리턴합니다.

➢ Format

```
object AddItem(string ItemText);
```

▪ ItemText : 아이템의 화면에 보여지는 텍스트

➢ Example

```
var objItem = ComboBox1.AddItem("a new item label");
```

- InsertItem

ComboBox에 특정 인덱스에 아이템을 추가하고, 아이템 객체를 리턴합니다.

➢ Format

```
object InsertItem(integer ItemIndex, string ItemText);
```

▪ ItemIndex : 아이템의 인덱스

▪ ItemText : 아이템의 화면에 보여지는 텍스트

➢ Example

```
var objItem = ComboBox1.InsertItem(5, "a new item label");
```

- RemoveItem

설정한 인덱스에 해당하는 아이템을 삭제합니다.

➢ Format

```
void RemoveItem(integer ItemIndex);
```

▪ ItemIndex : 삭제할 아이템의 인덱스

➢ Example

```
ComboBox1.RemoveItem(5);
```

- GetItemObject

ComboBox에 추가된 아이템 중 특정 인덱스의 아이템 객체를 얻습니다.

➢ Format

```
object GetItemObject(integer ItemIndex);
```

▪ ItemIndex : 아이템의 인덱스

➢ Example

```
var objItem = ComboBox1.GetItemObject(5);
objItem.ItemValue = "CODE0123";
```

■ Example

- 구현 기능

ComboBox에 아이템을 추가하고 선택된 아이템의 텍스트와 값을 표시하는 예제입니다.

- 디자인

➢ Board에 ComboBox를 추가한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
DropDownStyle	DropDown	**AutoSelect**	True

➢ 추가한 ComboBox(Name=ComboBox1)를 선택한 후 'OnInitialize' 이벤트를 선택하여 아래와 같이 코딩합니다.

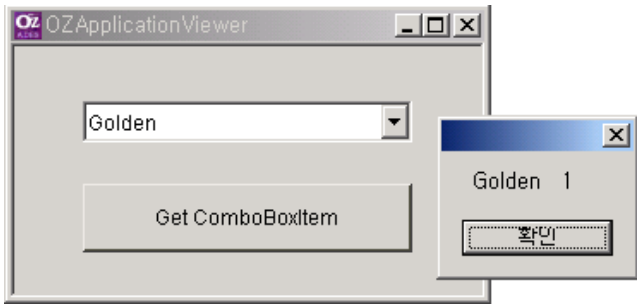
```
var objItem;
objItem = ComboBox1.AddItem("Golden");
objItem.ItemValue = 1;
objItem = ComboBox1.AddItem("Silver");
objItem.ItemValue = 2;
objItem = ComboBox1.AddItem("Bronze");
objItem.ItemValue = 3;
ComboBox1.SelectedIndex = 0;
```

➢ 선택된 아이템의 텍스트와 값을 표시하기 위하여 Button을 추가한 후 'Text' 속성 값을 'Get ComboBoxItem'으로 변경하고 추가한 Button의 'OnClick' 이벤트를 선택하여 아래와 같이 코딩합니다.

```
var objItem = ComboBox1.GetItemObject(ComboBox1.SelectedIndex);
_MessageBox(objItem.Text + "   " + objItem.ItemValue);
```

- 미리보기

  ComboBox에서 아이템을 선택하고 [Get ComboBoxItem] 버튼을 클릭하면 선택한 아이템의 텍스트(Text)와 아이템값(ItemValue)를 메시지박스로 표시합니다.



ListBox

다수의 아이템을 리스트 형태로 표시할 때 사용하는 컴포넌트입니다. ListBox에 나타낼 아이템 즉, ListBoxItem은 메소드 뿐만 아니라 마법사로도 추가할 수 있으며, 마법사 사용법은 본 매뉴얼의 Ⅶ.장 마법사 사용법을 참조하십시오.

■ Property

ListBox의 기본적인 속성은 아래와 같습니다.

Property	Description
ItemSelectionMode	아이템 선택 모드 * One : 하나의 아이템만 선택 가능 * MultiSimple : 여러 번의 마우스 클릭을 이용해 복수의 아이템 선택 가능 * MultiExtended : Shift, Ctrl키와 마우스 클릭을 이용해 복수의 아이템 선택 가능 * None: 아이템 선택 불가
FocusedIndex	포커스를 가진 아이템의 인덱스
FocusedText	포커스를 가진 아이템의 텍스트

■ Method

ListBox의 기본적인 메소드는 아래와 같습니다.

- AddItem

  ListBox 아이템의 마지막 인덱스에 설정한 텍스트의 아이템을 추가하고, 아이템 객체를 리턴합니다.

  ➢ Format

  ```
  object AddItem(string ItemText);
  ```

  ▪ ItemText : 아이템의 화면에 보여지는 텍스트

  ➢ Example

  ```
  var objItem = ListBox1.AddItem("a new item label");
  ```

- InsertItem

  ListBox에 특정 인덱스에 아이템을 추가하고, 아이템 객체를 리턴합니다.

  ➢ Format

  ```
  object InsertItem(integer ItemIndex, string ItemText);
  ```

  ▪ ItemIndex : 아이템의 인덱스

  ▪ ItemText : 아이템의 화면에 보여지는 텍스트

  ➢ Example

  ```
  var objItem = ListBox1.InsertItem(5, "a new item label");
  ```

- RemoveItem

  설정한 인덱스에 해당하는 아이템을 삭제합니다.

  ➢ Format

  ```
  void RemoveItem(integer ItemIndex);
  ```

  ▪ ItemIndex : 삭제할 아이템의 인덱스

  ➢ Example

  ```
  ListBox1.RemoveItem(5);
  ```

- GetItemObject

  ListBox에 추가된 아이템 중 특정 인덱스의 아이템 객체를 얻습니다.

  ➢ Format

  ```
  object GetItemObject(integer ItemIndex);
  ```

  ▪ ItemIndex : 아이템의 인덱스

➢ Example

```
var objItem = ListBox1.GetItemObject(5);
objItem.ItemValue = "CODE0123";
```

- GetSelectedIndexes

ListBox에 추가된 아이템 중 선택된 아이템의 인덱스를 배열로 얻습니다.

➢ Format

```
void GetSelectedIndexes(Array SelectedIndexArray);
```

▪ SelectedIndexArray : 선택된 아이템의 인덱스를 받을 배열

➢ Example

```
var SelectedIndexs = new Array();
ListBox1.GetSelectedIndexes(SelectedIndexs);
```

■ Example

- 구현 기능

ListBox에 아이템을 추가하고 선택된 아이템의 텍스트를 표시하는 예제입니다.

- 디자인

➢ Board에 ListBox를 추가한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
ItemSelectionMode	MultiExtended	**-**	-

➢ 추가한 ListBox(Name=ListBox1)를 선택한 후 'OnInitialize' 이벤트를 선택하여 아래와 같이 코딩합니다.

```
ListBox1.AddItem("Golden");
ListBox1.AddItem("Silver");
ListBox1.AddItem("Bronze");
ListBox1.AddItem("Normal");
```

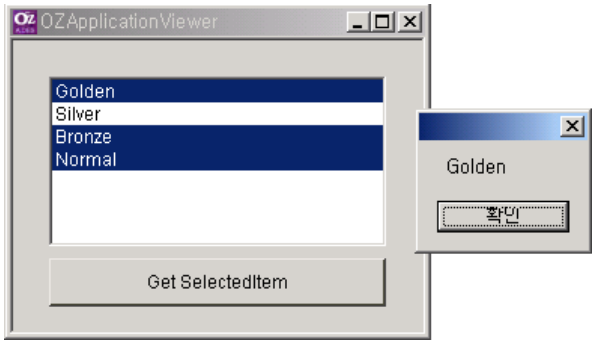
➢ 선택된 아이템의 텍스트를 표시하기 위하여 Button을 추가한 후 'Text' 속성 값을 'Get SelectedItem'으로 변경하고 추가한 Button의 'OnClick' 이벤트를 선택하여 아래와 같이 코딩합니다.

```
var SelectedIndexs = new Array();
```

```
ListBox1.GetSelectedIndexes(SelectedIndexs);
for(i=0;i<SelectedIndexs.length;i++) {
    _MessageBox(ListBox1.GetItemObject(SelectedIndexs[i]).Text)
    ;
}
```

- 미리보기

ListBox에서 Shift, Ctrl키와 마우스 클릭을 이용해 복수의 아이템을 선택하고 [Get SelectedItem] 버튼을 클릭하면 선택된 아이템의 텍스트(Text)를 메시지박스로 표시합니다.



CheckListBox

체크마크를 가진 아이템을 리스트 형태로 나타낼 때 사용하는 컴포넌트입니다. CheckListBox에 나타낼 아이템 즉, CheckListBoxItem은 메소드 뿐만 아니라 마법사로도 추가할 수 있으며, 마법사 사용법은 본 매뉴얼의 Ⅶ.장 마법사 사용법을 참조하십시오.

■ Property

CheckListBox의 기본적인 속성은 아래와 같습니다.

Property	Description
ItemSelectionMode	아이템 선택 모드 * One : 하나의 아이템만 선택 가능 * MultiSimple : 여러 번의 마우스 클릭을 이용해 복수의 아이템 선택 가능 * MultiExtended : Shift, Ctrl키와 마우스 클릭을 이용해 복수의 아이템 선택 가능 * None: 아이템 선택 불가

FocusedIndex	포커스를 가진 아이템의 인덱스
FocusedText	포커스를 가진 아이템의 텍스트

■ Method

CheckListBox의 기본적인 메소드는 아래와 같습니다.

- AddItem

CheckListBox 아이템의 마지막 인덱스에 설정한 텍스트의 아이템을 추가하고, 아이템 객체를 리턴합니다.

➢ Format

```
object AddItem(string ItemText);
```

▪ ItemText : 아이템의 화면에 보여지는 텍스트

➢ Example

```
var objItem = CheckListBox1.AddItem("a new item label");
```

- InsertItem

CheckListBox에 특정 인덱스에 아이템을 추가하고, 아이템 객체를 리턴합니다.

➢ Format

```
object InsertItem(integer ItemIndex, string ItemText);
```

▪ ItemIndex : 아이템의 인덱스

▪ ItemText : 아이템의 화면에 보여지는 텍스트

➢ Example

```
var objItem = CheckListBox1.InsertItem(5, "a new item label");
```

- GetItemObject

CheckListBox에 추가된 아이템 중 특정 인덱스의 아이템 객체를 얻습니다.

➢ Format

```
object GetItemObject(integer ItemIndex);
```

▪ ItemIndex : 아이템의 인덱스

➢ Example

```
var objItem = CheckListBox1.GetItemObject(5);
objItem.ItemValue = "CODE0123";
```

- AddCheck

  설정한 인덱스에 해당하는 아이템의 체크 표시를 Checked 상태로 설정합니다.

  ➢ Format

  ```
  void AddCheck(integer ItemIndex);
  ```

  ▪ ItemIndex : 아이템의 인덱스

  ➢ Example

  ```
  CheckListBox1.AddCheck(5);
  ```

- RemoveCheck

  설정한 인덱스에 해당하는 아이템의 체크 표시를 Unchecked 상태로 설정합니다.

  ➢ Format

  ```
  void RemoveCheck(integer ItemIndex);
  ```

  ▪ ItemIndex : 아이템의 인덱스

  ➢ Example

  ```
  CheckListBox1.RemoveCheck(5);
  ```

- GetCheckedIndexes

  CheckListBox에 추가된 아이템 중 Checked 상태인 아이템의 인덱스를 배열로 얻습니다.

  ➢ Format

  ```
  void GetCheckedIndexes(Array CheckedIndexArray);
  ```

  ▪ CheckedIndexArray : Checked 상태인 아이템의 인덱스를 받을 배열

  ➢ Example

  ```
  var CheckedIndexs = new Array();
  CheckListBox1.GetCheckedIndexes(CheckedIndexs);
  ```

■ Example

- 구현 기능

  CheckListBox에 아이템을 추가하고 체크된 아이템의 텍스트를 표시하는 예제입니다.

- 디자인

  ➢ Board에 CheckListBox(Name=CheckListBox1)를 추가하고, 추가한

CheckListBox를 선택한 후 'OnInitialize' 이벤트를 선택하여 아래와 같이 코딩합니다.

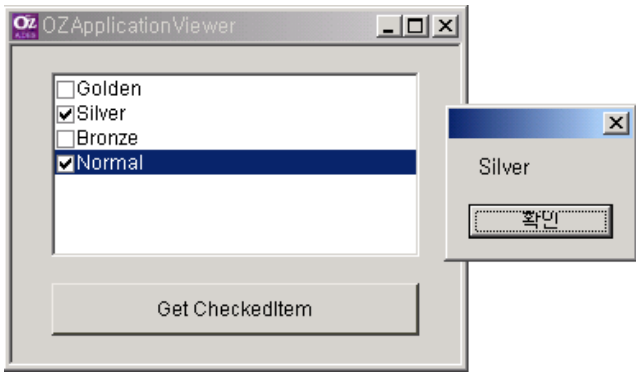
```
CheckListBox1.AddItem("Golden");
CheckListBox1.AddItem("Silver");
CheckListBox1.AddItem("Bronze");
CheckListBox1.AddItem("Normal");
```

➢ 체크된 아이템의 텍스트를 표시하기 위하여 Button을 추가한 후 'Text' 속성 값을 'Get CheckedItem'으로 변경하고 추가한 Button의 'OnClick' 이벤트를 선택하여 아래와 같이 코딩합니다.

```
var CheckedIndexs = new Array();
CheckListBox1.GetCheckedIndexes(CheckedIndexs);
for(i=0;i<CheckedIndexs.length;i++) {
    _MessageBox(CheckListBox1.GetItemObject(CheckedIndexs[i]).T
    ext);
}
```

- 미리보기

CheckListBox에서 각 아이템을 체크하고 [Get CheckedItem] 버튼을 클릭하면 Checked 상태인 아이템의 텍스트(Text)를 메시지박스로 표시합니다.



PictureBox

BMP, JPG, GIF 등 이미지 파일을 표시할 때 사용하는 컴포넌트입니다.

■ Property

PictureBox의 기본적인 속성은 아래와 같습니다.

Property	Description
Image	PictureBox에 표시할 이미지 파일명
ImageAlign	표시할 이미지의 정렬 방식 * None : 이미지 원래 크기대로 표시 * Fit : PictureBox의 크기에 맞게 이미지의 크기를 조절하여 표시 * Tile : 바둑판식으로 반복하여 표시 * TopLeft/TopCenter/TopRight : 상단좌측/상단중앙/상단우측을 기준으로 하여 원래의 이미지 크기대로 표시 * MiddleLeft/MiddleCenter/MiddleRight : 중단좌측/중단중앙/중단우측을 기준으로 하여 원래의 이미지 크기대로 표시 * BottomLeft/BottomCenter/BottomRight : 하단좌측/하단중앙/하단우측을 기준으로 하여 원래의 이미지 크기대로 표시
SizeToImage	이미지 파일의 크기에 맞게 PictureBox의 크기 조정 여부

■ Example

PictureBox를 이용하여 샘플 그림 파일을 표시하는 예제입니다.

- 구현 기능

  PictureBox를 이용하여 샘플 그림 파일인 'Image1.gif' 파일을 표시하는 예제입니다.

- 디자인

  ➢ Board에 PictureBox을 추가한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
Image	ozp:///Image1.gif	**ImageAlign**	None

- 미리보기



NumericUpDown

TextBox와 Up/Down 버튼이 조합된 컴포넌트로, 숫자값을 TextBox에 직접 입력하거나 Up/Down 버튼을 클릭하여 숫자값을 증감시키고자 할 때 사용하는 컴포넌트입니다.

■ Property

NumericUpDown의 기본적인 속성은 아래와 같습니다.

Property	Description
DecimalPlaces	소수점을 표시할 때 소수점 아래 자리수
Hexadecimal	숫자값을 16진수로 표시할지 여부
Increment	Up/Down 버튼을 클릭하여 숫자값을 증감시킬 때 증감시킬 값
MaxVal	표시되는 값의 최대값
MinVal	표시되는 값의 최소값
ThousandsSeparator	천단위 구분자 표시 여부
Value	표시되는 값

■ Example

NumericUpDown를 이용하여 경비를 입력받는 예제입니다.

- 구현 기능

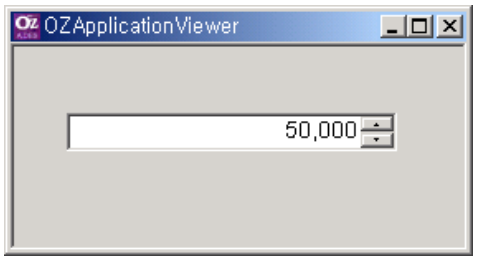
  경비를 만원부터 천만원사이의 값으로 입력할 수 있고, 값의 증감 단위는 천원으로 하며, 금액을 입력하므로 천단위 구분자를 표시하고, 기본 값으로 5만원이 표시되도록 합니다.

- 디자인

  ➢ Board에 NumericUpDown을 추가한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
Increment	1000	**MaxVal**	10000000
MinVal	10000	**ThousandsSeparator**	True
Value	50000	**-**	-

- 미리보기



MaskTextBox

날짜나 주민등록번호 등 특정한 형식을 지정하여 문자열을 입력받고자 할 때 사용하는 컴포넌트입니다.

■ Property

MaskTextBox의 기본적인 속성은 아래와 같습니다.

Property	Description

CharacterCasing	알파벳 문자를 대소문자로 변환하여 입력받을지 여부 * Normal:입력한 그대로 입력받음 * Upper:알파벳 문자를 모두 대문자로 변환하여 입력 * Lower:알파벳 문자를 모두 소문자로 변환하여 입력
Mask	마스크 문자열
Literal	리터럴 문자열
Default	기본 값
PromptChar	프롬프트 문자

■ Method

MaskTextBox의 기본적인 메소드는 아래와 같습니다.

- SetEditMask

MaskTextBox의 마스크 문자열, 리터럴 문자열, 기본 값을 설정합니다.

➢ Format

```
void SetEditMask(string Mask, string Literal, string Default);
```

▪ Mask : 마스크 문자열로 사용 가능한 마스크 문자는 아래와 같습니다.

Mask	Description
#	숫자, Space, +, -
L	알파벳 문자
?	알파벳 문자, Space
A	알파벳 문자, 숫자
a	알파벳 문자, 숫자, Space
&	프린트 문자
H	헥사값을 나타내는 숫자
X	헥사값을 나타내는 숫자, Space
>	알파벳(대문자로 변환)
<	알파벳(소문자로 변환)

▪ Literal : PromptChar와 리터럴 문자열

▪ Default : 기본 값

➢ Example

▪ 날짜

```
MaskTextBox1.SetEditMask("####/##/##", "____/__/__",
"2005/01/01");
```

- 시간

```
MaskTextBox1.SetEditMask(">>.##:##", "__.__:__",
"AM.09:00");
```

- 주민등록 번호

```
MaskTextBox1.SetEditMask("######-#######",
"______-_______", "800101-1234567")
```

- 전화 번호

```
MaskTextBox1.SetEditMask("###)####-####", "___)____-____",
"010)2222-3333");
```

■ Example

MaskTextBox를 이용하여 제품 번호를 입력받는 예제입니다.

- 구현 기능

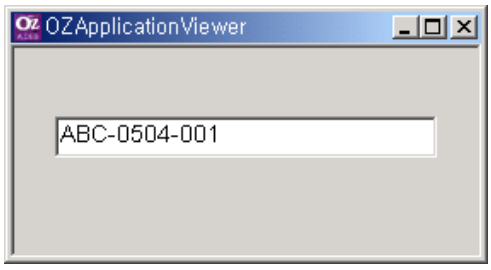
  제품 번호를 '알파벳대문자세자리-숫자네자리-숫자세자리'로 입력받습니다.

- 디자인

  ➢ Board에 MaskTextBox을 추가한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
Mask	>>>-####-###	**Literal**	___-____-__
Default	ABC-0504-001	**PromptChar**	_

- 미리보기



RichTextBox

글자별로 텍스트 형식을 달리 설정하여 문자열을 표시하고자할 때 사용하는 컴포넌트입니다. RichTextBox 마법사를 이용하여 포맷팅된 텍스트를 표시할 수 있으며, 마법사 사용법은 본 매뉴얼의 Ⅶ.장 마법사 사용법을 참조하십시오.

■ Property

RichTextBox의 기본적인 속성은 아래와 같습니다.

Property	Description
RichEditText	화면에 표시되는 텍스트, 마법사를 통해서만 입력됨 **※ 참고사항** 텍스트 포맷 정보를 모두 포함하여 리턴되므로, 화면에 보이는 텍스트만을 가져올 경우에는 GetSel() 메소드와 GetSelText() 메소드를 이용하기 바람

■ Method

RichTextBox의 기본적인 메소드는 아래와 같습니다.

- GetDefCharFormat

  RichTextBox의 기본 텍스트 포맷 객체를 가져옵니다.

  ➢ Format

  ```
  object GetDefCharFormat();
  ```

- GetSelectCharFormat

  RichTextBox의 선택된 영역의 텍스트 포맷 객체를 가져옵니다.

  ➢ Format

  ```
  object GetSelectCharFormat([integer StartIndex,
                              integer EndIndex]);
  ```

  ▪ StartIndex : 선택할 영역의 시작 인덱스

  ▪ EndIndex : 선택할 영역의 끝 인덱스

- SetDefCharFormat

  RichTextBox의 기본 텍스트 포맷을 설정합니다.

  ➢ Format

  ```
  void SetDefCharFormat(object RichCharFormat);
  ```

- SetSelectCharFormat

  영역을 선택하여 텍스트 포맷을 설정합니다.

  ➢ Format

  ```
  void SetSelectCharFormat(object RichCharFormat,
                           [integer StartIndex, integer EndIndex]);
  ```

  - StartIndex : 적용할 영역의 시작 인덱스
  - EndIndex : 적용할 영역의 끝 인덱스

■ Example

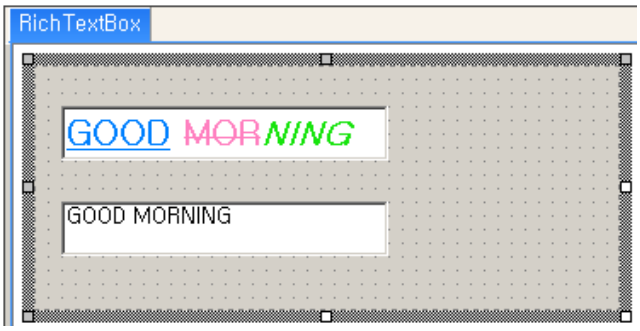
RichTextBox1에 있는 텍스트의 포맷을 RichTextBox2에 적용하는 예제입니다.

- 구현 기능

  RichTextBox1에서 선택한 영역의 텍스트 포맷을 RichTextBox2에 적용합니다. 만일 RichTextBox1에서 영역을 선택하지 않을 경우에는 RichTextBox1의 포맷된 텍스트를 그대로 RichTextBox2에 적용합니다.

- 디자인

  ➢ Board에 RichTextBox를 두 개 추가한 후 마법사를 이용하여 아래 그림과 같이 디자인합니다.

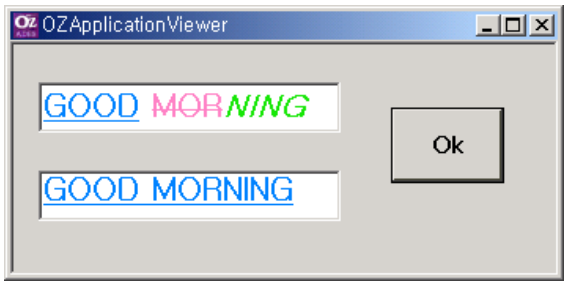


  ➢ Board에 Button을 추가하고, 추가한 Button을 선택한 후 'OnClick' 이벤트를 선택하여 아래와 같이 코딩합니다.

```
var SelText = RichTextBox1.GetSelText();
if(SelText == "") {
    RichTextBox2.RichEditText = RichTextBox1.RichEditText;
} else {
    var Pos = RichTextBox1.GetSel();
    var objCharFormat = RichTextBox1.GetSelectCharFormat(Pos.x,
    Pos.x + SelText.length);
    RichTextBox2.SetDefCharFormat(objCharFormat);
}
```

- 미리보기

  아래 그림은 RichTextBox1의 "GOOD"를 선택한 후 [Ok] 버튼을 클릭하여 텍스트 포맷을 적용한 모습입니다.



TreeView

다수의 아이템을 다단계의 트리 형태로 나타낼 때 사용하는 컴포넌트입니다. TreeView에 나타낼 아이템 즉, TreeNode는 메소드 뿐만 아니라 마법사로도 추가할 수 있으며, 마법사 사용법은 본 매뉴얼의 Ⅶ.장 마법사 사용법을 참조하십시오.

■ Property

TreeView의 기본적인 속성은 아래와 같습니다.

Property	Description
CheckBoxes	TreeNode를 체크박스 형태로 나타낼지 여부
ShowPlusMinus	자식 노드 펼침 기능인 PlusMinus 기호 표시 여부
ShowRootLines	루트의 형제 노드간을 선으로 연결하여 표시할지 여부

■ Method

TreeView, TreeNode의 기본적인 메소드는 아래와 같습니다.

- AddRootString (TreeView 메소드)

  RootNode를 추가하고 새로 추가된 RootNode 객체를 리턴합니다.

  ➢ Format

```
object AddRootString(string NodeText, [string NodeID,]
                     [integer ImageIndex,]
                     [integer SelectedImageIndex]);
```

  ▪ NodeText : 화면에 표시할 RootNode의 텍스트

- NodeID : RootNode의 ID
- ImageIndex : RootNode 앞에 표시할 이미지의 인덱스로 TreeView에 ImageList 컴포넌트가 설정되어 있는 경우에 가능함.
- SelectedImageIndex : RootNode가 선택된 상태일 경우에 표시할 이미지의 인덱스로 TreeView에 ImageList 컴포넌트가 설정되어 있는 경우에 가능함

➢ Example

```
var objRoot1 = TreeView1.AddRootString("Root1", "Node1", 1, 2);
var objRoot2 = TreeView1.AddRootString("Root2");
```

- AddNodeString (TreeNode 메소드)

자식 Node를 추가하고 새로 추가된 자식 Node 객체를 리턴합니다. 자식 Node는 RootNode에 추가될 수 있으므로 RootNode 객체를 얻어서 메소드를 호출합니다.

➢ Format

```
object AddNodeString(string NodeText, [string NodeID,]
                     [integer ImageIndex,]
                     [integer SelectedImageIndex]);
```

- NodeText : 화면에 표시할 자식 Node의 텍스트
- NodeID : 자식 Node의 ID
- ImageIndex : 자식 Node 앞에 표시할 이미지의 인덱스로 TreeView에 ImageList 컴포넌트가 설정되어 있는 경우에 가능함.
- SelectedImageIndex : 자식 Node가 선택된 상태일 경우에 표시할 이미지의 인덱스로 TreeView에 ImageList 컴포넌트가 설정되어 있는 경우에 가능함

➢ Example

```
var objRoot = TreeView1.AddRootString("Root", "Node1", 1, 2);
var objNode = objRoot.AddNodeString("Node");
```

- ExpandAll (TreeView, TreeNode 메소드)

모든 자식 Node를 펼쳐서 표시합니다.

➢ Format

```
void ExpandAll(bool ExpandFlag);
```

- ExpandFlag : 모든 자식 Node를 펼쳐서 표시할지 여부

➢ Example

```
TreeView1.ExpandAll(true);
```

■ Example

- 구현 기능

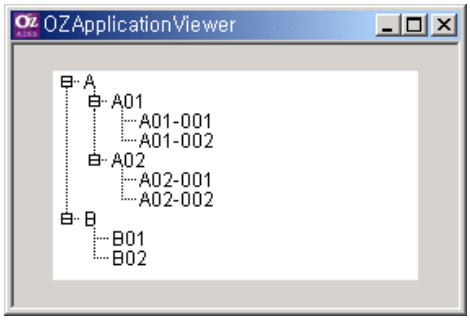
TreeView를 이용하여 다수의 아이템을 다단계의 트리 형태로 나타냅니다.

- 디자인

➢ Board에 TreeView(Name=TreeView1)를 추가하고, 추가한 TreeView를 선택한 후 'OnInitialize' 이벤트를 선택하여 아래와 같이 코딩합니다.

```
var objRoot;
var objNode;
objRoot = TreeView1.AddRootString("A");
objNode = objRoot.AddNodeString("A01");
objNode.AddNodeString("A01-001");
objNode.AddNodeString("A01-002");
objNode = objRoot.AddNodeString("A02");
objNode.AddNodeString("A02-001");
objNode.AddNodeString("A02-002");
objRoot = TreeView1.AddRootString("B");
objRoot.AddNodeString("B01");
objRoot.AddNodeString("B02");
TreeView1.ExpandAll(true);
```

- 미리보기



ListView

다수의 아이템을 컬럼과 서브 아이템이 있는 형태로 나타낼 때 사용하는 컴포넌트입니다. ListView에 나타낼 아이템(ListViewItem), 컬럼(ListViewColumnItem), 서브 아이템(ListViewSubItem)은 메소드 뿐만 아니라 마법사로도 추가할 수 있으며, 마법사 사용법은 본 매뉴얼의 Ⅶ.장 마법사 사용법을 참조하십시오.

■ Property

ListView의 기본적인 속성은 아래와 같습니다.

Property	Description
FullRowSelect	아이템 선택시 서브아이템까지 모두 선택할지 여부
AllowColumnReorder	마우스로 컬럼 타이틀을 클릭하여 Drag & Drop으로 컬럼 순서를 변경하게 할지 여부를 설정
HeaderStyle	컬럼 헤더 스타일을 설정 ∗ None : 헤더를 표시하지 않음 ∗ Clickable : 클릭 가능한 헤더로 설정

■ Method

ListView의 기본적인 메소드는 아래와 같습니다.

- AddItem (ListView 메소드)

  아이템을 추가하고 새로 추가된 아이템 객체를 리턴합니다.

  ➢ Format

  ```
  object AddItem(string ItemText, [integer ImageListIndex]);
  ```

  ▪ ItemText : 화면에 표시할 아이템의 텍스트

  ▪ ImageListIndex : 아이템에 표시할 이미지 리스트의 인덱스

- InsertItem (ListView 메소드)

  특정 인덱스에 아이템을 추가하고 새로 추가된 아이템 객체를 리턴합니다.

  ➢ Format

  ```
  object InsertItem(integer ItemIndex, string ItemText,
                    [integer ImageListIndex]);
  ```

  ▪ ItemIndex : 추가할 아이템의 인덱스

  ▪ ItemText : 추가할 아이템의 텍스트

  ▪ ImageListIndex : 아이템에 표시할 이미지 리스트의 인덱스

- InsertColumn (ListView 메소드)

  컬럼을 추가하고 새로 추가된 컬럼 객체를 리턴합니다.

  ➢ Format

  ```
  object InsertColumn(integer ColumnIndex, [string ColumnText],
                      [string ColumnAlignment],
                      [integer ColumnWidth]);
  ```

  - ColumnIndex : 추가할 컬럼의 인덱스
  - ColumnText : 추가할 컬럼의 텍스트
  - ColumnAlignment : 컬럼 텍스트의 정렬 방식
  - ColumnWidth : 컬럼의 너비

- AddSubItem (ListViewItem 메소드)

  해당 아이템에 서브 아이템을 추가하고 추가된 서브 아이템을 리턴합니다.

  ➢ Format

  ```
  object AddSubItem(string SubItemText);
  ```

  - SubItemText : 추가할 서브 아이템의 텍스트

■ Example

- 구현 기능

  ListView를 이용하여 아이템을 컬럼과 서브 아이템의 형태로 나타냅니다.

- 디자인

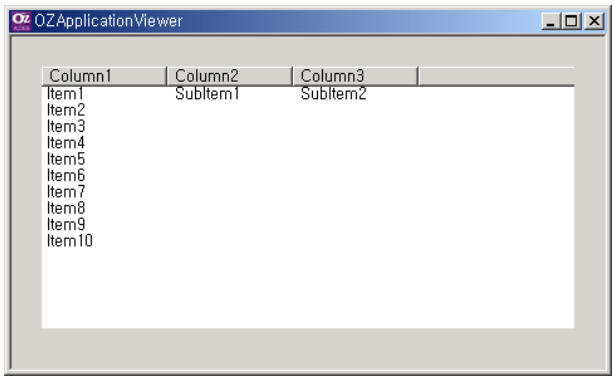
  ➢ Board에 ListView(Name=ListView1)를 추가하고, 추가한 ListView를 선택한 후 'OnInitialize' 이벤트를 선택하여 아래와 같이 코딩합니다.

```
var objColumn;
objColumn = ListView1.InsertColumn(1, "Column1");
objColumn = ListView1.InsertColumn(2, "Column2");
objColumn = ListView1.InsertColumn(3, "Column3");

var objItem;
objItem = ListView1.AddItem("Item1");
objItem.AddSubItem("SubItem1");
objItem.AddSubItem("SubItem2");

for(i=2;i<11;i++) {
      ListView1.InsertItem(i, "Item" + i);
}
```

- 미리보기



ProgressBar

진행 상태를 Bar 형태로 나타낼 때 사용하는 컴포넌트입니다.

■ Property

ProgressBar의 기본적인 속성은 아래와 같습니다.

Property	Description
BarColor	ProgressBar의 막대 색
Pos	Bar의 위치값
Maximum	Bar 위치값의 최대값
Minimum	Bar 위치값의 최소값
Step	Bar의 위치값을 증가시킬 기본 값
Text	ProgressBar의 안에 표시할 텍스트

■ Method

ProgressBar의 기본적인 메소드는 아래와 같습니다.

- IsFullPos

  Bar의 위치값이 Maximum값에 이르렀는지 체크합니다.

  ➢ Format

  ```
  bool IsFullPos();
  ```

➢ Example

```
if( ProgressBar1.IsFullPos() ) {
    return;
}
```

- OffsetPos

현재 Bar의 위치값에 설정한 값만큼 증가시켜서 Bar의 위치를 설정합니다.

➢ Format

```
void OffsetPos(OffsetValue);
```

▪ OffsetValue : 증가시킬 값

➢ Example

```
//현재 Bar의 위치값에 50을 증가합니다.
ProgressBar1.OffsetPos(50);
//만일 Bar의 위치값이 20인 경우 위 메소드를 실행하면 Bar의 위치값이 70이 됩니다.
```

- StepIt

현재 Bar의 위치값을 Step 속성에서 설정한 증가 기본 값만큼 증가시킵니다.

➢ Format

```
void StepIt();
```

➢ Example

```
//현재 ProgressBar의 위치값이 최대값이 아니면 증가 기본 값만큼 증가시킴
if( !ProgressBar1.IsFullPos() ) {
    ProgressBar1.StepIt();
}
```

■ Example

ProgressBar를 이용하여 진행 상태를 Bar 형태로 나타내는 예제입니다.

- 구현 기능

위치값 증가 버튼을 클릭하면 증가 기본 값만큼 ProgressBar의 위치값을 변경하고, 위치값이 최대값까지 증가되었을 경우 위치값을 최소값으로 변경하며, ProgressBar의 위치값을 텍스트로 나타내는 예제입니다.

- 디자인

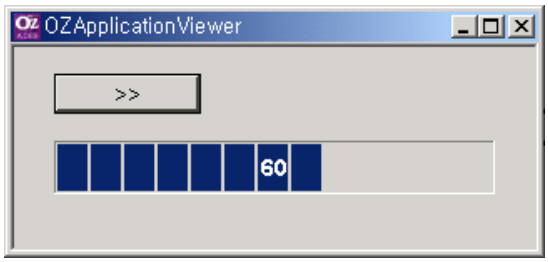
  ➢ Board에 ProgressBar를 추가한 후 속성 값을 아래와 같이 변경합니다

Property	Value	Property	Value
TextColor	255, 255, 255	**Font**	Arial, 14pt, Bold
TextAlign	MiddleCenter	**Pos**	0
Maximum	100	**Minimum**	0
Step	20	**-**	-

  ➢ Board에 Button을 추가한 후 'Text' 속성 값을 '>>'로 변경하고, 추가한 Button의 'OnClick' 이벤트를 선택하여 아래와 같이 코딩합니다.

```
if( !ProgressBar1.IsFullPos() ) {
    ProgressBar1.StepIt();
} else {
    ProgressBar1.Pos = ProgressBar1.Minimum;
}
ProgressBar1.Text = ProgressBar1.Pos;
```

- 미리보기



MonthCalendar

달력을 표시하고 날짜를 선택할 때 사용하는 컴포넌트입니다.

■ Property

MonthCalendar의 기본적인 속성은 아래와 같습니다.

Property	Description

SelectionStart	선택되거나 선택한 첫번째 날짜
SelectionEnd	선택되거나 선택한 마지막 날짜
MaxSelectionCount	여러 날을 선택할 경우 선택할 수 있는 최대 날짜 수
ShowToday	하단에 오늘 날짜 표시 여부
ShowTodayCircle	오늘 날짜에 붉은색 원 표시 여부
ShowWeekNumbers	달력에 주(Week)에 대한 일련 번호 표시 여부
TodayDate	달력의 오늘 날짜값

■ Example

MonthCalendar를 이용하여 달력을 표시합니다.

- 구현 기능

  MonthCalendar를 이용하여 달력을 표시하는 예제입니다.

- 디자인

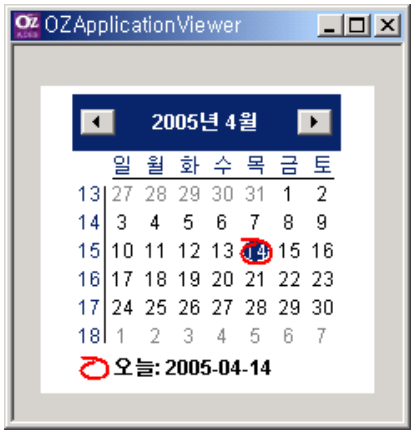
  ➢ Board에 MonthCalendar를 추가한 후 속성 값을 아래와 같이 변경합니다

Property	Value	Property	Value
ShowWeekNumbers	True	**-**	-

  ➢ MonthCalendar가 화면에 표시될 때 오늘 날짜를 시스템에서 설정된 날짜와 일치하도록 MonthCalendar의 'OnPostInitialize' 이벤트를 선택하여 아래와 같이 코딩합니다.

```
var sysTime = _GetSystemTime();
var date = sysTime.GetDate();
MonthCalendar1.TodayDate = date;
MonthCalendar1.SelectionStart = date;
MonthCalendar1.SelectionEnd = date;
```

- 미리보기



DateTimePicker

ComboBox와 MonthCalender가 결합된 컴포넌트로, 날짜 또는 시간을 입력하거나 표시할 때 사용하는 컴포넌트입니다.

■ Property

DataTimePicker의 기본적인 속성은 아래와 같습니다.

Property	Description
BackColor	배경색
Format	표시할 날짜의 포맷 * Long : Window의 지역 설정값의 Long 형식(32Bit Integer)으로 설정된 값으로 표시됨 * Short : Window의 지역 설정값의 Short 형식(16Bit Integer)으로 설정된 값으로 표시됨 * Time : Window의 지역 설정값의 Time 형식으로 설정된 값으로 표시됨 * Custom : 사용자 정의형 즉 DataTimePicker의 'CustomFormat' 속성 값에 설정된 값으로 표시됨

CustomFormat	'Format' 속성 값을 'Custom'으로 설정한 경우 적용될 포맷 문자열 입력할 수 있는 문자열은 h(12시간), H(24시간), m(분), s(초), t(오전,오후), y(년도), M(월), d(날짜), dddd(요일) 등 이며, 'hh, mm, ss'로 입력하면 앞에 0을 표시하고 'h, m, s'로 입력하면 앞에 0을 표시하지 않음
ValueDate	화면에 표시되는 날짜와 시각 정보

■ Method

DataTimePicker의 기본적인 메소드는 아래와 같습니다.

- GetCurDateTime

  선택된 날짜와 시각을 DateTime 객체로 가져옵니다. DateTime 객체의 메소드는 OZ Application Designer 도움말의 'General Script'의 'DateTime'을 참조하십시오.

  ➢ Format

  ```
  object GetCurDateTime();
  ```

  ➢ Example

  ```
  var SelectedDateTime = DateTimePicker1.GetCurDateTime();
  _MessageBox(SelectedDateTime.GetYear()); //선택된 년도
  _MessageBox(SelectedDateTime.GetMonth()); //선택된 월
  _MessageBox(SelectedDateTime.GetDay()); //선택된 날
  _MessageBox(SelectedDateTime.GetHour()); //선택된 시
  _MessageBox(SelectedDateTime.GetMinute()); //선택된 분
  _MessageBox(SelectedDateTime.GetSecond()); //선택된 초
  ```

■ Example

DataTimePicker를 이용하여 날짜와 시간을 표시합니다.

- 구현 기능

  DataTimePicker를 이용하여 사용자 정의형으로 날짜와 시간을 표시하는 예제입니다.

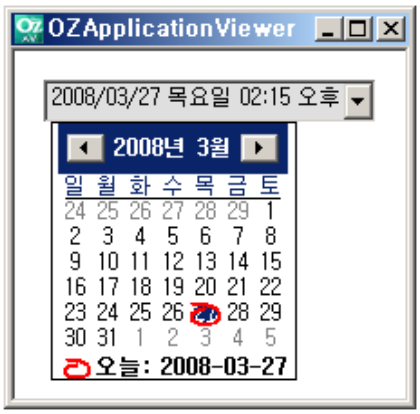
- 디자인

  ➢ Board에 DataTimePicker를 추가한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value

Format	Custom	**CustomFormat**	yyyy'/'MM'/'dd dddd hh':'mm tt
BackColor	223, 223, 223	**-**	-

- 미리보기



HTML View

웹 페이지를 표시할 때 사용하는 컴포넌트입니다.

■ Property

HTML View의 기본적인 속성은 아래와 같습니다.

Property	Description
URL	표시할 웹 페이지의 URL

■ Example

HTML View를 이용하여 특정 웹페이지를 표시합니다.

- 구현 기능

  HTML View를 이용하여 ㈜포시에스 홈페이지를 나타내는 예제입니다.

- 디자인

  ➢ Board에 HTML View를 추가한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
URL	http://www.forcs.com /index.asp	-	-

- 미리보기



Splitter

전체 화면을 분할하고 분할된 영역 사이의 크기를 유동적으로 조절하고자 할 때 사용하는 컴포넌트입니다.

■ Example

Splitter를 이용하여 분할된 영역 사이의 크기를 유동적으로 조절합니다.

- 구현 기능

  Splitter를 이용하여 폼에 Docking된 두 개의 Panel의 크기를 유동적으로 조절합니다.

- 디자인

  ➢ Board에 Panel을 추가한 후 크기를 적절하게 조절하고 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
BackColor	128, 128, 255	**Dock**	Left

➢ Board에 Splitter를 추가한 후 너비를 적절하게 조절하고 속성 값을 아래와 같이 변경합니다.

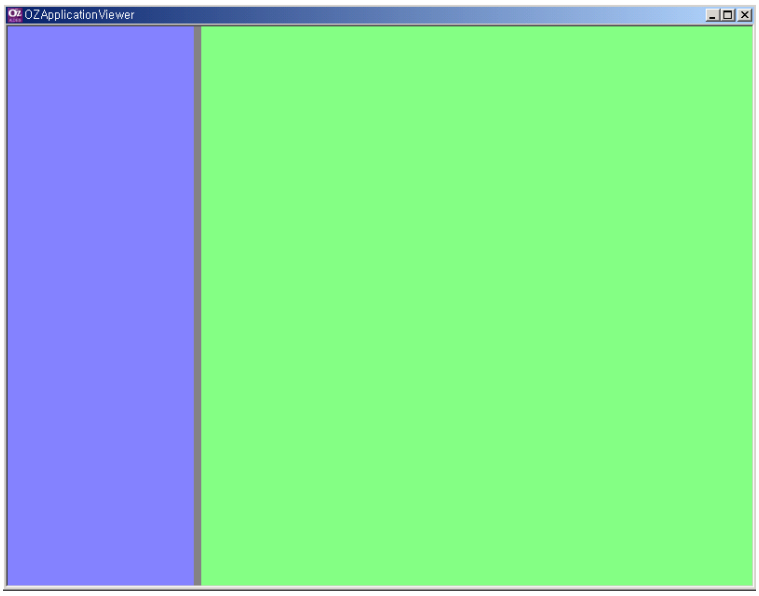
Property	Value	Property	Value
BackColor	128, 128, 128	**Cursor**	HSplit
Dock	Left	**-**	-

➢ Board에 Panel을 하나 더 추가한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
BackColor	128, 255, 128	**Dock**	Fill

- 미리보기

회색 부분의 Splitter에 마우스를 위치하면 마우스 커서가 화면 분할 모양으로 변경됩니다. 마우스로 분할 영역의 크기를 유동적으로 조절합니다.



Horizontal Navigator

Table 컴포넌트와 연동하여 Table에 표시될 Row수를 나누어 페이징시켜 데이터를 조회할 때 사용하는 컴포넌트로, 맨처음/이전/다음/맨뒤로 페이지를 이동시키는 Button과 페이지 번호를 직접 입력할 수 있는 TextBox로 구성된 컴포넌트입니다.

■ Property

Horizontal Navigator의 기본적인 속성은 아래와 같습니다.

Property	Description
LinkComponentName	Horizontal Navigato와 연동될 Table 컴포넌트명

■ Example

Horizontal Navigator를 이용하여 Table의 데이터를 페이징시켜 조회합니다.

- 구현 기능

  Horizontal Navigator를 이용하여 Table의 데이터를 한 페이지당 10개의 Row 데이터를 표시하도록 페이징시켜 조회합니다.

- 디자인

  ➢ 본 매뉴얼의 Ⅵ.장을 참조하여 프로젝트에 'Foodmart.odi' 파일을 추가하고 Board에 Table을 추가하여 Table에 'ODIKey' 속성 값은 ''Foodmart'로 'DataSet' 속성 값은 'ProductSales'로 설정한 후 Table 마법사를 실행하여 표시할 필드를 설정합니다.

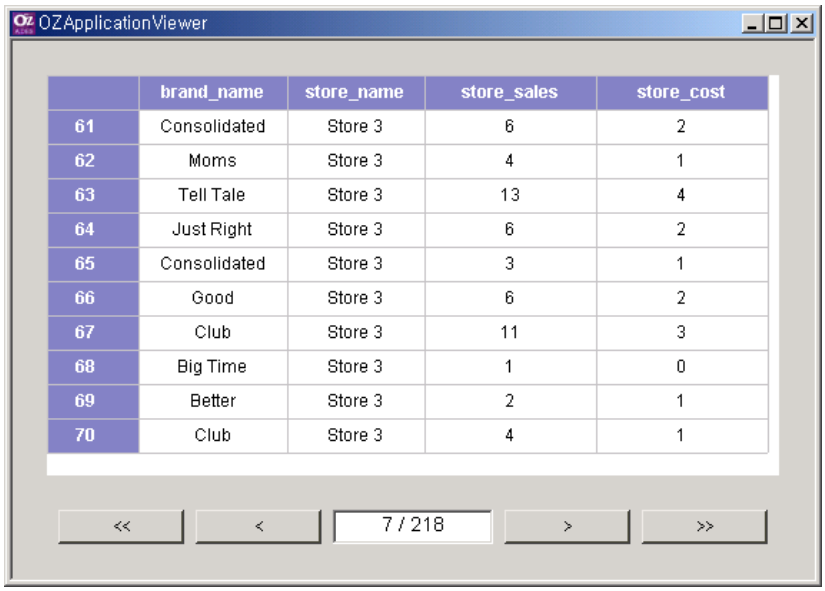
  ➢ Board에 추가된 Table(Name=Table1)을 선택한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
RowAutosize	FixCount	**RowCount**	10

  ➢ Board에 Horizontal Navigator를 추가한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
LinkComponentName	Table1	**-**	-

- 미리보기



Vertical Navigator

Table 컴포넌트와 연동하여 Table에 표시될 Row수를 나누어 페이징시켜 데이터를 조회할 때 사용하는 컴포넌트로, 맨처음/이전/다음/맨뒤로 페이지를 이동시키는 Button과 이전/다음으로 페이지를 이동시키는 VScrollBar로 구성된 컴포넌트입니다.

■ Property

Vertical Navigator의 기본적인 속성은 아래와 같습니다.

Property	Description
LinkComponentName	Horizontal Navigato와 연동될 Table 컴포넌트명

■ Example

Vertical Navigator를 이용하여 Table의 데이터를 페이징시켜 조회합니다.

- 구현 기능

  Vertical Navigator를 이용하여 Table의 데이터를 한 페이지당 10개의 Row 데이터를 표시하도록 페이징시켜 조회합니다.

- 디자인

  ➢ 본 매뉴얼의 Ⅵ.장을 참조하여 프로젝트에 'Foodmart.odi' 파일을 추가하고 Board에 Table을 추가하여 Table에 'ODIKey' 속성 값은 ''Foodmart'로 'DataSet' 속성 값은 'ProductSales'로 설정한 후 Table 마법사를 실행하여 표시할 필드를 설정합니다.

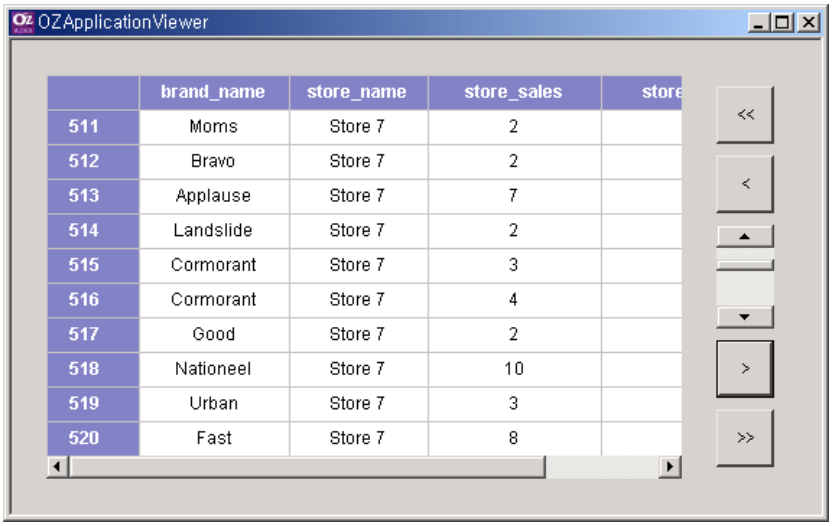
  ➢ Board에 추가된 Table(Name=Table1)을 선택한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
RowAutosize	FixCount	**RowCount**	10

  ➢ Board에 Vertical Navigator를 추가한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
LinkComponentName	Table1	**-**	-

- 미리보기



ActiveX Control

OZ Application Designer에서 제공하는 컴포넌트 외에 다른 객체를 사용할 수 있도록 하는 컴포넌트입니다. 컴포넌트를 Board에 추가하면 윈도우에 설치된 ActiveX 객체를 리스트 형태로 나타낸 다이얼로그가 표시되며, 해당 객체를 선택하면 선택된 객체별로 설정할 수 있는 속성이 'ActiveXControl' 카데고리의 하위 아이템으로 추가됩니다.

■ Method

ActiveX Control의 기본적인 메소드는 아래와 같습니다.

- GetOCX

  ActiveX Control의 OCX 객체를 가져옵니다.

  ➢ Format

  ```
  object GetOCX();
  ```

  ➢ Example

  ```
  var objOCX = ActiveXControl1.GetOCX();
  ```

■ Example

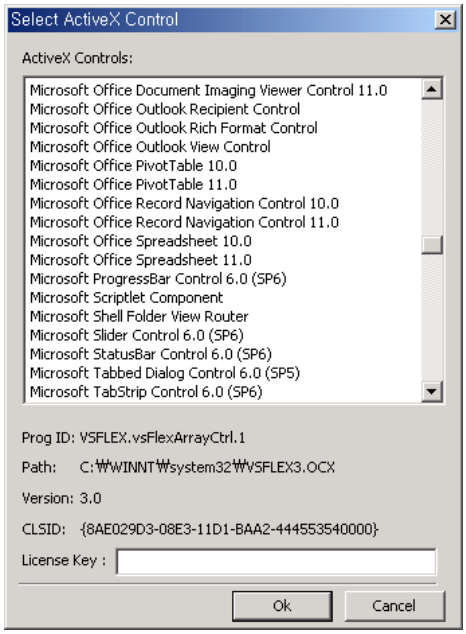
ActiveX Control를 이용하여 윈도우에 설치된 ActiveX 객체를 사용합니다.

- 구현 기능

  ActiveX Control를 이용하여 Microsoft Office Spreedsheet를 표시하고 셀에 숫자값을 입력하고 합계를 표시하는 예제입니다.

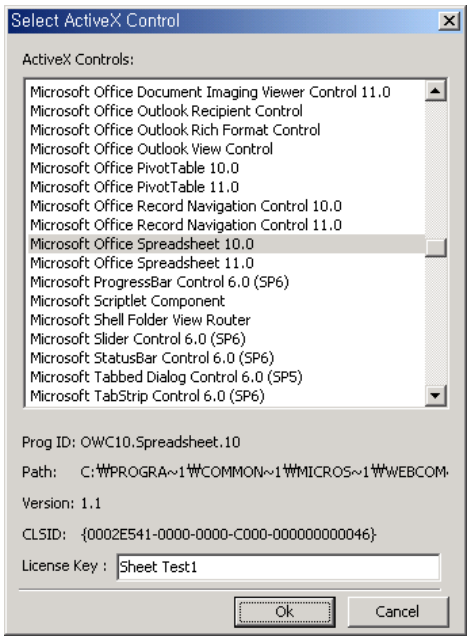
- 디자인

  ➢ Board에 ActiveX Control를 추가하면 아래와 같이 윈도우에 설치된 ActiveX 객체를 리스트 형태로 나타낸 다이얼로그가 표시됩니다.

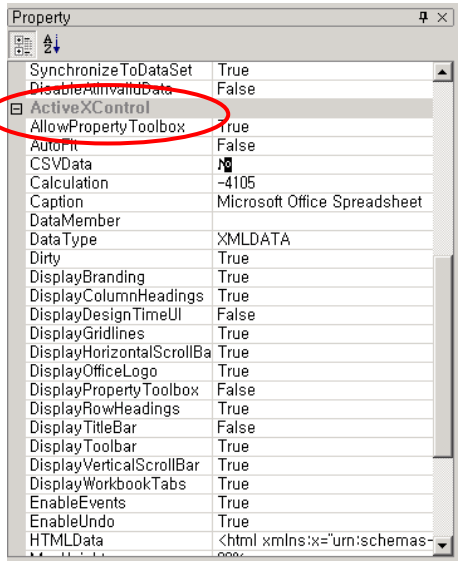


- License Key : 라이센스 키는 추가하려는 ActiveX의 키값 또는 고유한 이름으로, 같은 ClassID를 가진 다른 ActiveX와 구분하여 사용하고자 할 때 설정합니다.

➢ ActiveX 객체 중 'Microsoft Office Spreedsheet 10.0'을 선택하고, 필요에 따라 License Key를 입력한 후 [Ok] 버튼을 클릭합니다.



➢ 속성창에 'ActiveXControl' 카테고리로 'Microsoft Office Spreedsheet 10.0'에 대해 설정할 수 있는 속성이 표시됩니다.

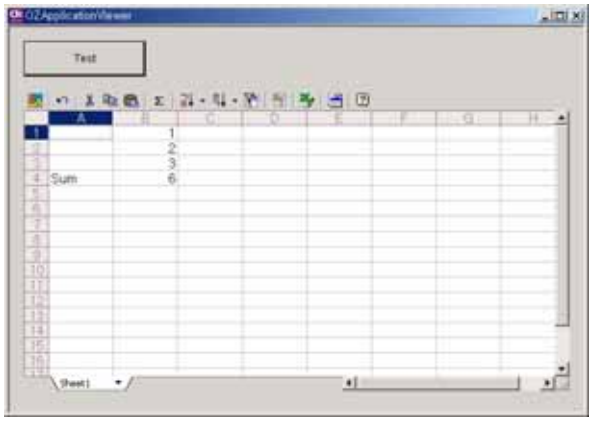


➢ Board에 Button을 추가하고, 추가한 Button의 'OnClick' 이벤트를 선택하여 아래와 같이 코딩합니다.

```
var activeX_Excel = ActiveXControl1.GetOCX();
activeX_Excel.Cells(1,2).Value = "1";
activeX_Excel.Cells(2,2).Value = "2";
activeX_Excel.Cells(3,2).Value = "3";
activeX_Excel.Cells(4,1).Value = "Sum";
activeX_Excel.Cells(4,2).Value = "=SUM(B1:B3)";
```

- 미리보기

'Microsoft Office Spreedsheet'를 표시하고, 버튼을 클릭하여 숫자를 입력하고 합계를 표시합니다.



Custom Control

OZ Application Designer에서 제공하는 컴포넌트 외에 사용자 컨트롤을 사용할 수 있도록 하는 컴포넌트입니다. 사용자 컨트롤 즉, 'CustomDll'이란 상용화된 pdf417.dll 파일과 같이 Drawing 로직을 구현한 Dll 파일을 말합니다. 특별하게 Drawing 로직을 구현해야 할 경우에는 Drawing 로직을 구현한 Dll 파일을 만들어 Custom Control을 사용하여 사용자가 원하는 형태로 그려지도록 할 수 있습니다. Custom Control에 연결할 수 있는 CustomDll 파일명은 반드시 'ozc_'로 시작하여야 하며, Custom Control에 연결할 수 있는 CustomDll 파일을 만드는 방법은 'API Developer's Guide'를 참조하십시오. Custom Control에 연결할 CustomDll 파일은 OZ Application Designer의 실행 파일이 있는 폴더 또는 윈도우의 시스템 폴더에 위치하면 됩니다. Custom Control에 연결할 사용자 컨트롤을 선택하면 선택된 사용자 컨트롤별로 설정할 수 있는 속성이 'CustomControl' 카테고리의 하위 아이템으로 추가됩니다.

■ Method

Custom Control의 기본적인 메소드는 아래와 같습니다.

- GetCustomProperty

  CustomDll의 설정한 속성에 해당되는 속성 값을 가져옵니다.

  ➢ Format

```
string GetCustomProperty(string PropertyName);
```

- SetCustomProperty

  CustomDll의 속성 값을 설정한 값으로 변경합니다.

  ➢ Format

```
void SetCustomProperty(string PropertyName,
                       string PropertyValue);
```

■ Example

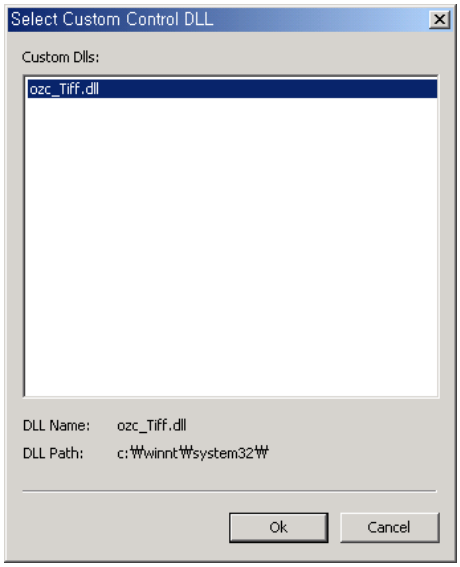
Custom Control을 이용하여 CustomDll 파일을 연동합니다.

- 구현 기능

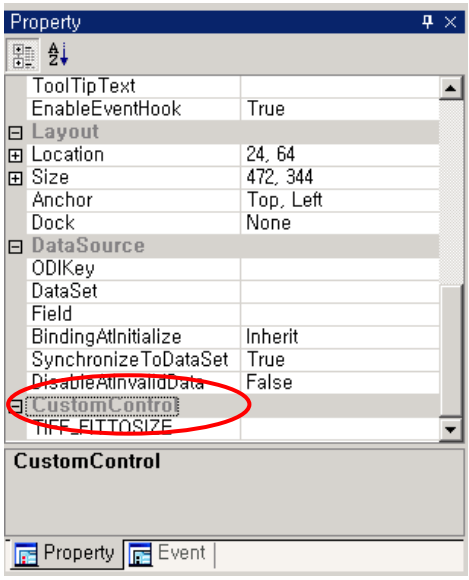
  버튼을 클릭하면 파일 선택 다이얼로그가 표시되고 TIFF 타입의 이미지 파일을 선택합니다. Custom Control에 TIFF 타입의 이미지 파일(.tif 파일)을 표시하고, Custom Control의 크기에 맞춰 이미지 크기를 조절하여 표시할지 여부를 선택하는 예제입니다. 본 매뉴얼의 예제는 'ozc_Tiff.dll' 이라는 CustomDll 파일을 사용하였으며, 'ozc_Tiff.dll' 파일은 고객 지원 센터로 문의 바랍니다.

- 디자인

➢ Board에 Custom Control(Name=CustomControl1)를 추가하면 아래와 같이 연동할 수 있는 CustomDll 파일을 리스트 형태로 나타낸 다이얼로그가 표시됩니다.



➢ ozc_Tiff.dll 파일을 선택한 후 [Ok] 버튼을 클릭하면, 속성창에 'CustomControl' 카테고리로 설정할 수 있는 속성이 표시됩니다.



➢ Board에 Button을 추가하고, Text를 'Select File'로 변경합니다.

➢ Board에 CheckBox(Name=CheckBox1)를 추가하고, 'Text'를 'Fit To Size'로 변경한 후 클릭하면 Custom Control의 크기에 맞춰 이미지 크기를 조절하여 표시하

도록 추가한 CheckBox의 'OnClick' 이벤트를 선택하여 아래와 같이 코딩합니다.

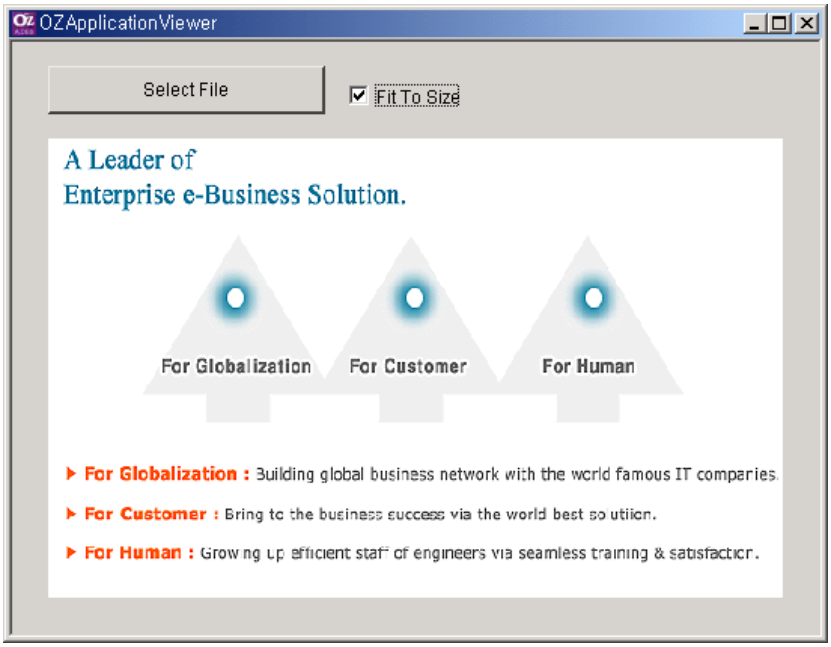
```
if(CheckBox1.Checked == true) {
    CustomControl1.SetCustomProperty("TIFF_FITTOSIZE", "true");
} else {
    CustomControl1.SetCustomProperty("TIFF_FITTOSIZE",
    "false");
}
```

➢ [Select File] 버튼을 클릭하면 파일 선택 다이얼로그를 표시하고 선택한 TIFF 이미지를 표시하도록 추가한 Button의 'OnClick' 이벤트를 선택하여 아래와 같이 코딩합니다.

```
var SelectedFile = _FileDialog(true, "", "", "","TIF Files
(*.tif)|*.tif");
if(CheckBox1.Checked == true) {
    CustomControl1.SetCustomProperty("TIFF_FITTOSIZE", "true");
} else {
    CustomControl1.SetCustomProperty("TIFF_FITTOSIZE",
    "false");
}
CustomControl1.Value = SelectedFile;
```

- 미리보기

[Select File] 버튼을 클릭하여 TIFF 이미지 파일을 선택하여 이미지를 표시하고, 'Fit To Size' 옵션도 변경합니다.



Report Control

Report Control은 OZ Report Designer로 작성된 보고서 파일(.ozr)을 리포트 뷰어와 연동하여 미리보기하는 컴포넌트입니다. 'ReportViewerOption' 속성 값을 변경하여 리포트 뷰어 옵션을 설정하거나 Report Control의 'OnPreApplyReport' 이벤트에 적용할 옵션을 스크립트로 코딩하고 'ApplyReport' 메소드를 호출하여 해당 옵션을 적용할 수 있으며, 적용할 옵션명과 해당 값에 대한 설명은 'OZ Report Viewer Guide'를 참조하십시오.
Report Control을 사용하기 위해서는 리포트 뷰어가 설치되어 있어야 하며, 리포트 뷰어 설치 방법은 'OZ Report Viewer Guide'를 참조하거나 고객 지원 센터로 문의 바랍니다.

■ Property

Property	Description
ReportViewerOption	리포트 뷰어의 옵션 설정 리포트 뷰어 옵션 설정 다이얼로그를 이용하여 화면에 표시할 리포트 뷰어의 연결 정보, 툴바 등의 옵션을 설정

■ Method

Report Control의 기본적인 메소드는 아래와 같습니다.

- AddReportParameter

  Report Control에 적용할 옵션 즉, Parameter를 추가합니다.

  ➢ Format

  ```
  void AddReportParameter(string ParameterName, string ParameterValue);
  ```

  ▪ ParameterName : 적용할 옵션명

  ▪ ParameterValue : 적용할 옵션값

  ➢ Example

  ```
  //리포트 뷰어창에 툴바 표시하지 않음
  ReportControl1.AddReportParameter("toolbar.all","false");
  //프린트 옵션창 없이 바로 인쇄하며 인쇄 과정을 ProgressBar로 표시
  ReportControl1.AddReportParameter("print.mode","showprogress");
  ```

- InitUserParameter

  사용자가 스크립트로 추가한 옵션 즉, AddReportParameter()메소드를 이용하여 추가한 옵션을 초기화합니다.

➢ Format

```
void InitUserParameter();
```

➢ Example

```
ReportControl1.InitUserParameter();
```

- ApplyReport

Report Control의 'OnPreApplyReport'에 스크립트로 코딩한 옵션을 적용하여 리포트를 바인딩합니다.

➢ Format

```
bool ApplyReport(bool RecreateReportFlag);
```

▪ RecreateReportFlag : 이전에 생성된 리포트 뷰어의 삭제 여부로, 만일 Parameter를 변경하여 리포트 뷰어의 툴바 표시 등 User Interface를 변경하려면 RecreateReportFlag 값을 반드시 true로 설정하십시오.

➢ Example

```
ReportControl1.ApplyReport();
```

■ Example

Report Control를 이용하여 리포트 파일을 표시합니다.

- 구현 기능

Report Control를 이용하여 ㈜포시에스의 샘플용 리포트 파일을 표시하는 예제입니다.

- 디자인

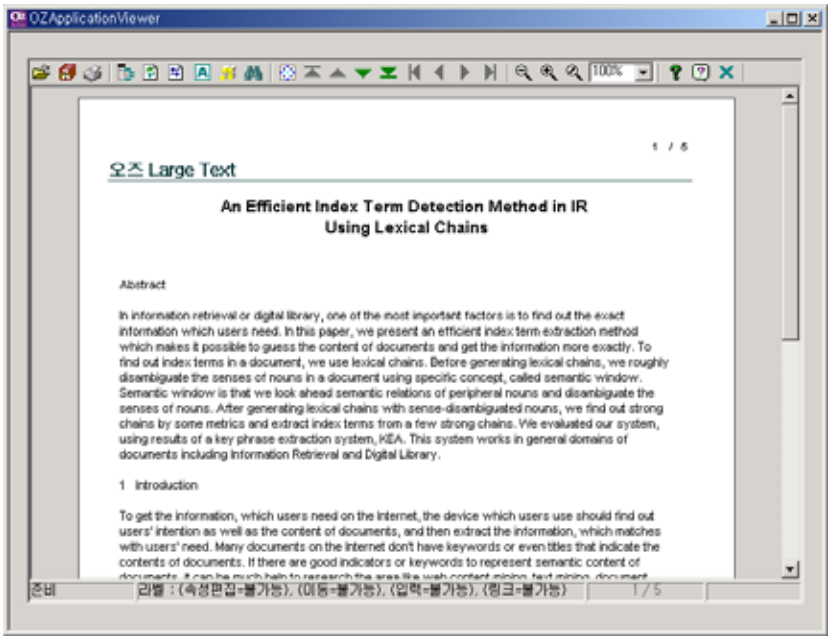
➢ Board에 Report Control을 추가한 후 'OnPreApplyReport' 이벤트를 선택하여 아래와 같이 코딩합니다.

```
ReportControl1.InitUserParameter();
ReportControl1.AddReportParameter("connection.servlet",
"http://127.0.0.1:8080/ozdemo/server");
ReportControl1.AddReportParameter("connection.reportname",
"/ozdemo/kr/A/08NOTE.ozr");
```

➢ Board가 화면에 표시될 때 Report Control에 설정한 옵션값이 적용되도록 Board의 'OnAttached' 이벤트를 선택하여 아래와 같이 코딩합니다.

```
ReportControl1.ApplyReport();
```

- 미리보기



ReportButton

ReportButton은 OZ Report Designer로 작성된 보고서 파일(.ozr)을 미리보기하거나 미리보기 하지 않고 바로 인쇄 또는 다른 파일로 Export하는 컴포넌트입니다. 'ReportViewerOption' 속성 값을 변경하여 인쇄 또는 Export 옵션을 설정하거나 ReportButonl의 'OnPreApplyReport' 이벤트에 적용할 옵션을 스크립트로 코딩하고 'ApplyReport' 메소드를 호출하여 해당 옵션을 적용할 수 있으며, 적용할 옵션명과 해당 값에 대한 설명은 'OZ Report Viewer Guide'를 참조하십시오.

ReportButton을 사용하기 위해서는 리포트 뷰어가 설치되어 있어야 하며, 리포트 뷰어 설치 방법은 'OZ Report Viewer Guide'를 참조하거나 고객 지원 센터로 문의 바랍니다.

■ Property

Property	Description
ReportViewerOption	미리보기, 인쇄, Export 옵션 설정 리포트 뷰어 옵션 설정 다이얼로그를 이용하여 미리보기, 인쇄, Export 등의 옵션을 설정

ReportViewerMode	리포트 버튼을 클릭하였을 경우 리포트 뷰어의 실행 형태를 설정 ∗ Print : 보고서 인쇄 다이얼로그 표시 ∗ Export : 보고서 저장 다이얼로그 표시 ∗ Preview : 리포트 뷰어에 보고서 표시 ※ 참고사항 : ReportViewerMode 속성 값을 Preview로 설정할 경우 웹 브라우저와 동일한 프로세스로 리포트 뷰어가 실행됩니다.

■ Method

ReportButton의 기본적인 메소드는 아래와 같습니다.

- AddReportParameter

  ReportButton에 적용할 옵션 즉, Parameter를 추가합니다. 적용할 옵션명과 해당 값에 대한 설명은 'OZ Report Viewer Guide'를 참조하십시오.

  ➢ Format

  ```
  void AddReportParameter(string ParameterName, string ParameterValue);
  ```

  - ParameterName : 적용할 옵션명
  - ParameterValue : 적용할 옵션값

  ➢ Example

  ```
  //프린트 옵션창 없이 바로 인쇄하며 인쇄 과정을 ProgressBar로 표시
  ReportButton1.AddReportParameter("print.mode","showprogress");
  ```

- InitUserParameter

  사용자가 스크립트로 추가한 옵션 즉, AddReportParameter()메소드를 이용하여 추가한 옵션을 초기화합니다.

  ➢ Format

  ```
  void InitUserParameter();
  ```

  ➢ Example

  ```
  ReportButton1.InitUserParameter();
  ```

- ApplyReport

  ReportButton의 'OnPreApplyReport'에 스크립트로 코딩한 옵션을 적용하여 리포트를 바인딩합니다.

➢ Format

```
bool ApplyReport();
```

➢ Example

```
ReportButton1.ApplyReport();
```

■ Example

ReportButton을 이용하여 리포트 파일을 다른 파일로 Export합니다.

- 구현 기능

ReportButton을 이용하여 ㈜포시에스의 샘플용 리포트 파일을 다른 파일로 Export하는 예제입니다.

- 디자인

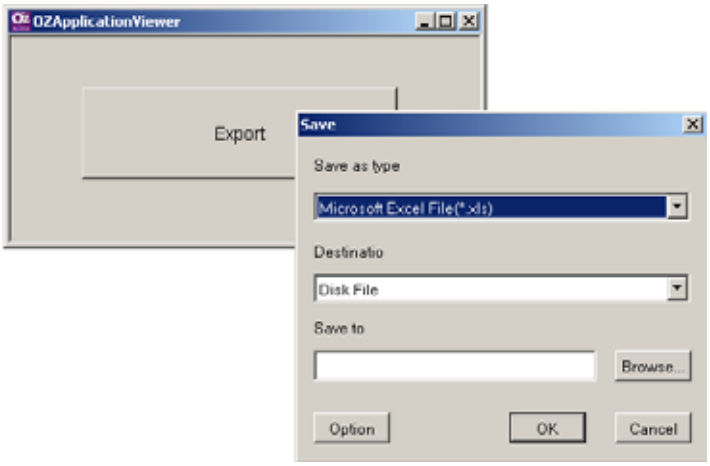
➢ Board에 ReportButton을 추가한 후 'Text' 속성 값을 'Export'로 변경하고 'ReportViewerMode' 속성 값을 'Export'로 변경합니다.

➢ ReportButton의 'OnPreApplyReport' 이벤트를 선택하여 아래와 같이 코딩합니다.

```
ReportButton1.InitUserParameter();
ReportButton1.AddReportParameter("connection.servlet",
"http://127.0.0.1:8080/oz/server");
ReportButton1.AddReportParameter("connection.reportname",
"/ozdemo30/kr/A/08NOTE.ozr");
```

- 미리보기

Export 버튼을 클릭하면 아래 그림과 같이 다른 파일로 Export할 수 있는 다이얼로그가 표시되며 리포트 파일을 다른 파일로 Export할 수 있습니다.



Invisibles 컴포넌트

Timer

규칙적인 시간 간격으로 특정한 동작을 실행하고자 할 때 사용하는 컴포넌트입니다. Timer의 동작 주기가 되었을 경우 Timer의 'OnTimer' 이벤트에 코딩한 스크립트 내용을 실행합니다.

■ Property

Timer의 기본적인 속성은 아래와 같습니다.

Property	Description
Enabled	Timer의 동작 여부
Interval	Timer의 동작 주기 즉 시간 간격 밀리세컨드(ms) 단위로 설정, 3초의 경우 3000으로 설정

■ Method

Timer의 기본적인 메소드는 아래와 같습니다.

- Start

  Timer의 동작을 시작합니다. Timer의 'Enabled' 속성 값을 'True'로 변경시킵니다.

  ➢ Format

  ```
  void Start();
  ```

  ➢ Example

  ```
  Timer1.Start();
  ```

- Stop

  Timer의 동작을 중지합니다. Timer의 'Enabled' 속성 값을 'False'로 변경시킵니다.

  ➢ Format

  ```
  void Stop();
  ```

  ➢ Example

  ```
  Timer1.Stop();
  ```

■ Example

Timer를 이용하여 ProgressBar의 위치값을 변경하는 예제입니다.

- 구현 기능

  버튼을 이용하여 Timer의 동작을 제어하고 0.5초를 시간 간격으로 하여 ProgressBar의 위치값을 변경하여 진행 상태를 표시합니다.

- 디자인

  ➢ Document에 Timer를 추가한 후 'Interval' 속성 값을 '500'으로 변경합니다.

  ➢ Board에 Button을 추가하고, 'Text' 속성 값을 'Start'라고 변경한 후 'OnClick' 이벤트를 선택하여 아래와 같이 코딩합니다.

  ```
  Timer1.Start();
  ```

  ➢ Board에 Button을 하나 더 추가하고, 'Text' 속성 값을 'Stop'이라고 변경한 후 'OnClick' 이벤트를 선택하여 아래와 같이 코딩합니다.

  ```
  Timer1.Stop();
  ```

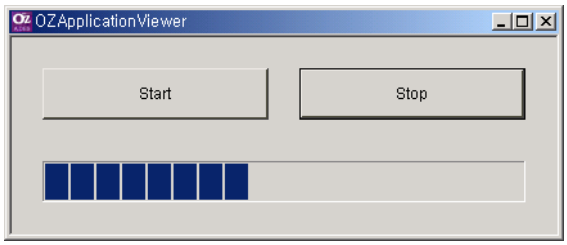
  ➢ Board에 ProgressBar를 추가합니다.

  ➢ Timer의 'OnTimer' 이벤트를 선택하여 아래와 같이 코딩합니다.

  ```
  var objProgressBar=_GetCurrentBoard().GetChild("ProgressBar1");
  if( !objProgressBar.IsFullPos() ) {
      objProgressBar.StepIt();
  } else {
      objProgressBar.Pos = objProgressBar.Minimum;
  }
  ```

- 미리보기

  [Start], [Stop] 버튼으로 Timer의 동작을 제어하고 시간 간격으로 진행바의 위치값을 이동합니다.



ContextMenu

팝업 메뉴를 표시할 때 사용하는 컴포넌트입니다. 마법사를 이용하여 메뉴의 아이템을 추가하며, 마법사에 대한 자세한 사용법은 본 매뉴얼의 Ⅶ.장 마법사 사용법을 참조하십시오. ContextMenu는 다른 컴포넌트에서 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴를 표시할 때 사용하며, 팝업 메뉴를 표시할 다른 컴포넌트의 속성 중 'ContextMenu' 속성 값에 해당 ContextMenu의 이름을 설정하면 자동으로 팝업 메뉴로 표시됩니다. ContextMenu의 아이템을 클릭하였을 때 실행할 스크립트는 각 메뉴 아이템의 'OnCommand' 이벤트에 코딩합니다.

■ Example

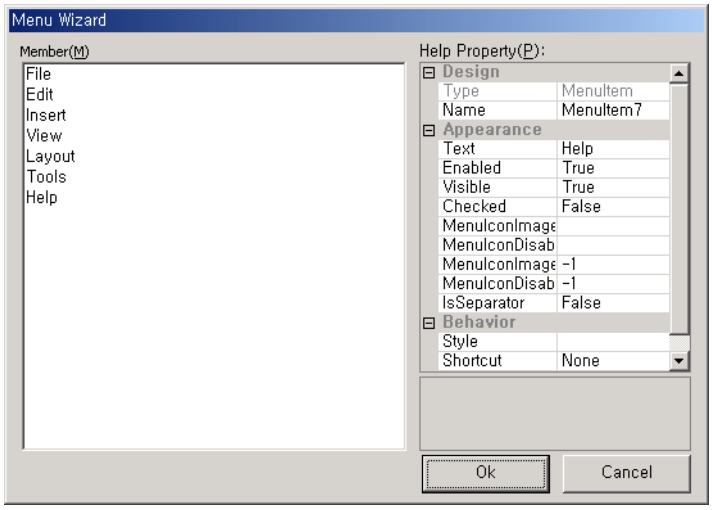
ContextMenu를 이용하여 Board에 팝업 메뉴를 표시하는 예제입니다.

- 구현 기능

  ContextMenu 마법사를 이용하여 메뉴 아이템을 추가하고 Board의 팝업 메뉴로 표시합니다.

- 디자인

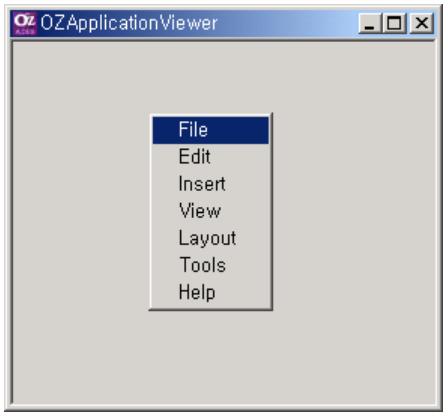
  ➢ Document에 ContextMenu(Name=ContextMenu1)를 추가한 후 마법사를 이용하여 메뉴 아이템을 아래 그림과 같이 추가합니다.



  ➢ Board의 속성 값 중 'ContextMenu' 속성 값을 'ContextMenu1'로 변경합니다.

- 미리보기

  Board에서 마우스 오른쪽 버튼을 클릭하면 아래 그림처럼 팝업 메뉴가 표시됩니다.



MainMenu

Board 상단에 메뉴를 표시할 때 사용하는 컴포넌트입니다. 마법사를 이용하여 메뉴의 아이템을 추가하며, 마법사에 대한 자세한 사용법은 본 매뉴얼의 Ⅶ.장 마법사 사용법을 참조하십시오. MainMenu를 표시할 Board의 속성 중 'MainMenu' 속성 값에 해당 MainMenu의 이름을 설정하면 화면 상단에 메뉴바로 표시됩니다. MainMenu의 아이템을 클릭하였을 때 실행할 스크립트는 각 메뉴 아이템의 'OnCommand' 이벤트에 코딩합니다.

■ Example

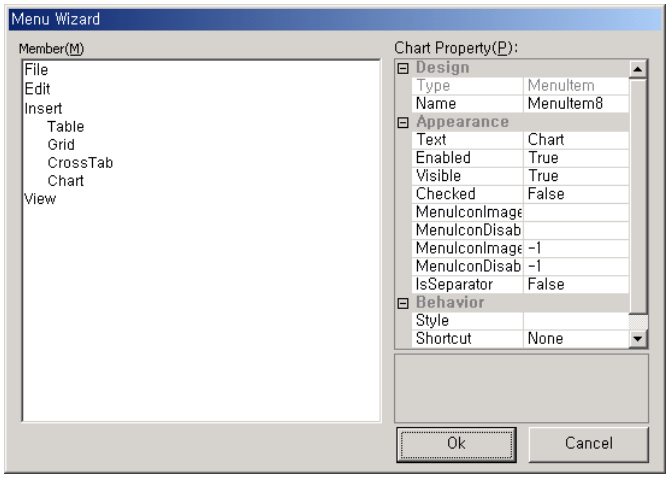
MainMenu를 이용하여 Board의 상단에 메뉴바를 표시하는 예제입니다.

- 구현 기능

  MainMenu 마법사를 이용하여 메뉴 아이템을 추가하고 Board의 메뉴바로 표시합니다.

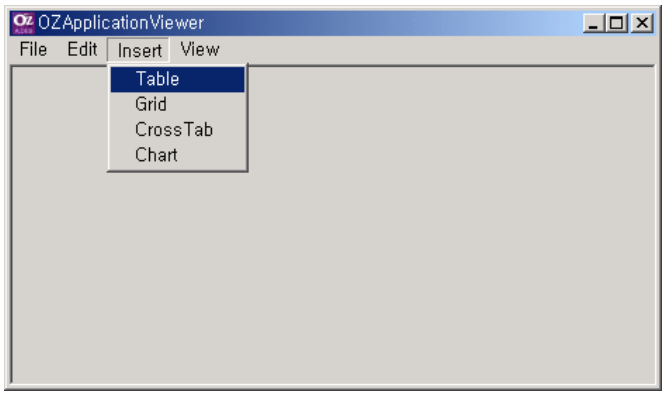
- 디자인

  ➢ Document에 MainMenu(Name=MainMenu1)를 추가한 후 마법사를 이용하여 메뉴 아이템을 아래 그림과 같이 추가합니다.



➢ Board의 속성 값 중 'MainMenu' 속성 값을 'MainMenu1'로 변경합니다.

- 미리보기



TrayContextMenu

뷰어가 종료될 때 뷰어를 종료하지 않고 트레이에 숨기고자 할 때 사용하는 컴포넌트입니다. 마법사를 이용하여 메뉴의 아이템을 추가하며, 마법사에 대한 자세한 사용법은 본 매뉴얼의 Ⅶ.장 마법사 사용법을 참조하십시오. 뷰어 트레이 기능을 사용하기 위해서는 Document의 'TrayContextMenu' 속성 값에 연결할 TrayContextMenu를 설정하고, TrayContextMenu 마법사를 이용하여 메뉴 아이템을 추가하여야 하며, 뷰어 패러미터 중 'ozadoc.use_tray' 패러미터 값을 true로 설정하여야 합니다.

■ Example

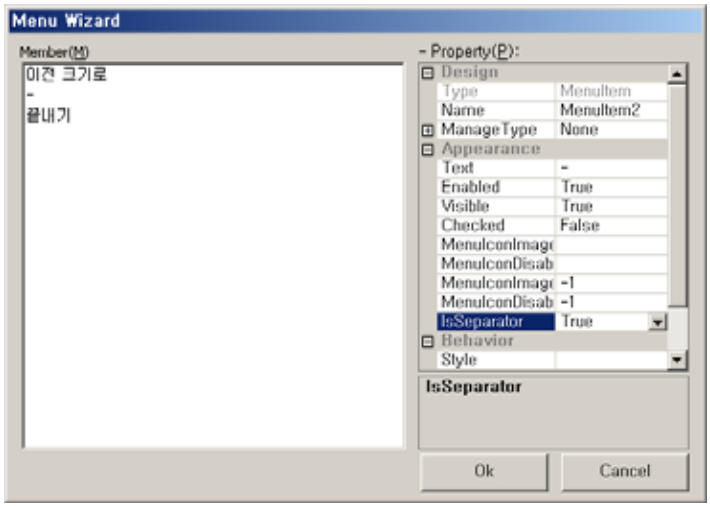
TrayContextMenu를 이용하여 뷰어를 종료하지 않고 트레이에 숨기는 예제입니다. 본 매뉴얼에서는 param.oz 파일을 이용하여 EXE 뷰어를 단독으로 실행하는 경우를 예로 들어 설명합니다.

- 구현 기능

  TrayContextMenu 마법사를 이용하여 메뉴 아이템을 추가하고 트레이에 숨깁니다.

- 디자인

  ➢ Document에 TrayContextMenu(Name=TrayContextMenu1)을 추가한 후 마법사를 이용하여 메뉴 아이템을 아래 그림과 같이 추가합니다.



※ 참고사항 : 두 번째 아이템은 메뉴 구분자로 설정하기 위해 'IsSeparator' 속성 값을 'True'로 변경하시기 바랍니다.

  ➢ TrayContextMenu1의 속성 창에서 'ToolTipText' 속성 값을 'G 프로젝트 전용 뷰어'로 설정하고 'TrayAction' 속성 값을 'HideAtClose'로 설정합니다.

  ➢ TrayContextMenu의 MenuItem1의 OnCommand 이벤트에 동작시킬 스크립트를 다음과 같이 입력합니다.

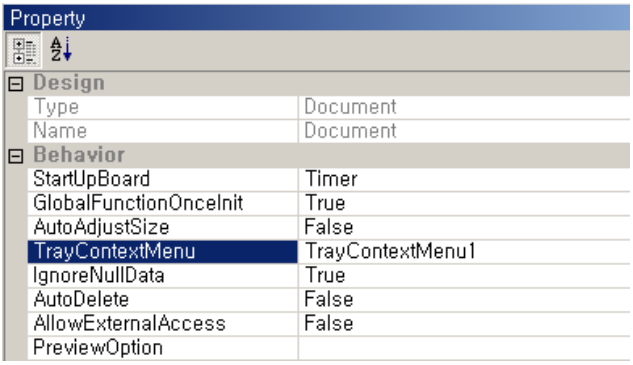
```
//이전 크기로
  _ShowOZAViewerFromTray();
```

  ➢ TrayContextMenu의 MenuItem3의 OnCommand 이벤트에 동작시킬 스크립트를 다음과 같이 입력합니다.

```
//끝내기
```

```
_ExitOZAViewer(false);
```

- Document의 'TrayContextMenu' 속성에 'TrayContextMenu1'을 설정합니다.



- 완성된 프로젝트를 'c:\sample.oza'로 저장합니다.

- 애플리케이션 EXE 뷰어(OZAViewer.exe) 파일이 설치된 경로에 메모장 등의 편집 프로그램을 이용하여 아래 내용을 입력한 후 param.oz 파일로 저장합니다.

```
repository_agent.local.root_path=c:/
ozadoc.path=sample.oza
ozadoc.use_tray=true
```

- 애플리케이션 EXE 뷰어(OZAViewer.exe) 파일을 더블클릭하여 뷰어를 실행한 후 뷰어 트레이에 뷰어 아이콘이 표시되고, 타이틀바에 있는 뷰어 종료 버튼[ ]을 클릭하여 뷰어 종료 시 뷰어가 종료되지 않고 트레이에 숨는 것을 확인합니다.

Tooltip

컴포넌트에 툴팁 즉, 풍선 도움말을 표시할 때 사용하는 컴포넌트입니다. 실제 컴포넌트에 툴팁을 표시하려면 컴포넌트의 'ToolTip' 속성 값에 Tooltip 컴포넌트의 이름을 설정하고, 'ToolTipText' 속성 값에 표시할 풍선 도움말을 입력합니다.

■ Example

Tooltip을 이용하여 Button에 풍선 도움말을 표시하는 예제입니다.

- 구현 기능

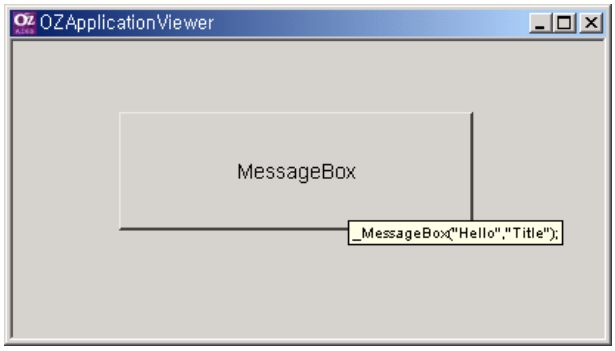
  Tooltip을 이용하여 Button에 풍선 도움말을 표시합니다.

- 디자인

  ➢ Document에 Tooltip(Name=ToolTip1)을 추가합니다.

  ➢ Board에 Button을 추가한 후 'ToolTip' 속성 값을 'ToolTip1'로 변경하고 'ToolTipText' 속성 값을 '_MessageBox("Hello","Title");'로 입력합니다.

- 미리보기



ImageList

ContextMenu, MainMenu, TreeView, CheckBox 등의 각 아이템 앞에 이미지를 표시할 때 사용하는 컴포넌트입니다. 각 아이템에 표시할 이미지를 동일한 크기로 배열해 놓은 하나의 이미지 파일을 'ImageSize' 속성 값에 설정한 크기대로 분할하여 인덱스 번호로 표시합니다. ImageList와 연동하여 이미지를 표시할 컴포넌트의 속성 중 'ImageList' 속성 값을 ImageList 이름으로 설정하고, 'MenuIconImageIndex' 또는 'MenuIconDisableImageIndex' 속성 값에 인덱스 번호를 입력하여 이미지를 표시합니다.

■ Property

ImageList의 기본적인 속성은 아래와 같습니다.

Property	Description

ImageSize	이미지를 분할할 크기
ImageListUrl	연동할 이미지 리스트 파일명

■ Example

ImageList를 이용하여 MainMenu의 아이템 앞에 아이콘을 표시하는 예제입니다.

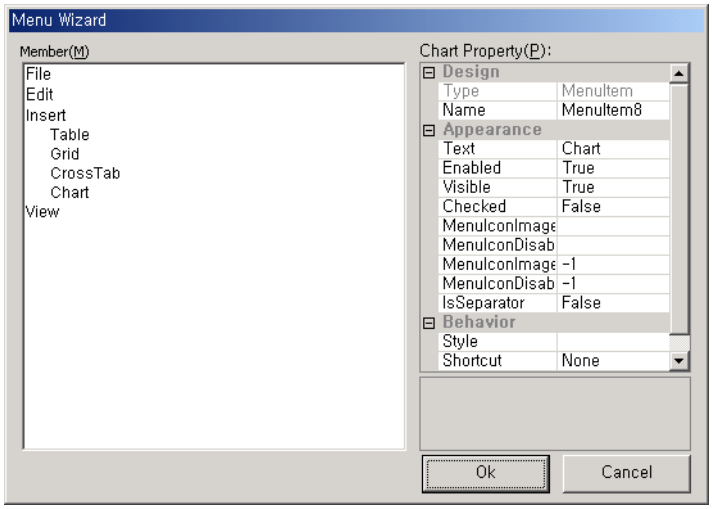
- 구현 기능

  ImageList를 이용하여 MainMenu의 아이템 앞에 아이콘을 표시합니다.

- 디자인

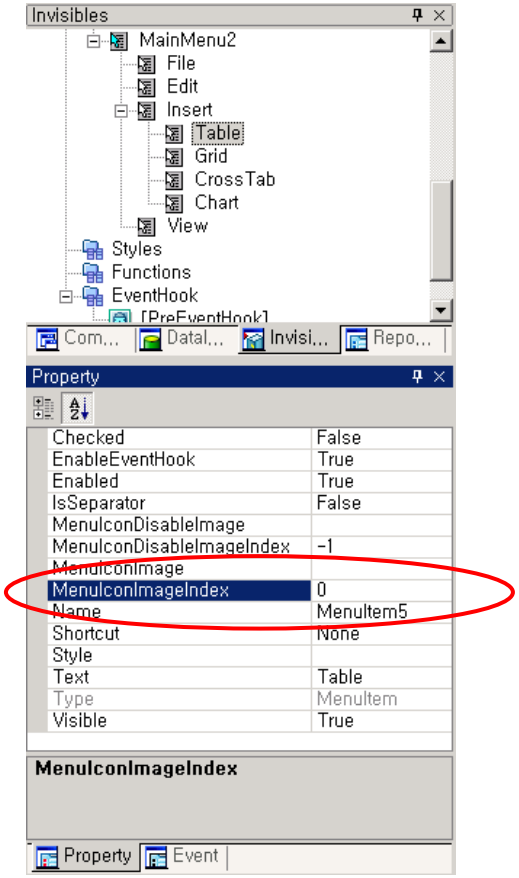
  ➢ Document에 ImageList(Name=ImageList1)을 추가한 후 'ImageSize' 속성 값을 '20, 20'으로, 'ImageListUrl' 속성 값을 'ozp:///ComplexControls.gif'로 변경합니다.

  ➢ Document에 MainMenu(Name=MainMenu2)를 추가한 후 마법사를 이용하여 메뉴 아이템을 아래 그림과 같이 추가합니다.



  ➢ MainMenu2의 속성 값 중 'ImageList' 속성 값을 'ImageList1'로 변경합니다.

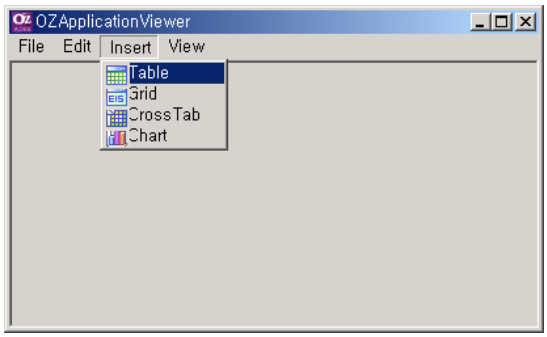
  ➢ Invisibles 트리창에서 MainMenu2에서 Table을 나타내는 메뉴 아이템을 선택한 후 속성창에서 'MenuIconImageIndex'의 속성 값을 '0'으로 변경합니다.



➢ 위와 같은 방법으로 Grid의 메뉴 아이템의 'MenuIconImageIndex'의 속성 값을 '1'로, CrossTab의 메뉴 아이템의 'MenuIconImageIndex'의 속성 값을 '2'로, Chart의 메뉴 아이템의 'MenuIconImageIndex'의 속성 값을 '3'으로 변경합니다.

➢ Board의 속성 값 중 'MainMenu' 속성 값을 'MainMenu2'로 변경합니다.

- 미리보기



TitleBar

Board, Panel, HBox, VBox, GroupBox, BoardView의 타이틀바에서 최소화, 최대화 기능 외에 아이콘 등을 표시하여 다른 기능을 구현할 때 사용하는 컴포넌트입니다. 마법사를 이용하여 타이틀바의 아이템을 추가하며, 마법사에 대한 자세한 사용법은 본 매뉴얼의 Ⅶ.장 마법사 사용법을 참조하십시오.

■ Example

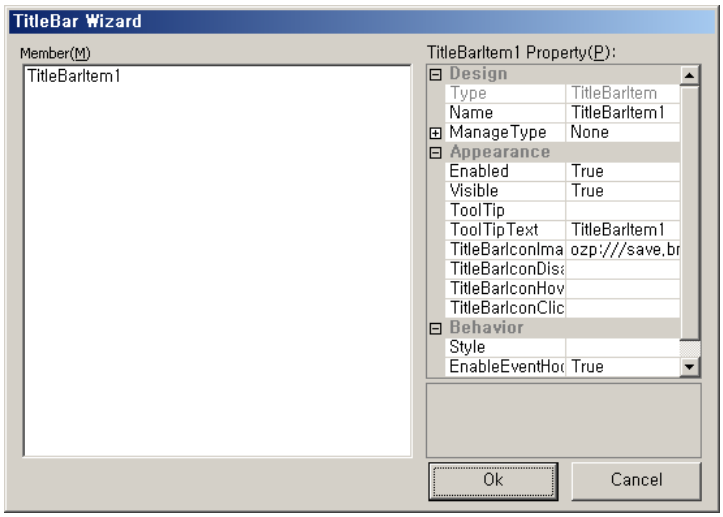
TitleBar를 이용하여 Panel의 Table을 엑셀 파일로 저장하는 예제입니다.

- 구현 기능

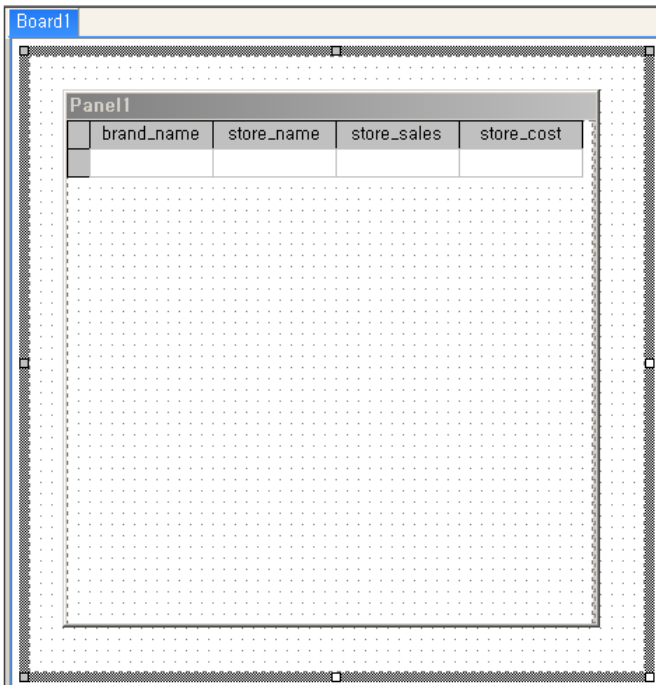
  TitleBar 마법사를 이용하여 타이틀바 아이템을 추가하고 Panel의 Table을 엑셀 파일로 저장합니다.

- 디자인

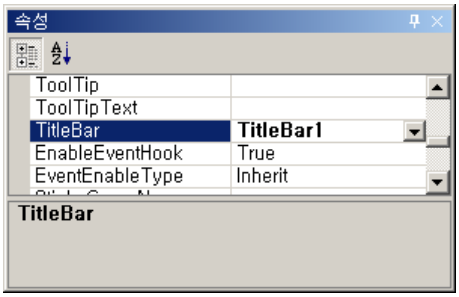
  ➢ Document에 TitleBar(Name=TitleBar1)를 추가한 후 마법사를 이용하여 타이틀바 아이템(Name=TitleBarItem1)을 아래 그림과 같이 추가합니다.



  ➢ Board에 Panel(Name=Panel1)을 추가한 후 Panel에 Table(Name=Table1)을 아래와 같이 추가합니다.



➢ Panel의 'TitielBar' 속성 값을 'TitleBar1'로 변경합니다.

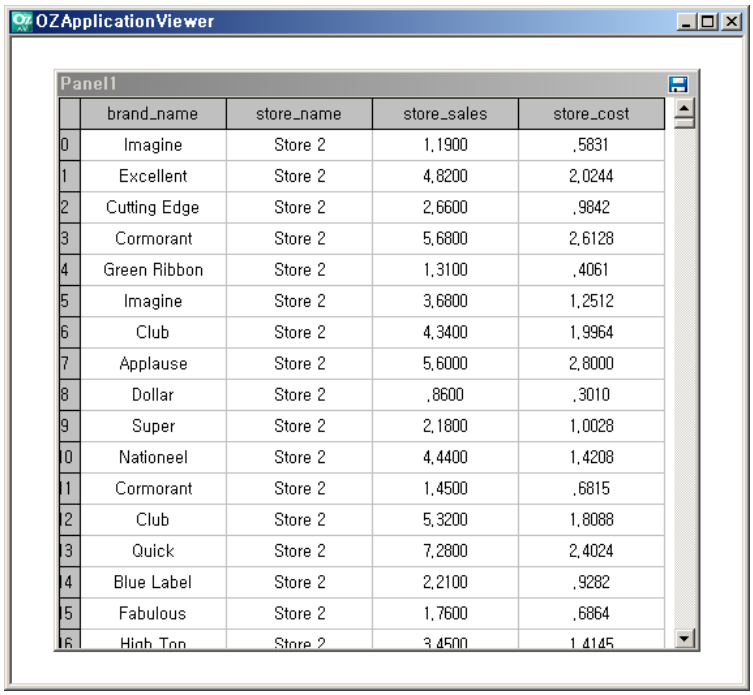


➢ TitleBarItem1의 'OnClick' 이벤트를 선택하여 아래와 같이 코딩합니다.

```
var tlb = Board1.GetChild("Table1");
tlb.SelectCellRange(-1,-1,-1,-1);
tlb.SaveExcelToFilePathInRange("C:/Temp/Sample.xls");
```

- 미리보기

TatleBar의 파일 저장 아이콘()을 클릭하여 Table을 엑셀 파일로 저장합니다.



	brand_name	store_name	store_sales	store_cost
0	Imagine	Store 2	1,1900	,5831
1	Excellent	Store 2	4,8200	2,0244
2	Cutting Edge	Store 2	2,6600	,9842
3	Cormorant	Store 2	5,6800	2,6128
4	Green Ribbon	Store 2	1,3100	,4061
5	Imagine	Store 2	3,6800	1,2512
6	Club	Store 2	4,3400	1,9964
7	Applause	Store 2	5,6000	2,8000
8	Dollar	Store 2	,8600	,3010
9	Super	Store 2	2,1800	1,0028
10	Nationeel	Store 2	4,4400	1,4208
11	Cormorant	Store 2	1,4500	,6815
12	Club	Store 2	5,3200	1,8088
13	Quick	Store 2	7,2800	2,4024
14	Blue Label	Store 2	2,2100	,9282
15	Fabulous	Store 2	1,7600	,6864
16	High Top	Store 2	3,4500	1,4145

Container Controls 컴포넌트

Panel, GroupBox

다수의 컴포넌트를 자식 컴포넌트로 포함시킬 때 사용하는 컴포넌트입니다. Board에 Panel을 추가하고 자식 컴포넌트로 포함시킬 컴포넌트를 Panel 영역 안에 추가하면 해당 컴포넌트는 Panel의 자식 컴포넌트로 추가됩니다. GroupBox도 Panel과 동일한 기능을 하는 컴포넌트이며, 본 매뉴얼에서는 Panel을 예로 들어 설명합니다.

■ Method

Panel의 기본적인 메소드는 아래와 같습니다.

- DeleteAllChild

  Panel에 추가된 모든 자식 컴포넌트를 삭제합니다.

  ➢ Format

  ```
  void DeleteAllChild();
  ```

  ➢ Example

  ```
  Panel1.DeleteAllChild();
  ```

- GetChild

  설정한 컴포넌트 이름에 해당하는 자식 컴포넌트를 얻어옵니다.

  ➢ Format

  ```
  object GetChild(string ComponentName);
  ```

  ➢ Example

  ```
  var objLabel = Panel1.GetChild("Label1");
  ```

- GetChildAt

  설정한 인덱스에 해당하는 자식 컴포넌트를 얻어옵니다.

  ➢ Format

  ```
  object GetChildAt(integer ComponentIndex);
  ```

  ➢ Example

```
var objComp = Panel1.GetChildAt(5);
```

- GetChildCount

  Panel에 포함된 자식 컴포넌트의 수를 얻어옵니다.

  ➢ Format

  ```
  integer GetChildCount();
  ```

  ➢ Example

  ```
  var ChildCount = Panel1.GetChildCount();
  ```

■ Example

- 구현 기능

  Panel에 Label, CheckBox, RadioButton을 추가하고, Button을 클릭하면 Panel에 포함된 자식 컴포넌트의 이름(Name)과 텍스트(Text)를 메시지로 표시하는 예제입니다.

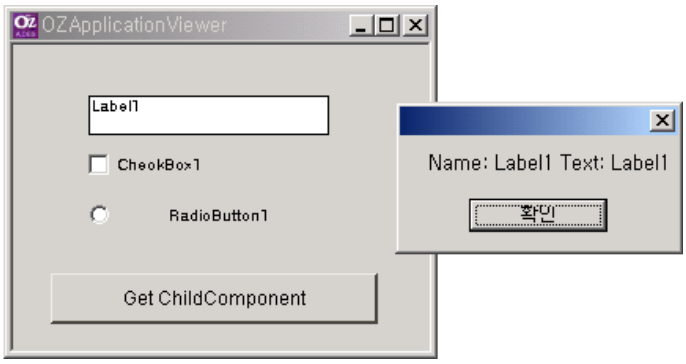
- 디자인

  ➢ Board에 Panel을 추가하고, 추가한 Panel에 Label, CheckBox, RadioButton을 자식 컴포넌트로 추가합니다.

  ➢ Board에 Button을 추가하고, 'Text' 속성 값을 'Get ChildComponent'라고 변경한 후 'OnClick' 이벤트를 선택하여 아래와 같이 코딩합니다.

  ```
  var objComp;
  var ChildCount = Panel1.GetChildCount();
  for(i=0;i<ChildCount;i++) {
      objComp = Panel1.GetChildAt(i);
      _MessageBox("Name:  "+  objComp.Name  +  "  Text:  "  +
      objComp.Text);
  }
  ```

- 미리보기



VBox, HBox

다수의 컴포넌트를 자식 컴포넌트로 포함시킬 때 추가된 자식 컴포넌트를 일정한 간격으로 위치시키고자 할 경우 사용하는 컴포넌트입니다. Board에 VBox를 추가하고 자식 컴포넌트로 포함시킬 컴포넌트를 VBox에 추가하면 자식 컴포넌트가 설정한 수직 간격으로 배치됩니다. HBox는 수평 간격으로 자식 컴포넌트를 배치시키는 컴포넌트이며, 본 매뉴얼에서는 VBox를 예로 들어 설명합니다.

■ Property

VBox의 기본적인 속성은 아래와 같습니다.

Property	Description
VerticalGap	VBox에 추가된 컴포넌트 간의 수직 간격

HBox의 기본적인 속성은 아래와 같습니다.

Property	Description
HorizontalGap	HBox에 추가된 컴포넌트 간의 수평 간격

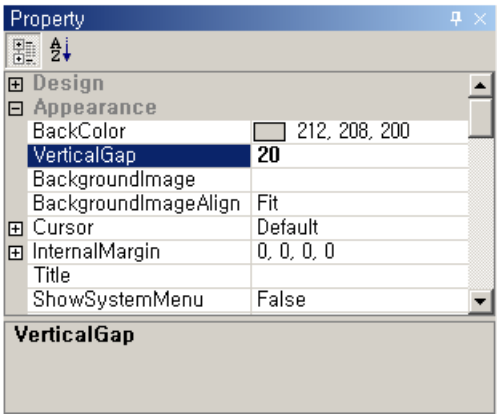
■ Example

- 구현 기능

  VBox에 TextBox, Button을 추가하고, VBox에 포함된 자식 컴포넌트를 일정한 수직 간격으로 배치시키는 예제입니다.

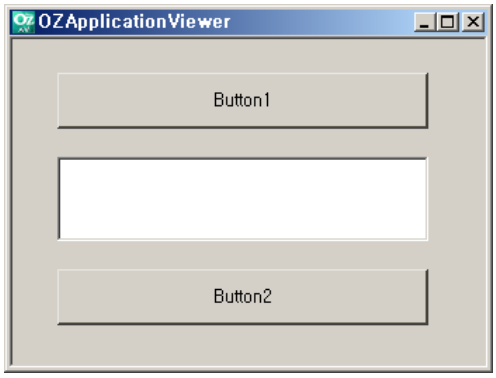
- 디자인

  ➢ Board에 VBox를 추가한 후 'VerticalGap' 속성 값을 '20'으로 변경합니다.



  ➢ VBox에 TextBox, Button을 추가합니다.

- 미리보기



TabControl

다수의 Panel을 탭 형태로 나타내며, 여러 개의 Panel을 탭으로 구분지어 나타낼 때 사용하는 컴포넌트입니다. Board에 TabControl을 추가하면 기본적으로 하나의 TabPanel이 생기며 TabControl 마법사를 이용하여 TabPanel을 추가하거나 삭제하며, 마법사 사용법은 본 매뉴얼의 Ⅶ.장 마법사 사용법을 참조하십시오.

■ Property

TabControl의 기본적인 속성은 아래와 같습니다.

Property	Description
SelectedIndex	선택되거나 선택할 Tab의 인덱스, 0부터 시작됨

■ Example

- 구현 기능

  4개의 TabPanel이 있는 TabControl이 화면에 표시될 때 두번째 Tab이 선택되어 표시되게 하는 예제입니다.

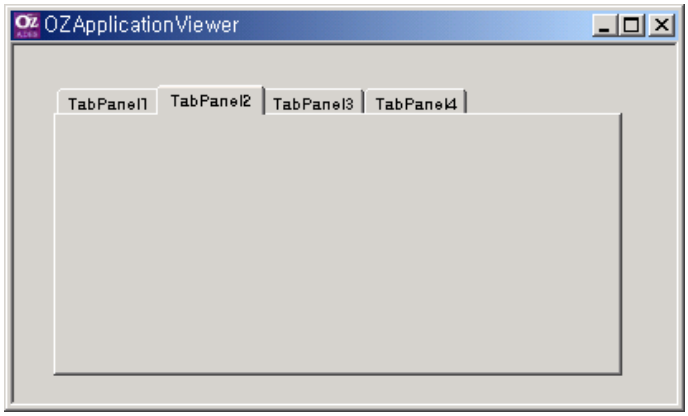
- 디자인

  ➢ Board에 TabControl(Name=TabControl1)을 추가하고, TabControl 마법사를 이용하여 TabPanel을 4개 추가합니다.

  ➢ Board의 'OnPostInitialize' 이벤트를 선택하여 아래와 같이 코딩합니다.

```
TabControl1.SelectedIndex = 1;
```

- 미리보기



BoardView

프로젝트 파일에 추가되어 있는 Board를 표시할 때 사용하는 컴포넌트입니다. 같은 OZA에 있는 Board 뿐만 아니라 리파지토리에 있는 다른 OZA에 있는 Board도 표시할 수 있으며, 만일 BoardView의 크기가 표시되는 Board보다 작을 경우 BoardView에 스크롤바가 생깁니다.

■ Property

BoardView의 기본적인 속성은 아래와 같습니다.

Property	Description
BoardName	표시할 Board 이름

같은 OZA에 있는 Board를 표시할 때에는 BoardName 속성에 보드 이름을 바로 설정하고, 다른 OZA에 있는 Board를 표시할 때에는 "/카테고리/OZA파일명/보드이름" 형태로 설정하여야 합니다.

- OZA의 Board를 표시할 경우 스크립트 작성 예

```
BoardView1.BoardName = "Board2";
```

- 다른 OZA의 Board를 표시할 경우 스크립트 작성 예

```
BoardView1.BoardName = "/oza/test.oza/Board1";
```

■ Example

- 구현 기능

TextBox에 Board 이름을 입력한 후 Button을 클릭하면 BoardView에 Board를 표시하는 예제입니다.

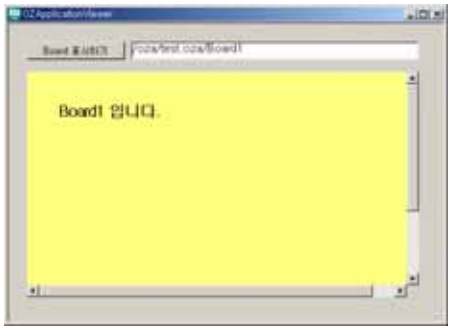
- 디자인

➢ 프로젝트에 여러 개의 Board를 추가한 후 Board에 TextBox(Name=TextBox1), Button(Name=Button1), BoardView(Name=BoardView1)를 추가합니다.

➢ Button의 'OnClick' 이벤트를 선택하여 아래와 같이 코딩합니다.

```
BoardView1.BoardName = TextBox1.Text;
```

- 미리보기



Complex Controls 컴포넌트

Table

데이터를 테이블 형태로 표시할 때 사용하는 컴포넌트로 마법사를 이용하여 별도의 코딩없이 쉽게 데이터를 표시할 수 있습니다. Table에 대한 설명은 본 매뉴얼의 Ⅵ.장 데이터 연동, Ⅶ.장 마법사 사용법을 참조하십시오.

Grid

헤더와 타이틀이 있고 행과 열로 구성되는 그리드를 표시할 때 사용하는 컴포넌트입니다. Grid에 대한 설명은 본 매뉴얼의 Ⅵ.장 데이터 연동을 참조하십시오.

CrossTab

데이터를 크로스탭 형태로 표시하고 그룹핑하여 합계등을 표시할 때 사용하는 컴포넌트입니다. CrossTab에 대한 설명은 본 매뉴얼의 Ⅶ.장 마법사 사용법을 참조하십시오.

Chart

데이터를 2차원 또는 3차원의 차트 형태로 표시할 때 사용하는 컴포넌트입니다. Chart에 대한 설명은 본 매뉴얼의 Ⅶ.장 마법사 사용법을 참조하십시오.

Drawings 컴포넌트

Line

직선, 점선 등 선을 표시할 때 사용하는 컴포넌트입니다.

■ Property

Line의 기본적인 속성은 아래와 같습니다.

Property	Description
LineDash	점선으로 그릴 때 화면에 표시할 선의 길이
LineGap	점선으로 그릴 때 선과 선 사이의 간격 0보다 큰 값으로 설정하여야 점선으로 표시됩니다.
LineType	선의 방향 ∗ Horizontal : 가로 방향 ∗ Vertical : 세로 방향 ∗ Slash : 오른쪽 위에서 왼쪽 아래 방향 ∗ BackSlash : 왼쪽 위에서 오른쪽 아래 방향
PenThickness	선의 두께

※ 참고사항 : Line 컴포넌트의 LineType 속성 값 자동 설정

Line 컴포넌트는 마우스의 드래그 방향에 따라 LineType 속성 값이 아래 표와 같이 자동으로 설정됩니다.

마우스 드래그 방향	LineType
왼쪽 → 오른쪽 오른쪽 → 왼쪽	Horizontal
위 → 아래 아래 → 위	Vertical
왼쪽 아래 → 오른쪽 위 오른쪽 위 → 왼쪽 아래	Slash
오른쪽 아래 → 왼쪽 위 왼쪽 위 → 오른쪽 아래	BackSlash

■ Example

- 구현 기능

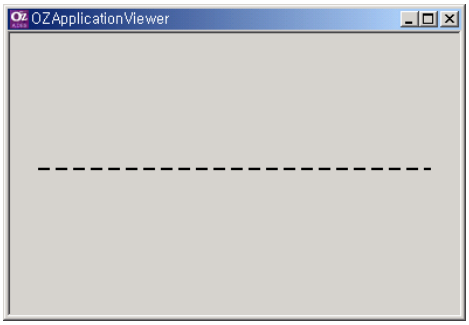
  Line을 이용하여 가로 방향의 점선을 그립니다.

- 디자인

  ➢ Board에 Line을 추가한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
LineDash	10	**LineGap**	5
LineType	Horizontal	**PenThickness**	2

- 미리보기



ArrowLine

화살표가 있는 선을 표시할 때 사용하는 컴포넌트입니다.

■ Property

ArrowLine의 기본적인 속성은 아래와 같습니다.

Property	Description

ArrowShape	화살표 모양 ∗ Circle : 원 모양 화살표 ∗ Diamond : 마름모 모양 화살표 ∗ OpenTriangle : 양 끝이 두툼한 화살표 ∗ SemiCircle : 반원 모양 화살표 ∗ SharpTriangle : 양 끝이 날까로운 화살표 ∗ Triangle : 삼각형 모양 화살표
ArrowSize	화살표 모양의 크기
ArrowType	화살표 방향 ∗ Normal : 정방향 ∗ Reverse : 역방향 ∗ Both : 양쪽 방향
LineType	선의 방향 ∗ Horizontal : 가로 방향 ∗ Vertical : 세로 방향 ∗ Slash : 오른쪽 위에서 왼쪽 아래 방향 ∗ BackSlash : 왼쪽 위에서 오른쪽 아래 방향

※ 참고사항 : ArrowLine 컴포넌트의 LineType 속성 값 자동 설정

ArrowLine 컴포넌트는 마우스의 드래그 방향에 따라 LineType 속성 값이 자동으로 설정되며, 자동으로 설정되는 LineType 속성 값은 Line 컴포넌트의 참고사항을 참조하시기 바랍니다.

※ 참고사항 : ArrowLine 컴포넌트의 ArrowType 속성 값 자동 설정

ArrowLine 컴포넌트는 마우스의 드래그 방향에 따라 ArrowType 속성 값이 아래 표와 같이 자동으로 설정됩니다.

마우스 드래그 방향	ArrowType
왼쪽 → 오른쪽 위 → 아래 왼쪽 아래 → 오른쪽 위 오른쪽 아래 → 왼쪽 위	Normal
오른쪽 → 왼쪽 아래 → 위 오른쪽 위 → 왼쪽 아래 왼쪽 위 → 오른쪽 아래	Reverse

■ Example

- 구현 기능

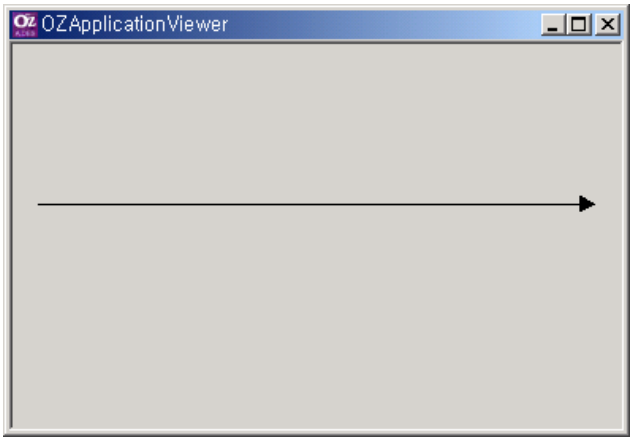
ArrowLine을 이용하여 가로 방향의 정방향 화살표 선을 그립니다.

- 디자인

  ➢ Board에 ArrowLine을 추가한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
ArrowShape	Triangle	**ArrowSize**	10
ArrowType	Normal	**LineType**	Horizontal

- 미리보기



Ellipse

원형, 타원형 등 원을 표시할 때 사용하는 컴포넌트입니다.

■ Property

Ellipse의 기본적인 속성은 아래와 같습니다.

Property	Description
BrushStyle	원을 그릴 브러쉬 스타일 ∗ Solid : 단선으로 원을 그려 원 내부를 채우지 않음 ∗ Hatch : 평행선으로 원을 그려 원 내부를 평행선으로 채움

PatternStyle	원 내부의 패턴 스타일, BrushStyle 속성 값을 Hatch로 설정했을 경우 적용됨 ∗ Horizontal : 가로 방향 선형 패턴 ∗ Vertical : 세로 방향 선형 패턴 ∗ FDiagonal : 왼쪽위에서 오른쪽아래 방향의 선형 패턴 ∗ BDiagonal : 오른쪽위에서 왼쪽아래 방향의 선형 패턴 ∗ Cross : Horizontal형과 Vertical형이 교차된 패턴 ∗ Diacross : FDiagonal형과 BDiagonal형이 교차된 다이아몬드형 패턴
PenThickness	원 테두리 선의 두께

■ Example

- 구현 기능

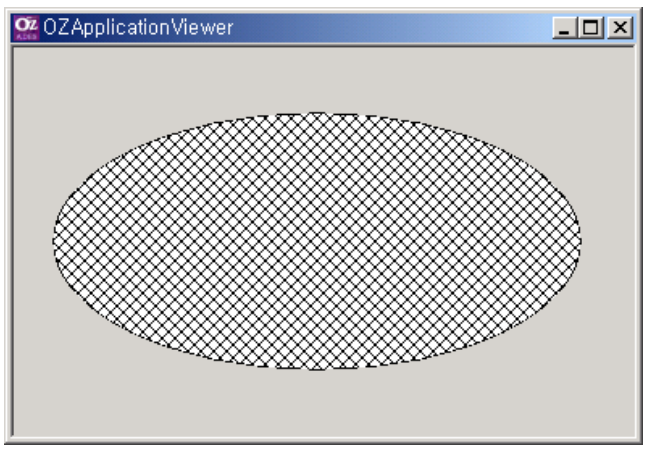
  Ellipse를 이용하여 다이아몬드형 패턴의 타원을 그립니다.

- 디자인

  ➢ Board에 Ellipse를 추가한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
BrushStyle	Hatch	**PatternStyle**	Diacross

- 미리보기



Rectangle

사각형을 표시할 때 사용하는 컴포넌트입니다.

■ Property

Rectangle의 기본적인 속성은 아래와 같습니다.

Property	Description
BrushStyle	사각형을 그릴 브러쉬 스타일 * Solid : 단선으로 사각형을 그려 내부를 채우지 않음 * Hatch : 평행선으로 사각형을 그려 내부를 평행선으로 채움
PatternStyle	사각형 내부의 패턴 스타일, BrushStyle 속성 값을 Hatch로 설정했을 경우 적용됨 * Horizontal : 가로 방향 선형 패턴 * Vertical : 세로 방향 선형 패턴 * FDiagonal : 왼쪽위에서 오른쪽아래 방향의 선형 패턴 * BDiagonal : 오른쪽위에서 왼쪽아래 방향의 선형 패턴 * Cross : Horizontal형과 Vertical형이 교차된 패턴 * Diacross : FDiagonal형과 BDiagonal형이 교차된 다이아몬드형 패턴
PenThickness	사각형 테두리 선의 두께

■ Example

- 구현 기능

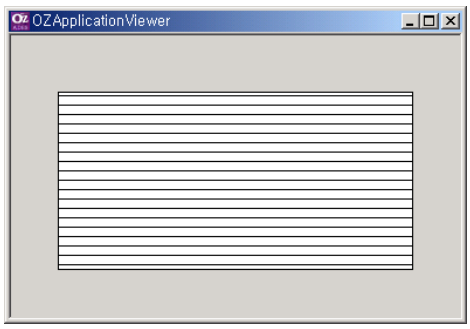
  Rectangle을 이용하여 가로선형 패턴의 사각형원을 그립니다.

- 디자인

  ➢ Board에 Rectangle을 추가한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
BrushStyle	Hatch	**PatternStyle**	Horizontal

- 미리보기

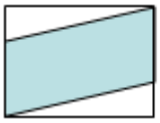
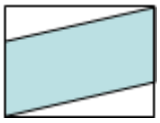
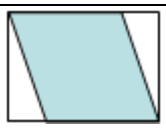
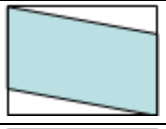
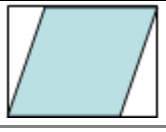


Parallelogram

평행사변형을 표시할 때 사용하는 컴포넌트입니다.

■ Property

Parallelogram의 기본적인 속성은 아래와 같습니다.

Property	Description	
Slope	평행사변형의 면 비율 (기본 값 : 75)  ※ 참고사항 : Slope 값이 0이면 평행사변형이 보이지 않고 100이면 사각형과 같습니다.	
ParallelogramType	평행사변형 타입	
	VSlash	 수직 슬래시 방향
	HBackSlash	 수평 역 슬래시 방향
	VBackSlash	 수직 역 슬래시 방향
	HSlash (기본 값)	 수평 슬래시 방향

■ Example

- 구현 기능

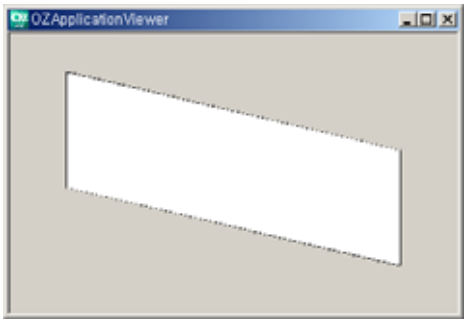
  Parallelogram을 이용하여 면 비율이 60인 수직 역슬래시 방향의 평행사변형을 그립니다.

- 디자인

  ➢ Board에 Parallelogram을 추가한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
Slope	60	**ParallelogramType**	VBackSlash

- 미리보기



Diamond

마름모형을 표시할 때 사용하는 컴포넌트입니다.

■ Property

Diamond의 기본적인 속성은 아래와 같습니다.

Property	Description
BrushStyle	마름모형을 그릴 브러쉬 스타일 ∗ Solid : 단선으로 마름모형을 그려 내부를 채우지 않음 ∗ Hatch : 평행선으로 마름모형을 그려 내부를 평행선으로 채움

PatternStyle	마름모형 내부의 패턴 스타일, BrushStyle 속성 값을 Hatch로 설정했을 경우 적용됨 ∗ Horizontal : 가로 방향 선형 패턴 ∗ Vertical : 세로 방향 선형 패턴 ∗ FDiagonal : 왼쪽위에서 오른쪽아래 방향의 선형 패턴 ∗ BDiagonal : 오른쪽위에서 왼쪽아래 방향의 선형 패턴 ∗ Cross : Horizontal형과 Vertical형이 교차된 패턴 ∗ Diacross : FDiagonal형과 BDiagonal형이 교차된 다이아몬드형 패턴
PenThickness	마름모형 테두리 선의 두께

■ Example

- 구현 기능

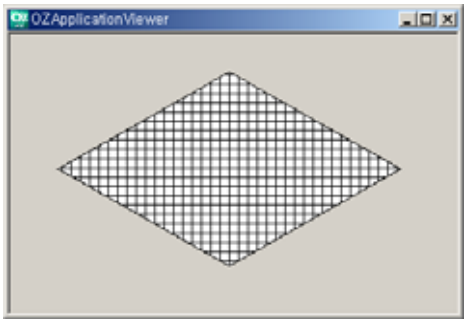
  Diamond를 이용하여 가로선형과 세로선형이 교차된 패턴의 마름모를 그립니다.

- 디자인

  ➢ Board에 Diamond를 추가한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
BrushStyle	Hatch	**PatternStyle**	Cross

- 미리보기

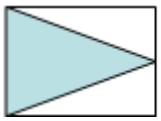
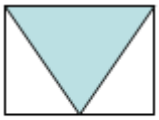


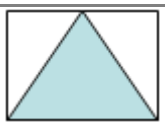
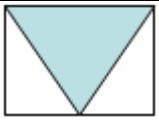
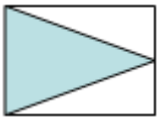
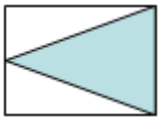
Triangle

삼각형을 표시할 때 사용하는 컴포넌트입니다.

■ Property

Triangle의 기본적인 속성은 아래와 같습니다.

<table>
<tr><th>Property</th><th colspan="2">Description</th></tr>
<tr><td>Slope</td><td colspan="2">삼각형의 면 비율 (기본 값 : 50)

비율


비율
※ 참고사항 : Slope 값이 0이면 직각삼각형이되고 50이면 이등변삼각형이 됩니다.</td></tr>
<tr><td rowspan="5">TriangleType</td><td colspan="2">평행사변형 타입</td></tr>
<tr><td>HNormal(기본 값)</td><td>
수평 삼각형</td></tr>
<tr><td>HReverse</td><td>
수평 역 삼각형</td></tr>
<tr><td>VNormal</td><td>
수직 삼각형</td></tr>
<tr><td>VReverse</td><td>
수직 역 삼각형</td></tr>
</table>

■ Example

- 구현 기능

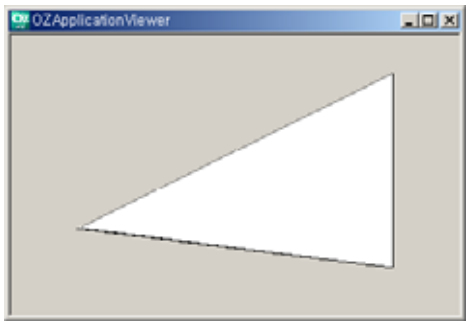
  Triangle을 이용하여 면 비율이 90인 수직 역 삼각형을 그립니다.

- 디자인

  ➢ Board에 Triangle을 추가한 후 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
Slope	90	**TriangleType**	VReverse

- 미리보기



OZC(OZ Component)

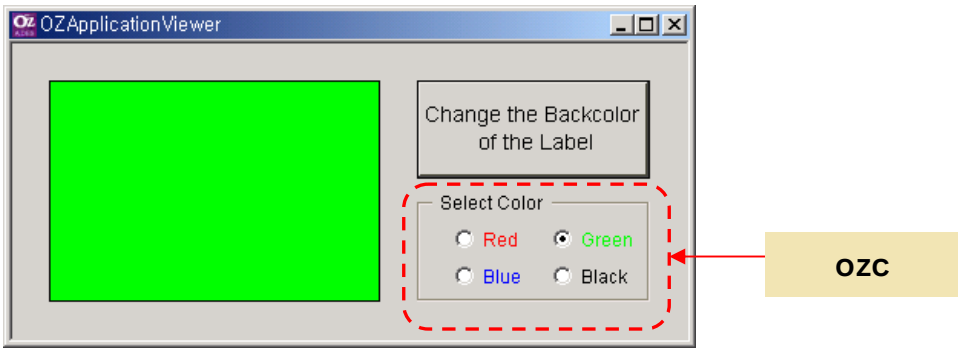
OZC는 OZ Application Designer의 컴포넌트를 개별적 또는 복합적으로 모듈화시킨 OZ Component 파일입니다.
Label, TextBox 등 OZ Application Designer에서 제공하는 각각의 컴포넌트를 OZC 파일로 개별적으로 모듈화시키거나 자식 컴포넌트를 가지는 Container Controls 컴포넌트 및 Board를 한꺼번에 그룹지어 OZC 파일로 모듈화시킬 수도 있습니다. OZC 파일은 [Export Component] 메뉴를 이용하여 만들고, [Import Component] 메뉴나 스크립트를 통해 사용합니다.

본 매뉴얼에서는 Container Controls 컴포넌트와 Board를 OZC 파일로 모듈화하고 사용하는 방법을 예제를 통해 설명합니다.

Container Controls 모듈화

Container Controls 컴포넌트 중 GroupBox와 GroupBox의 자식 컴포넌트를 OZC 파일로 만들고 사용하는 방법을 아래와 같은 예제를 통해 설명합니다.

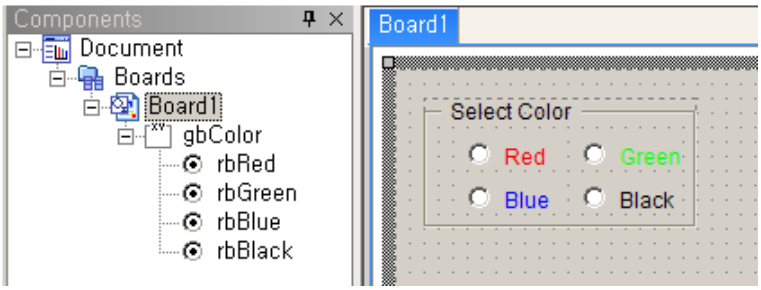
미리보기 화면



Step 1 OZC 만들기

OZC 파일로 만들 컴포넌트를 디자인하고 OZC 파일로 만듭니다.

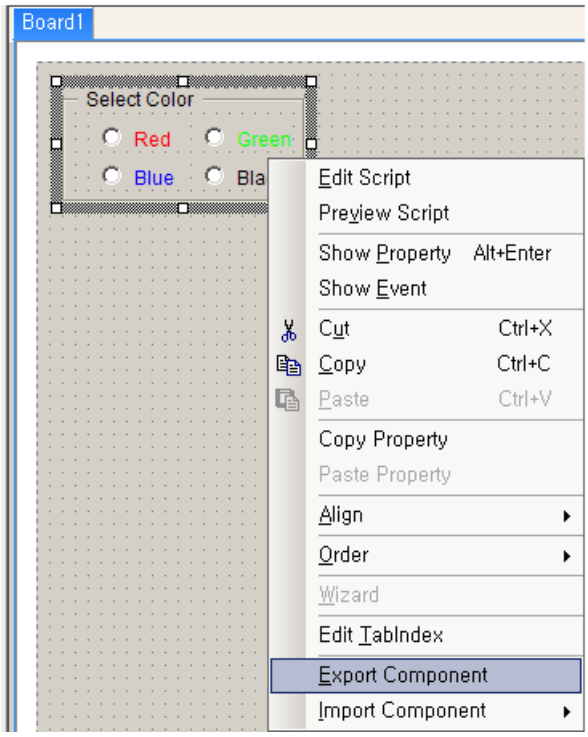
➢ Board에 GroupBox, RadioButton을 추가하고 아래 그림처럼 디자인합니다.



GroupBox와 RadioButton의 'Name' 속성 값을 컴포넌트 트리창에 표시된 값처럼 'gbColor', 'rbRed', 'rbGreen', 'rbBlue', 'rbBlack'로 변경하고, GroupBox의 'Text' 속성 값은 'Select Color'로, RadioButton 4개의 'Text' 속성 값을 'Red', 'Green', 'Blue', 'Black'로 변경합니다.

> ❖ **주의사항** ❖
>
> 만일 임포트되는 OZC 파일에 있는 컴포넌트 이름과 메인 Board에 있는 컴포넌트 이름을 비교하여 동일한 이름의 컴포넌트가 있는 경우 OZC 파일에 있는 컴포넌트의 이름은 기본 값 즉, '컴포넌트타입명' + '일련번호'로 자동 변경됩니다.
>
> 그러므로 OZC로 만들 컴포넌트의 이름 즉, 'Name' 속성 값은 반드시 고유한 값으로 설정하십시오. RadioButton을 예로 들면, RadioButton1, RadioButton2, ... 등의 일련 번호 형식의 컴포넌트 이름이 아니라 rbRed, rbGreen, ... 등 고유한 컴포넌트 이름으로 변경하십시오.

➢ GroupBox를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Export Component] 메뉴를 클릭합니다.

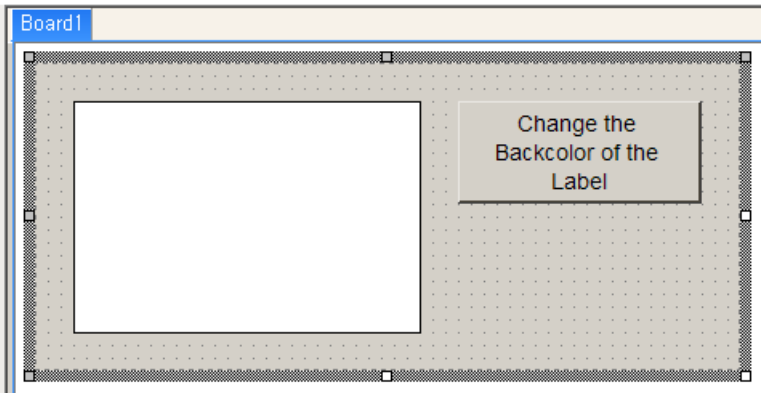


➢ '파일 저장 다이얼로그'에서 'OZC_GroupBox.ozc' 파일로 리피자토리 루트 폴더에 저장합니다. 본 매뉴얼에서 리파지토리 루트 폴더는 'C:\Repository' 입니다.

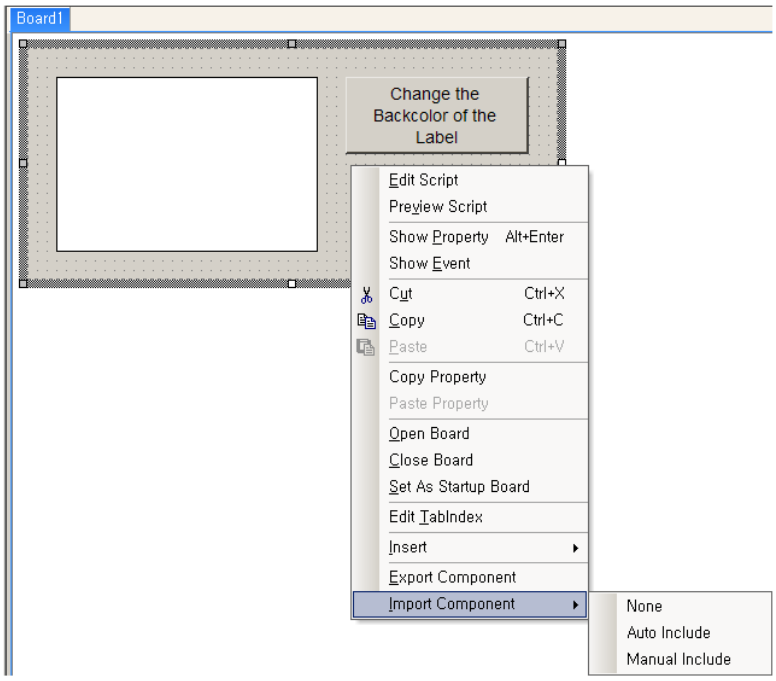
Step 2 OZC 추가

OZC 파일을 임포트하여 프로젝트에 추가합니다.

➢ 프로젝트를 새로 만들고, OZC를 추가할 메인 Board에 Label, Button을 추가한 후 아래 그림처럼 디자인합니다.



➢ Board를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Import Component] 메뉴를 선택합니다.

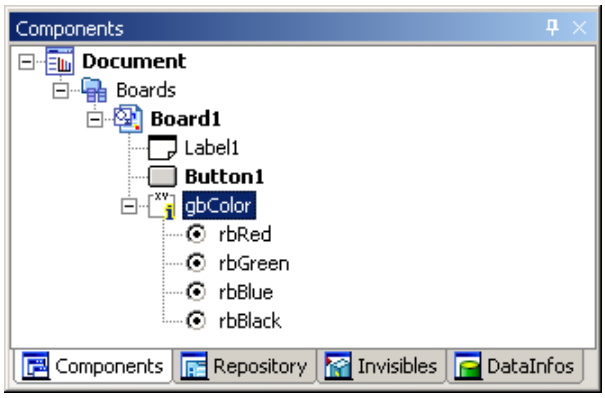


[Import Component] 메뉴는 세 개의 서브 메뉴로 구성되어 있으며, 각 서브 메뉴의 간략한 설명은 아래와 같습니다.

Menu	Description
None	임포트된 OZC 파일과 상관없이 일반적인 컴포넌트로 프로젝트에 추가합니다.
Auto Include	임포트된 OZC 파일의 정보를 저장하여 프로젝트에 추가합니다. "SaveType" 속성 값을 이용하여 OZC가 추가된 프로젝트가 저장될 때 원본 OZC의 내용도 저장할지 여부를 설정할 수 있습니다. 'Auto Include' 방식으로 임포트된 OZC 파일은 디자인시 뿐만 아니라 미리보기시에도 임포트한 OZC의 내용을 볼 수 있습니다.
Manual Include	'Auto Include' 방식처럼 원본 OZC 파일과 연동되는 것은 동일하나, 'Manual Include' 방식으로 임포트된 OZC 파일은 디자인시에는 임포트한 OZC의 내용이 보이나 미리보기시에는 ImportNewChild() 메소드를 호출하는 시점에 OZC의 내용을 볼 수 있습니다.

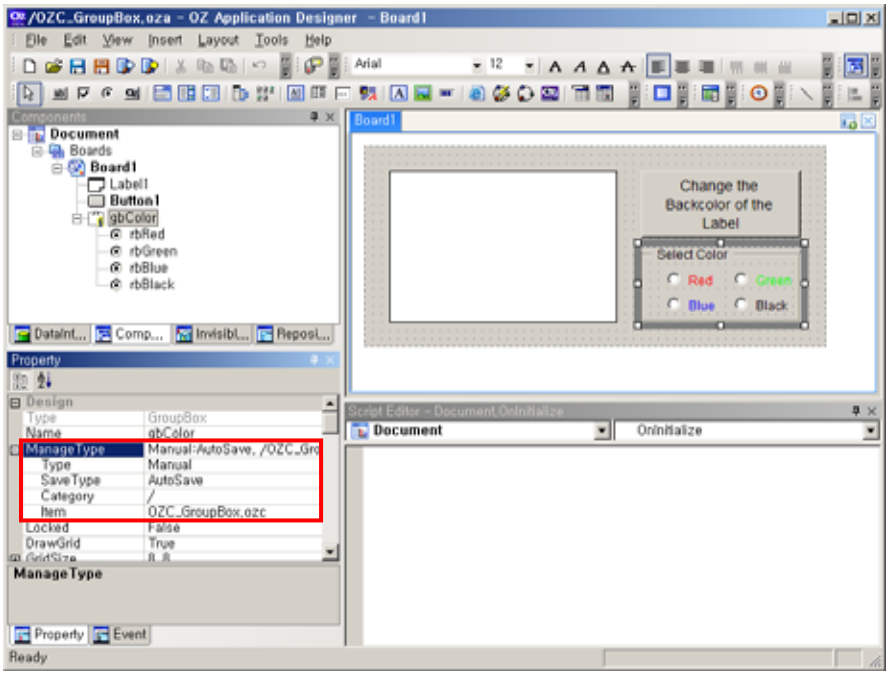
본 장에서는 [Manual Include] 방식으로 OZC 파일을 임포트하는 방법을 살펴봅니다.

➢ [Import Component] 메뉴의 [Manual Include] 메뉴를 클릭하여 OZC 파일을 Board에 임포트합니다.



[Auto Include]나 [Manual Include] 방식으로 OZC를 임포트할 경우 위의 그림과 같이 컴포넌트 창에 표시되는 컴포넌트 아이콘 옆에 영문 "i"자()가 표시됩니다.

➢ 임포트된 OZC의 위치를 아래 그림처럼 변경하고, 속성창에서 'Manage Type' 속성 값으로 임포트 방식과 연동 파일 정보를 확인합니다.



'Manage Type' 속성의 상세 설명은 아래와 같습니다.

Property	Description
Type	OZC 파일의 임포트 방식을 설정합니다.
SaveType	프로젝트가 저장될 때 원본 OZC의 내용도 저장할지 여부를 설정합니다.
Category	원본 OZC 파일의 리파지토리 경로를 설정합니다.
Item	원본 OZC 파일명을 설정합니다.

Step 3 스크립트 코딩

'Manual Include' 방식으로 임포트한 OZC 파일을 Board의 초기화가 종료될 때 Board에 표시하고 Label의 배경색을 변경하는 스크립트를 코딩합니다.

- 'Manual Include' 방식으로 임포트한 OZC 파일을 Board의 초기화가 종료될 때 표시하는 스크립트를 코딩합니다. 스크립트창에서 'Board1'의 'OnPostInitialize' 이벤트를 선택하고 아래와 같이 코딩합니다.

```
Board1.ImportNewChild("ozp:///OZC_GroupBox.ozc");
```

- 인자값으로 색 이름을 받아 Label의 배경색을 변경하는 함수를 전역 함수로 코딩합니다. 'Document'의 GlobalFunction'을 선택하고 아래와 같이 코딩합니다.

```
function changeBackColor(Color) {
    var myLabel = _GetBoard("Board1").GetChild("Label1");
    switch(Color) {
        case "Red":
            myLabel.BackColor = "255, 0, 0";
            break;
        case "Green":
            myLabel.BackColor = "0, 255, 0";
            break;
        case "Blue":
            myLabel.BackColor = "0, 0, 255";
            break;
        case "Black":
            myLabel.BackColor = "0, 0, 0";
            break;
        default:
            break;
    }
}
```

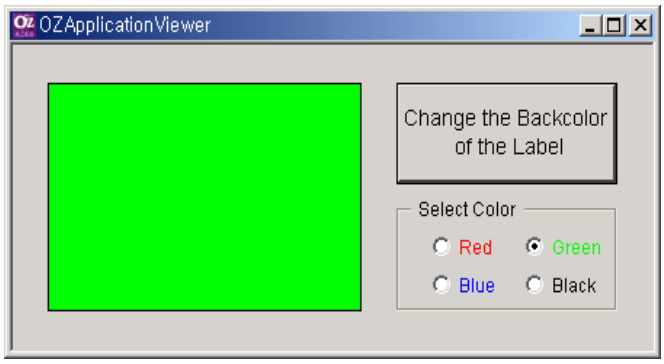
➢ [Change the Backcolor of the Label] 버튼을 클릭하면 Label의 배경색을 변경하는 전역 함수를 호출하는 스크립트를 코딩합니다. 스크립트창에서 'Button1'의 'OnClick' 이벤트를 선택하고 아래와 같이 코딩합니다.

```
if(gbColor.GetCheckedRadio() != null)  {
    var selectedColor = gbColor.GetCheckedRadio().Text;
    changeBackColor(selectedColor);
} else {
    _MessageBox("Select Color", "Error");
}
```

Step 4 미리보기

[File] 메뉴의 [Preview]를 클릭하거나 툴바의 미리보기 아이콘()을 클릭하여 미리보기합니다.

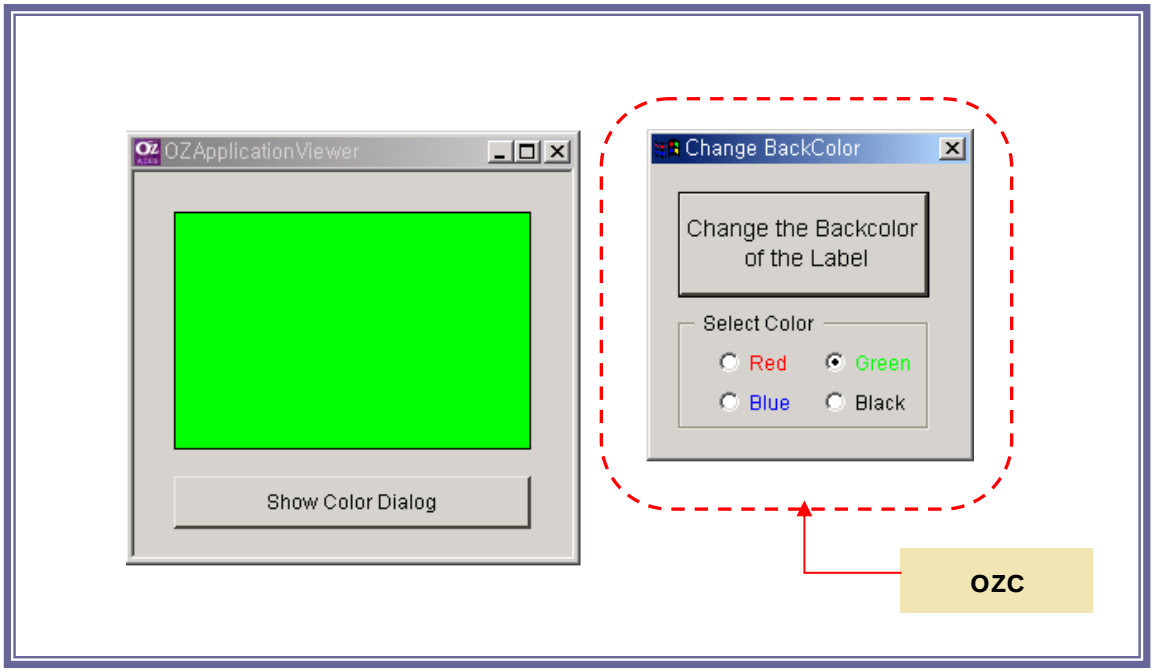
배경색을 선택하고 [Change the Backcolor of the Label] 버튼을 클릭하여 Label의 배경색을 변경합니다.



Board 모듈화

여러 개의 자식 컴포넌트를 가진 Board를 OZC 파일로 만들고 사용하는 방법을 아래와 같은 예제를 통해 설명합니다.

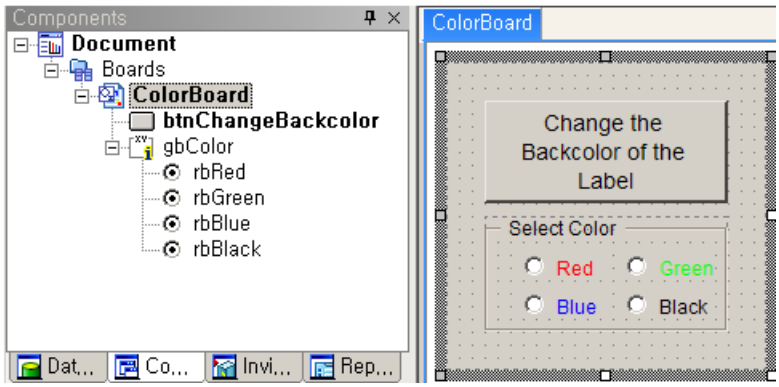
미리보기 화면



Step 1 OZC 만들기

OZC 파일로 만들 Board를 디자인하고 OZC 파일로 만듭니다.

➢ Board에 Button, GroupBox, RadioButton을 추가하고 아래 그림처럼 디자인합니다.



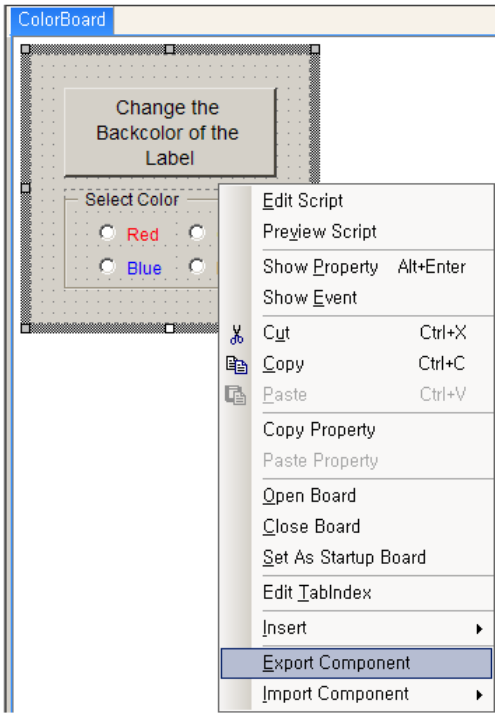
Button, GroupBox와 RadioButton의 'Name' 속성 값을 컴포넌트 트리창에 표시된 값처럼 변경하고, 'Text' 속성 값은 디자인 화면처럼 변경합니다.

➢ 버튼을 클릭하면 Label의 배경색을 변경하는 전역 함수를 호출하는 스크립트를 코딩합니다. 스크립트창에서 'btnChangeBackcolor'의 'OnClick' 이벤트를 선택하고

아래와 같이 코딩합니다.

```
if(gbColor.GetCheckedRadio() != null) {
    var selectedColor = gbColor.GetCheckedRadio().Text;
    changeBackColor(selectedColor);
} else {
    _MessageBox("Select Color", "Error");
}
```

➢ Board를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Export Component] 메뉴를 클릭합니다.

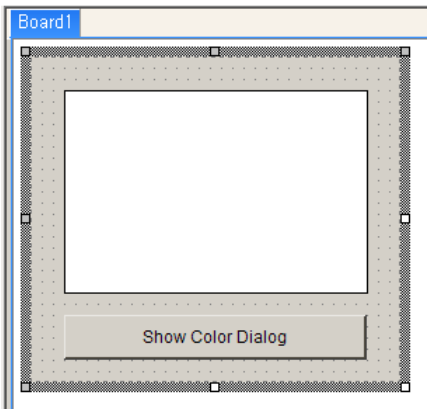


➢ '파일 저장 다이얼로그'에서 리파지토리 루트 폴더에 'OZC_Board.ozc' 파일로 저장합니다. 본 매뉴얼에서 리파지토리 루트 폴더는 'C:\Repository' 입니다.

Step 2 메인 Board 디자인

프로젝트의 메인 Board를 디자인하고, Board를 모듈화한 OZC 파일을 다이얼로그 형태로 표시하고 Label의 배경색을 변경하는 스크립트를 코딩합니다.

➢ 프로젝트를 새로 만들고, 프로젝트의 메인 Board를 아래 그림처럼 디자인합니다.



➢ [Show Color Dialog] 버튼을 클릭하면 Board를 모듈화한 OZC 파일을 다이얼로그 형태로 표시하는 스크립트를 코딩합니다. 스크립트창에서 'Button1'의 'OnClick' 이벤트를 선택하고 아래와 같이 코딩합니다.

```
var objBoard = _ImportNewBoard("ozp:///OZC_Board.ozc");
Board1.ShowDialog(objBoard.Name, "Change BackColor", true, 300, 0);
```

➢ Label의 배경색을 변경하는 스크립트를 전역 함수로 코딩합니다. 스크립트창에서 'Document'의 GlobalFunction'을 선택하고 아래와 같이 코딩합니다.

```
function changeBackColor(Color) {
    var myLabel = _GetBoard("Board1").GetChild("Label1");
    switch(Color) {
        case "Red":
            myLabel.BackColor = "255, 0, 0";
            break;
        case "Green":
            myLabel.BackColor = "0, 255, 0";
            break;
        case "Blue":
            myLabel.BackColor = "0, 0, 255";
            break;
        case "Black":
            myLabel.BackColor = "0, 0, 0";
            break;
        default:
            break;
    }
}
```

Step 3 미리보기

[File] 메뉴의 [Preview]를 클릭하거나 툴바의 미리보기 아이콘()을 클릭하여 미리보기합니다.

[Show Color Dialog] 버튼을 클릭하여 '배경색 선택 다이얼로그'를 표시하고 배경색을 선택한 후 [Change the Backcolor of the Label] 버튼을 클릭하여 Label의 배경색을 변경합니다.

EventHook

EventHook은 프로젝트를 만들면 자동으로 Document에 추가되는 컴포넌트로, PreEventHook과 PostEventHook으로 구성됩니다. EventHook 이벤트의 발생 허가 여부를 설정하는 속성인 'EnableEventHook'의 값이 'True'인 모든 컴포넌트에 대해 이벤트의 동작 전, 후로 특정한 작업을 실행하고자 할 때 사용됩니다.

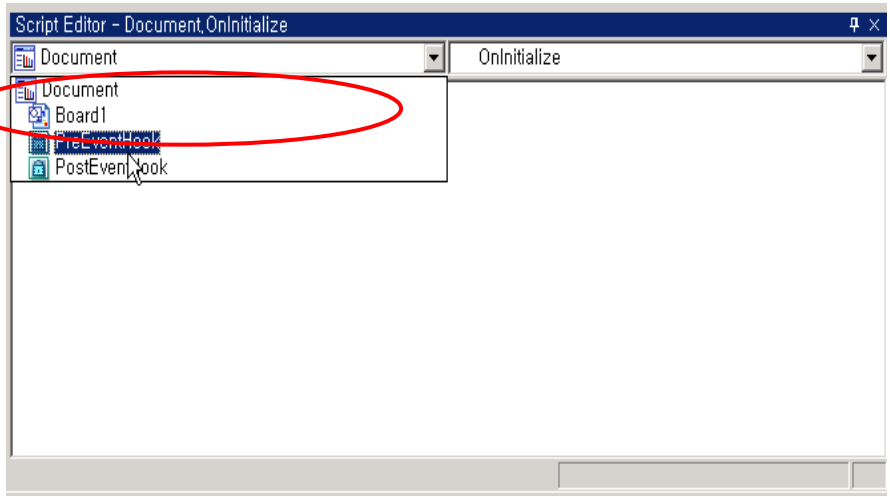
예를 들어, 'EnableEventHook'의 속성 값이 'True'인 Button1의 'OnClick' 이벤트가 발생했을 경우 스크립트가 실행되는 순서는 다음과 같습니다.

① PreEventHook의 'OnClick' 이벤트에 스크립트가 있으면 실행
② Button1의 'OnClick' 이벤트에 스크립트가 있으면 실행
③ PostEventHook의 'OnClick' 이벤트에 스크립트가 있으면 실행

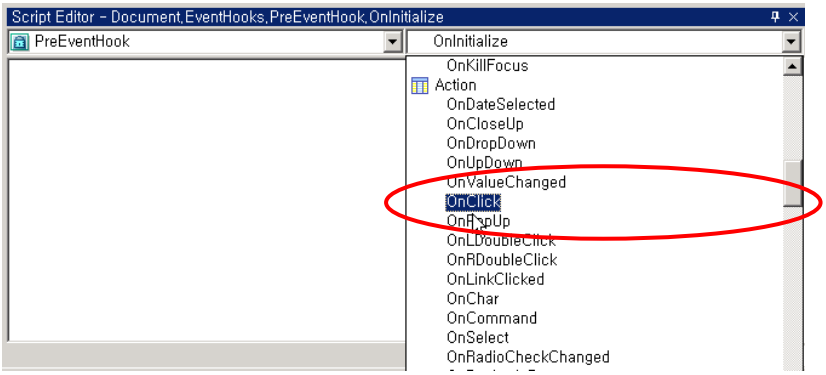
EventHook 이벤트를 발생시키지 않을 컴포넌트는 해당 컴포넌트의 'EnableEventHook' 속성 값(기본 값:True)을 'False'로 변경하여야 합니다.

PreEventHook 또는 PostEventHook에 실행할 스크립트는 스크립트창의 왼쪽 콤보 박스를 클릭하여 PreEventHook 또는 PostEventHook를 선택하고 오른쪽 콤보 박스에서 스크립트를 동작시킬 이벤트를 선택한 후 코딩합니다.

■ PreEventHook 선택



■ 스크립트를 동작시킬 이벤트(OnClick 등) 선택



Example

EventHook을 사용하는 방법을 Board에 Button과 Label을 'EnableEventHook' 속성 값에 따라 PreEventHook과 PostEventHook이 동작하는 과정을 예제를 통해 설명합니다.

■ 구현 기능

PreEventHook와 PostEventHook의 'OnClick' 이벤트에 메시지 박스를 표시하는 스크립트를 코딩합니다. Board에 Button, Label을 추가하여 'EnableEventHook' 속성 값을 설정하고, Button 또는 Label의 'OnClick' 이벤트에도 메시지 박스를 표시하는 스크립트를 코딩하여 'EnableEventHook' 속성 값에 따라 EventHook 이벤트의 발생 여부를 확인합니다.

■ 디자인

- 프로젝트를 새로 만들기한 후 스크립트창에서 PreEventHook의 'OnClick' 이벤트를 선택한 후 아래와 같이 코딩합니다.

```
_MessageBox("PreEventHook_OnClick");
```

- 스크립트창에서 PostEventHook의 OnClick 이벤트를 선택한 후 아래와 같이 코딩합니다.

```
_MessageBox("PostEventHook_OnClick");
```

- Board에 Button(Name=Button1)을 추가하고, 'EnableEventHook' 속성 값을 'True'로 설정한 후 추가한 Button의 'OnClick' 이벤트를 선택한 후 아래와 같이 코딩합니다.

```
_MessageBox("Button1_OnClick");
```

- Board에 Button(Name=Button2)을 하나 더 추가하고, 'EnableEventHook' 속성 값을 'False'로 설정한 후 추가한 Button의 'OnClick' 이벤트를 선택한 후 아래와 같이 코딩합니다.

```
_MessageBox("Button2_OnClick");
```

- Board에 Label(Name=Label1)을 추가하고, 'EnableEventHook' 속성 값을 'True'로 설정한 후 추가한 Label의 'OnClick' 이벤트를 선택한 후 아래와 같이 코딩합니다.

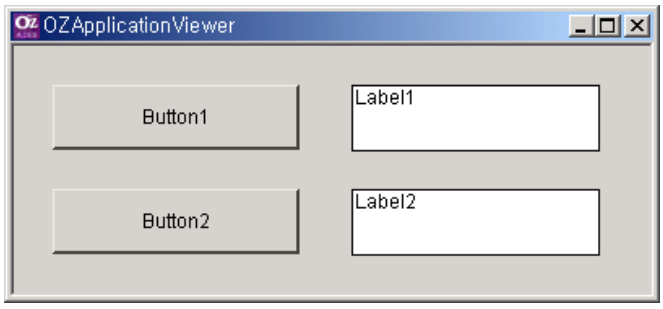
```
_MessageBox("Label1_OnClick");
```

- Board에 Label(Name=Label2)을 하나 더 추가하고, 'EnableEventHook' 속성 값을 'False'로 설정한 후 추가한 Label의 'OnClick' 이벤트를 선택한 후 아래와 같이 코딩합니다.

```
_MessageBox("Label2_OnClick");
```

■ 미리보기

- [File] 메뉴의 [Preview]를 클릭하거나 툴바의 미리보기 아이콘( )을 클릭하여 미리보기합니다.



- 'EnableEventHook' 속성 값이 'True'인 [Button1]을 클릭하면 PreEventHook의 OnClick -> Button1의 OnClick -> PostEventHook의 OnClick 이벤트의 스크립트가 모두 실행되어 아래 그림과 같이 메시지 박스가 세 번 표시됩니다.







- 'EnableEventHook' 속성 값이 'False'인 [Button2]를 클릭하면 PreEventHook과 PostEventHook의 스크립트는 실행되지 않으므로 Button2의 OnClick 이벤트의 스크립트만 실행되어 아래 그림과 같이 메시지 박스가 한 번 표시됩니다.



- 'EnableEventHook' 속성 값이 'True'인 [Label1]을 클릭하면 [Button1]과 마찬가지로 PreEventHook의 OnClick -> Label1의 OnClick -> PostEventHook의 OnClick 이벤트의 스크립트가 모두 실행되어 아래 그림과 같이 메시지 박스가 세 번 표시됩니다.







- 'EnableEventHook' 속성 값이 'False'인 [Label2]를 클릭하면 [Button2]와 마찬가지로 PreEventHook과 PostEventHook의 스크립트는 실행되지 않으므로 Label2의 OnClick 이벤트의 스크립트만 실행되어 아래 그림과 같이 메시지 박스가 한 번 표시됩니다.



Ⅵ. 데이터 연동

- 데이터 연동 가능 컴포넌트
- ODI 만들기, 추가
- 마법사로 Table에 데이터 표시
- 스크립트로 Grid에 데이터 표시
- DataAction : Insert, Delete, Update
- 수식 필드 사용

본 장에서는 데이터 셋을 연결하고 데이터를 조회, 추가, 삭제, 변경하는 방법과 수식 필드를 사용하는 방법을 예제를 통해 설명합니다.

데이터 연동 가능 컴포넌트

Controls 컴포넌트

Controls 컴포넌트 중 ODIKey, DataSet, Field 등 DataSource 속성을 가지는 컴포넌트는 데이터 연동이 가능합니다. 컴포넌트의 ODIKey, DataSet, Field 등의 속성 값을 설정하여 데이터를 표시할 수 있습니다.

Component	Description
CheckBox	옵션의 선택 여부를 선택하는 컴포넌트로, 부모와 상관없이 다수의 옵션을 선택할 수 있는 컴포넌트
RadioButton	옵션의 선택 여부를 설정하는 컴포넌트로, 한 부모 내에서는 하나의 옵션만을 선택할 수 있는 컴포넌트
ComboBox	다수의 아이템을 DropDown 형태로 나타내는 컴포넌트
ListBox	다수의 아이템을 리스트 형태로 나타내는 컴포넌트
CheckListBox	체크마크를 가진 아이템을 리스트 형태로 나타내는 컴포넌트
TextBox	숫자나 문자열을 입력받는 컴포넌트
NumericUpDown	숫자를 직접 입력하거나 증감시키는 컴포넌트
MaskTextBox	특정한 형식을 지정하여 텍스트를 입력받는 컴포넌트
RichTextBox	글자별로 텍스트 형식을 달리 설정하여 문자열을 표시할 수 있는 컴포넌트
Label	숫자나 문자열을 표시하는 컴포넌트
PictureBox	이미지를 표시하는 컴포넌트
ProgressBar	진행 상태를 Bar 형태로 나타내는 컴포넌트
HTML View	웹페이지를 표시하는 컴포넌트
ActiveX Control	ActiveX 컨트롤을 연결하는 컴포넌트
Custom Control	Custom 컨트롤을 연결하는 컴포넌트

DateTimePicker	날짜 또는 시간을 입력하거나 달력을 표시하는 컴포넌트
MonthCalendar	달력을 표시하는 컴포넌트

Complex Controls 컴포넌트

Complex Controls 컴포넌트 중 Table, Chart , CrossTab는 데이터 연동 가능합니다.

Component	Description
Table	데이터를 테이블 형태로 표시하는 컴포넌트
CrossTab	데이터를 크로스탭 형태로 표시하고 그룹핑하여 합계 등을 표시하는 컴포넌트
Chart	데이터를 2차원 또는 3차원의 차트 형태로 나타내는 컴포넌트

ODIKey, DataSet 등의 속성 값으로 연결할 데이터 셋을 설정하고, 마법사를 이용하여 별도의 코딩없이 쉽게 데이터를 표시할 수 있습니다.

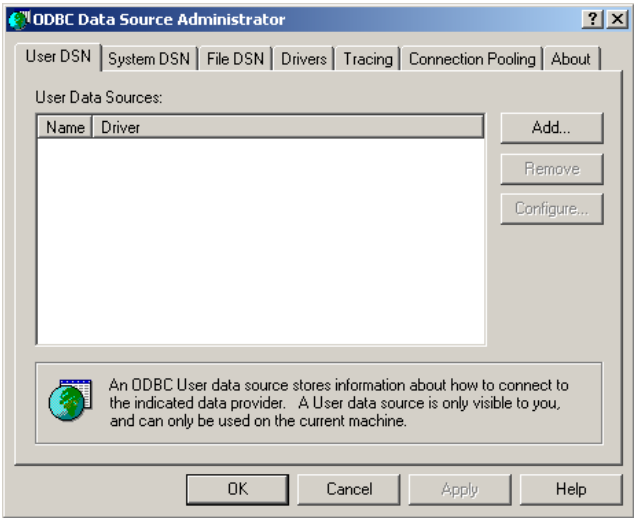
ODI 만들기, 추가

본 장은 ODI 파일을 만들고 프로젝트에 추가하는 방법에 대해 설명합니다.
ODI 파일을 만들기 위해서는 먼저 OZ Application Designer 설치시에 함께 제공되는 샘플용 Database 파일인 Foodmart.mdb 파일을 ODBC 데이터 소스로 등록해야 합니다. 다음으로 OZ Query Designer를 이용하여 ODI 파일을 만들고 프로젝트에 추가합니다. OZ Query Designer에 대한 자세한 설명은 'OZ Query Designer User's Guide'를 참조하십시오.

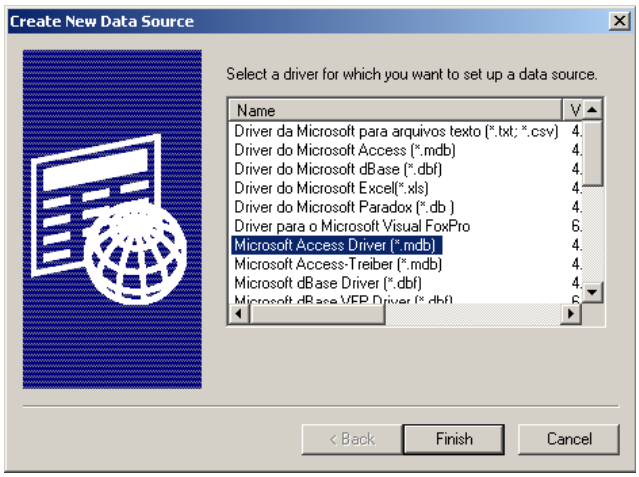
Step 1 ODBC 데이터 원본 설정

OZ Application Designer 설치시에 함께 제공되는 샘플용 Database 파일인 Foodmart.mdb 파일을 ODBC 데이터 소스로 등록합니다.

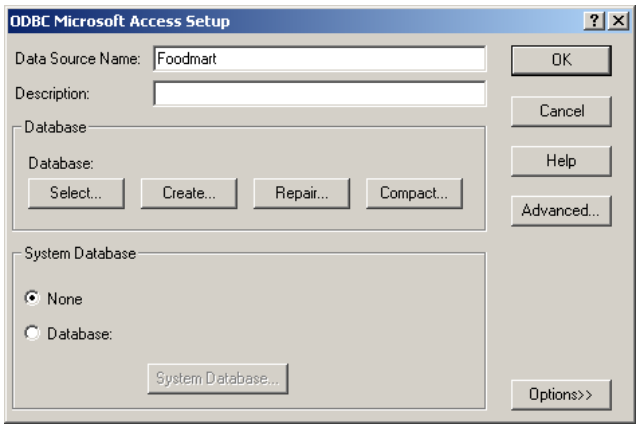
- [제어판]의 [관리 도구]에서 [데이터 원본(ODBC)]을 선택하여 ODBC 데이터 원본 관리자를 실행합니다.



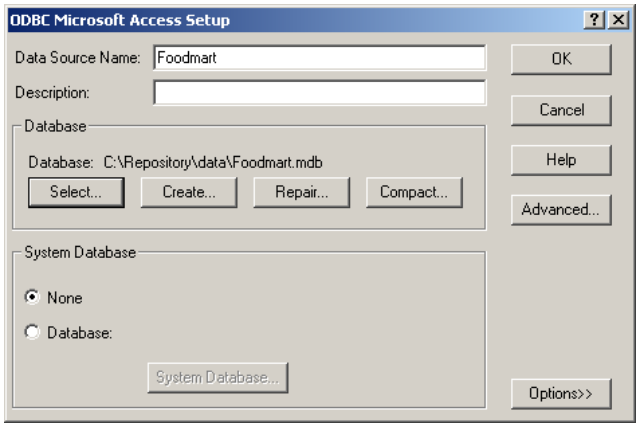
- [System DSN] 탭을 선택하고 [Add] 버튼을 클릭한 후 '데이터 원본 새로 만들기 다이얼로그'에서 'Microsoft Access Driver(*.mdb)'를 선택하고 [Finish] 버튼을 클릭합니다.



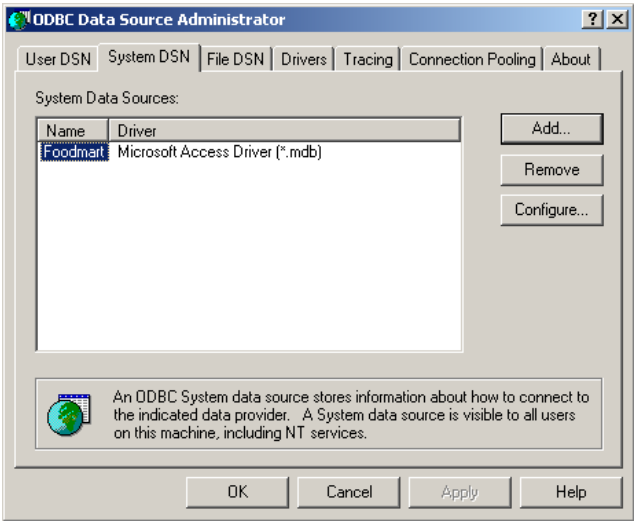
➢ 데이터 원본 이름을 'Foodmart'로 입력합니다.



➢ [Select] 버튼을 클릭하여 Database 파일 선택창에서 'Foodmart.mdb' 파일을 선택한 후 [OK] 버튼을 클릭하여, 데이터 원본에 연결할 Database 파일을 설정합니다.



➢ [OK] 버튼을 클릭하여 ODBC 데이터 원본을 추가합니다.

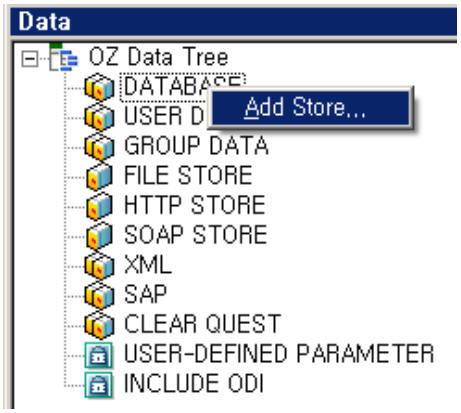


Step 2 ODI 파일 만들기

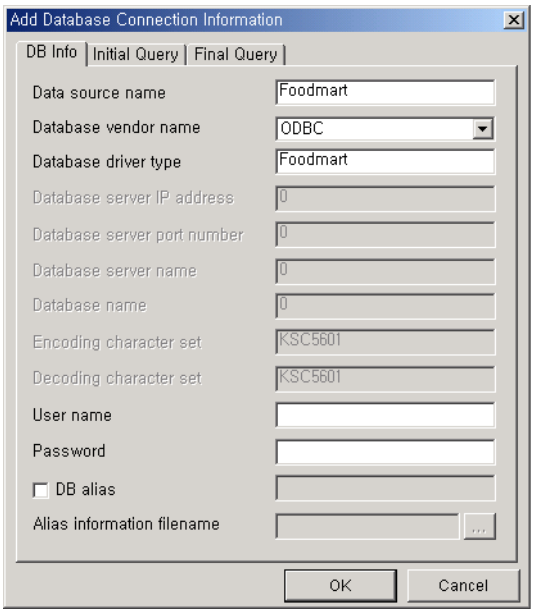
OZ Query Designer를 실행하여 ODI 파일을 만듭니다. 연결할 Database 정보를 추가한 후 필요한 데이터 셋을 추가하고 쿼리를 생성하여 ODI 파일을 만듭니다.

➢ 연결할 Database 정보를 'Foodmart'로 설정합니다.

- OZ Query Designer를 실행하여 'OZ Data Tree'의 'DATABASE'를 마우스 오른쪽 버튼으로 선택하여 나타나는 팝업 메뉴에서 [Add Database Info] 메뉴를 클릭합니다.

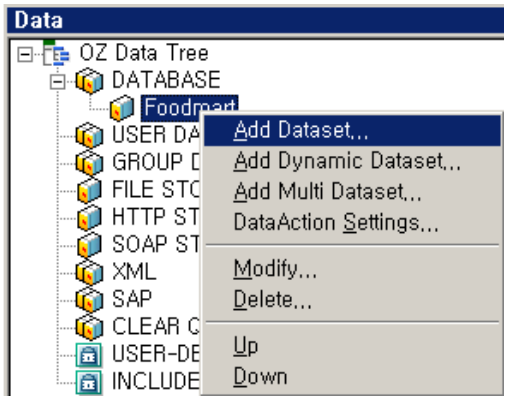


- 연결할 Database 정보를 아래와 같이 설정한 후 [OK] 버튼을 클릭합니다.



➢ 데이터 셋을 추가하고 쿼리를 생성합니다.

- 'OZ Data Tree'에서 'Foodmart'를 선택하고 마우스 오른쪽 버튼을 눌러 나타나는 팝업 메뉴에서 [Add Query Dataset] 메뉴를 클릭합니다.

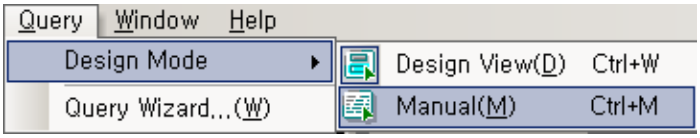


- '데이터 셋 추가 다이얼로그'에서 데이터 셋 이름을 'ProductSales'로 입력하고 [OK] 버튼을 클릭하여 데이터 셋을 추가합니다.



- 메뉴바에서 [Query] 메뉴의 [Change Design Mode] 메뉴의 서브 메뉴인

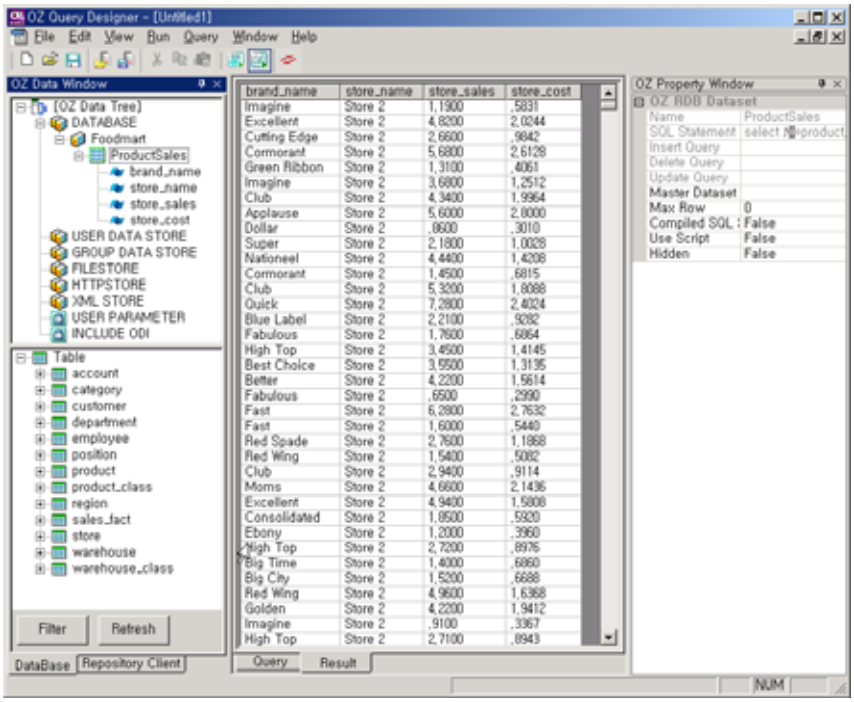
[Manual Mode]를 클릭하여, 쿼리를 직접 입력할 수 있는 매뉴얼 모드로 변경합니다.



- 'OZ Data Tree'에서 'ProductSales'를 선택하고 쿼리 프레임에 쿼리문을 아래와 같이 입력합니다.

```
select
    product.brand_name,
    store.store_name,
    sales_fact.store_sales,
    sales_fact.store_cost
from sales_fact, product, store
where sales_fact.product_id = product.product_id
and sales_fact.store_id = store.store_id
```

- 툴바에서 실행 아이콘( )을 클릭하여 쿼리 결과를 확인합니다.



➢ 작업 폴더에 저장합니다.

- 메뉴바에서 [File] 메뉴의 [Save] 메뉴나 툴바에서 저장 아이콘( )을 클릭하여 리파지토리 루트 폴더에 'Foodmart.odi' 파일로 저장합니다. 본 매뉴얼

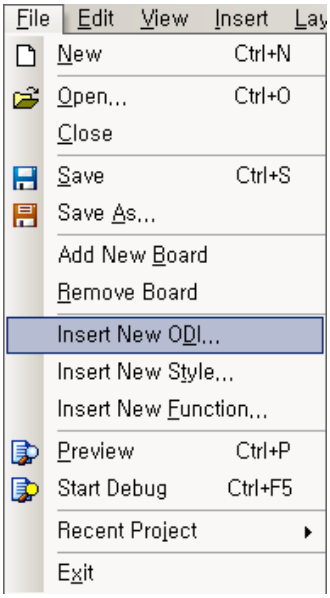
에서 리파지토리 루트 폴더는 'C:\Repository'입니다.

Step 3 ODI 파일 추가

메뉴바와 리파지토리창을 이용하여 ODI 파일을 프로젝트에 추가합니다.

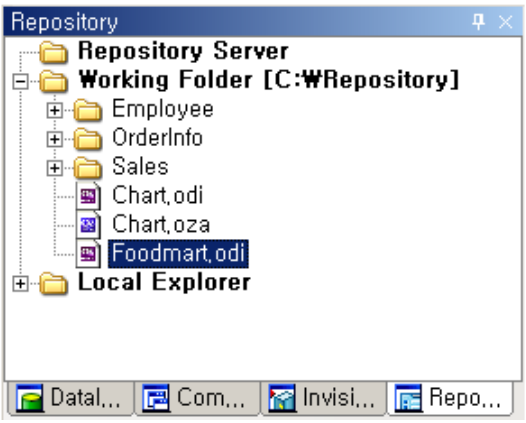
➢ 메뉴바 이용

[File] 메뉴의 [Insert New ODI] 메뉴를 클릭하여 리파지토리 파일창에서 ODI 파일을 선택한 후 [OK] 버튼을 클릭하여 추가합니다.



➢ 리파지토리창 이용

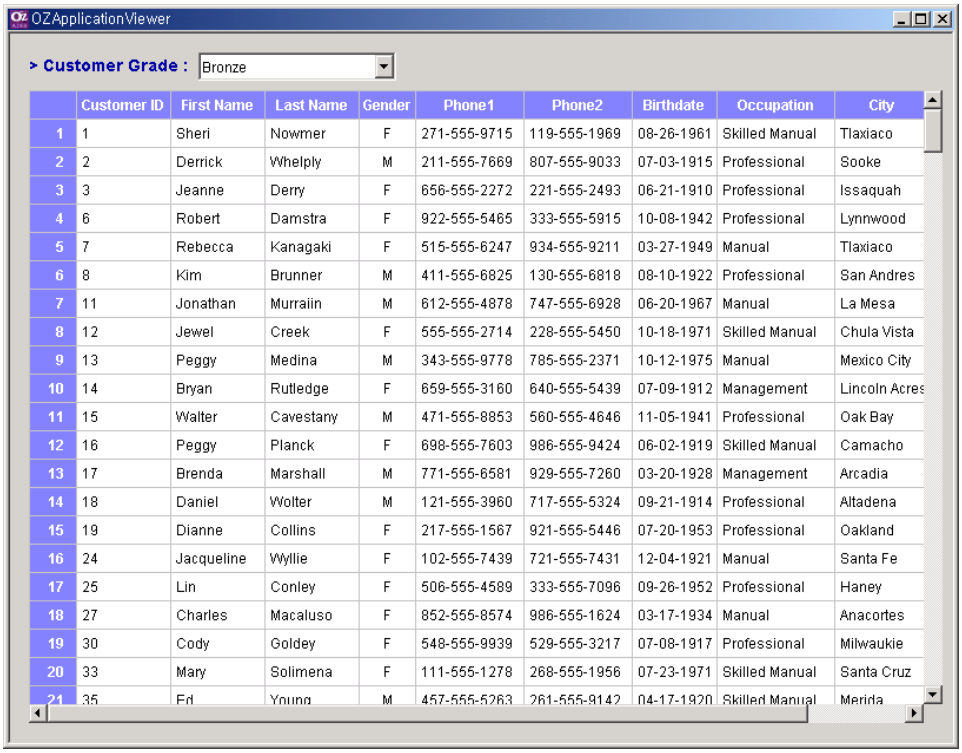
리파지토리창에서 추가할 ODI 파일을 더블클릭하여 추가합니다.



마법사로 Table에 데이터 표시

다음은 마법사를 이용하여 Table에 데이터를 표시하는 방법을 아래과 같은 예제를 통해 설명합니다.

미리보기 화면

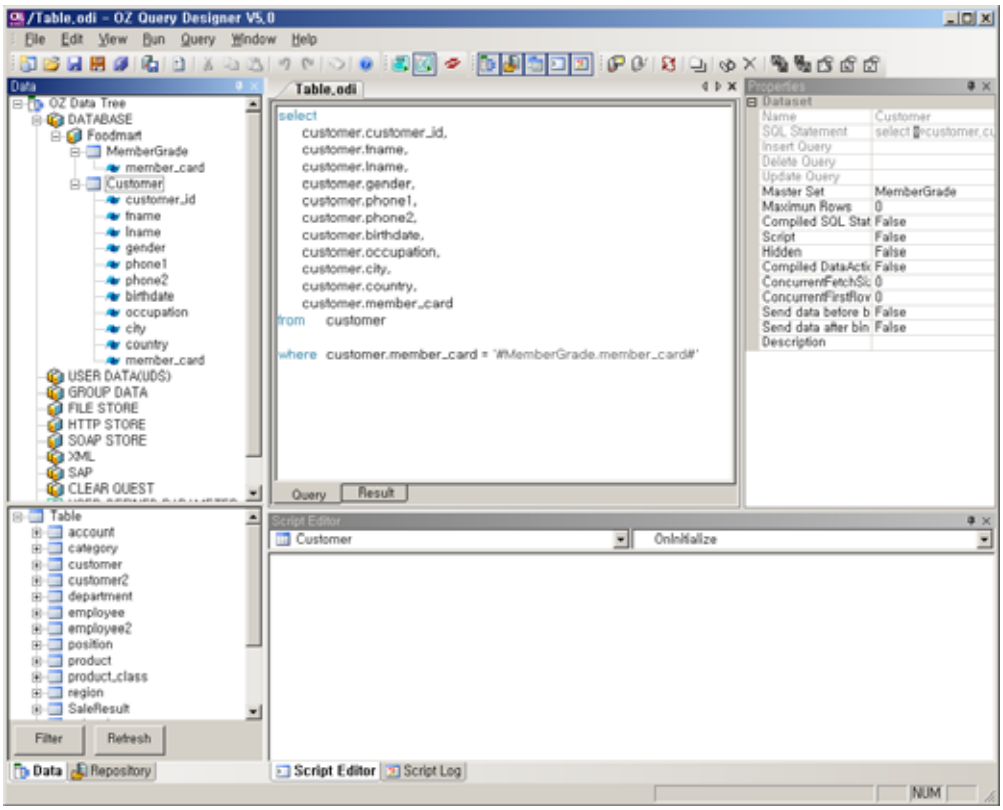


	Customer ID	First Name	Last Name	Gender	Phone1	Phone2	Birthdate	Occupation	City
1	1	Sheri	Nowmer	F	271-555-9715	119-555-1969	08-26-1961	Skilled Manual	Tlaxiaco
2	2	Derrick	Whelply	M	211-555-7669	807-555-9033	07-03-1915	Professional	Sooke
3	3	Jeanne	Derry	F	656-555-2272	221-555-2493	06-21-1910	Professional	Issaquah
4	6	Robert	Damstra	F	922-555-5465	333-555-5915	10-08-1942	Professional	Lynnwood
5	7	Rebecca	Kanagaki	F	515-555-6247	934-555-9211	03-27-1949	Manual	Tlaxiaco
6	8	Kim	Brunner	M	411-555-6825	130-555-6818	08-10-1922	Professional	San Andres
7	11	Jonathan	Murraiin	M	612-555-4878	747-555-6928	06-20-1967	Manual	La Mesa
8	12	Jewel	Creek	F	555-555-2714	228-555-5450	10-18-1971	Skilled Manual	Chula Vista
9	13	Peggy	Medina	M	343-555-9778	785-555-2371	10-12-1975	Manual	Mexico City
10	14	Bryan	Rutledge	F	659-555-3160	640-555-5439	07-09-1912	Management	Lincoln Acres
11	15	Walter	Cavestany	M	471-555-8853	560-555-4646	11-05-1941	Professional	Oak Bay
12	16	Peggy	Planck	F	698-555-7603	986-555-9424	06-02-1919	Skilled Manual	Camacho
13	17	Brenda	Marshall	M	771-555-6581	929-555-7260	03-20-1928	Management	Arcadia
14	18	Daniel	Wolter	M	121-555-3960	717-555-5324	09-21-1914	Professional	Altadena
15	19	Dianne	Collins	F	217-555-1567	921-555-5446	07-20-1953	Professional	Oakland
16	24	Jacqueline	Wyllie	F	102-555-7439	721-555-7431	12-04-1921	Manual	Santa Fe
17	25	Lin	Conley	F	506-555-4589	333-555-7096	09-26-1952	Professional	Haney
18	27	Charles	Macaluso	F	852-555-8574	986-555-1624	03-17-1934	Manual	Anacortes
19	30	Cody	Goldey	F	548-555-9939	529-555-3217	07-08-1917	Professional	Milwaukie
20	33	Mary	Solimena	F	111-555-1278	268-555-1956	07-23-1971	Skilled Manual	Santa Cruz
21	35	Ed	Young	M	457-555-5263	261-555-9142	04-17-1920	Skilled Manual	Merida

기능

- 고객의 멤버쉽 카드 등급을 ComboBox에 표시하고, 선택한 멤버쉽 카드 등급에 해당하는 고객을 검색하여 Table에 고객 정보를 표시합니다.
- 타이틀을 클릭하여 데이터를 정렬하여 표시합니다.
- 마스터-디테일 데이터 셋과 Table 마법사를 이용하여 별도의 코딩없이 구현합니다.

ODI 디자인



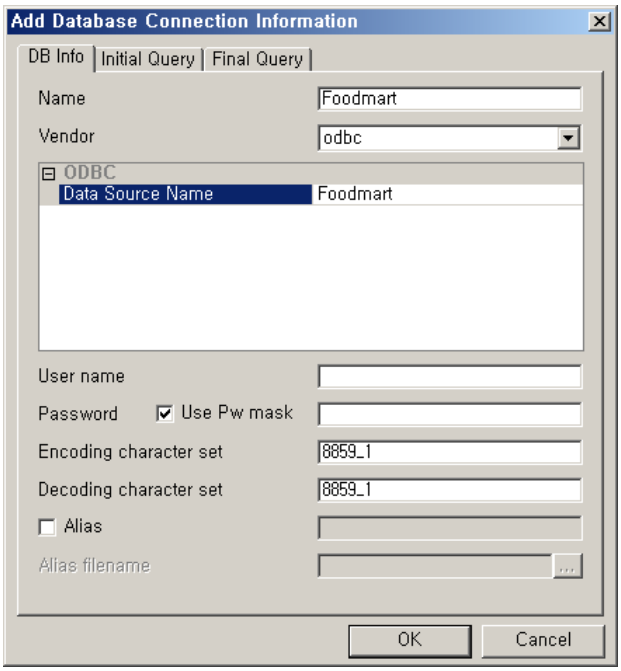
기능

- 연결할 Database는 ODBC 데이터 원본인 'Foodmart'이며, 고객의 멤버쉽 카드 등급 종류를 가져오는 데이터 셋인 'MemberGrade', 멤버쉽 카드 등급에 해당하는 고객 데이터를 가져오는 데이터 셋인 'Customer'로 구성됩니다.
- 'Customer' 데이터 셋의 속성 중 'Master Dataset' 속성 값을 'MemberGrade'로 설정하여 마스터-디테일 데이터 셋으로 설정합니다.

Step 1 ODI 파일 만들기

고객의 멤버쉽 카드 등급과 선택한 멤버쉽 카드 등급에 해당하는 고객 정보를 표시하는 데이터 셋을 포함하는 ODI 파일을 만듭니다. ODI 파일을 만드는 방법은 본 장의 'ODI 만들기, 추가' 부분을 참조하십시오.

- OZ Query Designer를 실행하여 'OZ Data Tree'의 'DATABASE'를 마우스 오른쪽 버튼으로 선택하여 나타나는 팝업 메뉴에서 [Add Database Info] 메뉴를 클릭합니다. Database 연결 정보를 'Foodmart'로 설정합니다.



- 'Foodmart'를 마우스 오른쪽 버튼으로 선택하여 나타나는 팝업 메뉴에서 [Add Query Dataset] 메뉴를 선택하여 데이터 셋을 추가하고, 데이터 셋 이름을 'MemberGrade'라고 입력합니다. 매뉴얼 모드로 변경한 후 쿼리문을 아래와 같이 입력한 후 실행 아이콘( )을 클릭하여 실행 결과를 확인합니다.

```
select distinct member_card
from customer
```

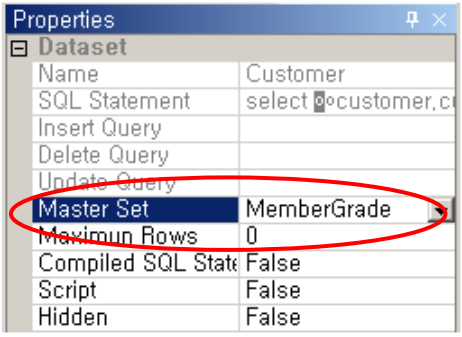
- 'Foodmart'를 마우스 오른쪽 버튼으로 선택하여 나타나는 팝업 메뉴에서 [Add Query Dataset] 메뉴를 선택하여 데이터 셋을 하나 더 추가하고, 데이터 셋 이름을 'Customer'라고 입력합니다. 매뉴얼 모드로 변경한 후 쿼리문을 아래와 같이 입력하고, 실행 아이콘( )을 클릭하여 실행 결과를 확인합니다.

```
select
    customer.customer_id,
```

```
    customer.fname,
    customer.lname,
    customer.gender,
    customer.phone1,
    customer.phone2,
    customer.birthdate,
    customer.occupation,
    customer.city,
    customer.country
from customer

where customer.member_card = '#MemberGrade.member_card#'
```

➢ 추가된 데이터 셋 중 'Customer'를 선택하고 왼쪽의 속성창에서 'Master Dataset' 속성 값을 'MemberGrade'로 설정하여 마스터-디테일 데이터 셋으로 설정합니다.



➢ 예제의 ODI 디자인 화면처럼 ODI 디자인이 완료되면, 리파지토리 루트 폴더에 'Table.odi' 파일로 저장합니다. 본 매뉴얼에서 리파지토리 루트 폴더는 'C:\Repository'입니다.

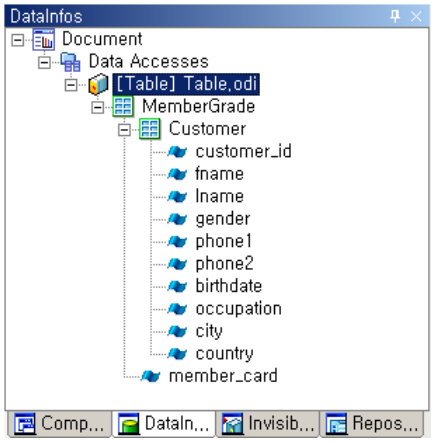
Step 2 ODI 파일 추가

OZ Application Designer를 실행하여 프로젝트를 새로 만들고 ODI 파일을 추가합니다.

➢ OZ Application Designer의 [File] 메뉴의 [New] 메뉴를 선택하거나 툴바의 새로 만들기 아이콘( )을 클릭하여 프로젝트를 새로 만듭니다.

➢ [File] 메뉴의 [Insert New ODI] 메뉴를 선택하거나 리파지토리창에서 ODI 파일을 더블클릭하여 'Table.odi' 파일을 추가합니다. 추가된 ODI 파일의 정보는 데이

터 트리창에서 확인할 수 있습니다.



Step 3 컴포넌트 추가, 데이터 셋 연결, 레이아웃

예제의 미리보기 화면처럼 Board에 Label, ComboBox, Table을 추가하고 데이터 셋 설정, 레이아웃 등을 디자인합니다.

- Label 추가, 디자인
  - Label을 추가하고 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
BackColor	212, 208, 200	**TextColor**	0, 0, 160
LineColor	212, 208, 200	**Font**	Arial, 14pt, Bold
Text	> Customer Grade :	**Location**	16, 18
Size	136, 16	**-**	-

- ComboBox 추가, 디자인, 데이터 셋 연결
  - 고객의 멤버쉽 카드 등급을 나타낼 ComboBox를 추가하고, 추가한 ComboBox의 속성 값을 아래와 같이 변경합니다.

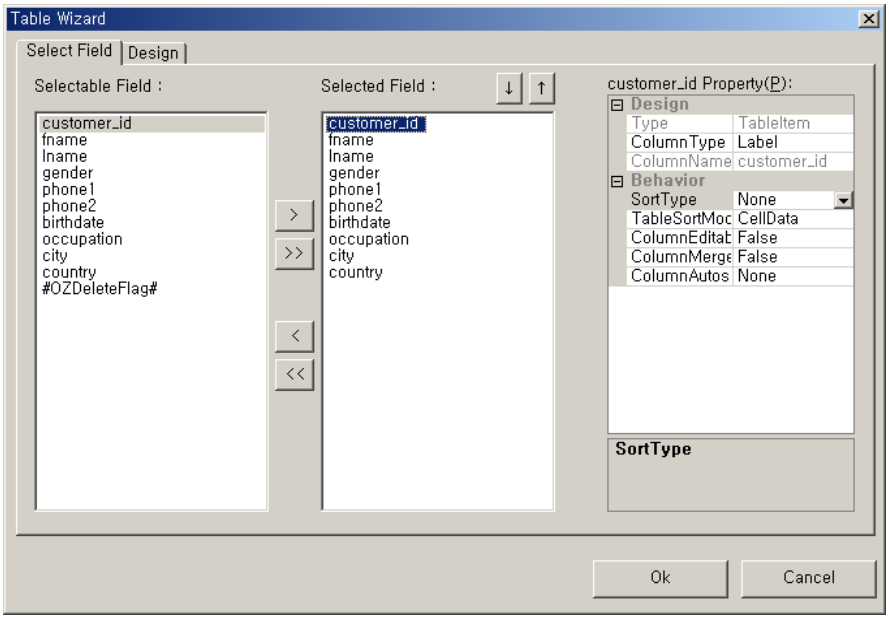
Property	Value	Property	Value
Font	Arial, 12pt	**DropDownStyle**	DropDownList
Location	160,16	**Size**	168, 24
ODIKey	Table	**DataSet**	MemberGrade
Field	member_card	**FireRowCursorChange**	True

➢ Table 추가, 디자인, 데이터 셋 연결

- 선택한 멤버쉽 카드 등급에 해당하는 고객 정보를 표시할 Table을 추가하고, 추가한 Table의 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
IndicatorStartIndex	1	**Location**	16, 48
Size	776, 536	**ODIKey**	Table
DataSet	Customer	**-**	-

- Table을 선택하고 마우스 오른쪽 버튼을 눌러 나타나는 팝업 메뉴에서 [Wizard]를 클릭하여 마법사를 실행한 후 아래 그림처럼 표시할 필드를 추가합니다.



추가된 필드의 'SortType' 속성 값을 정렬할 방식에 따라 'Numeric' 또는 'Text'로 설정하고 [Ok] 버튼을 클릭합니다.

- Table의 CellHeader, CellIndicator를 다중 선택하여 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
BackColor	128, 128, 255	**TextColor**	255, 255, 255
Font	Arial, 12pt, Bold	**TextAlign**	MiddleRight

CellFrameColor	192, 192, 192	Size	40, 24

- Table의 CellTitle1, ..., CellTitle10을 다중 선택하여 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
BackColor	128, 128, 255	TextColor	255, 255, 255
Font	Arial, 12pt, Bold	TextAlign	MiddleCenter
CellFrameColor	192, 192, 192	-	-

- Table의 CellTitle1, ..., CellTitle10의 속성 중 'Text' 속성 값을 'Customer ID, First Name, Last Name, Gender, Phone1, Phone2, Birthdate, Occupation, City, Country' 순서대로 변경합니다.

- Table의 CellLabel1, ..., CellLabel10을 다중 선택하여 속성 값을 아래와 같이 변경합니다.

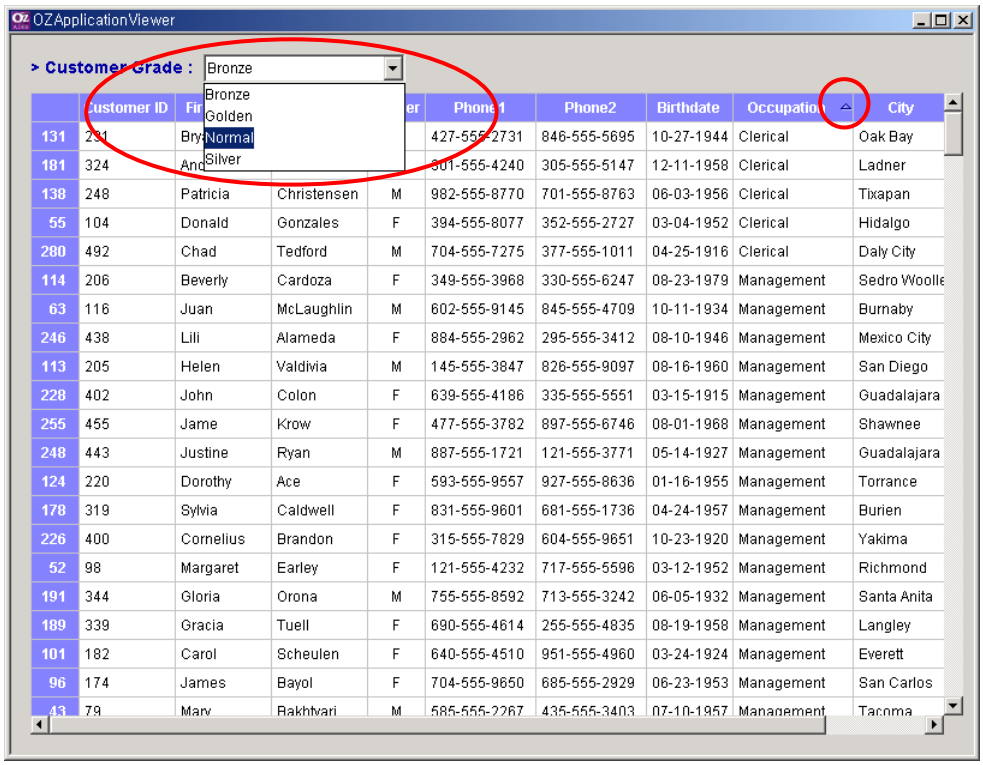
Property	Value	Property	Value
Font	Arial, 12pt	InternalMargin	4, 4, 4, 4

- Table의 CellLabel4를 선택하여 'TextAlign' 속성 값을 'MiddleCenter'로 변경하고, CellLabel7을 선택하여 'TextAlign' 속성 값을 'MiddleCenter'로 'Pattern' 속성 값을 'date_MM-dd-yyyy'로 변경하고, 그 외 CellLabel을 선택하여 'TextAlign' 속성 값을 'MiddleLeft'로 변경합니다.

Step 4 미리보기

[File] 메뉴의 [Preview]를 클릭하거나 툴바의 미리보기 아이콘()을 클릭하여 미리보기합니다.

고객 멤버쉽 카드 등급을 나타내는 ComboBox에서 다른 카드 등급을 선택하여 해당 고객을 검색하고 타이틀을 클릭하여 데이터를 정렬하여 조회합니다.

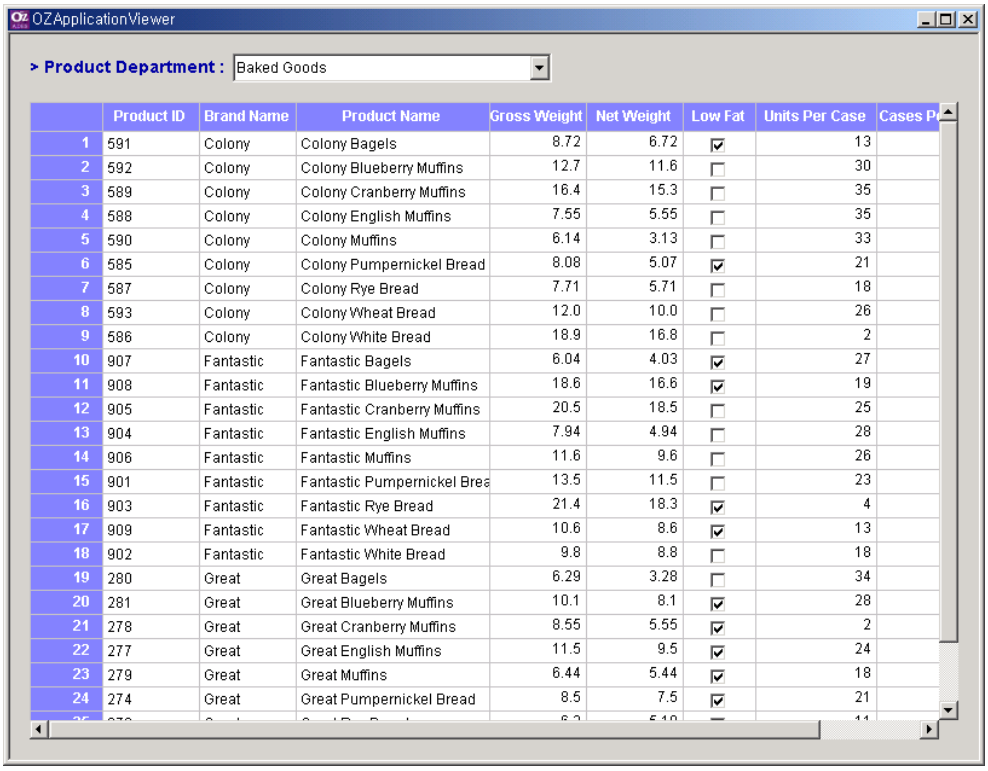


	Customer ID	Fir		er	Phone1	Phone2	Birthdate	Occupation	City
131	291	Bry			427-555-2731	846-555-5695	10-27-1944	Clerical	Oak Bay
181	324	And			301-555-4240	305-555-5147	12-11-1958	Clerical	Ladner
138	248	Patricia	Christensen	M	982-555-8770	701-555-8763	06-03-1956	Clerical	Tixapan
55	104	Donald	Gonzales	F	394-555-8077	352-555-2727	03-04-1952	Clerical	Hidalgo
280	492	Chad	Tedford	M	704-555-7275	377-555-1011	04-25-1916	Clerical	Daly City
114	206	Beverly	Cardoza	F	349-555-3968	330-555-6247	08-23-1979	Management	Sedro Woolle
63	116	Juan	McLaughlin	M	602-555-9145	845-555-4709	10-11-1934	Management	Burnaby
246	438	Lili	Alameda	F	884-555-2962	295-555-3412	08-10-1946	Management	Mexico City
113	205	Helen	Valdivia	M	145-555-3847	826-555-9097	08-16-1960	Management	San Diego
228	402	John	Colon	F	639-555-4186	335-555-5551	03-15-1915	Management	Guadalajara
255	455	Jame	Krow	F	477-555-3782	897-555-6746	08-01-1968	Management	Shawnee
248	443	Justine	Ryan	M	887-555-1721	121-555-3771	05-14-1927	Management	Guadalajara
124	220	Dorothy	Ace	F	593-555-9557	927-555-8636	01-16-1955	Management	Torrance
178	319	Sylvia	Caldwell	F	831-555-9601	681-555-1736	04-24-1957	Management	Burien
226	400	Cornelius	Brandon	F	315-555-7829	604-555-9651	10-23-1920	Management	Yakima
52	98	Margaret	Earley	F	121-555-4232	717-555-5596	03-12-1952	Management	Richmond
191	344	Gloria	Orona	M	755-555-8592	713-555-3242	06-05-1932	Management	Santa Anita
189	339	Gracia	Tuell	F	690-555-4614	255-555-4835	08-19-1958	Management	Langley
101	182	Carol	Scheulen	F	640-555-4510	951-555-4960	03-24-1924	Management	Everett
96	174	James	Bayol	F	704-555-9650	685-555-2929	06-23-1953	Management	San Carlos
43	79	Mary	Bakhtyari	M	585-555-2267	435-555-3403	07-10-1957	Management	Tacoma

스크립트로 Grid에 데이터 표시

다음은 스크립트를 코딩하여 Grid에 데이터를 표시하는 방법을 아래와 같은 예제를 통해 설명합니다.

미리보기 화면

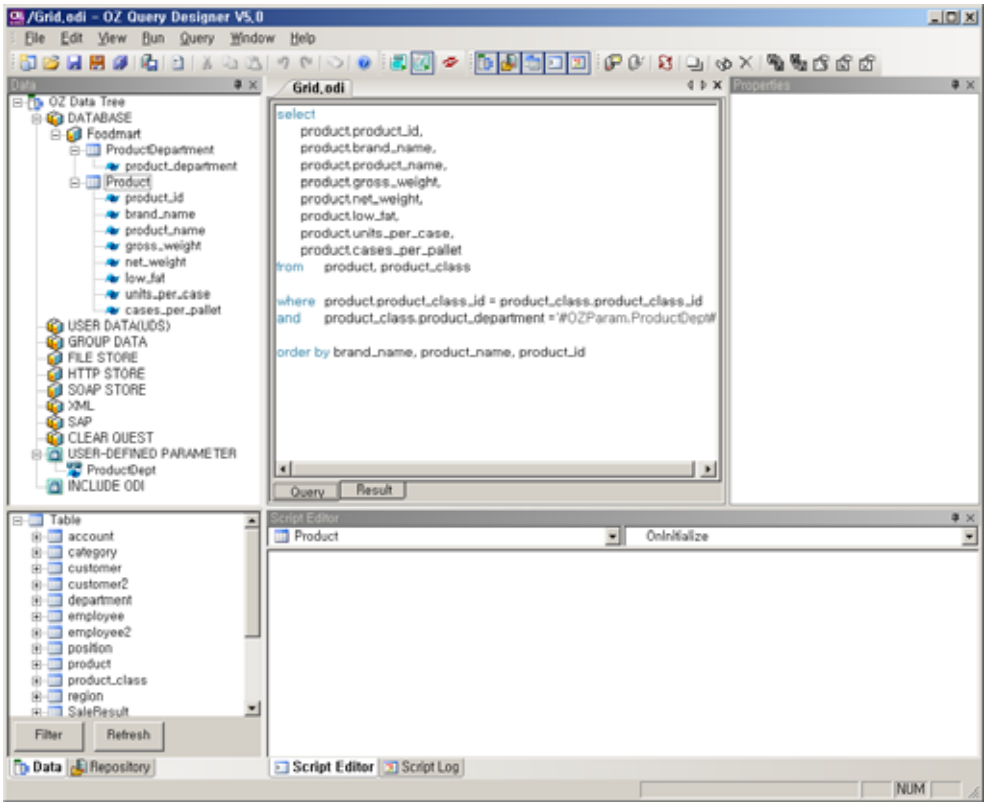


	Product ID	Brand Name	Product Name	Gross Weight	Net Weight	Low Fat	Units Per Case	Cases P
1	591	Colony	Colony Bagels	8.72	6.72	☑	13	
2	592	Colony	Colony Blueberry Muffins	12.7	11.6	☐	30	
3	589	Colony	Colony Cranberry Muffins	16.4	15.3	☐	35	
4	588	Colony	Colony English Muffins	7.55	5.55	☐	35	
5	590	Colony	Colony Muffins	6.14	3.13	☐	33	
6	585	Colony	Colony Pumpernickel Bread	8.08	5.07	☑	21	
7	587	Colony	Colony Rye Bread	7.71	5.71	☐	18	
8	593	Colony	Colony Wheat Bread	12.0	10.0	☐	26	
9	586	Colony	Colony White Bread	18.9	16.8	☐	2	
10	907	Fantastic	Fantastic Bagels	6.04	4.03	☑	27	
11	908	Fantastic	Fantastic Blueberry Muffins	18.6	16.6	☑	19	
12	905	Fantastic	Fantastic Cranberry Muffins	20.5	18.5	☐	25	
13	904	Fantastic	Fantastic English Muffins	7.94	4.94	☐	28	
14	906	Fantastic	Fantastic Muffins	11.6	9.6	☐	26	
15	901	Fantastic	Fantastic Pumpernickel Brea	13.5	11.5	☐	23	
16	903	Fantastic	Fantastic Rye Bread	21.4	18.3	☑	4	
17	909	Fantastic	Fantastic Wheat Bread	10.6	8.6	☑	13	
18	902	Fantastic	Fantastic White Bread	9.8	8.8	☐	18	
19	280	Great	Great Bagels	6.29	3.28	☐	34	
20	281	Great	Great Blueberry Muffins	10.1	8.1	☑	28	
21	278	Great	Great Cranberry Muffins	8.55	5.55	☑	2	
22	277	Great	Great English Muffins	11.5	9.5	☑	24	
23	279	Great	Great Muffins	6.44	5.44	☑	18	
24	274	Great	Great Pumpernickel Bread	8.5	7.5	☑	21	

기능

- 제품 종류를 ComboBox에 표시하고, 스크립트를 코딩하여 선택한 제품 종류에 해당하는 제품 정보를 Grid에 표시합니다.

ODI 디자인



기능

- 연결할 Database는 ODBC 데이터 원본인 'Foodmart'이며, 제품 종류를 가져오는 데이터 셋인 'ProductDepartment', 제품 종류에 해당하는 제품 데이터를 가져오는 데이터 셋인 'Product'로 구성됩니다.
- 사용자 패러미터인 'ProductDept'를 이용하여 제품 종류를 조건값으로 설정합니다.

Step 1 ODI 파일 만들기

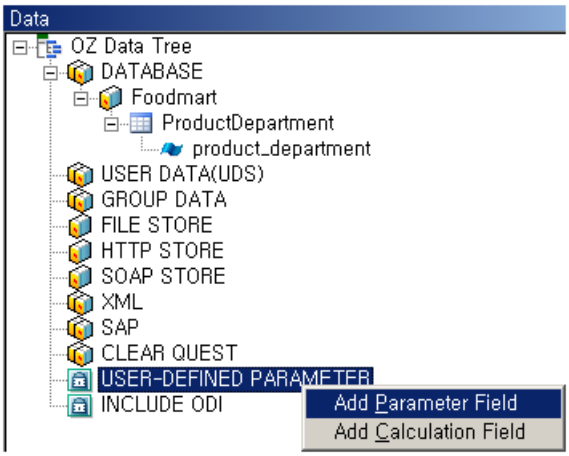
제품의 종류와 선택한 제품 종류에 해당하는 제품 정보를 표시하는 데이터 셋을 포함하는 ODI 파일을 만듭니다.

- OZ Query Designer를 실행하여 연결할 Database 정보를 'Foodmart'로 설정합니다.

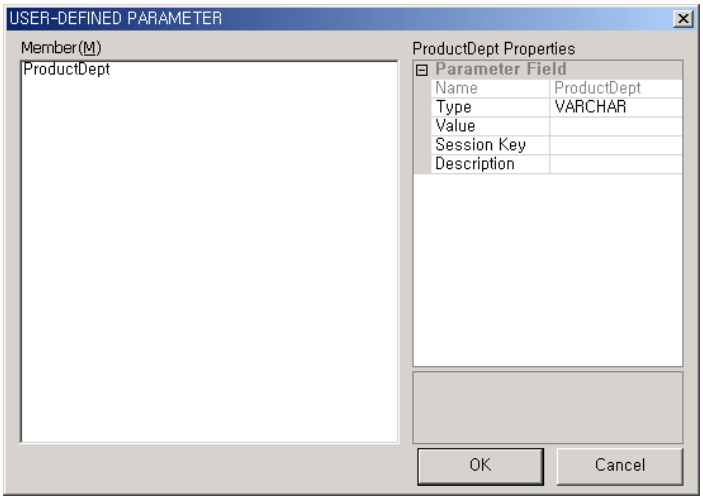
- [Add Query Dataset] 메뉴를 이용하여 데이터 셋을 추가합니다. 데이터 셋 이름을 'ProductDepartment'라고 입력한 후 쿼리문을 아래와 같이 입력하고, 실행 아이콘( )을 클릭하여 실행 결과를 확인합니다.

```
select distinct product_department
from product_class
```

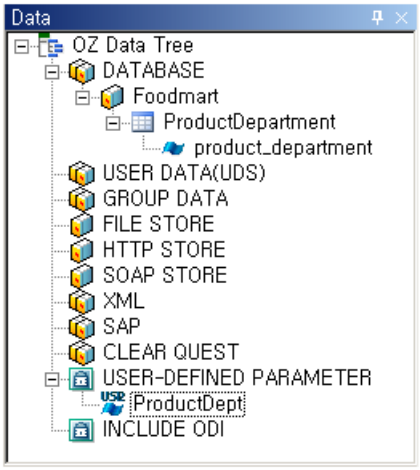
- 제품 종류를 선택할 수 있도록 사용자 패러미터를 추가합니다.
  - 'OZ Data Window'에서 'USER PARAMETER'를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Add Parameter Field] 메뉴를 선택합니다.



  - '패러미터 추가 다이얼로그'에서 패러미터 이름을 'ProductDept'로 입력하고 데이터 타입은 'VARCHAR'로 선택합니다.



- [OK] 버튼을 클릭하여 패러미터를 추가하고, 'OZ Data Window'에서 추가된 패러미터를 확인합니다.



➢ [Add Query Dataset] 메뉴를 이용하여 데이터 셋을 추가합니다. 데이터 셋 이름을 'Product'라고 입력한 후 쿼리문을 아래와 같이 입력하고, 실행 아이콘( )을 클릭하여 실행 결과를 확인합니다.

```
select
    product.product_id,
    product.brand_name,
    product.product_name,
    product.gross_weight,
    product.net_weight,
    product.low_fat,
    product.units_per_case,
    product.cases_per_pallet
```

```
from product, product_class

where product.product_class_id = product_class.product_class_id
and product_class.product_department ='#OZParam.ProductDept#'

order by brand_name, product_name, product_id
```

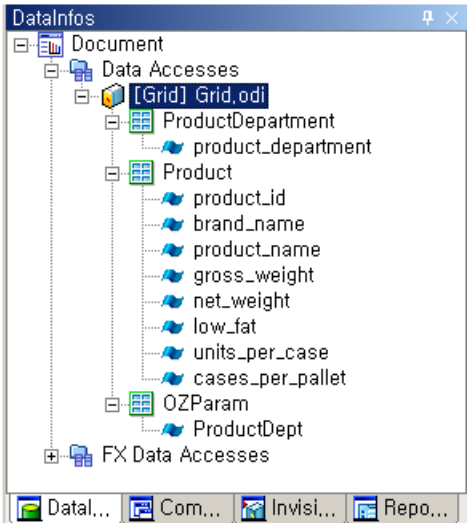
➢ 예제의 ODI 디자인 화면처럼 ODI 디자인이 완료되면, 리파지토리 루트 폴더에 'Grid.odi' 파일로 저장합니다. 본 매뉴얼에서 리파지토리 루트 폴더는 'C:\Repository'입니다.

Step 2 ODI 파일 추가

OZ Application Designer를 실행하여 프로젝트를 새로 만들고 ODI 파일을 추가합니다.

➢ OZ Application Designer의 [File] 메뉴의 [Create New Project] 메뉴를 선택하거나 툴바의 새로 만들기 아이콘( )을 클릭하여 프로젝트를 새로 만듭니다.

➢ [File] 메뉴의 [Insert New ODI] 메뉴를 선택하거나 리파지토리창에서 ODI 파일을 더블클릭하여 'Grid.odi' 파일을 추가합니다. 추가된 ODI 파일의 정보는 데이터 트리창에서 확인할 수 있습니다.



Step 3 컴포넌트 추가, 데이터 셋 연결, 레이아웃

예제의 미리보기 화면처럼 Board에 Label, ComboBox, Grid를 추가하고 데이터 셋 연결, 레이아웃 등을 디자인합니다.

➢ Label 추가, 디자인

- Label을 추가하고, 디자인 화면처럼 글꼴(Font), 글자색(TextColor), 위치(Location), 크기(Size), 텍스트(Text) 등을 변경합니다.

➢ ComboBox 추가, 디자인, 데이터 셋 연결

- 제품 종류를 나타낼 ComboBox를 추가하고, 디자인 화면처럼 글자색(TextColor), 위치(Location), 크기(Size) 등을 변경한 후 속성 값을 아래와 같이 설정합니다.

Property	Value	Property	Value
DropDownStyle	DropDownList	**ODIKey**	Grid
DataSet	ProductDepartment	**Field**	product_department

➢ Grid 추가, 디자인

- 선택한 제품 종류에 해당하는 제품 정보를 표시할 Grid를 추가하고, 디자인 화면처럼 위치(Location), 크기(Size) 등을 변경한 후 속성 값을 아래와 같이 설정합니다.

Property	Value	Property	Value
IndicatorStartIndex	1	**FixedColCount**	3

- Grid의 CellHeader, CellIndicator, CellTitle을 다중 선택하여, 디자인 화면처럼 배경색(BackColor), 셀테두리색(CellFrameColor), 글꼴(Font), 글자색(TextColor), 크기(Size) 등을 변경합니다.

- Grid의 CellHeader, CellIndicator를 다중 선택하여, 텍스트를 중앙 오른쪽 정렬(TextAlign = MiddleRight)로 변경하고, CellTitle을 선택하여 텍스트를 중앙 가운데 정렬(TextAlign=MiddleCenter)로 변경합니다.

- Grid의 CellLabel을 선택하여, 디자인 화면처럼 글꼴(Font)과 내부 여백(InternalMargin = '4,4,4,4') 등을 변경합니다.

Step 4 스크립트 코딩

ComboBox로 선택한 제품 종류에 따라 Grid에 제품 정보를 표시하기 위해 스크립트를

코딩합니다.

- Grid에 제품 정보를 표시하고 Grid의 Title, Label의 레이아웃을 디자인하는 스크립트를 전역 함수로 코딩합니다. 전역 함수로 코딩하기 위해 스크립트창에서 'Document'의 'GlobalFunction'을 선택하고 아래와 같이 코딩합니다.

```
function displayProduct(ProductDeptValue) {

//ODI, DataSet
    var DataManager = _GetDataManager();
    var DataModule = DataManager.GetDataModule("Grid");
    DataModule.UpdateOZParam("ProductDept", ProductDeptValue);
    var DataSet = DataModule.GetDataSet("Product", null, true);
    var DataCount = DataSet.GetRowCount();

    var grdProduct = _GetBoard("Board1").GetChild("Grid1");
//Reset Grid
    if(grdProduct.GetRowCount() > 0) {
       grdProduct.SelectCellRange(-1, -1, -1, -1);
       grdProduct.DeleteCellInRange();
    }
    grdProduct.AddColumn(8);
    grdProduct.AddRow(DataCount);

    grdProduct.SelectCellRange(0, 5, DataCount, 5);
    if(DataCount < 1) { return;  }
       grdProduct.ChangeCellTypeInRange(
                                      CellComponentTypeConst.CheckBox);

//Display Data
    for(i=0; i<DataCount; i++) {
       grdProduct.GetCell(i,0).Text =
                  DataSet.GetStringData(i, "product_id");
       grdProduct.GetCell(i,1).Text =
                  DataSet.GetStringData(i, "brand_name");
       grdProduct.GetCell(i,2).Text =
                  DataSet.GetStringData(i, "product_name");
       grdProduct.GetCell(i,3).Text =
                  DataSet.GetStringData(i, "gross_weight");
       grdProduct.GetCell(i,4).Text =
                  DataSet.GetStringData(i, "net_weight");
       grdProduct.GetCell(i,5).Checked =
                  DataSet.GetStringData(i, "low_fat");
       grdProduct.GetCell(i,6).Text =
                  DataSet.GetStringData(i, "units_per_case");
       grdProduct.GetCell(i,7).Text =
                  DataSet.GetStringData(i, "cases_per_pallet");
    }
```

```
}

function formatGrid(GridRowCount) {
    var grdProduct = _GetBoard("Board1").GetChild("Grid1");
//Grid Title
    grdProduct.GetTitle(0).Text = "Product ID";
    grdProduct.GetTitle(1).Text = "Brand Name";
    grdProduct.GetTitle(2).Text = "Product Name";
    grdProduct.GetTitle(3).Text = "Gross Weight";
    grdProduct.GetTitle(4).Text = "Net Weight";
    grdProduct.GetTitle(5).Text = "Low Fat";
    grdProduct.GetTitle(6).Text = "Units Per Case";
    grdProduct.GetTitle(7).Text = "Cases Per Pallet";

    grdProduct.GetTitle(0).SetSize(80,24);
    grdProduct.GetTitle(1).SetSize(80,24);
    grdProduct.GetTitle(2).SetSize(160,24);
    grdProduct.GetTitle(3).SetSize(80,24);
    grdProduct.GetTitle(4).SetSize(80,24);
    grdProduct.GetTitle(5).SetSize(60,24);
    grdProduct.GetTitle(6).SetSize(100,24);
    grdProduct.GetTitle(7).SetSize(100,24);

//Grid Label
    for(i=0; i<GridRowCount; i++) {
    grdProduct.GetCell(i,0).TextAlign = AlignmentConst.MiddleLeft;
    grdProduct.GetCell(i,1).TextAlign = AlignmentConst.MiddleLeft;
    grdProduct.GetCell(i,2).TextAlign = AlignmentConst.MiddleLeft;
    grdProduct.GetCell(i,3).TextAlign = AlignmentConst.BottomRight;
    grdProduct.GetCell(i,4).TextAlign = AlignmentConst.BottomRight;
    grdProduct.GetCell(i,5).CheckAlign = AlignmentConst.BottomCenter;
    grdProduct.GetCell(i,6).TextAlign = AlignmentConst.BottomRight;
    grdProduct.GetCell(i,7).TextAlign = AlignmentConst.BottomRight;
}

//Refresh ScrollBars
    grdProduct.RefreshScrollBars();

}
```

- 제품 종류를 나타내는 ComboBox를 선택하면 선택된 제품 종류에 해당하는 제품 정보를 Grid에 표시하고 Grid의 포맷을 설정하는 전역 함수를 호출하는 스크립트를 코딩합니다. 스크립트창에서 ComboBox의 'OnSelectedIndexChanged' 이벤트를 선택하고 아래와 같이 코딩합니다.

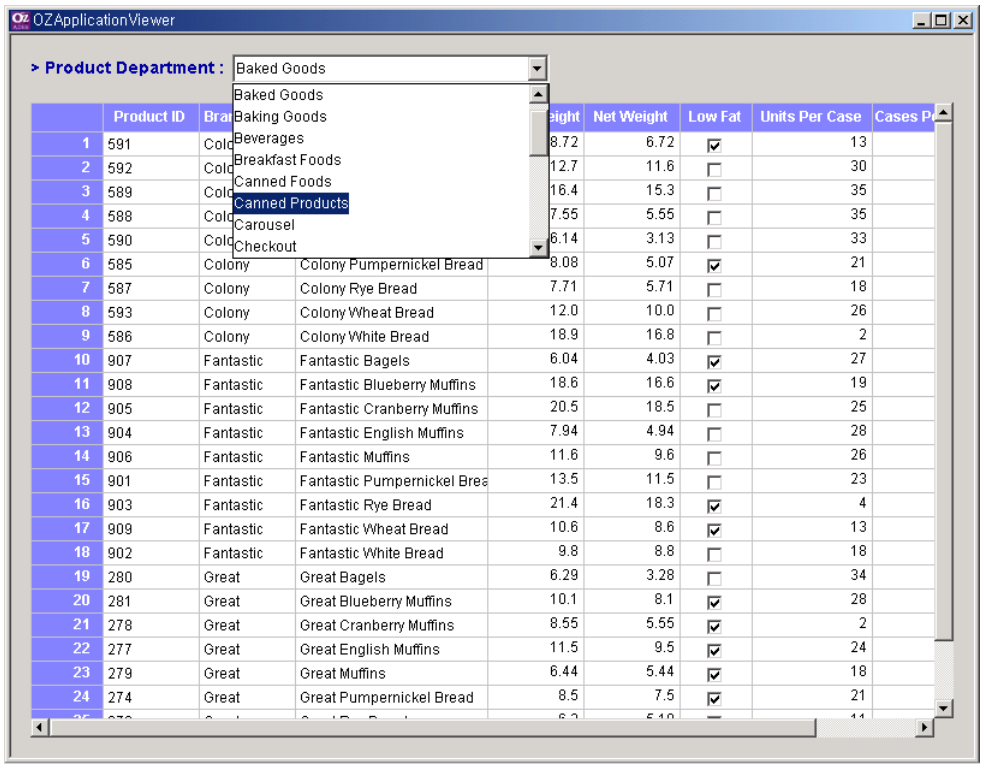
```
displayProduct(ComboBox1.Value);
```

```
formatGrid(Grid1.GetRowCount());
```

Step 5 미리보기

[File] 메뉴의 [Preview]를 클릭하거나 툴바의 미리보기 아이콘()을 클릭하여 미리보기합니다.

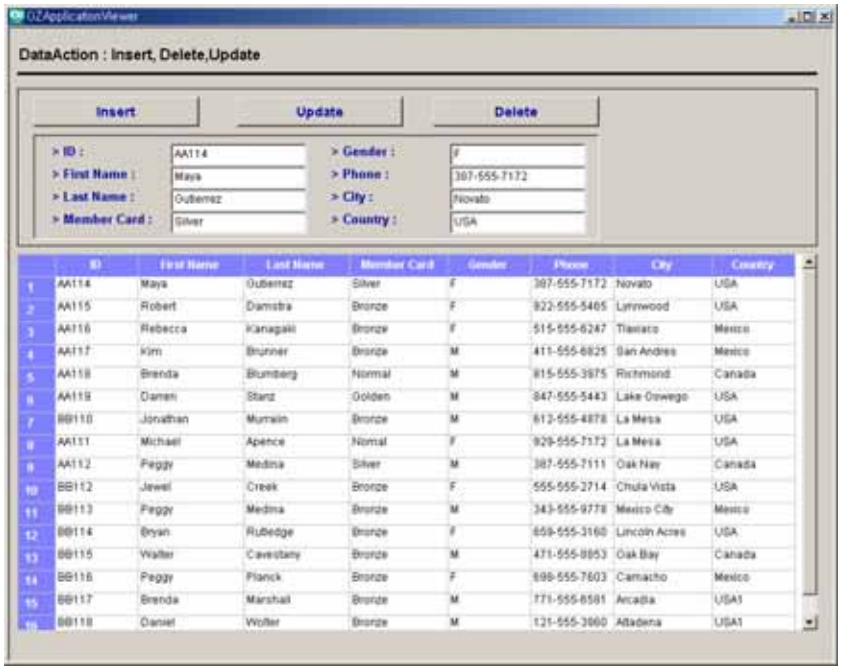
제품 종류를 나타내는 ComboBox에서 다른 제품 종류를 선택하여 해당 제품 정보를 검색합니다.



DataAction : Insert, Delete,Update

다음은 데이터를 추가, 삭제, 변경하는 방법을 아래와 같은 예제를 통해 설명합니다.

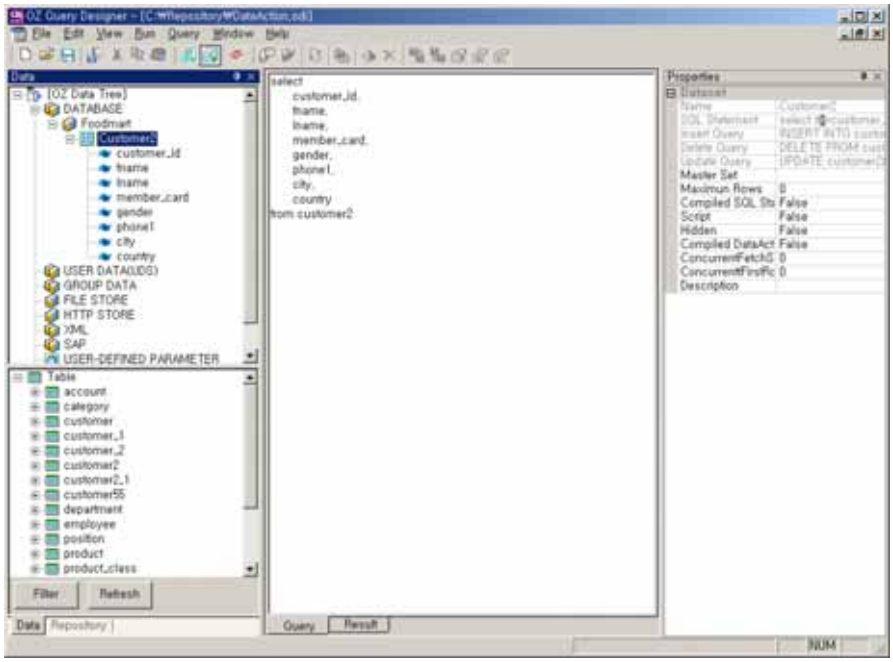
미리보기 화면



기능

- 고객 정보를 Table에 표시합니다.
- 추가할 고객 정보를 텍스트 박스에 직접 입력하고 [Insert] 버튼을 클릭하면 Table에 행이 추가되면서 텍스트 박스에 입력한 고객 정보가 추가됩니다.
- 변경할 고객 정보를 텍스트 박스에서 변경하고 [Update] 버튼을 클릭하면 고객 정보가 변경됩니다.
- 삭제할 고객 정보를 Table에서 선택한 후 [Delete] 버튼을 클릭하면 삭제됩니다.

ODI 디자인



기능

- 연결할 Database는 ODBC 데이터 원본인 'Foodmart'이며, 고객 정보를 가져오는 데이터 셋인 'Customer2'로 구성됩니다.
- 고객 정보를 추가, 삭제, 변경하기 위해 'Customer2' 데이터 셋에 삽입, 삭제, 변경 쿼리문을 추가합니다.

Step 1 ODI 파일 만들기

고객 정보를 가져오는 데이터 셋과 삽입, 삭제, 변경 쿼리문을 포함하는 ODI 파일을 만듭니다. 삽입, 삭제, 변경 쿼리문을 만드는 자세한 방법은 'OZ Query Designer User's Guide'를 참조하십시오.

- OZ Query Designer를 실행하여 연결할 Database 정보를 'Foodmart'로 설정합니다.

- [Add Query Dataset] 메뉴를 이용하여 데이터 셋을 추가합니다. 데이터 셋 이름을 'Customer2' 라고 입력한 후 쿼리문을 아래와 같이 입력하고, 실행 아이콘

()을 클릭하여 실행 결과를 확인합니다.

```
select
customer_id, fname, lname, member_card, gender, phone1, city,
country
from customer2
```

➢ 고객 정보를 추가, 삭제, 변경하기 위해 'Customer2' 데이터 셋에 삽입, 삭제, 변경 쿼리문을 추가합니다.

- 삽입 쿼리

  'OZ Data Tree'에서 'Customer2' 데이터 셋을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Insert Query] 메뉴를 클릭합니다. 삽입 쿼리문 입력창에 쿼리문을 아래와 같이 입력하고 [OK] 버튼을 클릭하여 삽입 쿼리문을 입력합니다.

```
INSERT INTO customer2
(#@ARG_SF1#,#@ARG_SF2#,#@ARG_SF3#,#@ARG_SF4#,#@ARG_SF5#,#@ARG_SF6#,
#@ARG_SF7#,#@ARG_SF8#)
VALUES
('#@ARG_SV1#','#@ARG_SV2#','#@ARG_SV3#','#@ARG_SV4#','#@ARG_SV5#','
#@ARG_SV6#','#@ARG_SV7#','#@ARG_SV8#')
```

- 삭제 쿼리

  'OZ Data Tree'에서 'Customer2' 데이터 셋을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Delete Query] 메뉴를 클릭합니다. 삭제 쿼리문 입력창에 쿼리문을 아래와 같이 입력하고 [OK] 버튼을 클릭하여 삭제 쿼리문을 입력합니다.

```
DELETE FROM customer2
WHERE #@ARG_DF1# = '#@ARG_DV1#'
```

- 변경 쿼리

  'OZ Data Tree'에서 'Customer2' 데이터 셋을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Update Query] 메뉴를 선택합니다. 변경 쿼리문 입력창에 쿼리문을 아래와 같이 입력하고 [OK] 버튼을 클릭하여 변경 쿼리문을 입력합니다.

```
UPDATE customer2
SET #@ARG_SF1# = '#@ARG_SV1#',
     #@ARG_SF2# = '#@ARG_SV2#',
     #@ARG_SF3# = '#@ARG_SV3#',
     #@ARG_SF4# = '#@ARG_SV4#',
     #@ARG_SF5# = '#@ARG_SV5#',
     #@ARG_SF6# = '#@ARG_SV6#',
```

```
      #@ARG_SF7# = '#@ARG_SV7#',
      #@ARG_SF8# = '#@ARG_SV8#'
WHERE #@ARG_DF1# = '#@ARG_DV1#'
```

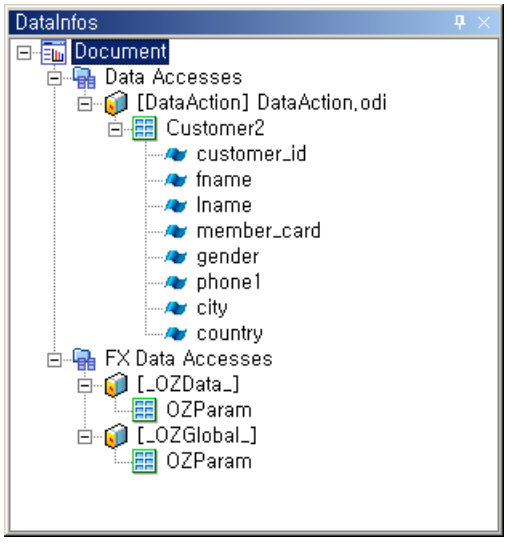
- 예제의 ODI 디자인 화면처럼 ODI 디자인이 완료되면, 리파지토리 루트 폴더에 'DataAction.odi' 파일로 저장합니다. 본 매뉴얼에서 리파지토리 루트 폴더는 'C:\Repository'입니다.

Step 2 ODI 파일 추가

OZ Application Designer를 실행하여 프로젝트를 새로 만들고 ODI 파일을 추가합니다.

- OZ Application Designer의 [File] 메뉴의 [Create New Project] 메뉴를 선택하거나 툴바의 새로 만들기 아이콘( )을 클릭하여 프로젝트를 새로 만듭니다.

- [File] 메뉴의 [Insert New ODI] 메뉴를 선택하거나 리파지토리창에서 ODI 파일을 더블클릭하여 'DataAction.odi' 파일을 추가합니다. 추가된 ODI 파일의 정보는 데이터 트리창에서 확인할 수 있습니다.



Step 3 컴포넌트 추가, 데이터 셋 연결, 레이아웃

예제의 미리보기 화면처럼 Board에 Label, Panel, TextBox, Button, Table을 추가하고 데이터 셋 연결, 레이아웃 등을 디자인합니다.

➢ Label 추가, 디자인

- Label을 추가하고, 디자인 화면처럼 글꼴(Font), 글자색(TextColor), 위치(Location), 크기(Size), 텍스트(Text) 등을 변경합니다.

➢ Panel 추가, 디자인

- Panel을 추가하고, 디자인 화면처럼 위치(Location), 크기(Size) 등을 변경합니다.

➢ Button 추가, 디자인

- Button을 3개 추가하고, 디자인 화면처럼 글꼴(Font), 위치(Location), 크기(Size), 텍스트(Text) 등을 변경합니다.

➢ TextBox 추가, 디자인, 데이터 셋 연결

- 고객 정보를 표시할 TextBox를 추가하고, 디자인 화면처럼 위치(Location), 크기(Size)등을 변경한 후 추가한 TextBox의 속성 값을 아래와 같이 각각 변경합니다.

Property	Value	Property	Value
ODIKey	DataAction	**DataSet**	Customer2
Field	customer_id, fname,lname, member_card, gender, phone1, city, county	**-**	-

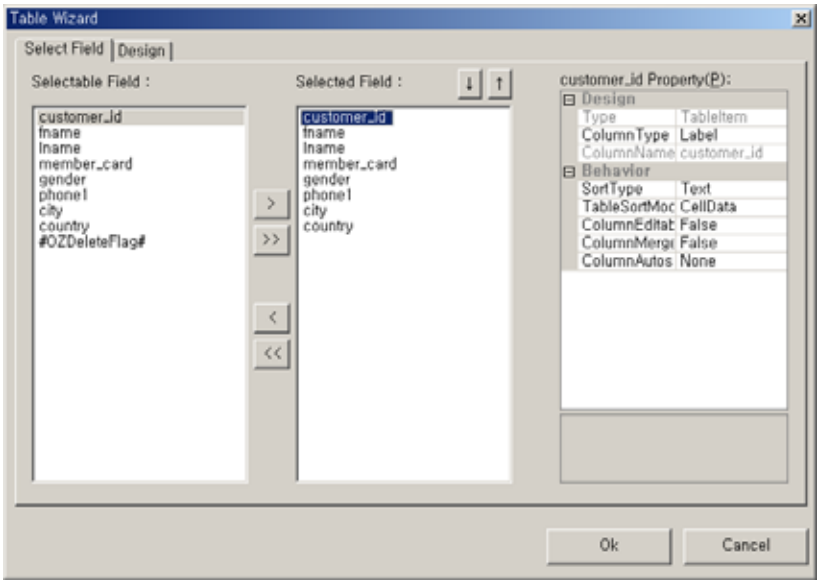
➢ Table 추가, 디자인, 데이터 셋 연결

- 고객 정보를 표시할 Table을 추가하고, 디자인 화면처럼 위치(Location), 크기(Size)를 변경한 후 추가한 Table의 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
IndicatorStartIndex	1	**ODIKey**	DataAction
DataSet	Customer2	**FireRowCursorChange**	True

- Table을 선택하고 마우스 오른쪽 버튼을 눌러 나타나는 팝업 메뉴에서 [Wizard]를 클릭하여 마법사를 실행하여 아래 그림처럼 표시할 필드를 추가

합니다.



추가된 모든 필드의 속성 값 중 'SortType' 속성 값을 'Text'로 변경하고 [Ok] 버튼을 클릭합니다

- Table의 CellHeader, CellIndicator, CellTitle, CellLabel을 선택하여 디자인 화면처럼 배경색(BackColor), 셀테두리색(CellFrameColor), 글꼴(Font), 글자색(TextColor), 크기(Size), 텍스트(Text), 텍스트 정렬(TextAlign) 등을 변경합니다.

Step 4 스크립트 코딩

고객 정보를 추가, 삭제, 변경하는 스크립트를 코딩합니다.

➢ [Insert] 버튼의 'OnClick' 이벤트를 선택하여 텍스트 박스에 입력한 고객 정보가 삽입되도록 스크립트를 코딩합니다.

```
//Create DataSet Object
   var DataModule = _GetDataManager().GetDataModule("DataAction");
   var DataSet = DataModule.GetDataSet("Customer2");
   var DataAction = new Array();
   DataAction[0] = DataSet.MakeDefaultDataAction();

//Create Insert Action
   DataAction[0].DataActionType = DataActionTypeConst.Insert;
   DataModule.CleanUpQueuedActions();

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

```
    arrField[0] = "customer_id";  arrValue[0] = TextBox_ID.Text;
    arrField[1] = "fname";        arrValue[1] = TextBox_FirstName.Text;
    arrField[2] = "lname";        arrValue[2] = TextBox_LastName.Text;
    arrField[3] = "member_card";  arrValue[3] = TextBox_MemberCard.Text;
    arrField[4] = "gender";       arrValue[4] = TextBox_Gender.Text;
    arrField[5] = "phone1";       arrValue[5] = TextBox_Phone.Text;
    arrField[6] = "city";         arrValue[6] = TextBox_City.Text;
    arrField[7] = "country";      arrValue[7] = TextBox_Country.Text;

//Create & Commit Queue DataAction
    DataAction[0].SetSourceNames(arrField);
    DataAction[0].SetSourceValues(arrValue);

    DataSet.QueueActionObjects(DataAction);
    var Result = DataModule.CommitQueuedActions();//Commit
    if(Result == "") {
         DataSet = DataModule.GetDataSet("Customer2", null, true);
         Table1.RefreshDataSet();
    } else {
         _MessageBox(Result, "DataActionError");
    }
```

➢ [Update] 버튼의 'OnClick' 이벤트를 선택하여 텍스트 박스에 입력한 고객 정보가 변경되도록 스크립트를 코딩합니다.

```
//Create DataSet Object
    var DataModule = _GetDataManager().GetDataModule("DataAction");
    var DataSet = DataModule.GetDataSet("Customer2");
    var DataAction = new Array();
    DataAction[0] = DataSet.MakeDefaultDataAction();

//Create Update Action
    DataAction[0].DataActionType = DataActionTypeConst.RowUpdate;

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

    arrField[0] = "customer_id";  arrValue[0] = TextBox_ID.Text;
    arrField[1] = "fname";        arrValue[1] = TextBox_FirstName.Text;
    arrField[2] = "lname";        arrValue[2] = TextBox_LastName.Text;
    arrField[3] = "member_card";  arrValue[3] = TextBox_MemberCard.Text;
    arrField[4] = "gender";       arrValue[4] = TextBox_Gender.Text;
    arrField[5] = "phone1";       arrValue[5] = TextBox_Phone.Text;
    arrField[6] = "city";         arrValue[6] = TextBox_City.Text;
    arrField[7] = "country";      arrValue[7] = TextBox_Country.Text;

//Create & Commit Queue DataAction
```

```
    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();//Commit
    if(Result == "") {
         DataSet = DataModule.GetDataSet("Customer2", null, true);
         Table1.RefreshDataSet();
    } else {
         _MessageBox(Result, "DataActionError");
    }
```

➢ [Delete] 버튼의 'OnClick' 이벤트를 선택하여 Table에서 선택한 고객 정보가 삭제되도록 스크립트를 코딩합니다.

```
//Create DataSet Object
    var DataModule = _GetDataManager().GetDataModule("DataAction");
    var DataSet = DataModule.GetDataSet("Customer2");
    var DataAction = new Array();
    DataAction[0] = DataSet.MakeDefaultDataAction();

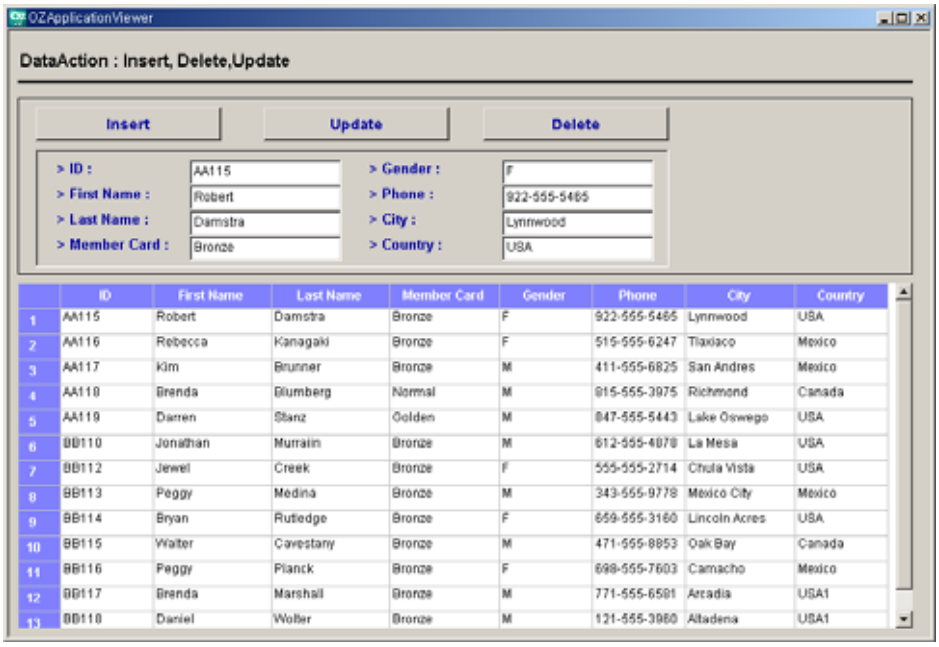
//Create Update Action
    DataAction[0].DataActionType = DataActionTypeConst.Delete;

//Create & Commit Queue DataAction
    DataAction[0].SetTargetValue("customer_id",
    Table1.GetCellInPage(Table1.GetFocusRowIndex,0).Value);
    DataSet.QueueActionObjects(DataAction);
    var Result = DataModule.CommitQueuedActions();//Commit
    if(Result == "") {
         DataSet = DataModule.GetDataSet("Customer2", null, true);
         Table1.RefreshDataSet();
    } else {
         _MessageBox(Result, "DataActionError");
    }
```

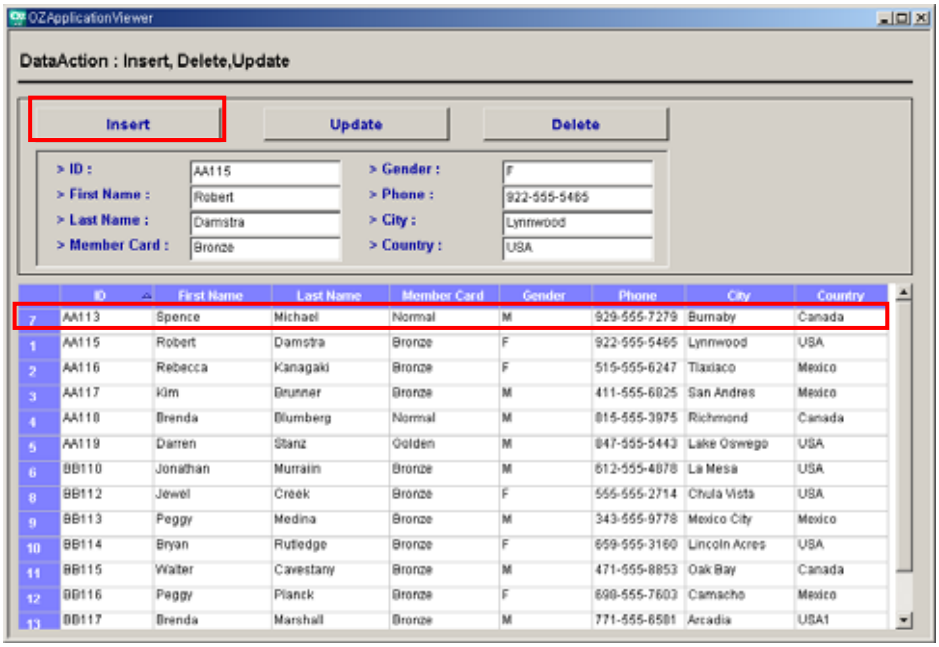
Step 5 미리보기

[File] 메뉴의 [Preview]를 클릭하거나 툴바의 미리보기 아이콘()을 클릭하여 미리보기합니다.

새로운 고객 정보를 추가하거나 삭제 또는 변경합니다.

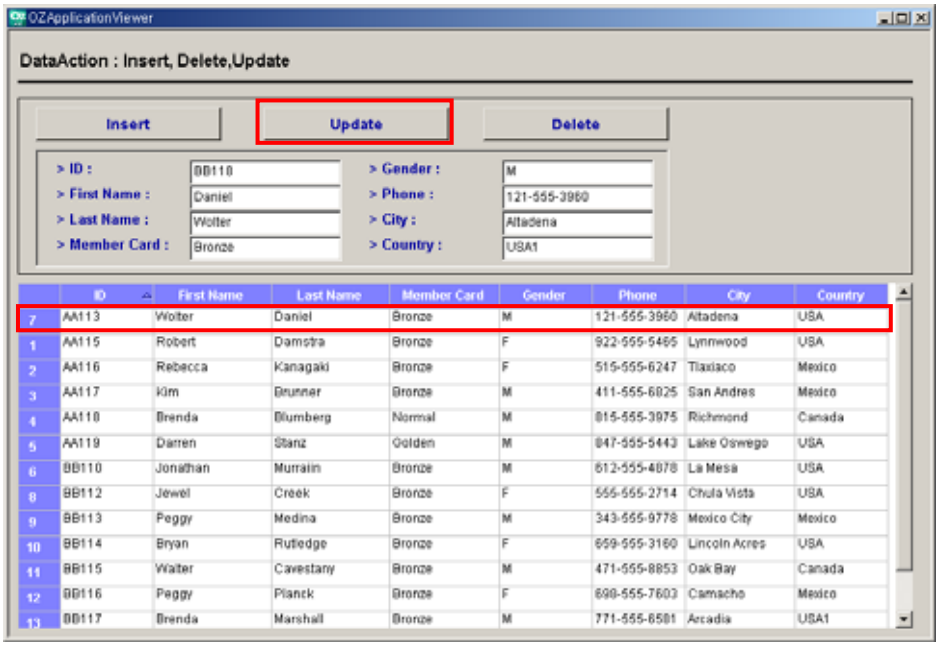


➢ 삽입



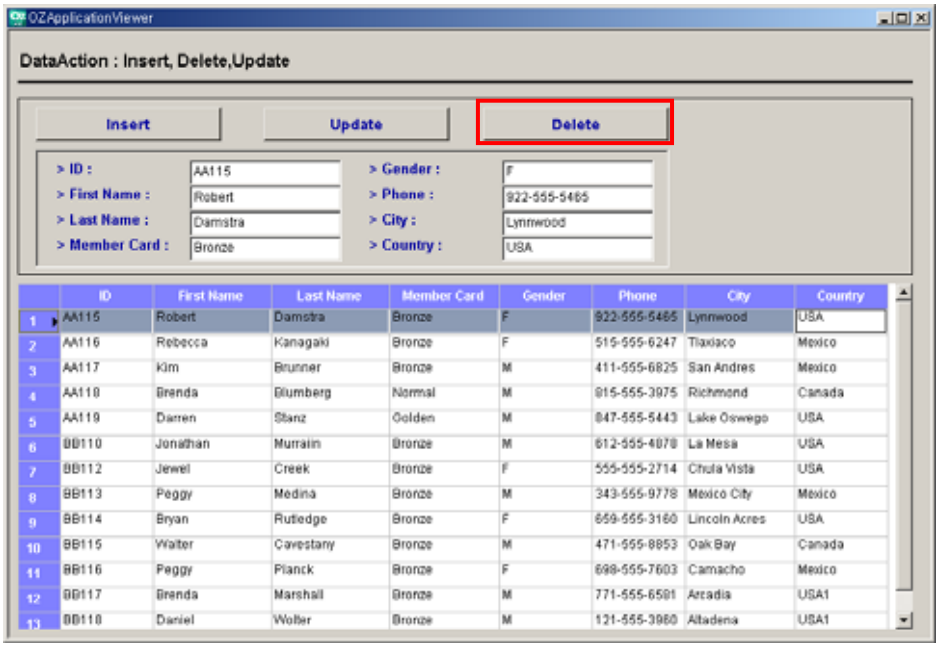
새로운 고객 정보(ID:AA113)가 추가된 것을 확인할 수 있습니다.

➢ 변경



고객 정보가 변경(ID:AA113)된 것을 확인할 수 있습니다.

➢ 삭제



고객 정보가 삭제(ID:AA113)된 것을 확인할 수 있습니다.

수식 필드 사용

본 장은 수식 필드를 만들고 사용하는 방법을 예제를 통해 간략히 설명합니다. 수식 필드는 DataBase의 테이블에 없는 필드로 수식이나 함수를 코딩하여 임의로 만든 필드입니다.

수식 필드 추가

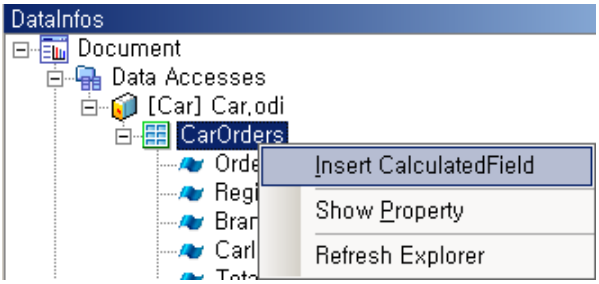
수식 필드는 사용할 스크립트 타입에 따라 자바 스크립트 타입 또는 오즈 스크립트 타입으로 추가할 수 있습니다.
본 장에서는 두 개의 데이터 필드 값을 연산하여 수식 필드에 값에 설정하는 방법을 예를 들어 살펴보겠습니다.

※ 주의사항

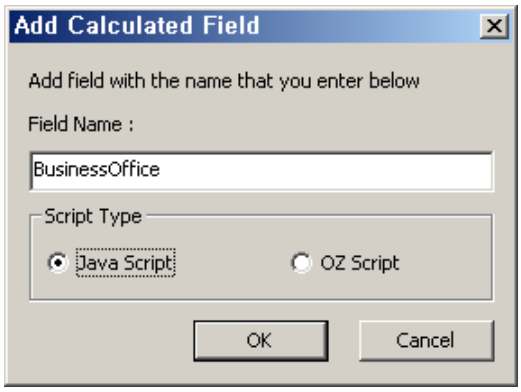
- 수식 필드 생성 시 수식 필드에 사용할 스크립트 타입을 먼저 선택하여야 하며, 스크립트 타입은 수식 필드를 추가한 후에는 변경할 수 없습니다.
- 오즈 쿼리 디자이너에서 만든 수식 필드는 ODI 파일에 저장되고, 오즈 애플리케이션 디자이너에서 만든 수식 필드는 해당 프로젝트 파일에만 저장됩니다. 즉, 오즈 쿼리 디자이너에서 만든 수식 필드는 해당 ODI를 추가한 모든 보고서에서도 사용할 수 있는 반면, 오즈 애플리케이션 디자이너에서 만든 수식 필드는 해당 프로젝트에서만 사용 가능합니다.

■ 자바 스크립트 타입

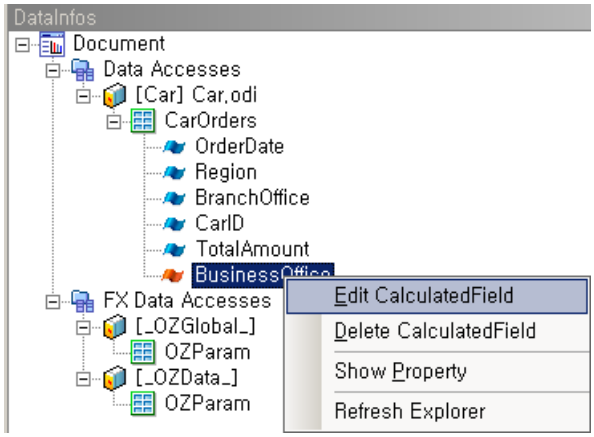
➢ 데이터 정보 창에서 데이터셋을 선택한 후 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Insert CalculatedField] 메뉴를 클릭합니다.



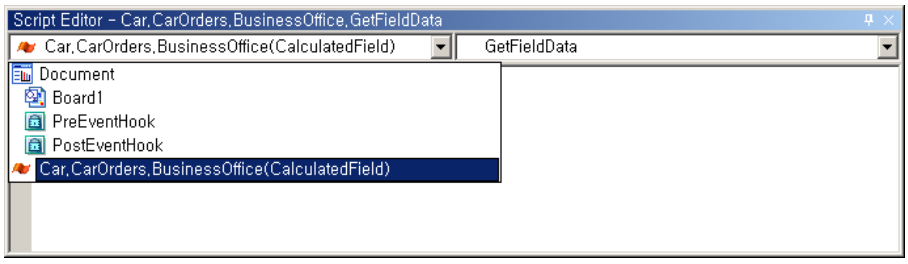
➢ [Insert CalculatedField] 메뉴를 클릭하여 나타난 'Add Calculated Field 다이얼로그'에서 필드 이름을 입력하고 "Script Type"을 "Java Script"로 선택한 후 [OK] 버튼을 클릭하여 수식 필드를 추가합니다.



➢ 추가된 수식 필드에 데이터 값을 연산하는 자바 스크립트를 입력합니다. 추가된 수식 필드를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Edit CalculatedField] 메뉴를 클릭하거나



스크립트 편집기에서 컴포넌트 콤보박스를 펼친 후 수식 필드를 클릭합니다.

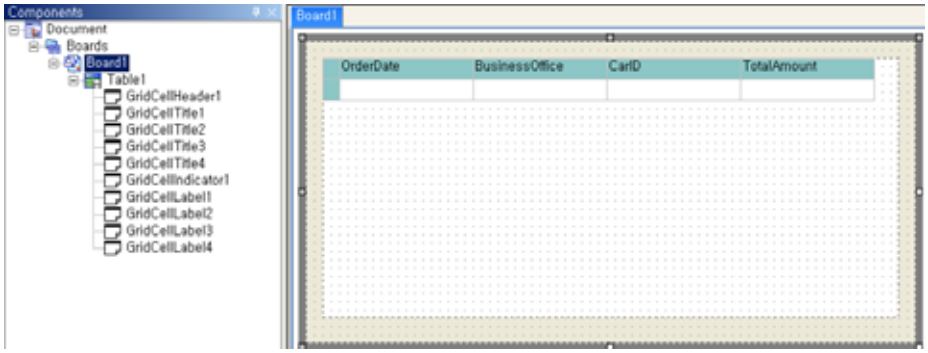


➢ 데이터 필드 중 Region과 BranchOffice를 더하여 수식 필드의 값으로 설정하도록 아래와 같이 스크립트를 입력합니다.

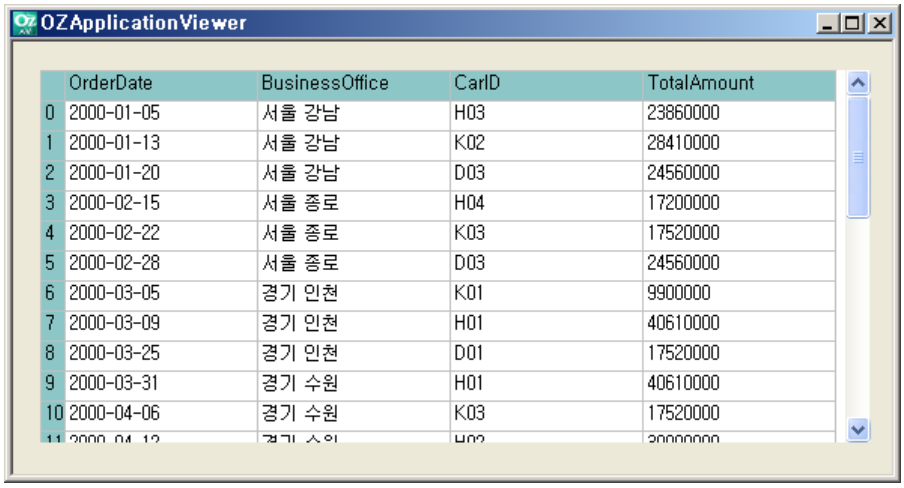
```
var BusinessOffice;
BusinessOffice = This.GetDataSetValue("Region") + " " +
This.GetDataSetValue("BranchOffice");
```

```
This.SetFieldData(BusinessOffice);
```

➢ 보드에 테이블을 아래 그림과 같이 추가한 후



➢ 프로젝트를 미리보기하여 수식 필드의 값을 확인합니다.

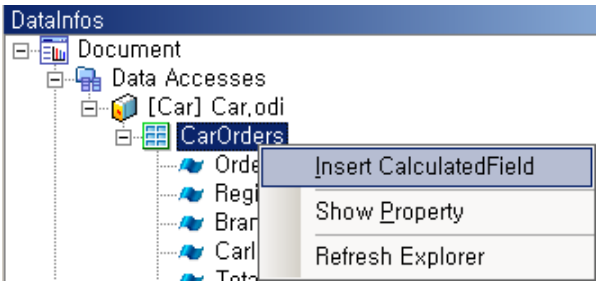


	OrderDate	BusinessOffice	CarID	TotalAmount
0	2000-01-05	서울 강남	H03	23860000
1	2000-01-13	서울 강남	K02	28410000
2	2000-01-20	서울 강남	D03	24560000
3	2000-02-15	서울 종로	H04	17200000
4	2000-02-22	서울 종로	K03	17520000
5	2000-02-28	서울 종로	D03	24560000
6	2000-03-05	경기 인천	K01	9900000
7	2000-03-09	경기 인천	H01	40610000
8	2000-03-25	경기 인천	D01	17520000
9	2000-03-31	경기 수원	H01	40610000
10	2000-04-06	경기 수원	K03	17520000

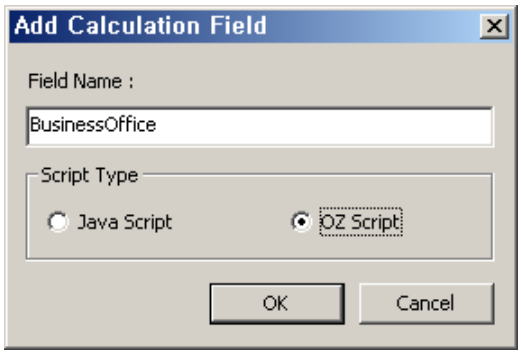
■ 오즈 스크립트 타입

수식 필드에 작성할 오즈 스크립트의 문법 및 함수는 'OZ Script Guide'를 참조하시기 바랍니다.

➢ 데이터 정보 창에서 데이터셋을 선택한 후 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Insert CalculatedField] 메뉴를 클릭합니다.

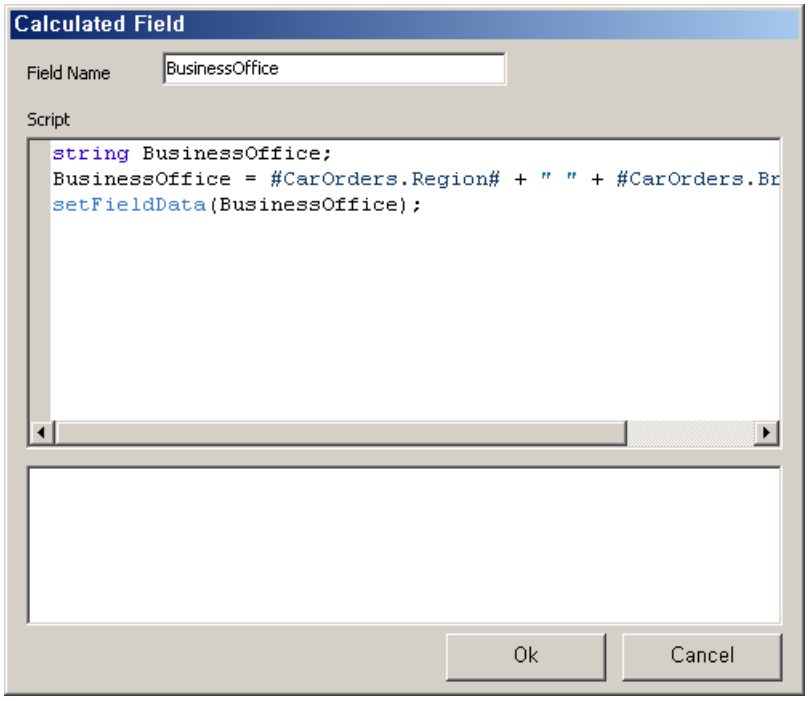


➢ [Insert CalculatedField] 메뉴를 클릭하여 나타난 'Add Calculated Field 다이얼로그'에서 필드 이름을 입력하고 "Script Type"을 "OZ Script"로 선택한 후 [OK] 버튼을 클릭하면



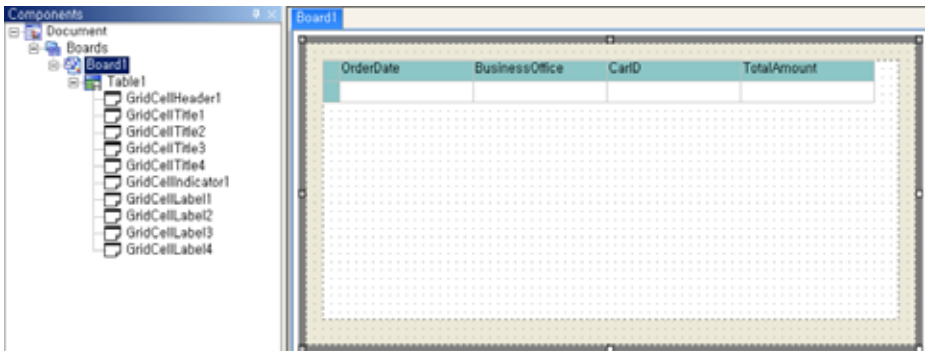
'Calculation Field 다이얼로그'가 나타납니다.

Script 입력 창에 데이터 필드 중 Region과 BranchOffice를 더하여 수식 필드의 값으로 설정하도록 아래와 같이 스크립트를 입력합니다.

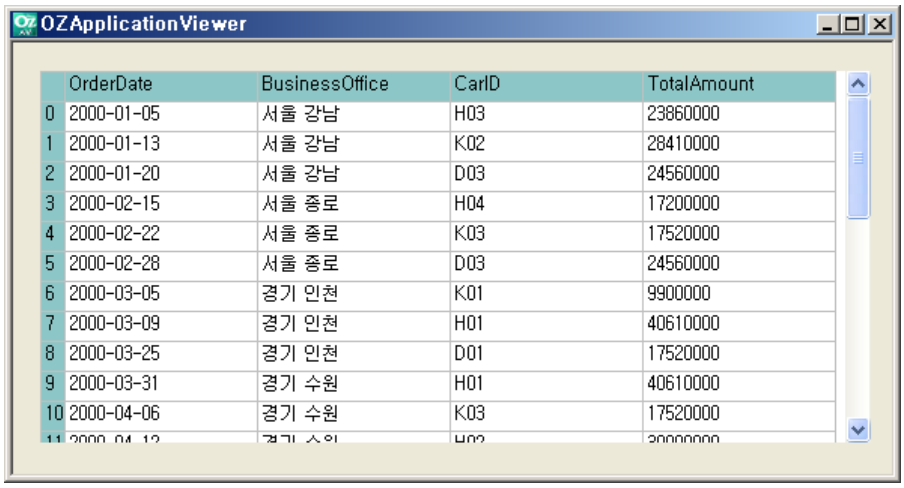



```
string BusinessOffice;
BusinessOffice    =    #CarOrders.Region#    +    "    "    +
#CarOrders.BranchOffice#;
setFieldData(BusinessOffice);
```

➢ 보드에 테이블을 아래 그림과 같이 추가한 후



➢ 프로젝트를 미리보기하여 수식 필드의 값을 확인합니다.



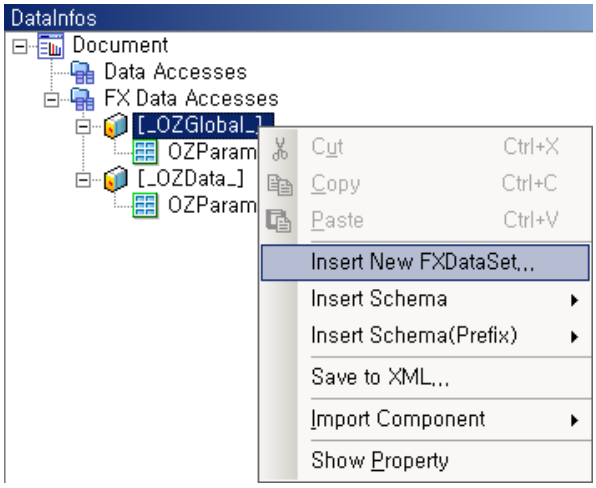
	OrderDate	BusinessOffice	CarID	TotalAmount
0	2000-01-05	서울 강남	H03	23860000
1	2000-01-13	서울 강남	K02	28410000
2	2000-01-20	서울 강남	D03	24560000
3	2000-02-15	서울 종로	H04	17200000
4	2000-02-22	서울 종로	K03	17520000
5	2000-02-28	서울 종로	D03	24560000
6	2000-03-05	경기 인천	K01	9900000
7	2000-03-09	경기 인천	H01	40610000
8	2000-03-25	경기 인천	D01	17520000
9	2000-03-31	경기 수원	H01	40610000
10	2000-04-06	경기 수원	K03	17520000

FXData 모듈 사용

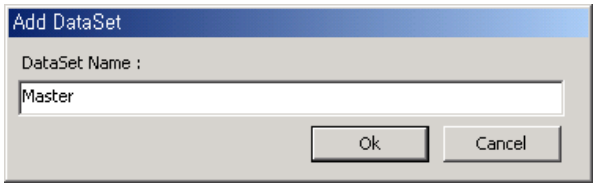
본 장은 FXData 모듈을 사용하여 마스터-디테일 관계의 데이터셋과 데이터 필드를 추가하고 Table에 표시하는 방법을 예제를 통해 간략히 설명합니다.

Step 1 마스터셋 추가하기

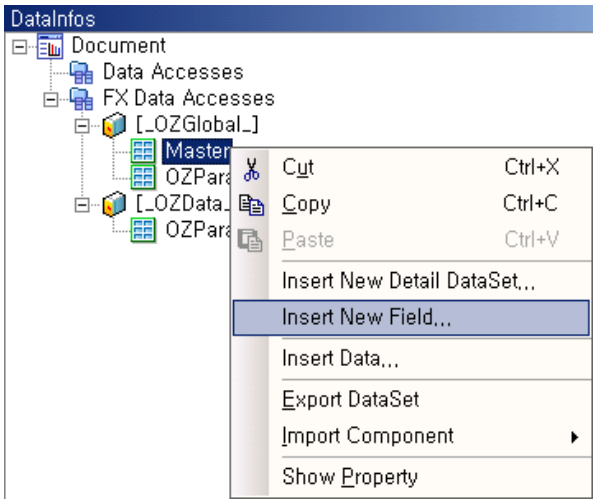
➢ FX Data Accesses의 "_OZGlobal_"을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Insert New FXDataSet] 메뉴를 클릭합니다.



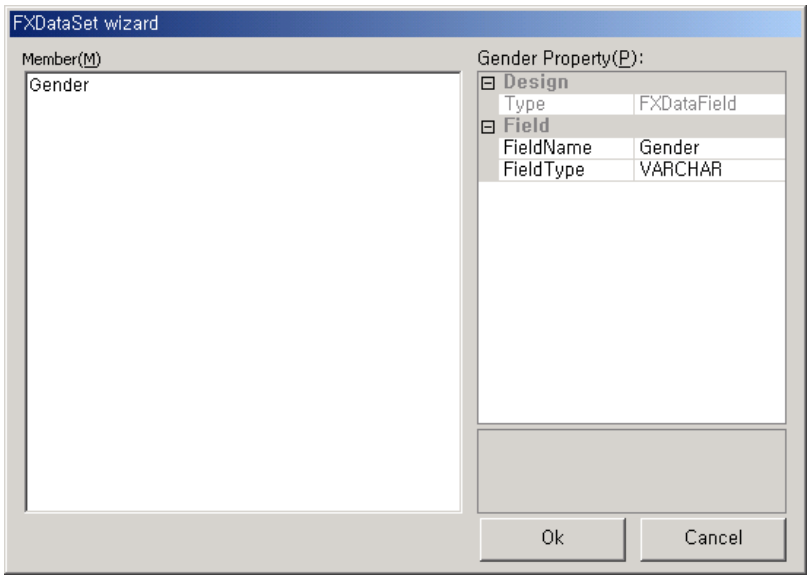
➢ 데이터셋 추가 다이얼로그에서 데이터셋 이름을 입력한 후 [Ok] 버튼을 클릭하여 데이터셋을 추가합니다.

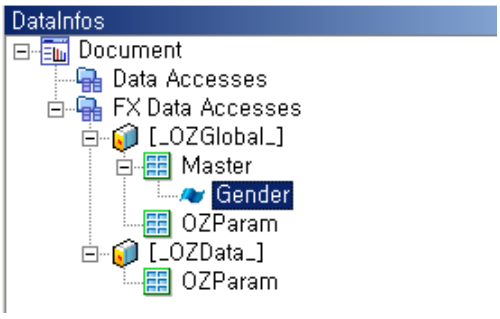


➢ 추가된 데이터셋을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Insert New Field] 메뉴를 클릭합니다.



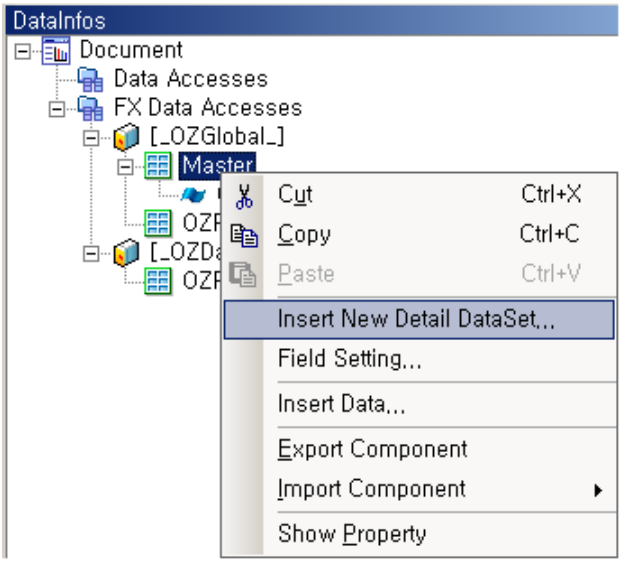
➢ 필드 추가 다이얼로그에서 마스터셋에 추가할 필드명을 입력하고 필드 타입을 설정한 후 [Ok] 버튼을 클릭하여 필드를 추가합니다.



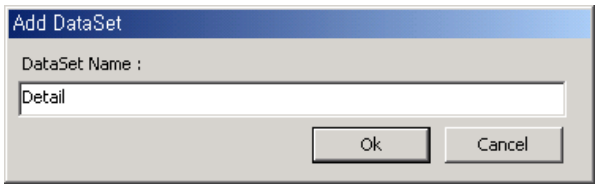


Step 2 디테일셋 추가하기

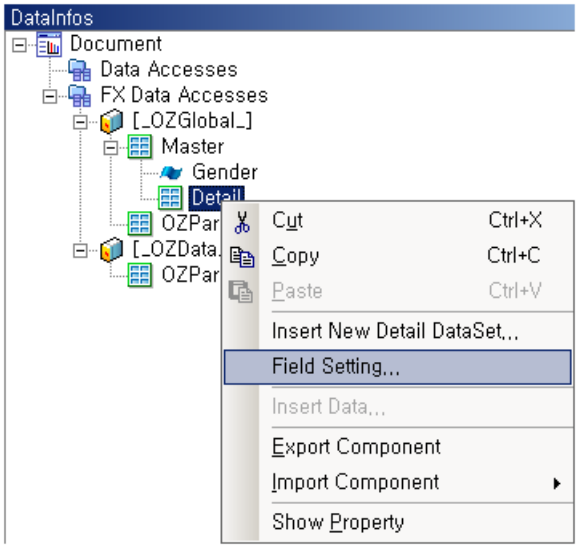
➢ 추가된 마스터셋을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Insert New Detail DataSet] 메뉴를 클릭합니다.



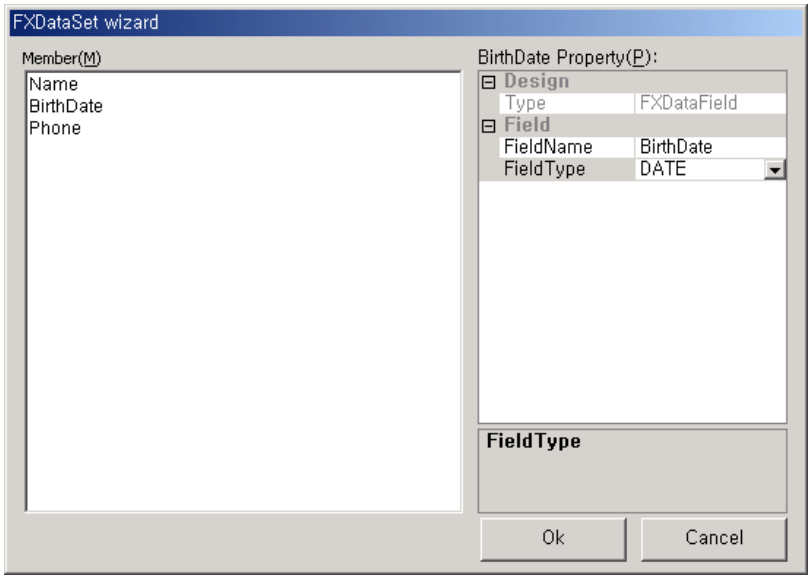
➢ 데이터셋 추가 다이얼로그에서 데이터셋 이름을 입력한 후 [Ok] 버튼을 클릭하여 데이터셋을 추가합니다.

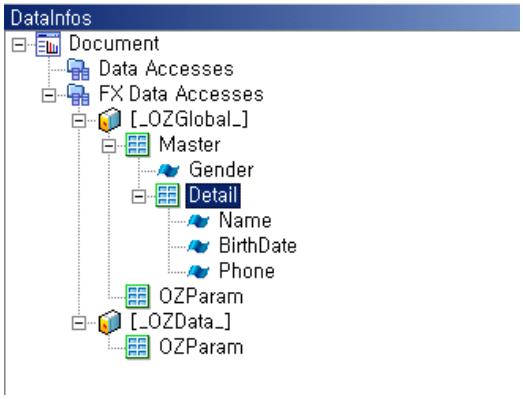


➢ 추가된 데이터셋을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Insert New Field] 메뉴를 클릭합니다.



➢ 필드 추가 다이얼로그에서 마스터셋에 추가할 필드명을 입력하고 필드 타입을 설정한 후 [Ok] 버튼을 클릭하여 필드를 추가합니다.

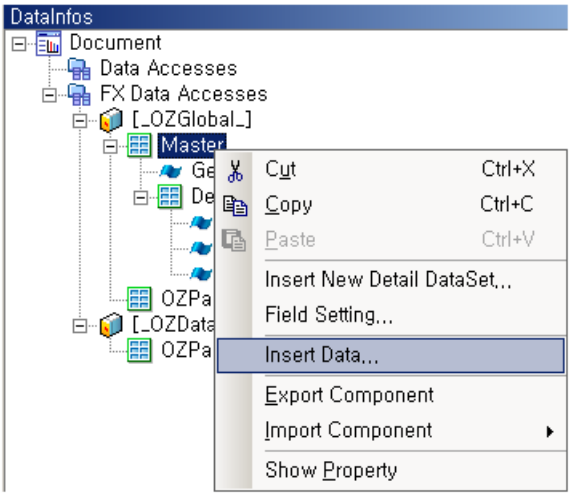




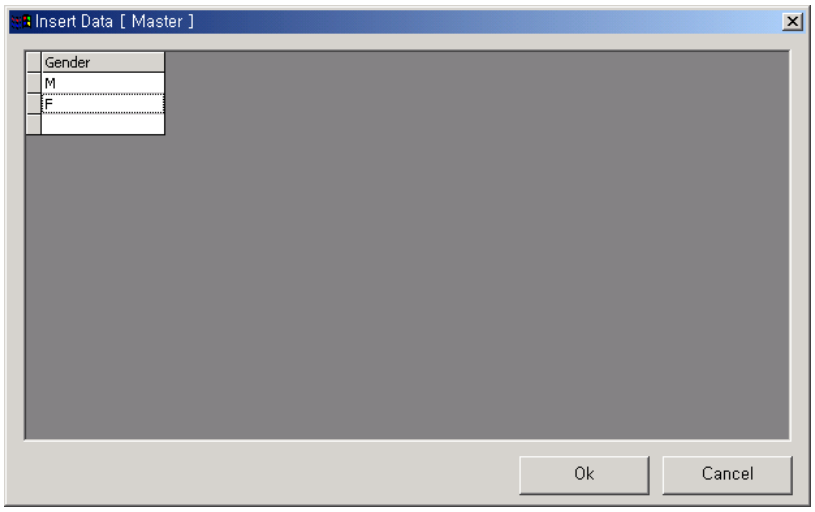
Step 3 마스터셋, 디테일셋의 데이터 값 입력하기

- 먼저 마스터셋 필드의 데이터 값을 입력한 후 각각의 마스터셋 필드의 데이터 값에 해당하는 디테일셋 필드의 데이터값을 입력하여야 합니다.

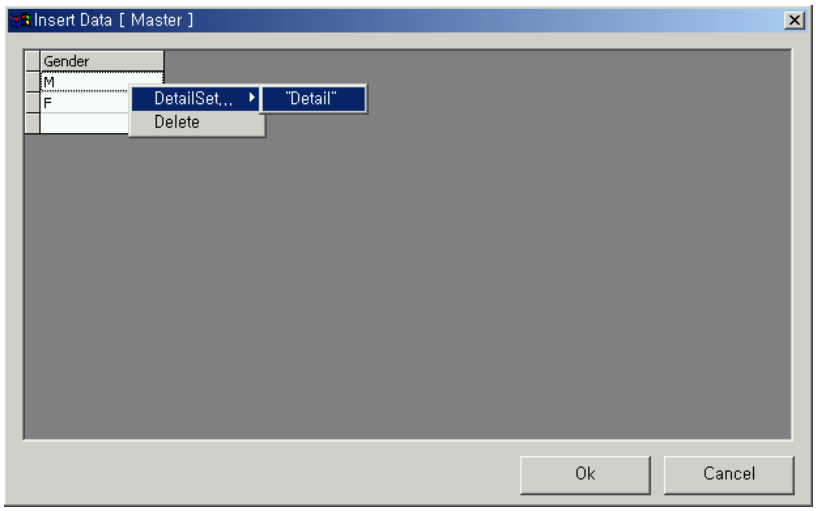
- 먼저 마스터셋의 필드값을 입력합니다. 마스터셋을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Insert Data] 메뉴를 클릭합니다.



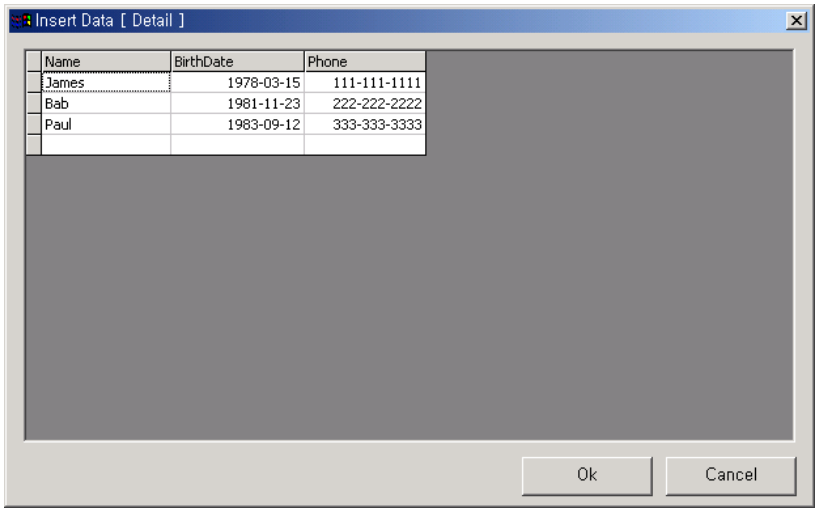
- 데이터 추가 다이얼로그에서 마스터셋에 추가된 필드의 데이터 값을 입력합니다.



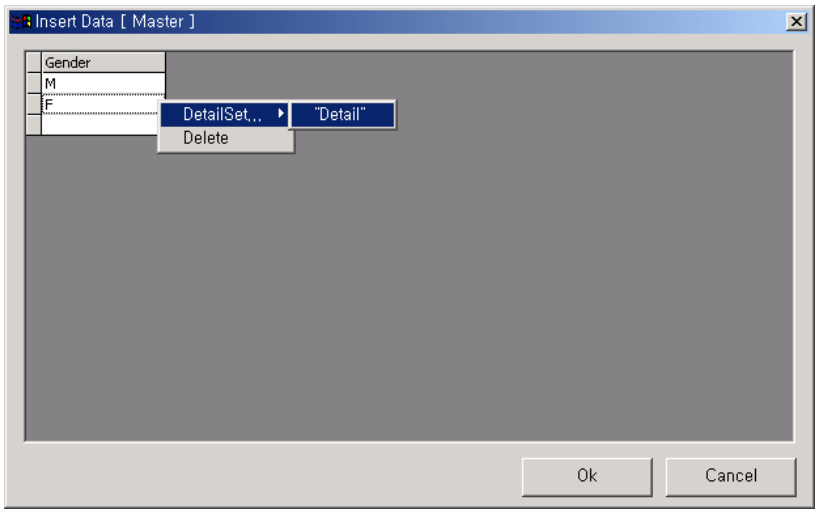
- 마스터셋 필드의 값을 입력한 후에는 각각의 마스터셋 필드의 값에 해당하는 디테일 필드값을 입력합니다. 위 그림과 같은 상태에서는 마스터셋의 필드 값 중 "M"을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [DetailSet] → ["Detail"] 메뉴를 클릭하여 디테일셋 필드 값을 입력하여야 합니다.

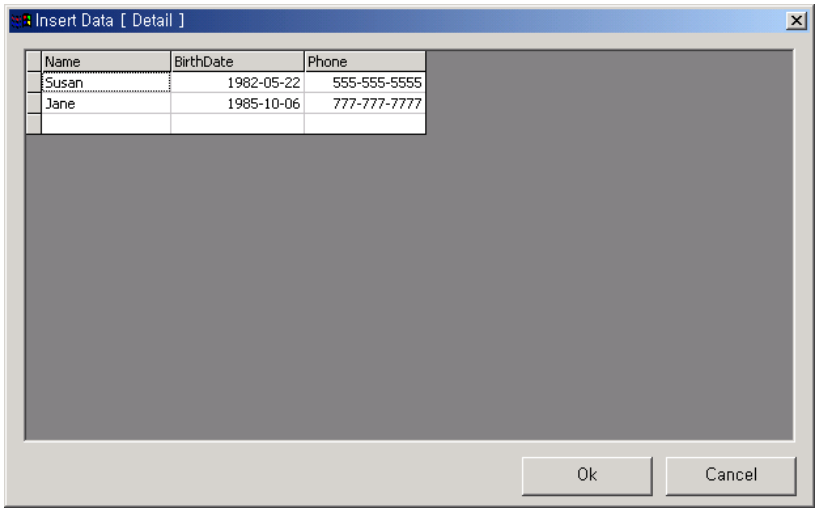


- 데이터 추가 다이얼로그에서 디테일셋에 추가된 필드의 데이터 값을 입력한 후 [Ok] 버튼을 클릭합니다.



➢ 마스터셋 필드 값을 나타내는 다이얼로그가 다시 나타나면 두 번째 마스터셋 필드 값에 해당하는 디테일셋의 필드 값을 입력합니다.





➢ 마스터셋의 필드 값과 디테일셋의 필드 값을 모두 입력한 후 [Ok] 버튼을 클릭합니다.

Step 4 FXData 표시하기

ComboBox에 마스터셋 데이터를 표시하고 Table에 마스터셋에 해당하는 디테일셋 데이터를 표시하는 방법을 살펴보겠습니다.

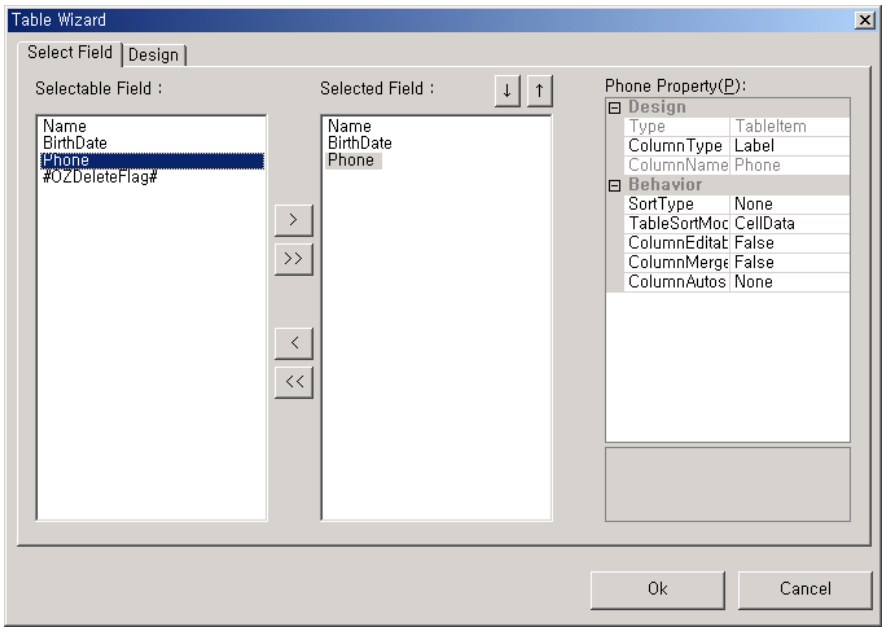
➢ Board에 ComboBox를 추가한 후 추가한 ComboBox의 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
ODIKey	_OZGlobal_	**DataSet**	Master
Field	Gender	**FireRowCursorChange**	True

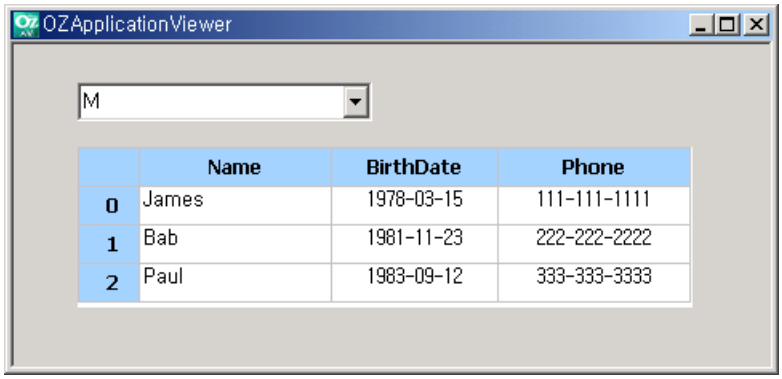
➢ Board에 Table을 추가한 후 추가한 Table의 속성 값을 아래와 같이 변경합니다.

Property	Value	Property	Value
ODIKey	_OZGlobal_	**DataSet**	Detail

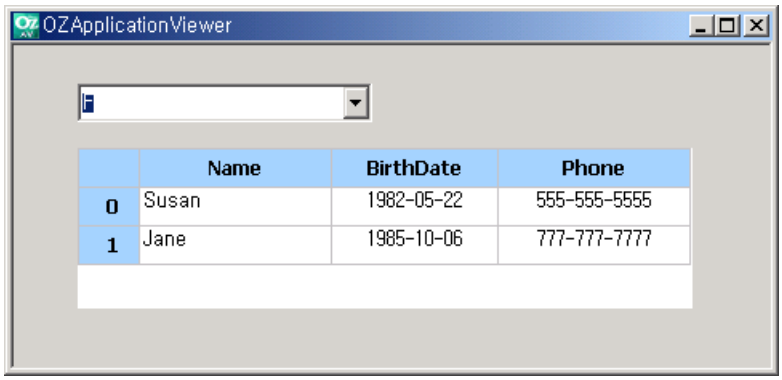
➢ Table을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Wizard] 메뉴를 클릭하여 테이블 마법사를 실행한 후 표시할 필드를 추가합니다.



➢ 필드를 추가한 후 [Ok] 버튼을 클릭합니다. 테이블의 타이틀, 폰트 등을 디자인한 후 미리보기합니다.



마스터셋의 필드 값을 바꿔서 데이터가 제대로 표시되는지 확인합니다.



Ⅶ. 마법사 사용법

- Chart 마법사
- CheckListBox 마법사
- ComboBox 마법사
- ContextMenu/MainMenu/TrayContextMenu 마법사
- CrossTab 마법사
- ListBox 마법사
- ListView 마법사
- RichTextBox 마법사
- TabControl 마법사
- Table 마법사
- TreeView 마법사

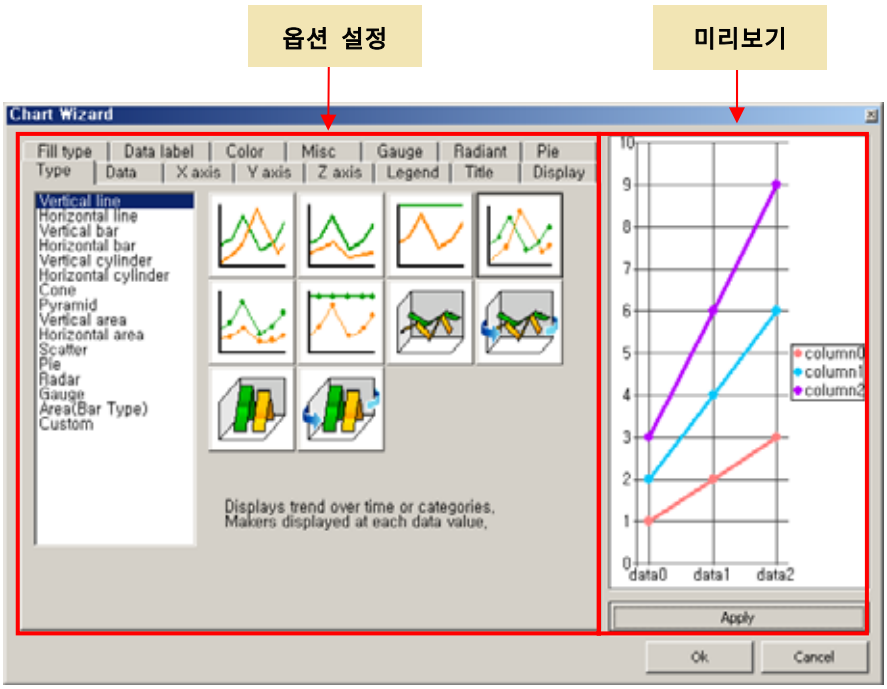
본 장에서는 OZ Application Designer에서 제공하는 마법사를 사용하는 방법에 대해 설명합니다.

Chart 마법사

Chart에 데이터를 표시하기 위해 Chart 마법사를 이용합니다. Chart 마법사를 실행하기 위해서는 먼저 Board에 Chart를 추가하고, Chart에 표시할 ODI 파일과 데이터 셋을 설정하여야 합니다. Chart 마법사에서 설정한 내용은 Chart의 속성창에서도 확인할 수 있습니다.

새로운 프로젝트에 Ⅵ.장에서 만든 'Foodmart.odi' 파일과 Chart를 추가하고, Chart의 속성 중 'ODIKey'는 'Foodmart'로 'DataSet'은 'ProductSales'로 설정합니다. Chart를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Wizard]를 클릭하여 Chart 마법사를 실행합니다.

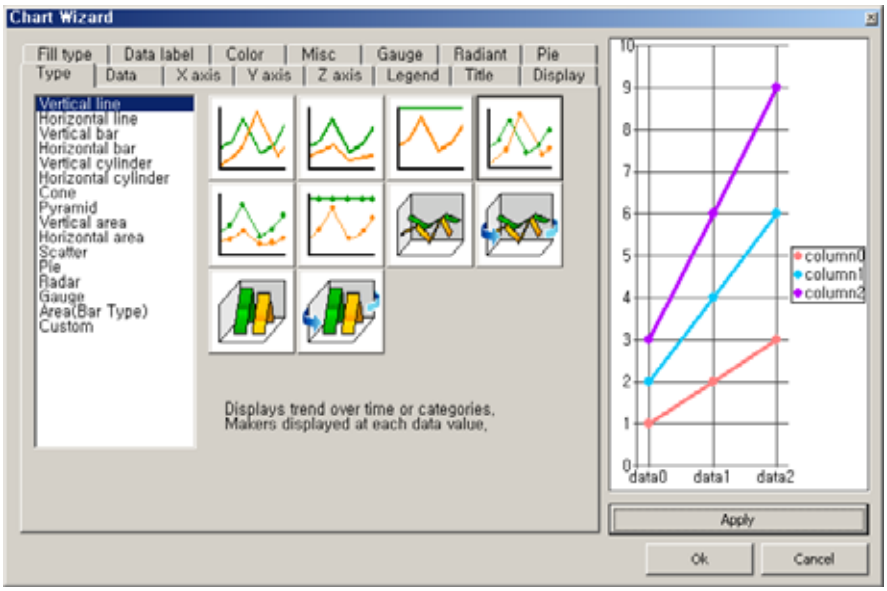
차트 마법사의 왼쪽 영역에서는 차트의 옵션을 설정하고 오른쪽 영역에서는 설정한 옵션이 적용된 모습을 미리보기 할 수 있습니다.



※ 옵션 설정 후 [Apply] 버튼을 클릭하면 미리보기창에 적용됩니다.

Type

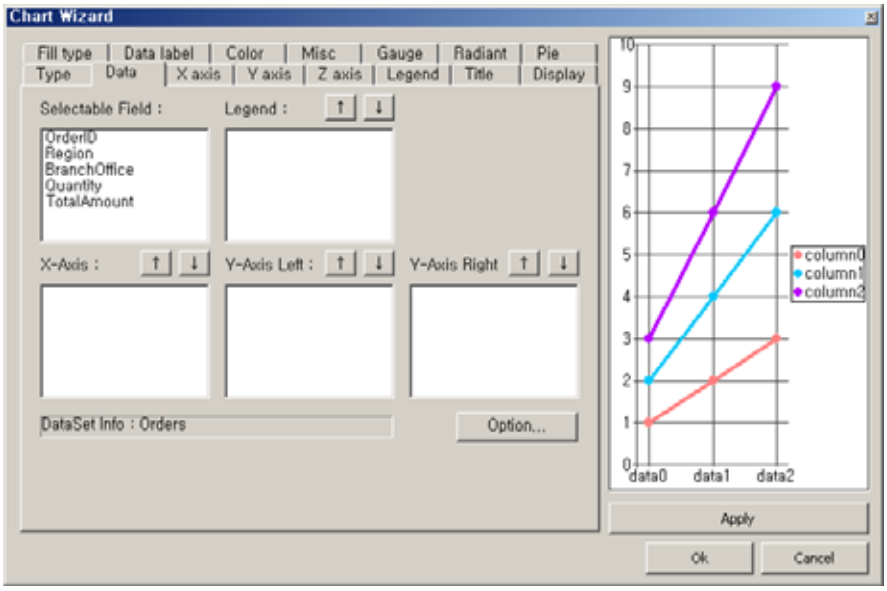
Chart의 타입을 설정합니다.



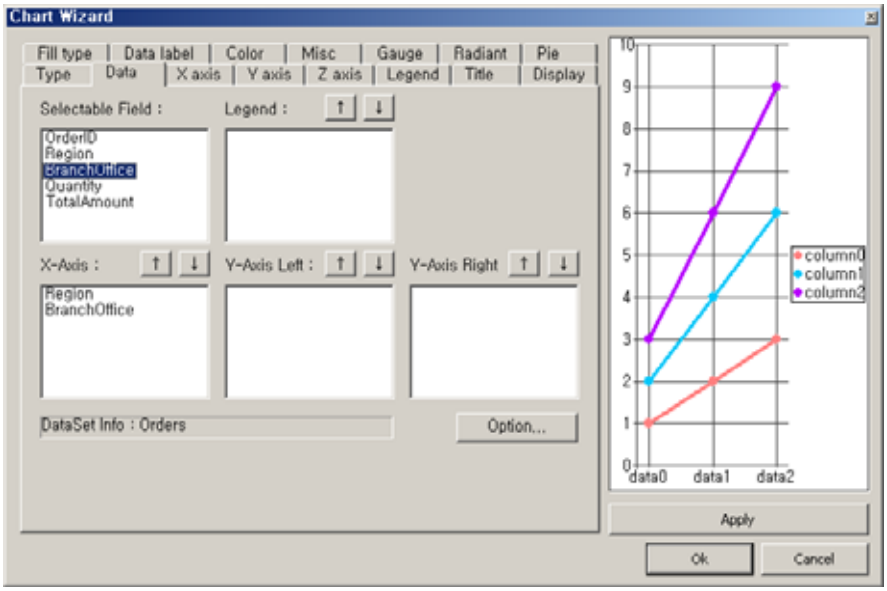
OZ Application Designer에서는 꺾은선형, 막대형, 원통형, 원뿔형, 피라미드형, 영역형, 점형, 원형, 방산형, 게이지형, 계단형, 사용자정의형 등 여러 가지 형태의 Chart 타입을 지원하며, 원하는 타입의 Chart 아이콘을 클릭하여 Chart의 타입을 설정합니다.

Data

Chart에 표시할 X축, Y축 데이터와 계열를 설정합니다.



'Selectable Field'에는 Chart에 바인딩되어 있는 DataSet의 필드가 리스트 형태로 표시됩니다. X축 항목, 왼쪽 Y축, 오른쪽 Y축, 계열로 설정할 데이터를 '선택 가능한 필드창'에서 마우스로 선택한 후 해당 영역에 Drag & Drop하여 설정합니다.



설정한 데이터 필드의 순서를 변경하려면 순서를 변경할 데이터 필드를 선택한 후 업/다운 버튼 (↑ ↓)으로 순서를 변경합니다.

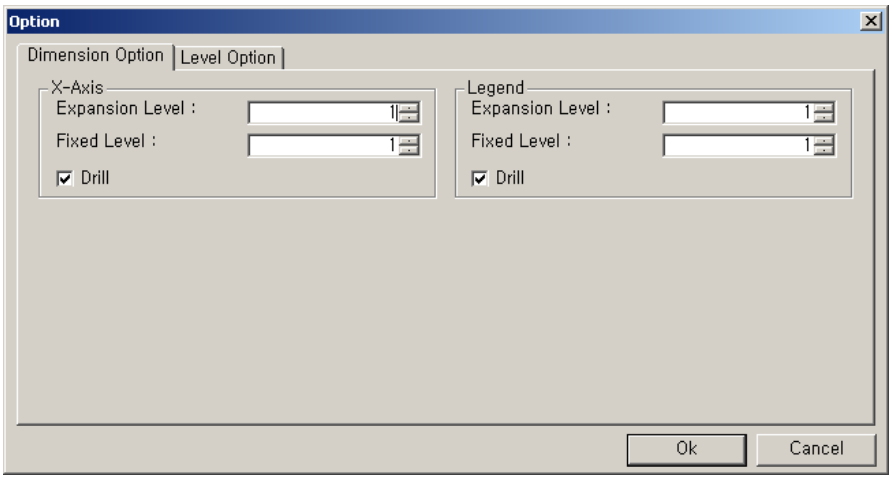
설정한 데이터 필드를 해제하려면 해제할 데이터 필드를 선택한 후 '선택 가능한 필드창'으로 다시 Drag & Drop하거나 키보드의 [Delete] 버튼을 클릭하여 해제합니다.

■ Option

[Option] 버튼을 클릭하면 축 옵션과 레벨 옵션을 설정할 수 있는 다이얼로그가 나타납니다.

- Dimension Option

[Dimension Option] 탭에서는 X축과 계열의 옵션을 설정합니다.

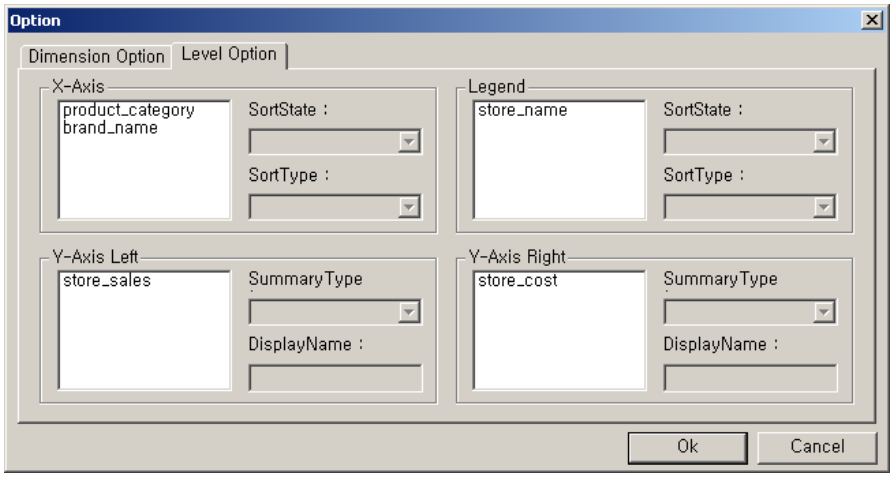


확장 레벨과 고정 레벨을 설정하고, 실행시에 마우스 더블 클릭으로 드릴 업/다운하게 할지 여부를 설정합니다.

※ 참고사항 : 확장 레벨과 고정 레벨은 Chart의 속성 중 "ChartExpansionType" 속성 값이 "Tree Type"인 경우에만 활성화됩니다.

- Level Option

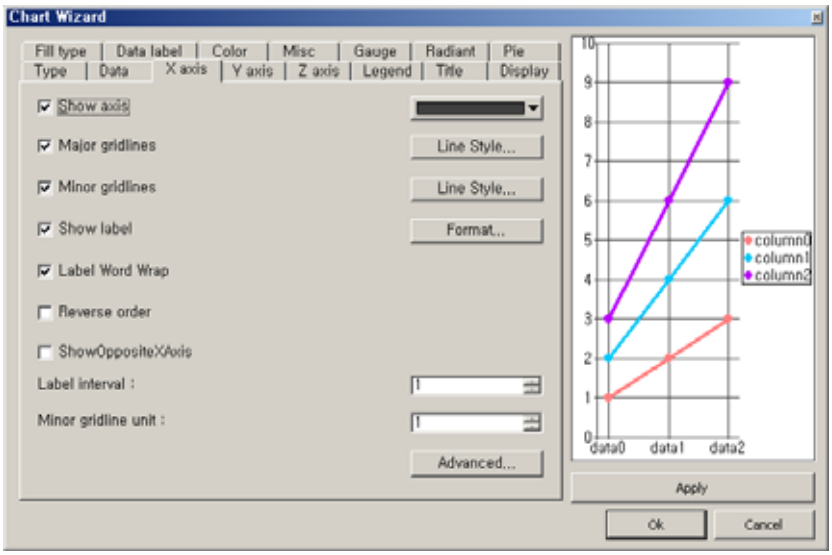
[Level Option] 탭에서는 X축과 계열의 정렬 상태와 정렬 타입을 설정하고 Y축의 써머리 타입과 표시 명을 설정합니다.



써머리 타입은 Default, Sum, Max, Min, Avg, Cnt 중의 하나로 선택할 수 있으며, Default는 Chart의 속성 중 "DefaultSummaryType"에 설정된 값을 적용하겠다는 뜻입니다.

X Axis

X축의 기본선, 주 눈금선, 보조 눈금선, 라벨 표시 등을 설정합니다.



- Show Axis

  X축 기본선의 표시 여부를 설정하고, 색 선택 콤보 박스를 이용하여 기본선의 색을 변경합니다.

- Major Gridlines

  X축의 주 눈금선의 표시 여부를 설정하고, [Line Style]버튼을 이용하여 눈금선의 스타일 즉 선 두께, 선 색 등을 설정합니다. [Line Style] 버튼 사용법은 아래 'Gridline Style Dialog' 부분을 참조하십시오.

- Minor Gridlines

  X축의 보조 눈금선의 표시 여부를 설정하고, [Line Style] 버튼을 이용하여 눈금선의 스타일 즉 선 두께, 선 색 등을 설정합니다. [Line Style] 버튼 사용법은 아래 'Gridline Style Dialog' 부분을 참조하십시오.

- Show Label

  X축의 라벨 즉, X축에 설정한 데이터 필드값의 표시 여부를 설정하고, [Format] 버튼을 이용하여 라벨의 포맷을 설정합니다. [Format] 버튼 사용법은 아래 'Format Dialog' 부분을 참조하십시오.

- Label Word Wrap

  X축의 라벨의 자동 줄바꾸기 여부를 설정합니다.

- Reverse Order

  X축의 값을 역순으로 표시할지 여부를 설정합니다. 옵션을 체크하지 않으면 데이터셋에서 가져온 순서대로 표시되고 체크하면 역순으로 표시됩니다.

- Show Oppposite XAxis

  X축의 반대편에도 축을 표시할지 여부를 설정합니다.

- Label Interval

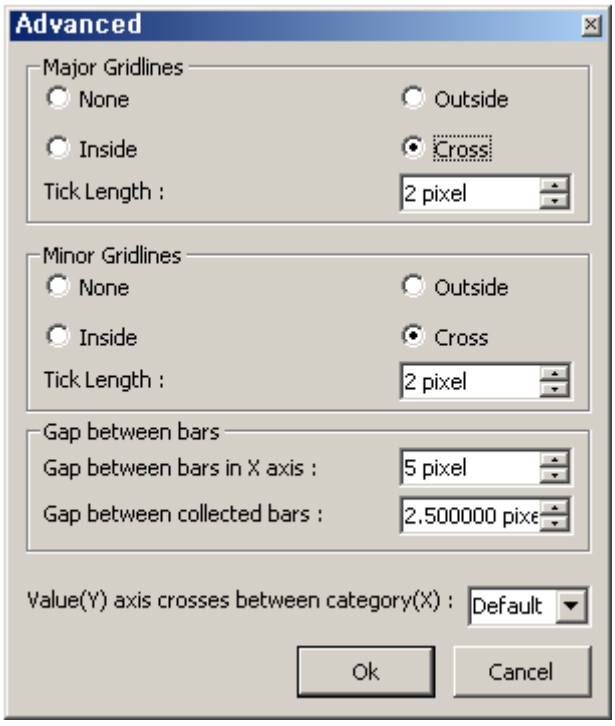
  X축 라벨의 표시 주기를 설정합니다. 표시 주기를 '1'로 설정한 경우에는 모든 라벨이 표시되며, '2'로 설정하면 첫번째, 세번째, 다섯번째, ... 주기로 표시됩니다.

- Minor Gridline Unit

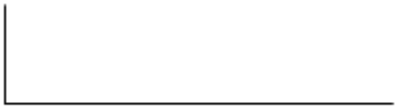
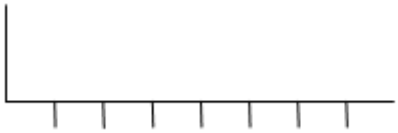
  X축 보조 눈금선의 간격을 설정합니다..

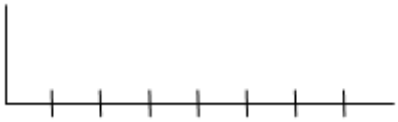
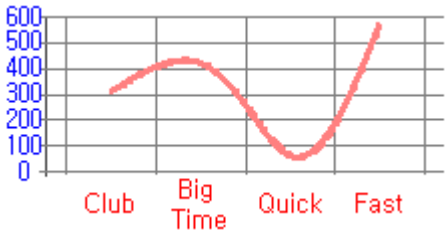
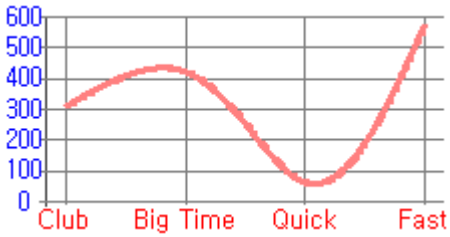
- Advanced

  X축의 기본선에 대한 주 눈금선과 보조 눈금선의 표시 위치, 눈금 길이, 막대간 간격 등을 설정합니다.



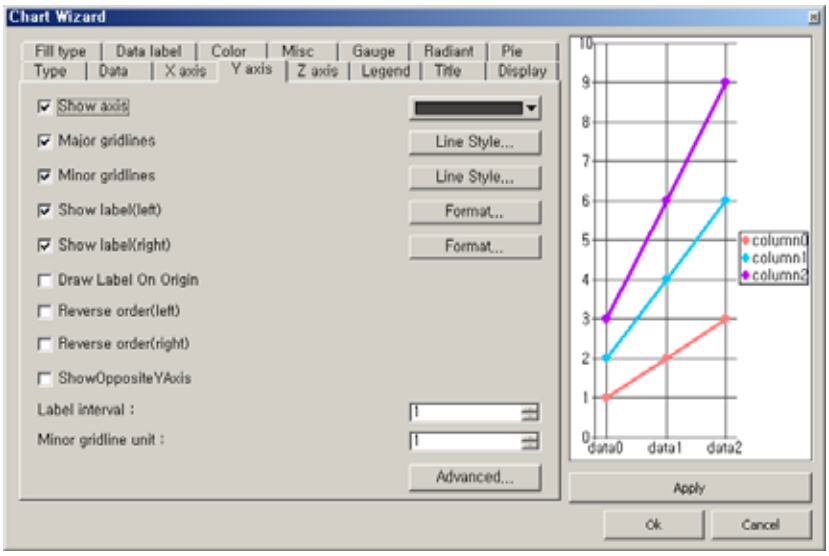
각 옵션과 기능은 아래와 같습니다.

Option		Description
Major / Minor Gridlines	**None**	기본선에 눈금선을 표시하지 않습니다. 
	Outside	기본선의 바깥쪽에 눈금선을 표시합니다. 
	Inside	기본선의 안쪽에 눈금선을 표시합니다. 

<table>
<tr><td rowspan="2"></td><td>Cross</td><td>기본선에 교차하여 눈금선을 표시합니다.
</td></tr>
<tr><td>Tick Length</td><td>눈금의 길이를 설정합니다.
※ 참고사항 : "눈금 길이"의 기본 값은 2픽셀이며, 값의 범위는 1 이상인 정수 값만 입력이 가능합니다.</td></tr>
<tr><td rowspan="2">Gap Between Bars</td><td>Gap Between Bars In X Axis</td><td>X축 항목 간 막대 간격을 설정합니다.</td></tr>
<tr><td>Gap Between Collected Bars</td><td>계열 간 막대 간격을 설정합니다.</td></tr>
<tr><td rowspan="3">Value(Y) axis crosses between category(X)</td><td>Default</td><td>설정된 차트 타입의 기본 값에 따라 정렬됩니다
(차트 타입이 막대형이나 계단형일 경우에는 기본 값이 "Ok"이고 그 밖의 차트 타입의 기본 값은 "Canel"입니다.)</td></tr>
<tr><td>Ok</td><td>Y축 값이 차트 타입에 관계 없이 X축 항목 사이에 표시됩니다.

</td></tr>
<tr><td>Cancel</td><td>Y축 값이 차트 타입에 관계 없이 X축선에 맞추어 표시됩니다.

</td></tr>
</table>

Y Axis

왼쪽 Y축, 오른쪽 Y축의 기본선, 주 눈금선, 보조 눈금선, 라벨 표시, 최대값, 최소값 등을 설정합니다.



- Show Axis

  Y축 기본선의 표시 여부를 설정하고, 색 선택 콤보 박스를 이용하여 기본선의 색을 변경합니다.

- Major Gridlines

  Y축의 주 눈금선의 표시 여부를 설정하고, [Line Style] 버튼을 이용하여 눈금선의 스타일 즉 선 두께, 선 색 등을 설정합니다. [Line Style] 버튼 사용법은 아래 'Gridline Style Dialog' 부분을 참조하십시오.

- Minor Gridlines

  Y축의 보조 눈금선의 표시 여부를 설정하고, [Line Style] 버튼을 이용하여 눈금선의 스타일 즉 선 두께, 선 색 등을 설정합니다. [Line Style] 버튼 사용법은 아래 'Gridline Style Dialog' 부분을 참조하십시오.

- Show Label(Left)

  왼쪽 Y축의 라벨 즉, 왼쪽 Y축에 설정한 데이터 필드값의 표시 여부를 설정하고, [Format] 버튼을 이용하여 라벨의 포맷을 설정합니다. [Format] 버튼 사용법은 아래 'Format Dialog' 부분을 참조하십시오.

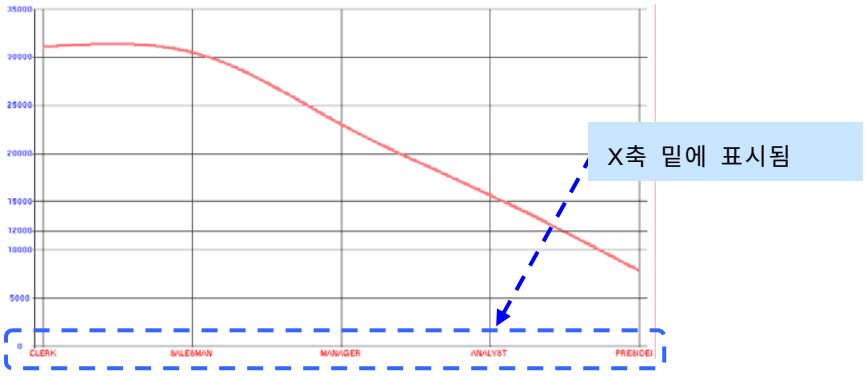
■ Show Label(Right)

오른쪽 Y축의 라벨 즉, 오른쪽 Y축에 설정한 데이터 필드값의 표시 여부를 설정하고, [Format] 버튼을 이용하여 라벨의 포맷을 설정합니다. [Format] 버튼 사용법은 아래 'Format Dialog' 부분을 참조하십시오.

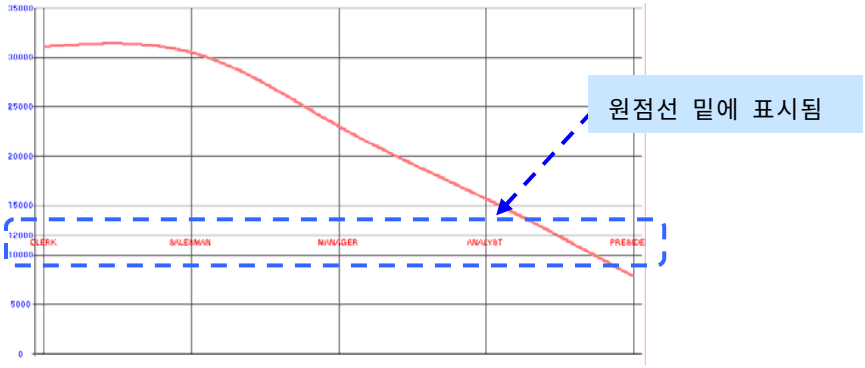
■ Draw Label On Origin

원점선에 맞추어 X축 라벨을 표시할지 여부를 설정합니다. 옵션을 체크하면 X축 라벨을 X축 밑에 표시하는 것이 아니라 원점선 바로 밑에 표시합니다.

- Draw Label On Origin 옵션을 체크하지 않은 경우



- Draw Label On Origin 옵션을 체크한 경우



■ Reverse Order(Left)

왼쪽 Y축의 값을 역순으로 표시할지 여부를 설정합니다. 옵션을 체크하지 않으면 왼쪽 Y축 값이 순서대로 표시되고 체크하면 역순으로 표시됩니다.

■ Reverse Order(Right)

오른쪽 Y축의 값을 역순으로 표시할지 여부를 설정합니다. 옵션을 체크하지 않으면 오른쪽 Y축 값이 순서대로 표시되고 체크하면 역순으로 표시됩니다.

※ 주의사항

- "Reverse Order(Right)" 옵션만 체크할 경우 차트 마법사의 미리보기창에는 적용되지 않습니다.
- 방사형 타입과 같이 좌우 구분이 없는 경우 "Reverse Order(Left)" 속성 값에 따라 오른쪽 축의 역순 여부가 결정됩니다.

■ Show Oppposite YAxis

Y축의 반대편에도 축을 표시할지 여부를 설정합니다.

■ Label Interval

Y축 라벨의 표시 주기를 설정합니다. 표시 주기를 '1'로 설정한 경우에는 모든 라벨이 표시되며, '2'로 설정하면 첫번째, 세번째, 다섯번째, ... 주기로 표시됩니다.

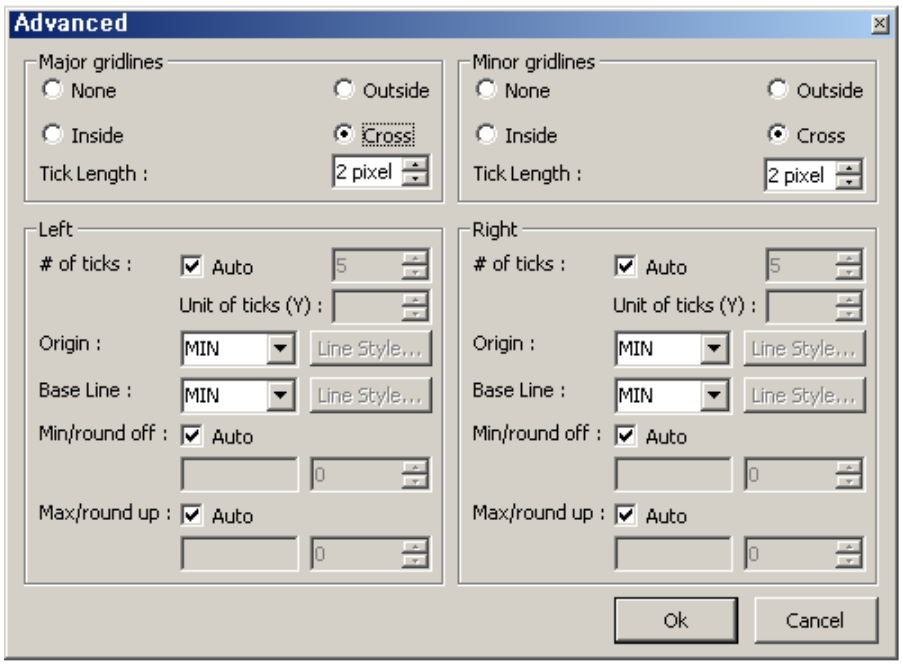
■ Minor Gridline Unit

Y축 보조 눈금선의 간격을 설정합니다.

■ Advanced

Y축의 기본선에 대한 주 눈금선과 보조 눈금선의 표시 위치와 왼쪽 Y축, 오른쪽 Y축의 눈금 단위, 눈금 수, 원점, 최소값, 버림 자리수, 최대값, 올림 자리수를 설정합니다.

Tick Length	눈금의 길이를 설정합니다. ※ 주의사항 : "눈금 길이"의 기본 값은 2픽셀이며, 값의 범위는 1 이상인 정수 값만 입력이 가능합니다.



각 옵션과 기능은 아래와 같습니다.

Option		Description
Major / Minor Gridlines	None	기본선에 눈금선을 표시하지 않습니다.
	Outside	기본선의 바깥쪽에 눈금선을 표시합니다.
	Inside	기본선의 안쪽에 눈금선을 표시합니다.
	Cross	기본선에 교차하여 눈금선을 표시합니다.
	Tick Length	눈금의 길이를 설정합니다. ※ 주의사항 : "눈금 길이"의 기본 값은 2픽셀이며, 값의 범위는 1 이상인 정수 값만 입력이 가능합니다.
# of Ticks		왼쪽 Y축, 오른쪽 Y축의 눈금수를 설정합니다. Auto옵션을 체크하여 눈금수를 자동으로 설정하거나 텍스트 박스에 직접 입력하여 설정합니다. 설정한 눈금 수가 눈금 단위보다 많은 경우 눈금 단위가 우선적으로 적용됩니다.
Unit of Ticks		왼쪽 Y축, 오른쪽 Y축의 축의 눈금 단위를 설정합니다.
Origin		X축과 교차되는 Y축의 값을 설정합니다. 값을 콤보 박스에 직접 입력하거나 Y축의 최소값(MIN) 또는 최대값(MAX) 중 하나를 선택합니다.
Base Line		Y축의 기준선을 설정합니다.

Line Style	[Line Style] 버튼을 이용하여 눈금선의 스타일 즉 선 두께, 선 색 등을 설정합니다. [Line Style] 버튼 사용법은 아래 'Gridline Style Dialog' 부분을 참조하십시오.
Min **Round Off**	Y축 눈금값의 최소값을 설정합니다. Auto옵션을 체크하여 최소값을 자동으로 설정하거나 텍스트 박스에 최소값을 직접 입력 또는 버림 자리수를 입력하여 설정합니다. 버림 자리수를 입력하면 실제 데이터의 최소값에서 입력한 자리수만큼 버림 처리하여 자동으로 최소값을 설정합니다. 버림 자리수를 입력한 것보다 직접 입력한 최소값이 우선적으로 적용됩니다.
Max **Round Up**	Y축 눈금값의 최대값을 설정합니다. Auto옵션을 체크하여 최대값을 자동으로 설정하거나 텍스트 박스에 최대값을 직접 입력 또는 올림 자리수를 입력하여 설정합니다. 올림 자리수를 입력하면 실제 데이터의 최대값에서 입력한 자리수만큼 올림 처리하여 자동으로 최대값을 설정합니다. 올림 자리수를 입력한 것보다 직접 입력한 최대값이 우선적으로 적용됩니다.

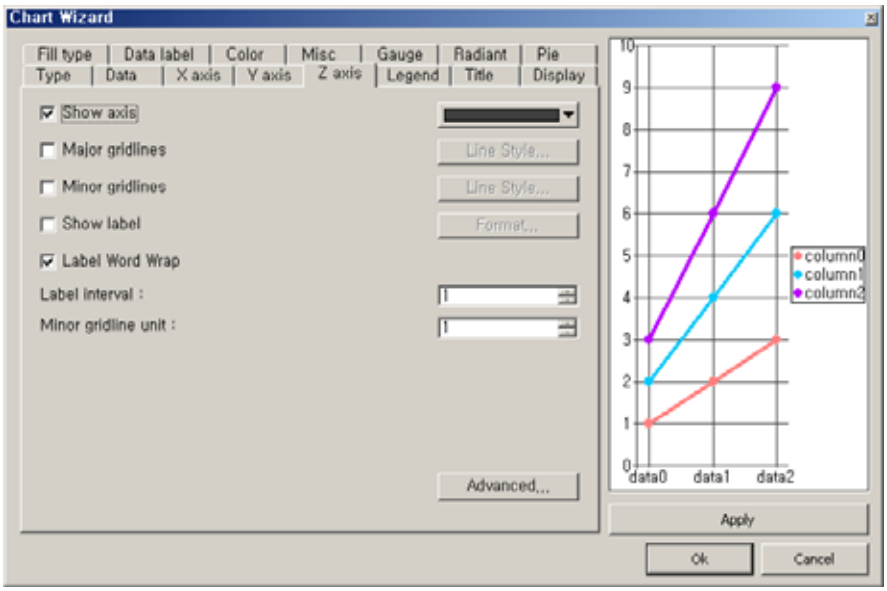
버림/올림 자리수는 음수 또는 양수로 입력하며, 소수점 위의 자리를 버림/올림할 경우에는 양수로 소수점 아래의 자리를 버림/올림할 경우에는 음수로 입력합니다.

실제 데이터의 최소값	버림 자리수	Y축 최소값
1276	2	1200
-1276	2	-1300
12.76	-2	12.7
-12.76	-2	-12.8

실제 데이터의 최대값	올림 자리수	Y축 최대값
1216	2	1300
-1216	2	-1200
12.16	-2	12.2
-12.16	-2	-12.1

Z Axis

Z축의 기본선, 주 눈금선, 보조 눈금선, 라벨 표시 등을 설정합니다.



- Show Axis

  Z축 기본선의 표시 여부를 설정하고, 색 선택 콤보 박스를 이용하여 기본선의 색을 변경합니다.

- Major Gridlines

  Z축의 주 눈금선의 표시 여부를 설정하고, [Line Style] 버튼을 이용하여 눈금선의 스타일 즉 선 두께, 선 색 등을 설정합니다. [Line Style] 버튼 사용법은 아래 'Gridline Style Dialog' 부분을 참조하십시오.

- Minor Gridlines

  Z축의 보조 눈금선의 표시 여부를 설정하고, [Line Style] 버튼을 이용하여 눈금선의 스타일 즉 선 두께, 선 색 등을 설정합니다. [Line Style] 버튼 사용법은 아래 'Gridline Style Dialog' 부분을 참조하십시오.

- Show Label

  Z축의 라벨 즉, Z축에 설정한 데이터 필드값의 표시 여부를 설정하고, [Format] 버튼을 이용하여 라벨의 포맷을 설정합니다. [Format] 버튼 사용법은 아래 'Format Dialog' 부분을 참조하십시오.

- Label Word Wrap

  Z축의 라벨의 자동 줄바꾸기 여부를 설정합니다.

- Label Interval

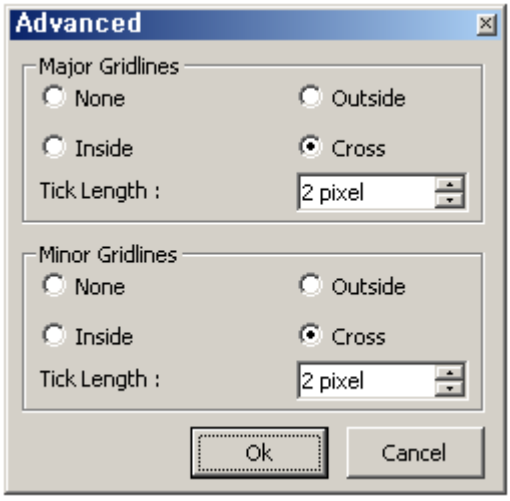
  Z축 라벨의 표시 주기를 설정합니다. 표시 주기를 '1'로 설정한 경우에는 모든 라벨이 표시되며, '2'로 설정하면 첫번째, 세번째, 다섯번째, ... 주기로 표시됩니다.

- Minor Gridline Unit

  Z축 보조 눈금선의 간격을 설정합니다.

- Advanced

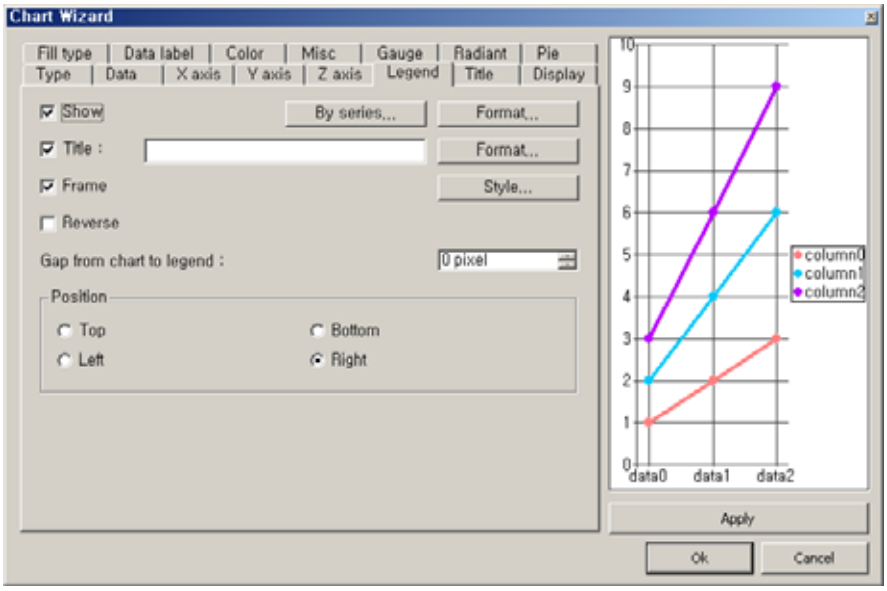
  Z축의 기본선에 대한 주 눈금선과 보조 눈금선의 표시 위치를 설정합니다.



※ 참고사항 : 각 옵션과 기능은 X축 기본선과 동일합니다.

Legend

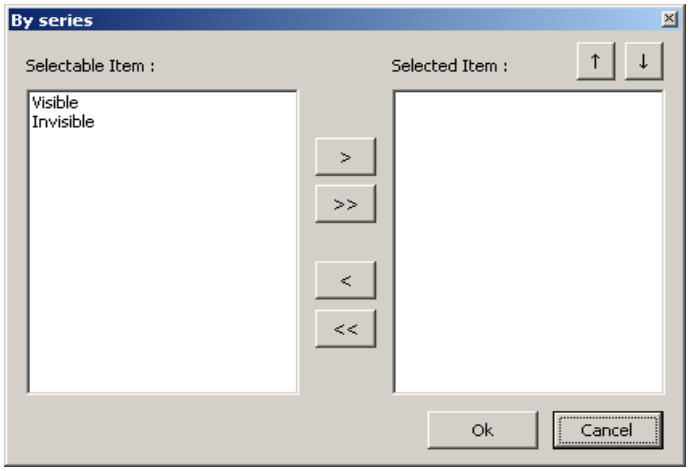
Chart의 범례의 표시 여부, 표시 주기, 제목, 프레임, 표시 위치 등을 설정합니다.



■ Show

범례의 전체 표시 여부를 설정합니다. [By Series] 버튼으로 범례로 설정된 필드의 값의 표시 여부를 개별적으로 설정하며, [Format] 버튼으로는 범례의 포맷을 설정합니다. [Format] 버튼 사용법은 아래 'Format Dialog' 부분을 참조하십시오.

[By Series] 버튼을 클릭하면 아래와 같은 다이얼로그가 표시됩니다.

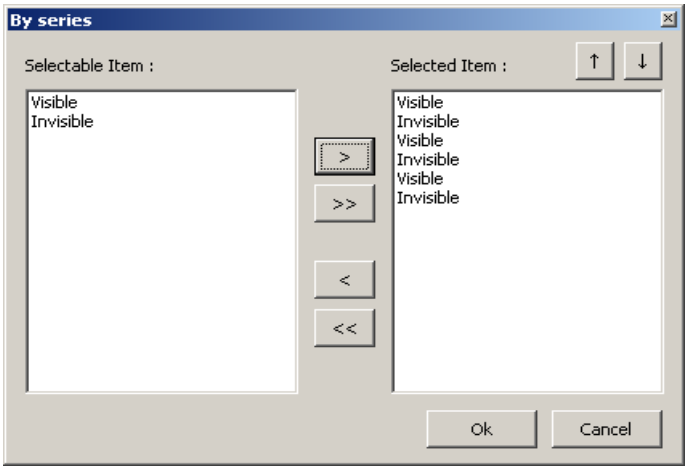


해당 순서의 범례 아이템의 표시 여부를 설정하는 방법은 화면 '선택 가능한 항목'창에서 'Visible' 또는 'Invisible'을 마우스로 선택하여 화면 '선택한 항목'창에 Drag & Drop하거나 아이템 이동 버튼(>, >>, <, <<)을 클릭하여 설정합니다. 선택된 아이템

의 순서를 변경하려면 순서를 변경할 아이템을 선택한 후 업/다운 버튼(↑ ↓)으로 순서를 변경합니다.

기본 값은 'Visible'이며, 만일 화면 오른쪽 표시 순서 설정창에 아무값도 설정되어 있지 않으면 각 순서의 범례 아이템을 모두 표시합니다.

예를 들어, 범례 아이템이 모두 10개라고 가정했을 때, 아래 그림과 같이 설정하면



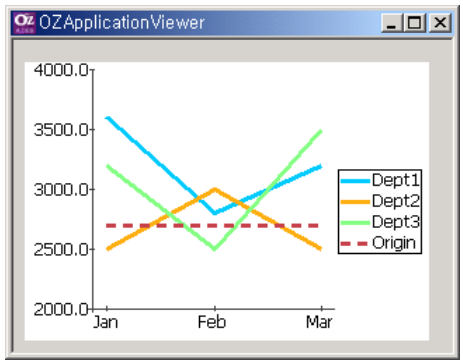
'Invisible'로 설정한 두번째, 네번째, 여섯번째 범례 아이템은 표시하지 않고, 'Visible'로 설정한 첫번째, 세번째, 다섯번째 범례 아이템은 표시합니다. 또한 표시 여부를 설정하지 않은 일곱번째부터 열번째 범례 아이템은 기본 값인 'Visible'로 설정한 것으로 간주되어 표시됩니다.

※ 참고사항 : 기본적으로 범례에 설정된 필드의 값을 그룹핑하여 런타임시에 동적으로 범례를 생성하기 때문에 디자인 시점에서는 실제로 범례에 들어가는 값을 표시할 수 없습니다. 그러므로 마법사에서도 단순히 'Visible', 'Invisible' 순서대로 나열만 가능합니다.

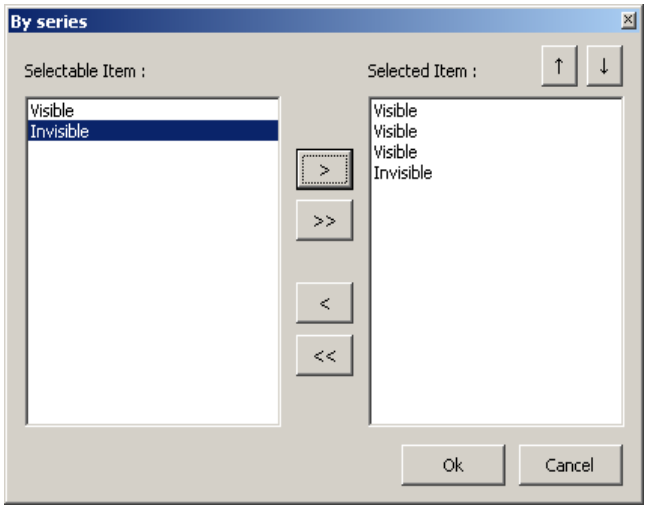
예를 들어 Chart의 기준선이나 평균값 등이 범례 필드의 값으로 설정되어 있을 경우 화면에 표시할 범례에는 '기준선', '평균값' 이런 텍스트가 표시되지 않기를 원할 경우 유용하게 사용할 수 있습니다.

만약 Chart에 표시할 데이터가 아래 그림과 같을 경우 X축 데이터에는 'Month'를 설정하고 Y축 데이터에는 'Sales'를 설정하고 범례에는 'Department'를 설정하면 범례에 기준값을 나타내는 'Origin'이라는 내용도 표시됩니다.

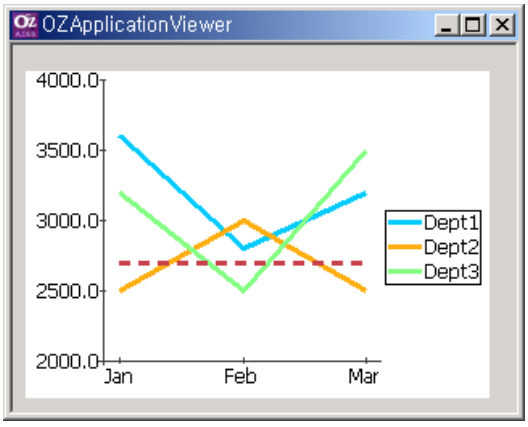
Month	Sales	Department
Jan	3600	Dept1
Jan	2500	Dept2
Jan	3200	Dept3
Feb	2800	Dept1
Feb	3000	Dept2
Feb	2500	Dept3
Mar	3200	Dept1
Mar	2500	Dept2
Mar	3500	Dept3
Jan	2700	Origin
Feb	2700	Origin
Mar	2700	Origin



이 때 범례에 'Origin'은 표시하고 싶지 않을 경우에는 범례의 [By Series] 버튼을 클릭하여 아래 그림처럼 설정하면 됩니다.



즉, 범례 필드로 설정한 값의 네번째 값을 표시하지 않게 설정하면 Chart의 범례는 아래 그림처럼 표시됩니다.

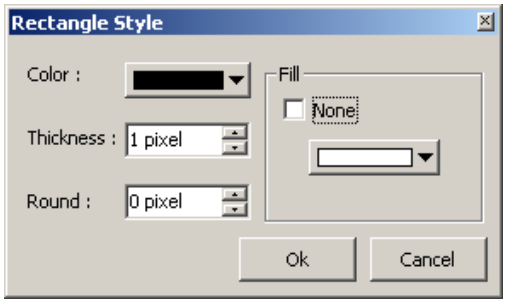


■ Title

범례의 제목 표시 여부를 설정하고, 텍스트 박스에 범례의 제목을 입력합니다. [Format] 버튼을 이용하여 제목의 포맷을 설정하며, [Format] 버튼 사용법은 아래 'Format Dialog' 부분을 참조하십시오.

■ Frame

범례 영역의 테두리 표시 여부를 설정하고, [Style] 버튼을 이용하여 테두리의 스타일을 설정합니다. [Style] 버튼을 클릭하면 아래와 같이 'Rectangle Style Dialog'가 나타납니다.



각 옵션과 기능은 아래와 같습니다.

Option	Description
Color	테두리의 선 색을 설정합니다.
Fill	범례 영역의 배경색과 색 채움 여부를 설정합니다. 'None' 옵션을 체크하면 배경을 채우지 않습니다.
Thickness	테두리선이 두께를 설정합니다.
Round	테두리선의 굴림 정도를 설정합니다.

■ Reverse

범례의 순서를 역순으로 표시할지 여부를 설정합니다. 체크하지 않으면 데이터셋에서 가져온 순서대로 표시되고, 체크하면 역순으로 표시합니다.

■ Gap from chart to legend

차트가 그려지는 영역과 범례 영역 사이의 간격을 설정합니다.

■ Position

Chart 전체 표시 영역에서 범례의 표시 위치를 위(Top), 아래(Bottom), 왼쪽(Left), 오른쪽(Rignt)으로 설정합니다.

Title

Chart의 제목을 설정합니다.

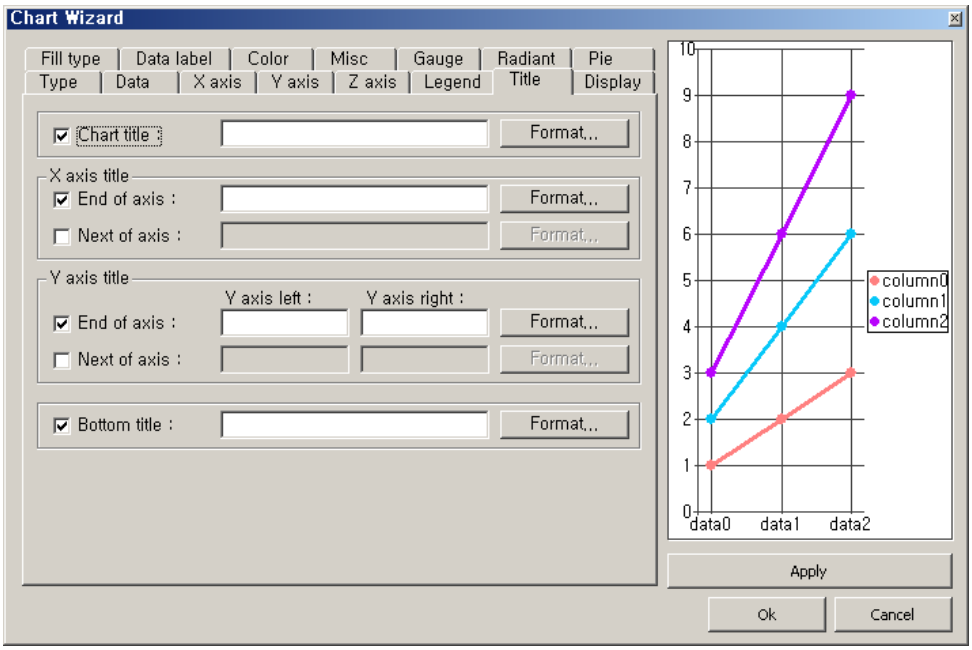


Chart 상단, 하단, X축, Y축의 제목을 텍스트 박스에 입력하고 [Format] 버튼을 클릭하여 각각의 제목에 해당하는 문자열의 포맷을 설정합니다. [Format] 버튼 사용법은 아래 'Format Dialog' 부분을 참조하십시오.

- Chart Title

  Chart 상단에 표시할 제목을 설정합니다.

- X Axis Title

  X축의 끝에 표시할 제목은 'End of Axis'를 체크하여 텍스트 박스에 입력하고, X축의 아래에 표시할 제목은 'Next of Axis'를 체크하고 텍스트 박스에 입력합니다.

- Y Axis Title

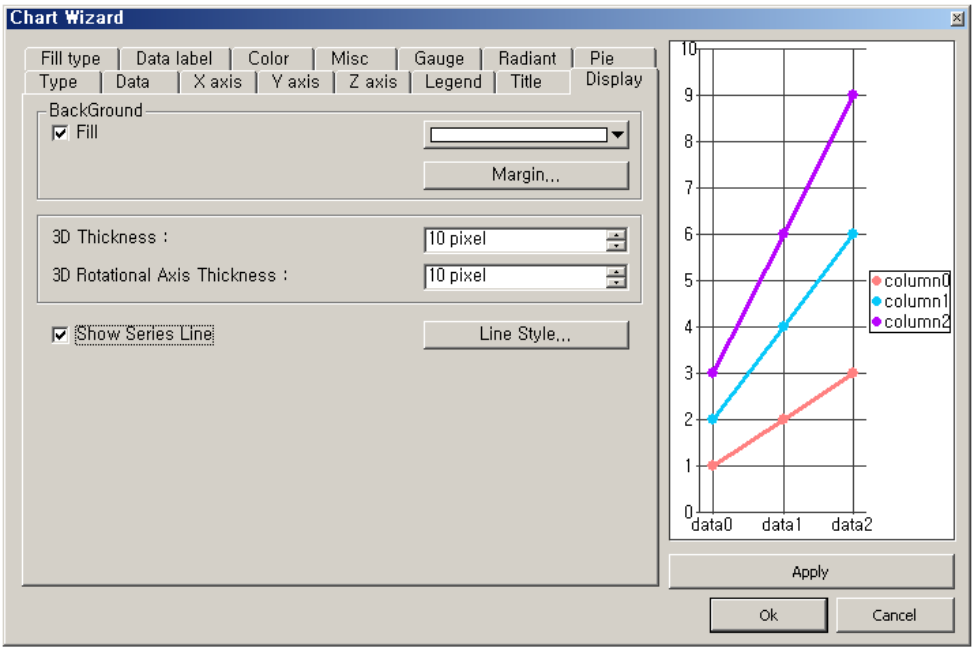
  Y축의 위에 표시할 제목은 'End of Axis'를 체크한 후 Y Axis Left(첫번째 텍스트 박스) 아래에는 왼쪽 Y축의 제목을, Y Axis Right(두번째 텍스트 박스) 아래에는 오른쪽 Y축의 제목을 입력합니다. Y축의 옆면에 표시할 제목은 'Next of Axis'를 체크한 후 Y Axis Left(첫번째 텍스트 박스) 아래에는 왼쪽 Y축의 제목을, Y Axis Right(두번째 텍스트 박스) 아래에는 오른쪽 Y축의 제목을 입력합니다.

■ Bottom Title

Chart 하단에 표시할 제목을 설정합니다.

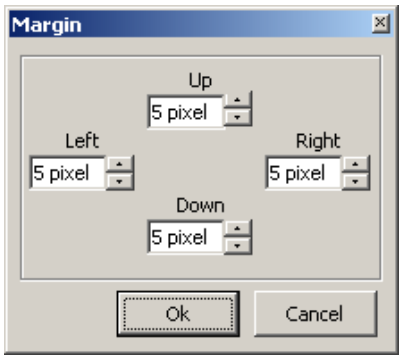
Display

Chart의 배경색, 여백, 두께 등을 설정합니다.



■ BackGround

- Chart 표시 영역의 배경색과 색 채움 여부를 설정합니다. 배경색을 채울 경우 Fill 옵션을 체크합니다. Chart 표시 영역 내에서 그래프와 Chart 영역간의 위, 아래, 왼쪽, 오른쪽의 여백을 설정하려면 [Margin] 버튼을 클릭합니다. [Margin] 버튼을 클릭하면 아래와 같이 'Margin Dialog'가 나타납니다.



■ 3D Thickness

3D 막대 그래프일 경우 막대의 두께 등 Chart에 입체적 효과를 줄 경우 입체 효과의 깊이 수준을 설정합니다.

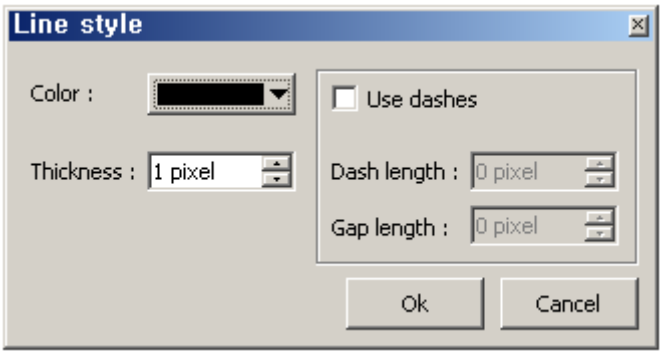
■ 3D Rotational Axis Thickness

Chart에 입체적 효과를 줄 경우 축의 두께를 설정합니다.

■ Show Series Line

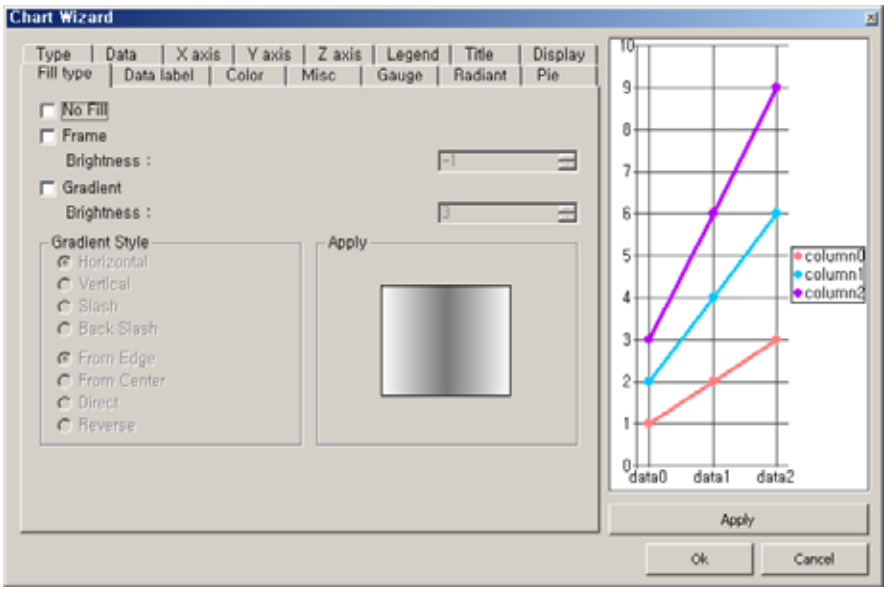
계열선 표시 여부를 설정합니다. 계열선 보기를 설정할 경우 Show Series Line 옵션을 체크합니다. 계열열선의 색상, 두께, 점선, 간격을 설정하려면 [Line Style] 버튼을 클릭합니다. [Line Style] 버튼을 클릭하면 아래와 같이 'Line Style Dialog'가 나타납니다.

※ 주의사항 : 계열선 보기 기능은 차트 타입이 누적형의 가로/세로 평면 막대형, 가로/세로 평면 원통형, 사용자형의 막대형인 경우에만 적용됩니다.



Fill Type

막대 그래프인 경우에만 적용되는 옵션으로 막대의 채움 속성을 설정합니다.



- No Fill

  막대의 면을 채우지 않습니다. 옵션을 체크하면 막대의 테두리는 표시되고 막대의 면은 투명하게 표시됩니다.

- Frame

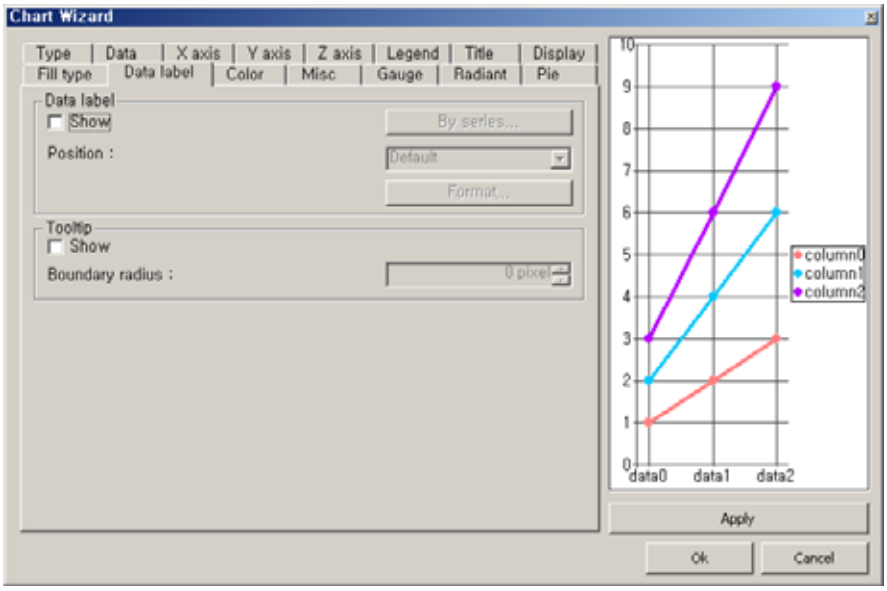
  테두리의 색 밝기를 조절할지 여부를 설정합니다. Frame 옵션을 체크한 후 Brightness 옵션으로 밝기 정도를 설정합니다. Brightness 옵션값을 양수로 설정하면 더 밝아지고, 음수로 설정하면 더 어두워집니다.

- Gradient

  막대의 면을 그라데이션을 주어 표시할지 여부를 설정합니다. Gradient 옵션을 체크한 후 Gradient Style을 선택하고, Brightness 옵션으로 밝기 정도를 설정합니다.

Data Label

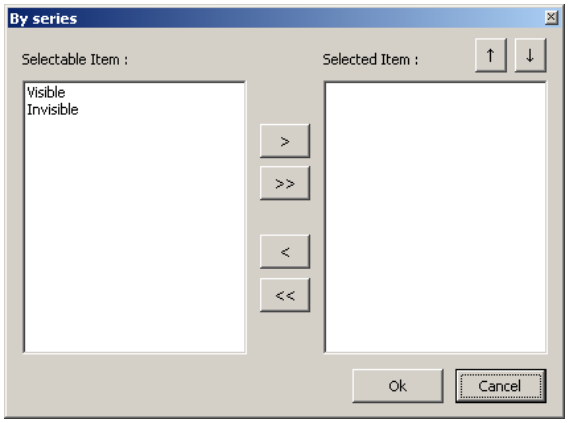
그래프의 데이터 값을 라벨 또는 툴팁으로 표시할지를 설정합니다.



■ Data Label

데이터 값을 라벨로 표시할지 여부, 표시 주기, 표시 위치, 포맷 등을 설정합니다. 'Show' 옵션을 체크하면 데이터 값을 라벨로 표시합니다. [By Series] 버튼으로 데이터 라벨의 주기를 설정하고, 'Position' 옵션을 변경하여 표시 위치를 설정합니다. [Format] 버튼을 이용하여 라벨의 포맷을 설정하며, [Format] 버튼 사용법은 아래 'Format Dialog' 부분을 참조하십시오.

[By Series] 버튼을 클릭하면 아래와 같은 다이얼로그가 표시됩니다.



위 다이얼로그의 기능은 Legend의 'By Series' 다이얼로그와 동일합니다.

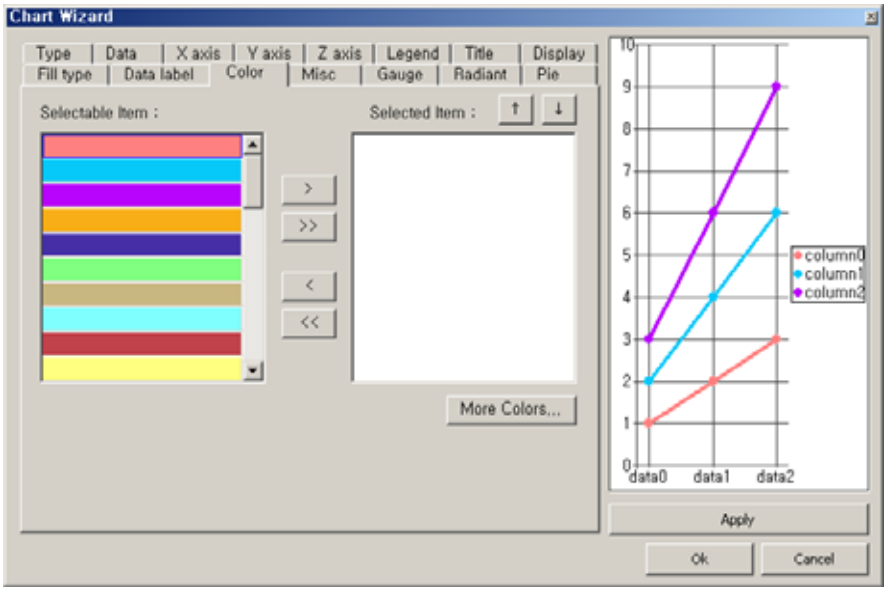
■ Tooltip

데이터 값을 툴팁으로 표시할지 여부와 툴팁 경계선을 설정합니다. 'Show' 옵션을 체크하면

데이터 값을 툴팁으로 표시하며, 'Boundary Radius' 옵션을 이용하여 툴팁 경계선을 설정합니다. 툴팁 경계선은 마우스가 그래프의 막대나 점 등에 어느 정도 근접했을 때 툴팁을 표시할지를 설정하는 값으로, 막대나 점에서의 반경을 픽셀 단위로 설정합니다.

Color

그래프를 표시하는 막대나 선의 색을 설정합니다.



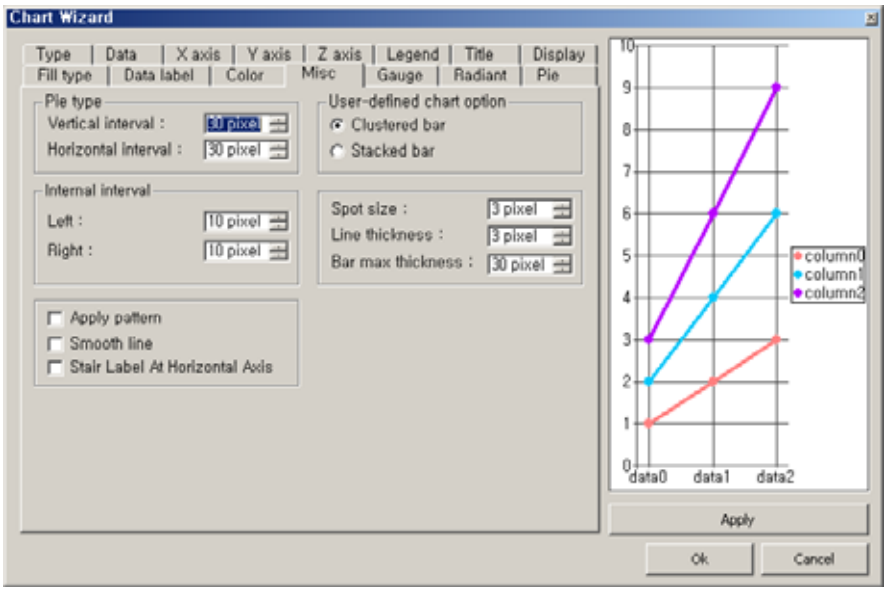
'선택 가능한 항목'창에서 적용할 색을 선택한 후 '선택한 항목'창으로 Drag & Drop하거나 아이템 이동 버튼(>, >>, <, <<)을 클릭하여 색을 설정합니다.

선택된 색상의 순서를 변경하려면 순서를 변경할 색상을 선택한 후 업/다운 버튼(↑ ↓)으로 순서를 변경합니다.

기본색이 아닌 다른 색을 설정할 경우에는 [User Defined] 버튼을 클릭하여 색을 추가할 수 있습니다.

Misc

그 밖의 원형 타입, 간격, 두께, 패턴 등을 설정합니다.



- Pie Type

  원형 Chart가 한 Chart 영역 내에 여러 개가 표시되는 경우 수직, 수평 간격을 설정하는 것으로, 'Vertical Interval'에는 수직 간격을 'Horizontal Interval'에는 수평 간격을 각각 입력합니다.

- User-Defined Chart Option

  사용자 정의 Chart에서 막대가 2개 이상인 경우 막대의 표시 형태를 설정하는 것으로, 'Clustered Bar' 옵션을 선택하면 2개 이상의 막대를 옆으로 묶어 놓은 묶음 막대형으로 'Stacked Bar' 옵션을 선택하면 하나의 막대에 값을 누적시킨 누적 막대형으로 표시합니다.

- Internal Interval

  Chart 내부의 왼쪽, 오른쪽 간격을 설정합니다.

- Spot Size

  점선일 경우 점의 크기를 설정합니다.

- Line Thickness

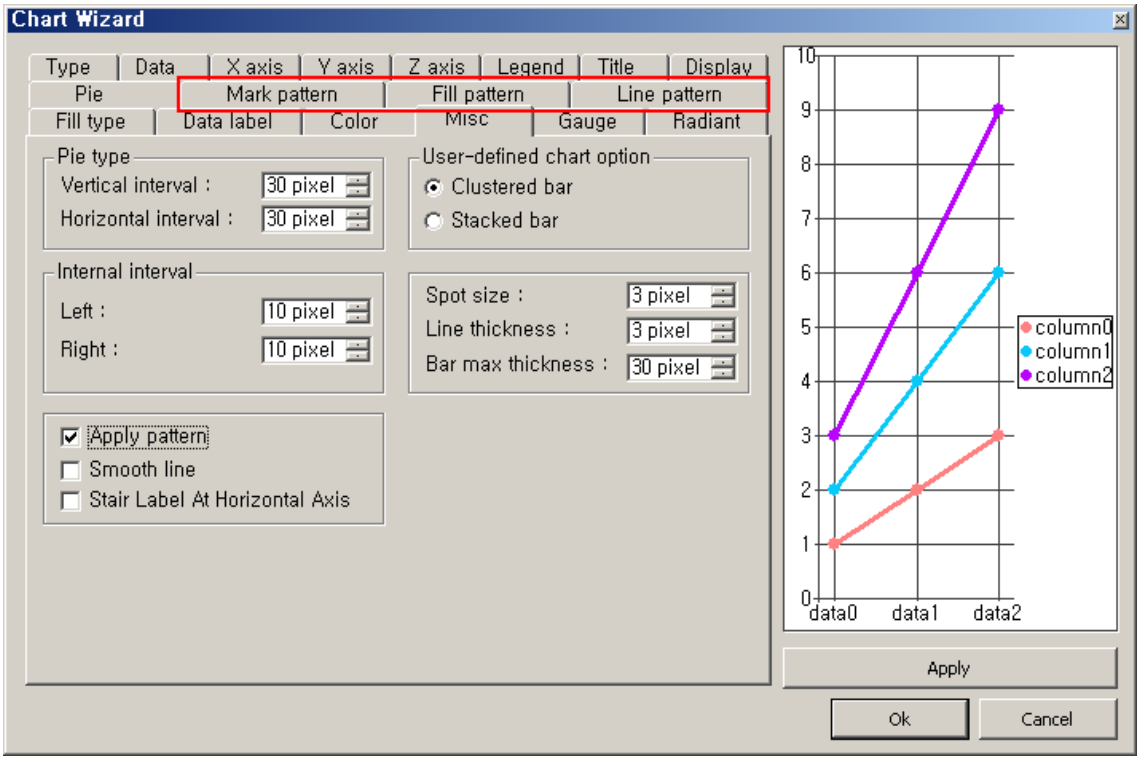
  선의 두께를 설정합니다.

■ Bar Max Thickness

막대의 최대 두께를 설정합니다.

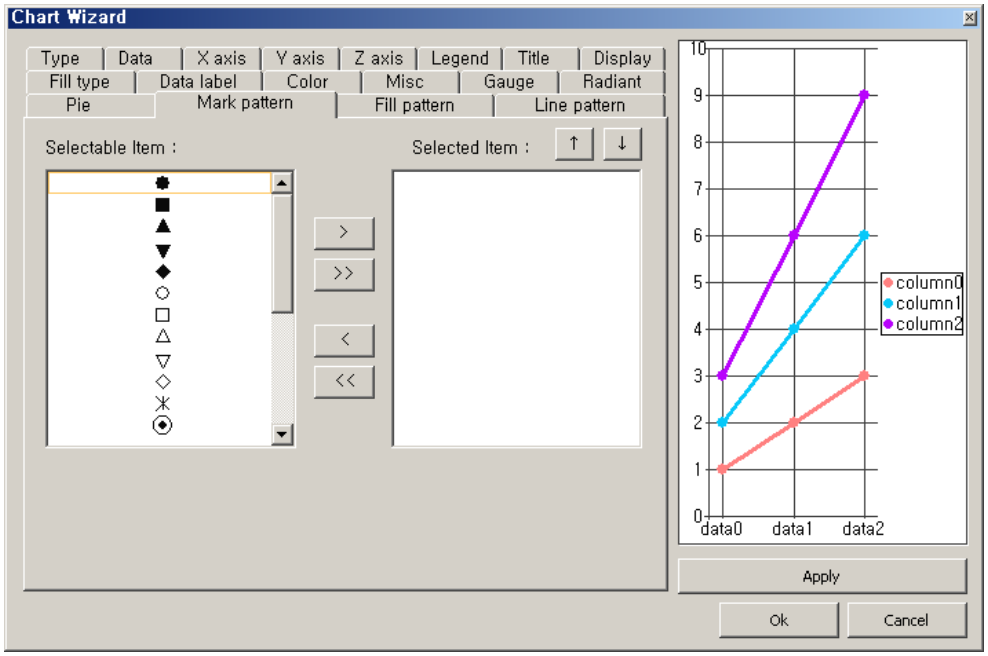
■ Apply Pattern

그래프의 점, 선, 면의 패턴 적용 여부를 설정합니다. 'Apply Pattern' 옵션을 체크하면 그래프의 점, 선, 면에 패턴이 적용되고 Chart 마법사에는 점, 선, 면의 패턴을 설정할 수 있는 'Mark Pattern', 'Fill Pattern', 'Line Pattern'의 3개의 탭이 추가되어 표시됩니다.



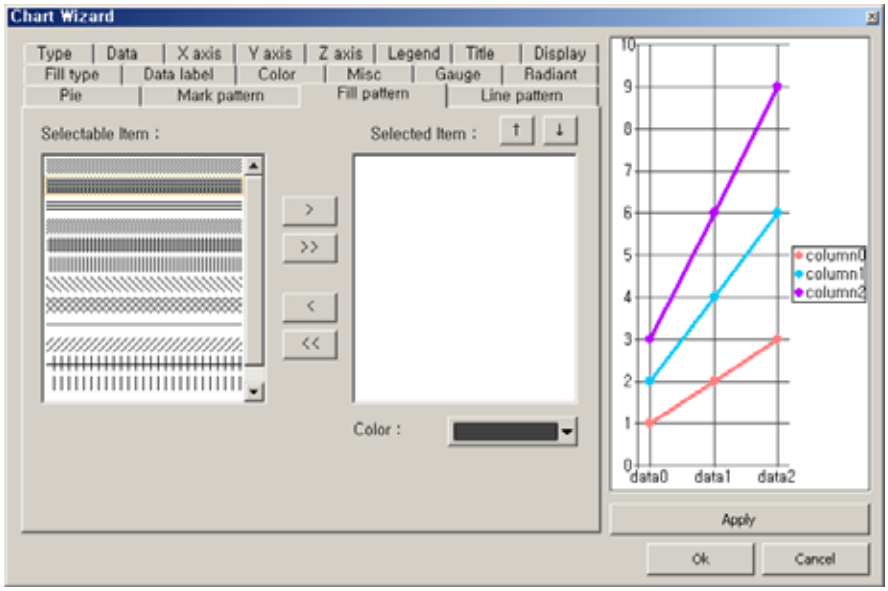
- Mark Pattern

Chart 타입이 표식이 있는 꺽은 선형이나 산포형인 경우 점의 표식을 설정합니다.



'선택 가능한 항목'창에서 적용할 표식을 선택한 후 '선택한 항목'창으로 Drag & Drop 하거나 아이템 이동 버튼(, , , )을 클릭하여 설정합니다.
선택된 아이템의 순서를 변경하려면 순서를 변경할 아이템을 선택한 후 업/다운 버튼 ()으로 순서를 변경합니다.

- Fill Pattern

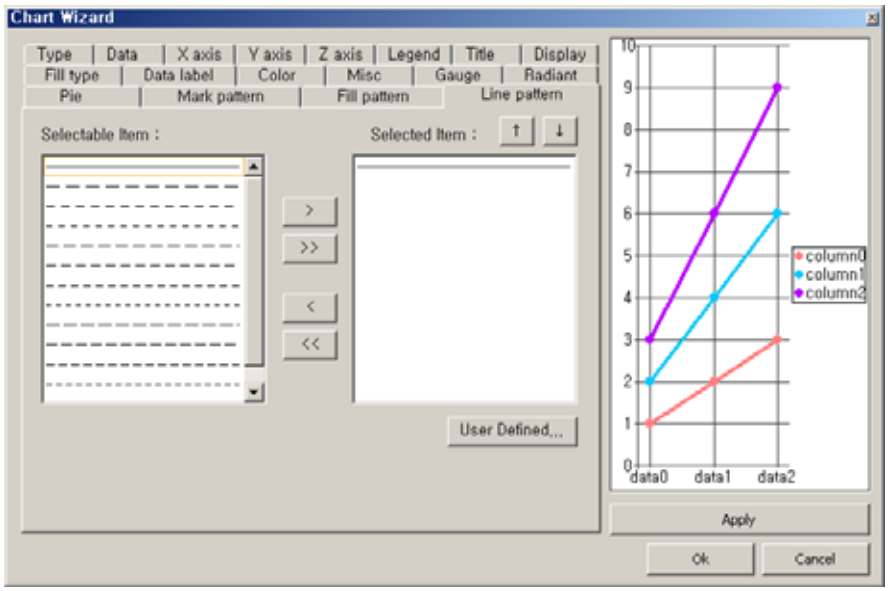


'선택 가능한 항목'창에서 적용할 표식을 선택한 후 '선택한 항목'창으로 Drag & Drop 하거나 아이템 이동 버튼(> , >> , < , <<)을 클릭하여 설정합니다.
선택된 아이템의 순서를 변경하려면 순서를 변경할 아이템을 선택한 후 업/다운 버튼 (↑ ↓)으로 순서를 변경합니다.

'Color' 옵션에서 면 패턴의 색을 설정합니다.

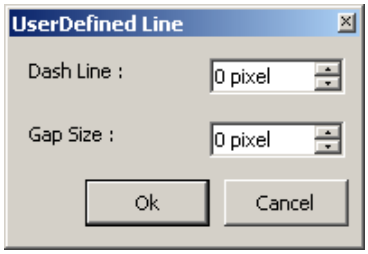
- Line Pattern

선 모양을 설정합니다.



'선택 가능한 항목'창에서 적용할 표식을 선택한 후 '선택한 항목'창으로 Drag & Drop 하거나 아이템 이동 버튼(> , >> , < , <<)을 클릭하여 설정합니다.
선택된 아이템의 순서를 변경하려면 순서를 변경할 아이템을 선택한 후 업/다운 버튼 (↑ ↓)으로 순서를 변경합니다.

[User Defined] 버튼으로 사용자가 선 모양을 추가할 수 있으며, [User Defined] 버튼을 클릭하면 아래와 같이 선 모양을 설정할 수 있는 다이얼로그가 표시됩니다.



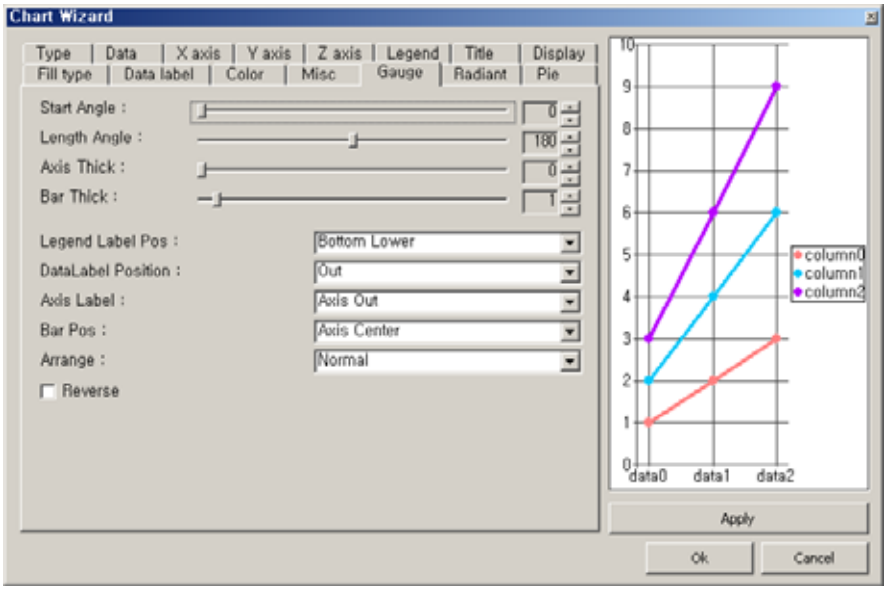
'Dash Line'에는 화면에 표시할 선의 길이를 'Gap Size'에는 선과 선사이의 간격을 입력한 후 [OK] 버튼을 클릭하면 선 모양이 추가됩니다. 선 모양을 실선으로 설정할 경우에는 'Gap Size' 옵션값을 '0'으로 입력하고, 점선으로 설정할 경우에는 'Gap Size' 옵션값을 '0'보다 큰 값으로 입력하십시오.

- Smoth Line
  꺽은선 형태의 Chart일 경우 꺽은선 또는 2D 영역선을 완만하게 표시할지 여부를 설정합니다.

- Stair Label At Horizontal Axis
  수평 축에 표시되는 라벨의 텍스트가 라벨 너비보다 클 경우 2단으로 표시할지 여부를 설정합니다.

Gauge

게이지 Chart의 축의 각도, 두께, 라벨 위치, 정렬 방식 등을 설정합니다.



- Start Angle

  축이 시작되는 각도를 설정합니다.

- Length Angle

  게이지 Chart가 표시되는 각도 크기를 설정합니다.

- Axis Thick

  Y축으로 나타내어지는 축의 두께를 설정합니다.

- Bar Thick

  게이지 Chart에 표시되는 막대의 두께를 설정합니다.

- Legend Label Pos

  X축 라벨 표시를 선택했을 때 표시되는 계열 데이터의 라벨 위치를 설정합니다.

- DataLabel Position

  데이터 라벨의 위치를 설정합니다.

- Axis Label

  Y축 라벨의 위치를 설정합니다.

- Bar Pos

  게이지 막대를 축의 어느 위치까지 그려줄 것인지를 설정합니다.

- Arrange

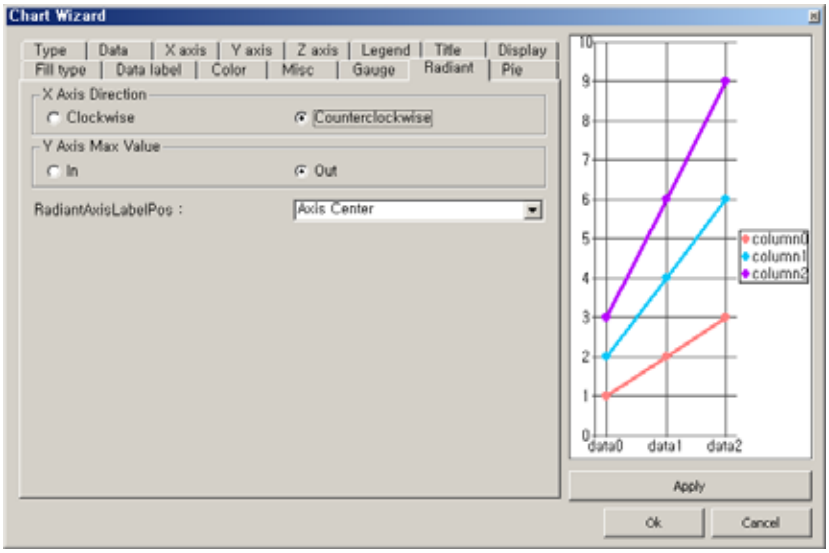
  Chart 영역 안에 표시되는 Chart의 정렬 방식을 설정합니다.

- Reverse

  데이터를 게이지 Chart가 그려지는 시작 위치에서부터 그려줄 것인지 끝 위치에서 그려줄 것인지를 설정합니다. 'Reverse' 옵션을 체크하면 반시계 방향으로 그려집니다.

Radiant

방사형 Chart의 X축 방향, Y축 최대값의 위치, 축 라벨 위치 등을 설정합니다.



- X Axis Direction

  X축 방향을 시계 방향 또는 반시계 방향으로 설정합니다.

- Y Axis Max Value

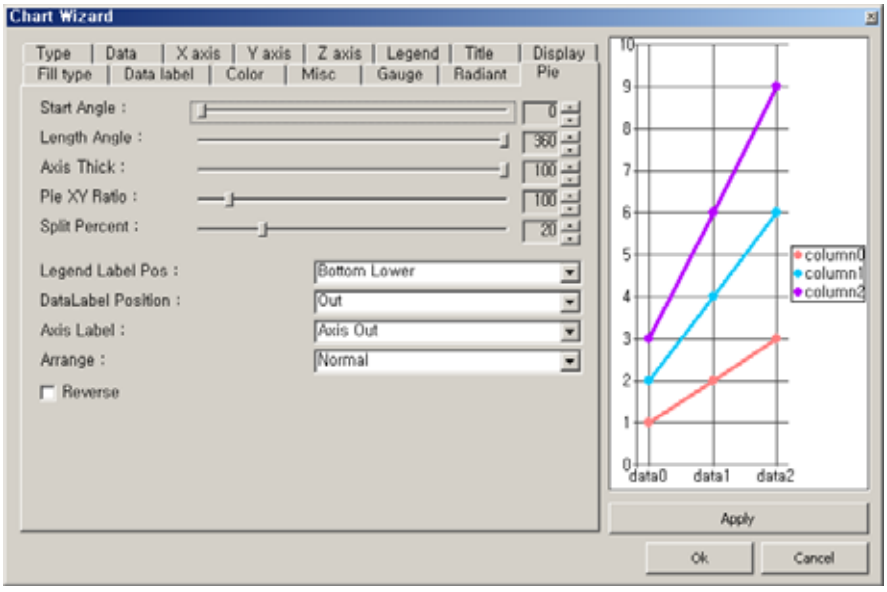
  Y축 최대값의 위치를 안쪽 또는 바깥쪽으로 설정합니다. 최대값을 안쪽에 표시하면 자연히 최소값은 바깥쪽에 표시됩니다.

■ Radiant Axis Label Position

축 라벨의 위치를 축 밖, 축 안, 축 중앙으로 설정합니다.

Pie

원형 Chart의 축의 각도, 두께, 라벨 위치, 정렬 방식 등을 설정합니다.



■ Start Angle

축이 시작되는 각도를 설정합니다.

■ Length Angle

원형 Chart가 시작되는 각도 크기를 설정합니다.

■ Axis Thick

원형 중심으로부터 어느 정도의 축 두께를 빼고 나타낼 것인지를 설정합니다. 축 두께를 적절하게 설정하면 도넛 형태의 파이 차트를 만들 수 있습니다.

■ Pie XY Ratio

원형 Chart의 가로 세로 비율을 설정합니다.

- Split Percent

  분리된 원형 차트의 분리된 정도를 설정합니다.

- Legend Label

  데이터 값을 표시할 계열 데이터의 라벨 위치를 설정합니다.

- Axis Label

  Z축 라벨 표시를 선택하였을 때 Z축 라벨의 위치를 설정합니다.

- Arrange

  Chart 영역 안에 표시되는 Chart의 정렬 방식을 설정합니다.

- Reverse

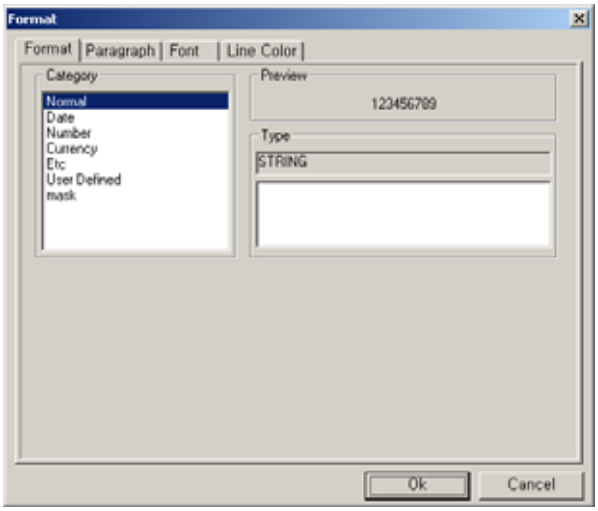
  데이터를 원형 Chart가 그려지는 시작 위치에서부터 그려줄 것인지 끝 위치에서 그려줄 것인지를 설정합니다. 'Reverse' 옵션을 체크하면 시계 방향으로 그려집니다.

Format Dialog

제목, 축 라벨, 범례 등 화면에 표시되는 텍스트의 포맷을 설정합니다.

- Format

  텍스트의 표시 형식을 설정합니다.



범주에서 일반, 날짜, 숫자, 통화, 기타, 마스크를 선택한 후 형식에서 원하는 형식을 선택합니다. 원하는 형식이 없을 경우에는 범주에서 사용자 지정을 선택한 후 형식의 텍스트 박스에 원하는 형식을 입력합니다.

※ 참고사항 : Mask에서 패턴 입력 형식은 다음 설정 방법을 참조하시기 바랍니다.

Mask_[PromptChar]_[User define mask]

PromptChar는 한 글자만 입력해야합니다.

한글자 이상 입력하는 경우 Mask가 적용되지 않습니다.

예) mask_#_(###)####-#### -> 입력값 : 01122221234

적용값 : (011)2222-1234

1) 마스크 캐릭터 수 보다 입력한 값이 작은 경우

입력값 : 123

적용값 : ()-123

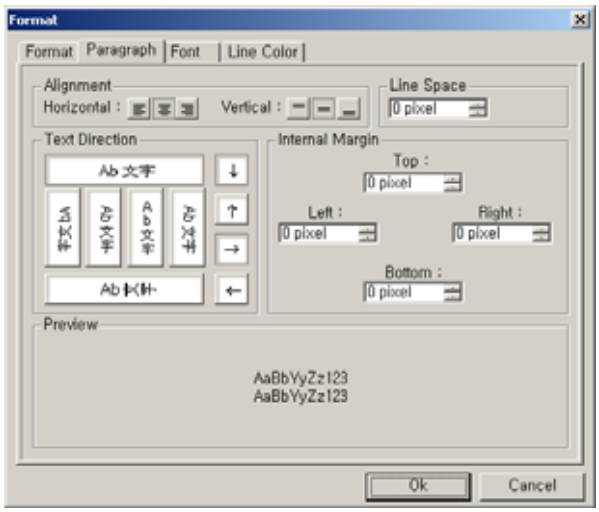
2) 마스크 케릭터 수 보다 입력한 값이 많은 경우

입력값 : 123456789123456

적용값 : (1234567)8912-3456

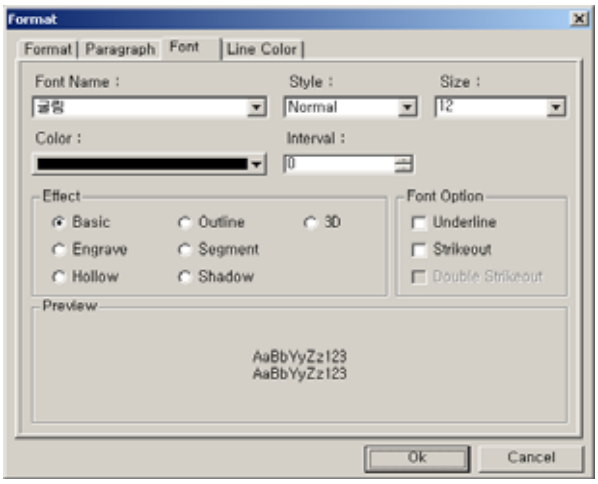
■ Paragraph

텍스트의 정렬 방식, 텍스트 방향, 여백, 줄간격을 설정합니다.



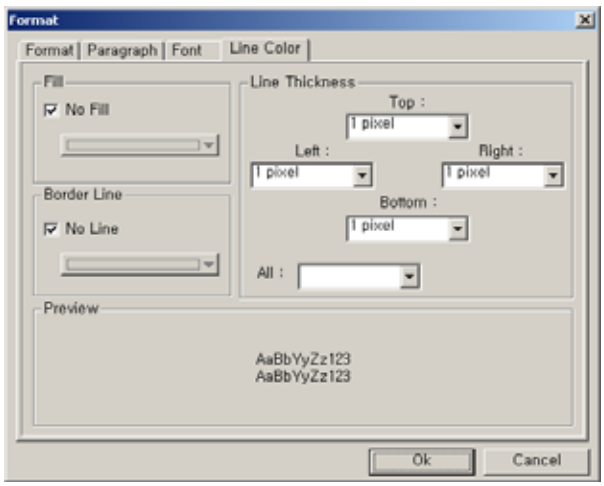
■ Font

텍스트의 글꼴, 스타일, 크기, 간격, 색을 설정합니다.



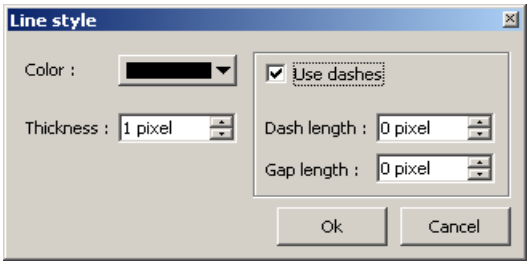
■ Line Color

텍스트의 테두리선과 배경 채우기 여부를 설정합니다.



Gridline Style Dialog

축의 주 눈금선과 보조 눈금선의 스타일을 설정합니다.



- Color

  눈금선의 색을 설정합니다.

- Use Dashes

  점선 여부를 설정합니다. 'Use Dashes' 옵션을 체크하면, 'Dash Length', 'Gap Length' 옵션이 활성화되며, 'Dash Length' 옵션에는 점선의 길이를 'Gap Length' 옵션에는 선과 선 사이의 간격을 입력합니다.

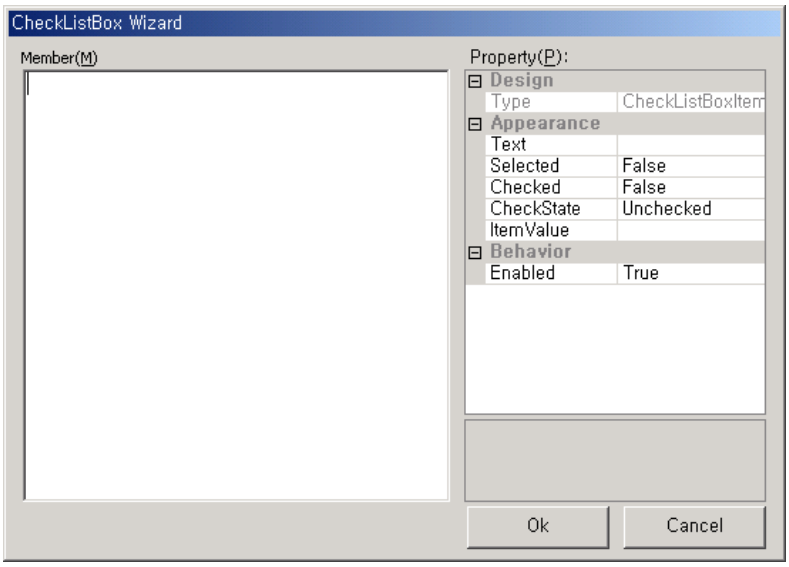
- Thickness

  선의 두께를 설정합니다.

CheckListBox 마법사

CheckListBox에 아이템을 추가하고 표시하기 위해 CheckListBox 마법사를 이용합니다. CheckListBox 마법사를 실행하기 위해서는 먼저 Board에 CheckListBox를 추가하고, CheckListBox를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Wizard]를 클릭하여 CheckListBox 마법사를 실행합니다.

CheckListBox 마법사는 아래 그림과 같이 CheckListBox의 아이템을 추가하는 화면 왼쪽의 'Member창'과 각각 아이템의 속성을 설정하는 화면 오른쪽의 '속성창'으로 구성되어 있습니다.



Member

CheckListBox의 멤버 즉, 아이템을 입력하여 추가합니다.

Property

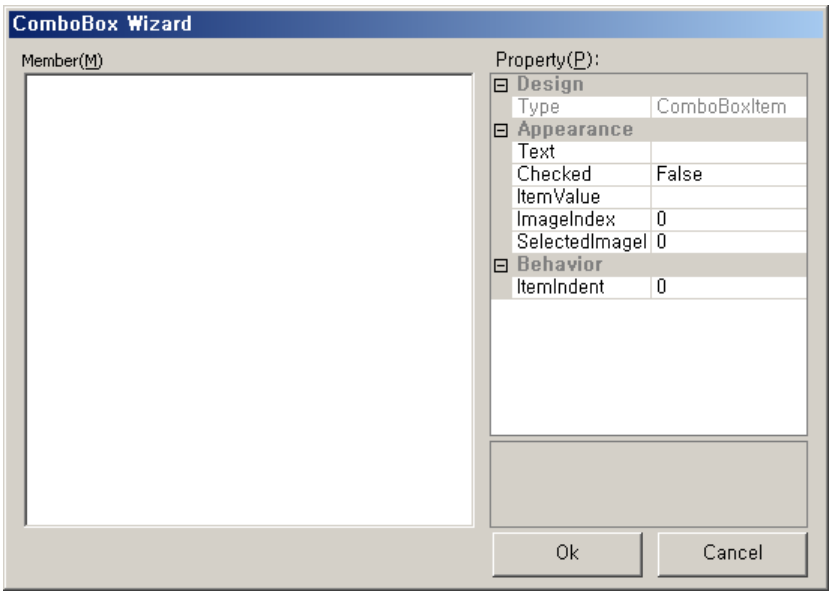
해당 아이템 즉, CheckListBoxItem의 속성을 설정합니다.

Property	Description
Type	아이템의 컴포넌트 타입입니다. CheckListBox 아이템의 컴포넌트 타입은 CheckListBoxItem으로 설정됩니다.
Text	화면에 보이는 CheckListBoxItem의 텍스트를 설정합니다.
Selected	CheckListBoxItem 선택 여부를 설정합니다. 디자인시에는 보이지 않으며 실행시 속성 값을 'True'로 설정한 CheckListBoxItem이 선택 상태로 표시됩니다. CheckListBox 별로 하나의 선택 상태로 설정할 수 있습니다.
Checked	CheckListBoxItem의 체크 여부를 설정합니다.
CheckState	CheckListBoxItem의 체크 상태를 설정합니다.
ItemValue	CheckListBoxItem의 값을 설정합니다.
Enabled	CheckListBoxItem의 활성 여부를 설정합니다. 속성 값이 'False'인 경우에는 실행시에 체크 상태를 변경할 수 없습니다.

ComboBox 마법사

ComboBox에 아이템을 추가하고 표시하기 위해 ComboBox 마법사를 이용합니다. ComboBox 마법사를 실행하기 위해서는 먼저 Board에 ComboBox를 추가하고, ComboBox를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Wizard]를 클릭하여 ComboBox 마법사를 실행합니다.

ComboBox 마법사는 아래 그림과 같이 ComboBox의 아이템을 추가하는 화면 왼쪽의 'Member창'과 각각 아이템의 속성을 설정하는 화면 오른쪽의 '속성창'으로 구성되어 있습니다.



Member

ComboBox의 멤버 즉, 아이템을 입력하여 추가합니다.

Property

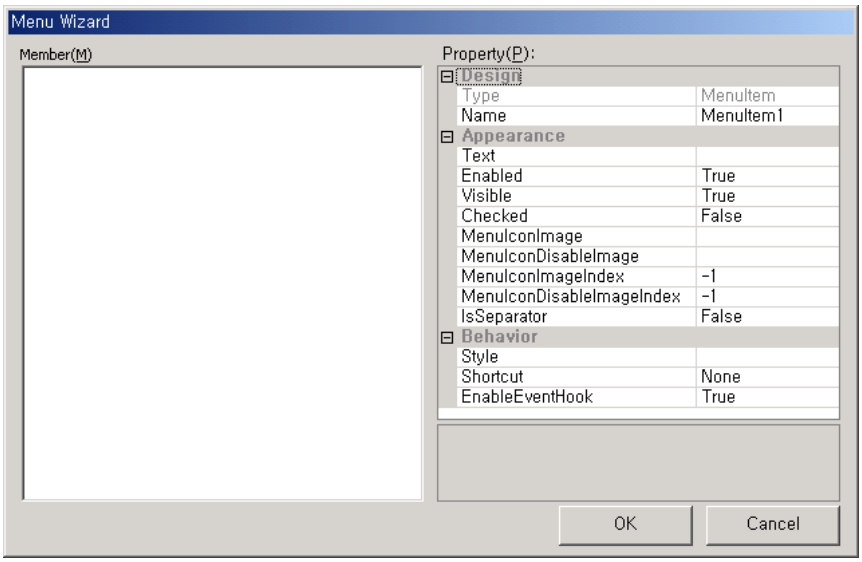
해당 아이템 즉, ComboBoxItem의 속성을 설정합니다.

Property	Description
Type	아이템의 컴포넌트 타입입니다. ComboBox 아이템의 컴포넌트 타입은 ComboBoxItem으로 설정됩니다.
Text	화면에 보이는 ComboBoxItem의 텍스트를 설정합니다.
ItemValue	ComboBoxItem의 값을 설정합니다.
ImageIndex	아이템 앞에 표시할 이미지의 인덱스를 설정합니다. 이 옵션은 ComboBox에 ImageList 컴포넌트가 설정되어 있는 경우에 가능합니다.
SelectedImageIndex	아이템이 선택된 상태일 경우에 표시할 이미지의 인덱스를 설정합니다. 이 옵션은 ComboBox에 ImageList 컴포넌트가 설정되어 있는 경우에 가능합니다.
ItemIndent	ComboBoxItem의 들여쓰기 간격을 설정합니다. 1은 10픽셀에 해당합니다.

ContextMenu/MainMenu/TrayContextMenu 마법사

ContextMenu 마법사와 MainMenu 마법사, TrayContextMenu 마법사의 사용법은 동일하므로 본 매뉴얼에서는 ContextMenu 마법사를 기준으로 설명합니다.
ContextMenu에 메뉴 아이템을 추가하고 표시하기 위해 ContextMenu 마법사를 이용합니다. ContextMenu 마법사를 실행하기 위해서는 먼저 Document에 ContextMenu를 추가하고, 'Invisibles 트리창'에서 ContextMenu를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Wizard]를 클릭하여 ContextMenu 마법사를 실행합니다.

ContextMenu 마법사는 아래 그림과 같이 ContextMenu의 메뉴 아이템을 추가하는 화면 왼쪽의 'Member창'과 각각 메뉴 아이템의 속성을 설정하는 화면 오른쪽의 '속성창'으로 구성되어 있습니다.



Member

ContextMenu의 멤버 즉, 메뉴 아이템을 입력하여 추가합니다.
메뉴 아이템 추가시에 서브 메뉴로 아이템을 추가할 경우에는 해당 아이템 문자열 앞에 [Tab]키를 입력하면 바로 위에 입력한 메뉴 아이템의 서브 메뉴로 추가됩니다.

Property

해당 아이템 즉, MenuItem의 속성을 설정합니다.

Property	Description
Type	메뉴 아이템의 컴포넌트 타입입니다. ContextMenu 아이템의 컴포넌트 타입은 MenuItem으로 설정됩니다.
Name	MenuItem의 이름을 설정합니다.
Text	화면에 보이는 MenuItem의 텍스트를 설정합니다.
Enabled	MenuItem의 활성 여부를 설정합니다. 속성 값이 'False'인 경우에는 실행시에 해당 메뉴를 선택할 수 없습니다.
Visible	MenuItem 표시 여부를 설정합니다.
Checked	MenuItem 앞에 체크 표시 여부를 설정합니다.
MenuIconImage	MenuItem이 활성화 상태이었을 때 해당 MenuItem 앞에 표시할 이미지 파일을 설정합니다.
MenuIconDisableImage	MenuItem이 비활성화 상태이었을 때 해당 MenuItem 앞에 표시할 이미지 파일을 설정합니다.
MenuIconImageIndex	MenuItem이 활성화 상태이었을 때 해당 MenuItem 앞에 표시할 이미지를 가져올 ImageList의 이미지 인덱스를 설정합니다. ContextMenu에 ImageList를 연결했을 경우에 적용되며, 'MenuIconImage' 속성 값에 설정한 이미지 파일 설정값보다 우선 적용됩니다.
MenuIconDisableImageIndex	MenuItem이 비활성화 상태이었을 때 해당 MenuItem 앞에 표시할 이미지를 가져올 ImageList의 이미지 인덱스를 설정합니다. ContextMenu에 ImageList를 연결했을 경우에 적용되며, 'MenuIconImage' 속성 값에 설정한 이미지 파일 설정값보다 우선 적용됩니다.
IsSeparator	MenuItem을 구분선으로 표시할지 여부를 설정합니다. 속성 값을 'True'로 설정하면 텍스트로 표시되지 않고 구분선으로 표시됩니다.

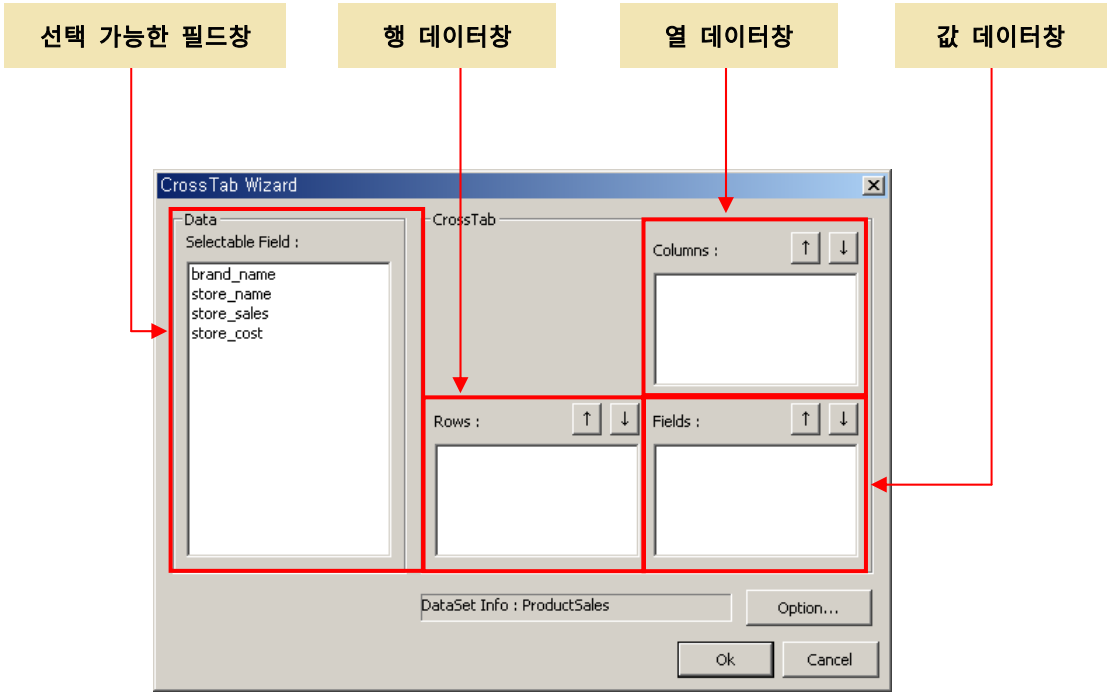
Style	MenuItem에 적용할 OZ Style을 설정합니다.
Shortcut	MenuItem의 단축키를 설정합니다.
EnableEventHook	EventHook 이벤트 발생 여부를 설정합니다.

CrossTab 마법사

CrossTab에 데이터를 표시하기 위해 CrossTab 마법사를 이용합니다. CrossTab 마법사를 실행하기 위해서는 먼저 Board에 CrossTab를 추가하고, CrossTab에 표시할 ODI 파일과 데이터 셋을 설정하여야 합니다.

새로운 프로젝트에 Ⅵ.장에서 만든 'Foodmart.odi' 파일과 CrossTab을 추가하고 CrossTab의 속성 중 'ODIKey'는 'Foodmart'로 'DataSet'은 'ProductSales'로 설정합니다. CrossTab을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Wizard]를 클릭하여 CrossTab 마법사를 실행합니다.

CrossTab 마법사는 아래 그림과 같이 CrossTab에 연동된 DataSet의 필드를 리스트 형태로 나타낸 '선택 가능한 필드창', 행에 표시될 데이터를 설정하는 '행 데이터창', 열에 표시될 데이터를 설정하는 '열 데이터창', 값으로 표시될 데이터를 설정하는 '값 데이터창'으로 구성되어 있습니다.



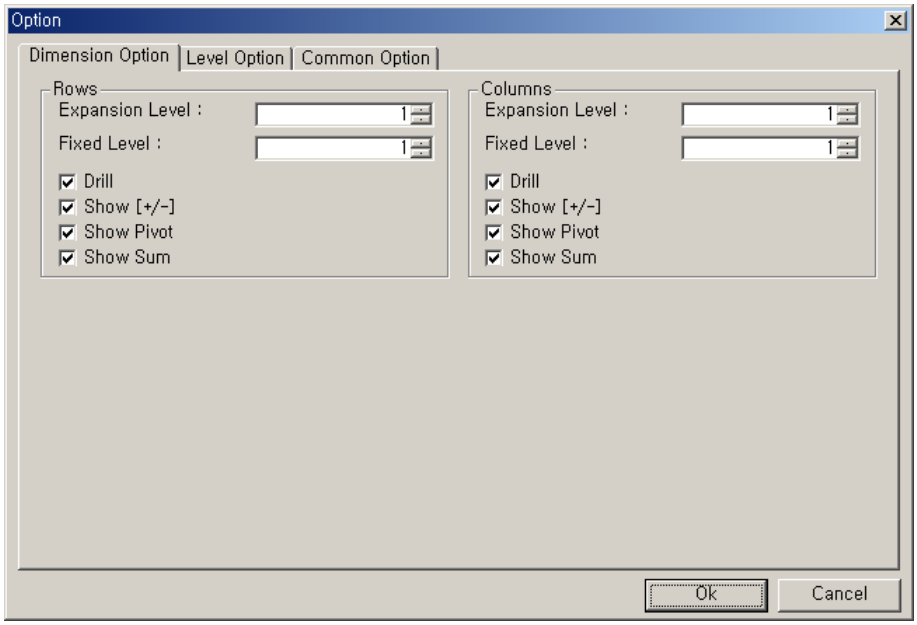
행, 열, 값으로 설정할 데이터를 '선택 가능한 필드창'에서 마우스로 선택한 후 해당 영역에 Drag & Drop하여 설정하고, 설정한 데이터 필드를 해제하려면 해제할 데이터 필드를 선택한 후 데이터 필드창으로 다시 Drag & Drop하거나 키보드의 [Delete] 버튼을 클릭하여 해제합니다.

■ Option

[Option...] 버튼을 클릭하면 CrossTab의 축 옵션과 드릴다운, 피봇팅 기능 들을 설정할 수 있는 다이얼로그가 나타납니다.

- Dimension Option

[Dimension Option] 탭에서는 CrossTab의 드릴다운, 확장 기호, 합계 표시 등을 설정합니다.



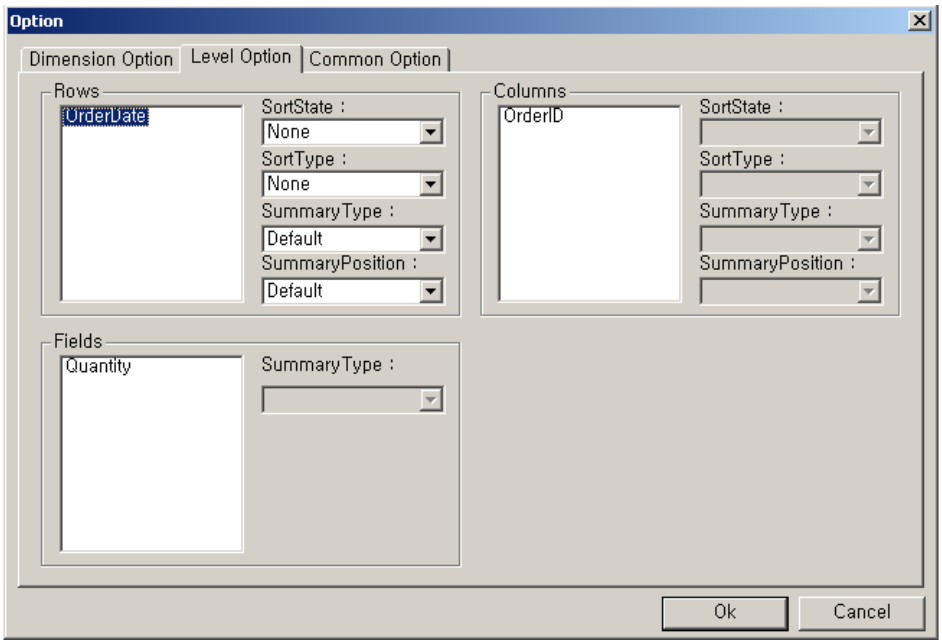
각 옵션과 기능은 아래와 같습니다.

Option	Description
Expansion Level	CrossTab을 화면에 표시할 때 기본적으로 확장하여 표시할 단계를 설정합니다. Expansion Level은 행 또는 열에 설정된 데이터 필드 수만큼 값을 증가하여 설정할 수 있습니다. 예를 들어 열 데이터창에 표시할 필드를 3개 설정하였을 경우 Expansion Level은 최대 3까지 설정할 수 있으며, 만일 2로 설정하면 CrossTab이 화면에 표시될 때 2단계까지 자동으로 확장하여 표시됩니다.

Fixed Level	고정하여 표시할 단계를 설정합니다. 고정할 단계를 설정하면 고정한 단계 이상으로 드릴업할 수 없습니다. Fixed Level은 행 또는 열에 설정된 데이터 필드 수만큼 값을 증가하여 설정할 수 있습니다. 예를 들어 열 데이터창에 표시할 필드를 3개 설정하였을 경우 Fixed Level은 최대 3까지 설정할 수 있으며, 만일 2로 설정하면 CrossTab에서 2단계까지는 고정되어 표시되기 때문에 그 이상 드릴업할수 없습니다.
Drill	드릴다운 또는 드릴업 여부를 설정합니다. 확장 기호를 더블클릭하여 드릴다운 또는 드릴업합니다.
Show [+/-]	드릴다운(+) 또는 드릴업(-)할 확장 기호의 표시 여부를 설정합니다.
Show Pivot	피봇팅 표시 여부를 설정합니다.
Show Sum	합계 표시 여부를 설정합니다.

- Level Option

[Level Option] 탭에서는 '선택 가능한 필드창'에서 설정한 필드의 정렬 상태와 정렬 타입, 써머리 타입을 설정합니다.



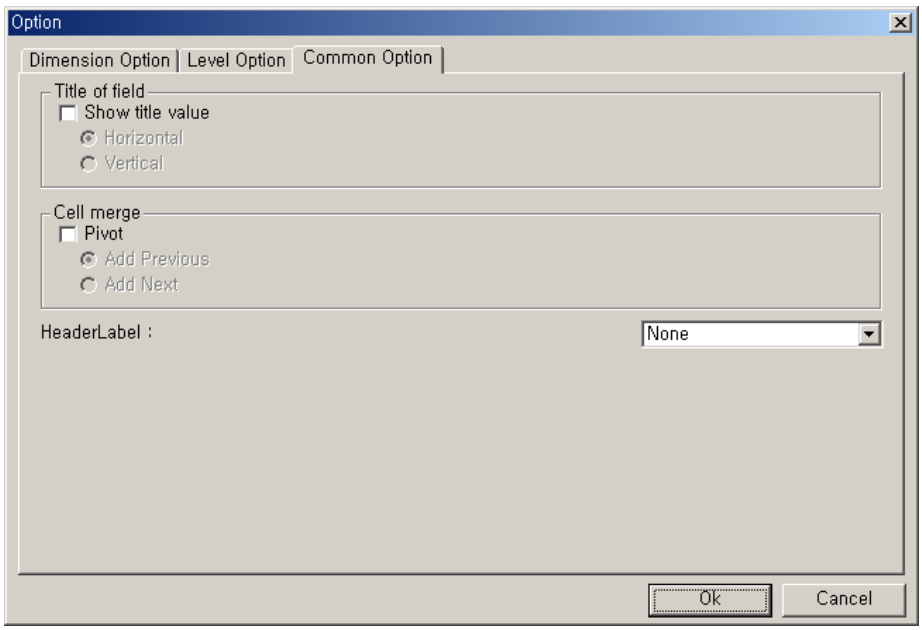
각 옵션과 기능은 아래와 같습니다.

Option	Description

SortState	선택된 Pivot의 정렬 상태를 설정합니다. 정렬 상태는 None(정렬안함), Ascending(오름차순), Descending(내림차순)으로 설정할 수 있으며, 기본 값은 "None"입니다.
SortType	선택된 Pivot를 정렬할 때 정렬할 데이터 타입을 설정합니다. 정렬 타입은 None(지정안함), Text(문자), Numeric(숫자)로 설정할 수 있으며. 기본 값은 "None"입니다.
Summary Type	선택된 Pivot의 써머리 타입을 설정합니다. 써머리 타입은 Default(기본), Sum(합계), Max(최대), Min(최소), Avg(평균), Cnt(개수)로 설정할 수 있으며. 기본 값은 "Default "입니다.
SummaryPosition	선택된 Pivot의 써머리 위치를 설정합니다. 써머리 위치는 Default(기본), Before(피봇 전), After(피봇 후), Hide(감추기)로 설정할 수 있으며 기본 값은 "Default" 입니다. ※ 참고사항 : Default로 설정하면 행 또는 열의 "합계 보이기" 옵션 값에 따라 써머리의 표시 여부가 결정됩니다. 피봇 전 또는 후로 설정하면 써머리 위치를 피봇의 전 또는 후에 표시하며, 감추기로 설정하면 써머리를 표시하지 않습니다.

- Common Option

  [Common Option] 탭에서는 '선택 가능한 필드창'에서 설정한 필드의 제목 표시 여부와 피봇팅 기능을 설정합니다.

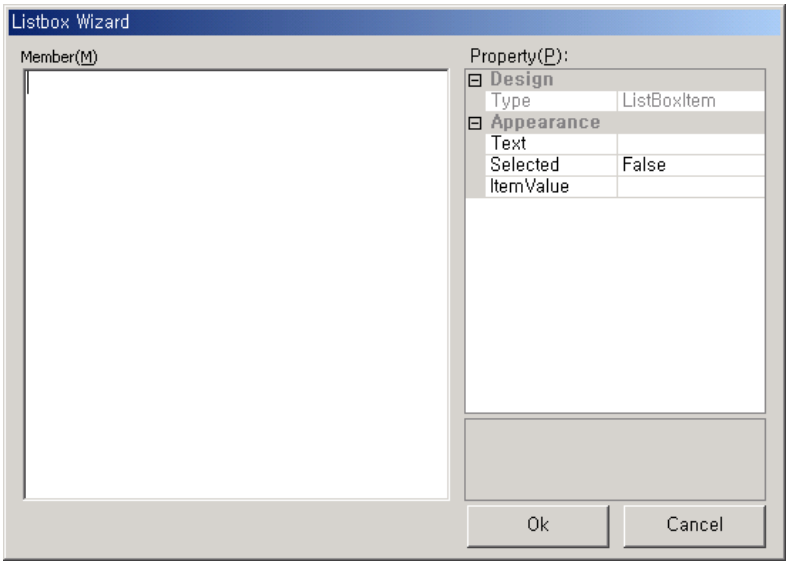


Option	Description
Show Title Value	'선택 가능한 필드창'에 설정한 필드의 제목 표시 여부를 설정하며, 제목 표시 방향을 수평 또는 수직으로 설정합니다.
Pivot	축 이동이 가능한 피봇팅 기능 사용 여부를 설정하며, 축 이동시 해당 항목의 앞쪽으로 피봇팅을 할 경우 'Add Previous' 옵션으로, 해당 항목의 뒤쪽으로 피봇팅을 할 경우 'Add Next' 옵션으로 설정합니다.

ListBox 마법사

ListBox에 아이템을 표시하기 위해 ListBox 마법사를 이용합니다. ListBox 마법사를 실행하기 위해서는 먼저 Board에 ListBox를 추가하고, ListBox를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Wizard]를 클릭하여 ListBox 마법사를 실행합니다.

ListBox 마법사는 아래 그림과 같이 ListBox의 아이템을 나타내는 화면 왼쪽의 'Member창'과 각각 아이템의 속성을 설정하는 화면 오른쪽의 '속성창'으로 구성되어 있습니다.



Member

ListBox의 멤버 즉, 아이템을 입력하여 추가합니다.

Property

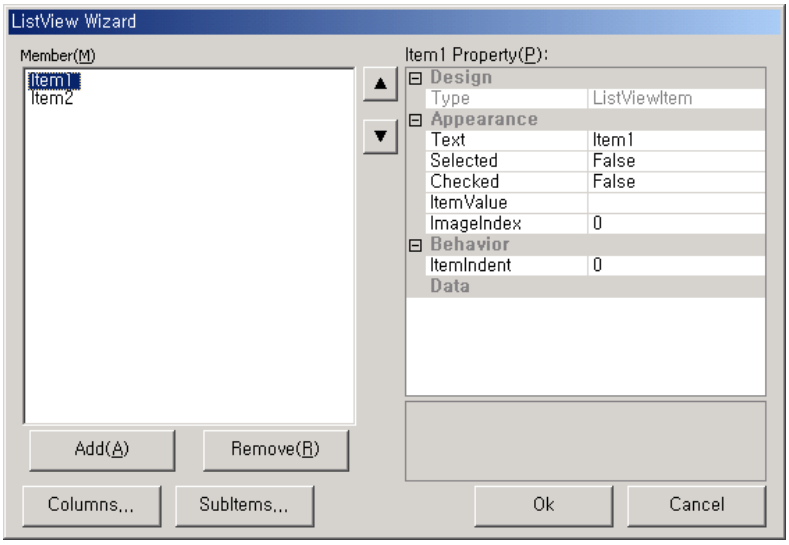
해당 아이템 즉, ListBoxItem의 속성을 설정합니다.

Property	Description
Type	아이템의 컴포넌트 타입입니다. ListBox 아이템의 컴포넌트 타입은 ListBoxItem으로 설정됩니다.
Text	화면에 보이는 ListBoxItem의 텍스트를 설정합니다.
Selected	ListBoxItem의 선택 여부를 설정합니다. 속성 값을 'True'로 설정한 ListBoxItem이 선택 상태로 표시됩니다. ListBox 별로 하나의 선택 상태로 설정할 수 있습니다.
ItemValue	ListBoxItem의 값을 설정합니다.

ListView 마법사

ListView에 아이템, 컬럼, 서브 아이템을 표시하기 위해 ListView 마법사를 이용합니다. ListView 마법사를 실행하기 위해서는 먼저 Board에 ListView를 추가하고, ListView를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Wizard]를 클릭하여 ListView 마법사를 실행합니다.

ListView 마법사는 아래 그림과 같이 ListView의 아이템을 나타내는 화면 왼쪽의 'Member창'과 각각 아이템의 속성을 설정하는 화면 오른쪽의 '속성창'으로 구성되어 있습니다.



[Add] 버튼을 클릭하여 ListView의 아이템을 추가하고, [Columns] 버튼을 ListView의 컬럼을 추가합니다. 아이템을 삭제할 경우에는 [Remove] 버튼을 클릭합니다. 아이템별로 서브 아이템을 만들 경우에는 해당 아이템을 클릭한 후 [SubItems] 버튼을 클릭하여 서브 아이템을 추가합니다.

Member

ListView에 추가된 아이템을 표시합니다.

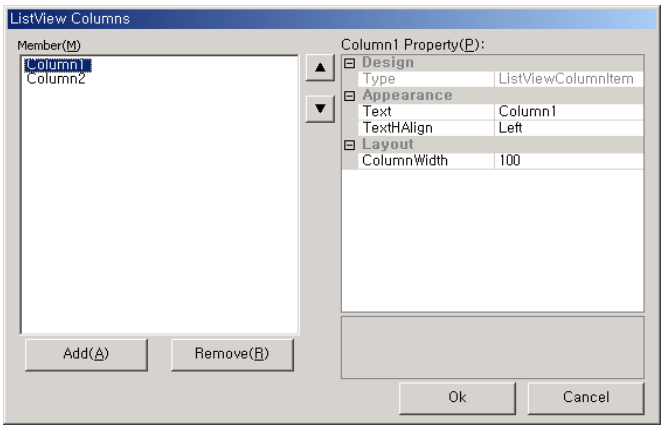
Property

선택한 아이템의 속성을 설정합니다.

Property	Description
Type	아이템의 컴포넌트 타입입니다. ListView 아이템의 컴포넌트 타입은 ListViewItem으로 설정됩니다.
Text	화면에 보이는 아이템의 텍스트를 설정합니다.
Selected	아이템의 선택 여부를 설정합니다. 속성 값을 'True'로 설정한 아이템이 선택된 상태로 표시됩니다. ListView의 속성 중 'HideSelection' 속성 값을 'False'로 설정하여야 제대로 동작합니다.
Checked	아이템의 체크 여부를 설정합니다. 이 옵션은 체크 박스 형태로 나타낸 ListView 즉, ListView 속성 중 'CheckBoxes' 속성 값을 'True'로 설정한 경우에 적용됩니다.
ItemValue	아이템의 값을 설정합니다.
ImageIndex	아이템 앞에 표시할 이미지의 인덱스를 설정합니다. 이 옵션은 ListView에 ImageList 컴포넌트가 설정되어 있는 경우에 가능합니다.
ItemIndent	아이템에 표시할 텍스트의 들여쓰기 간격을 설정합니다. 1은 10픽셀에 해당합니다.

컬럼 추가

[Columns] 버튼을 ListView의 컬럼을 추가할 수 있는 다이얼로그가 표시됩니다.



- Member

  ListView에 추가된 컬럼을 표시합니다.

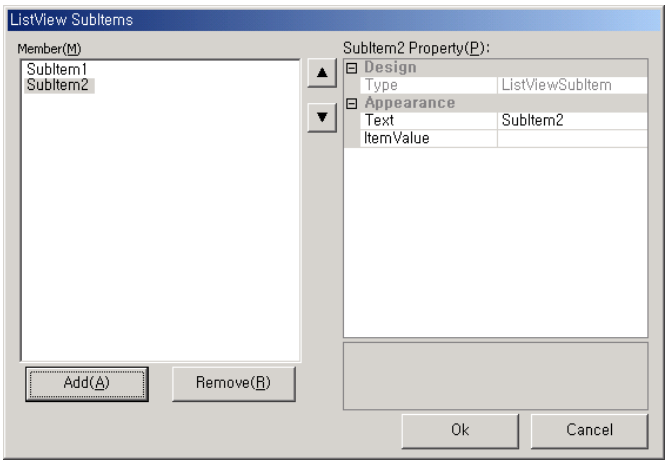
- Property

  선택한 컬럼의 속성을 설정합니다.

Property	Description
Type	컬럼의 컴포넌트 타입입니다. 컬럼의 컴포넌트 타입은 ListViewColumnItem으로 설정됩니다.
Text	화면에 보이는 컬럼의 텍스트를 설정합니다.
TextHAlign	컬럼의 가로 정렬 방식을 설정합니다.
ColumnWidth	컬럼의 너비를 설정합니다.

서브 아이템 추가

[SubItems] 버튼을 클릭하면 ListView 아이템의 서브 아이템을 추가할 수 있는 다이얼로그가 표시됩니다.



- Member

  ListView에 추가된 서브 아이템을 표시합니다.

  서브 아이템을 추가된 컬럼 개수보다 더 많이 추가하여도 화면에는 표시되지 않습니다.

■ Property

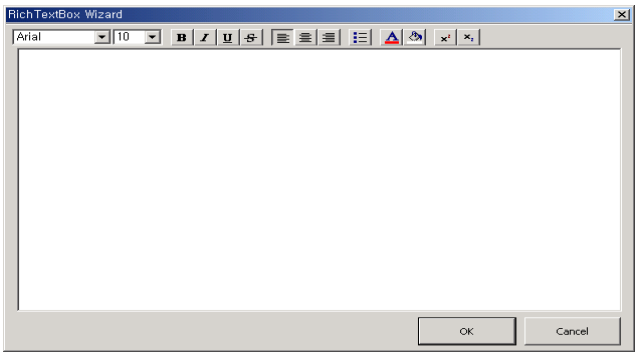
선택한 서브 아이템의 속성을 설정합니다.

Property	Description
Type	서브 아이템의 컴포넌트 타입입니다. 서브 아이템의 컴포넌트 타입은 ListViewSubItem으로 설정됩니다.
Text	화면에 보이는 서브 아이템의 텍스트를 설정합니다.
ItemValue	서브 아이템의 값을 설정합니다.

RichTextBox 마법사

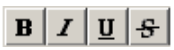
RichTextBox에 표시할 텍스트를 나타내기 위해 RichTextBox 마법사를 이용합니다. RichTextBox 마법사를 실행하기 위해서는 먼저 Board에 RichTextBox를 추가하고, RichTextBox를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Wizard]를 클릭하여 RichTextBox 마법사를 실행합니다.

RichTextBox 마법사는 아래 그림과 같이 텍스트의 포맷을 설정할 수 있는 '툴바'와 텍스트를 입력하는 '텍스트창'으로 구성되어 있습니다.



툴바

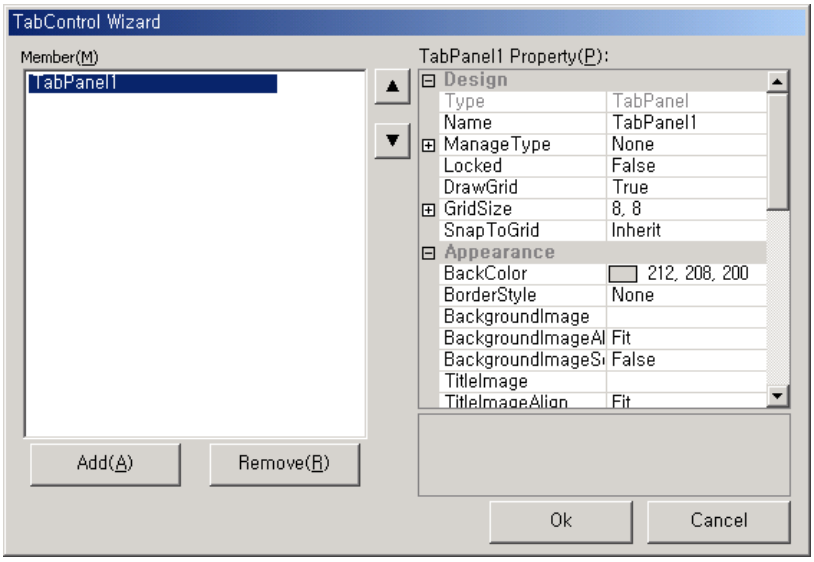
텍스트창에 입력한 텍스트의 글꼴, 크기, 타입, 정렬 방식, 글자색, 배경색 등을 설정합니다.

Icon	Description
	글꼴, 글자 크기를 설정합니다.
	글자 타입을 굵게, 기울임꼴, 밑줄, 취소선으로 설정합니다.
	문자열을 왼쪽, 가운데, 오른쪽으로 정렬합니다.
	글머리 기호를 추가합니다.
	글자색, 배경색을 설정합니다.
	글자를 위첨자, 아래첨자로 표시합니다.

TabControl 마법사

TabControl을 TabPanel을 추가하기 위해서 TabControl 마법사를 이용합니다. TabControl 마법사를 실행하기 위해서는 먼저 Board에 TabControl을 추가하고, TabControl을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Wizard]를 클릭하여 TabControl 마법사를 실행합니다.

TabControl 마법사는 아래 그림과 같이 추가된 TabPanel을 나타내는 화면 왼쪽의 'Member창'과 해당 TabPanel의 속성을 설정하는 화면 오른쪽의 '속성창'으로 구성되어 있습니다.



TabControl 마법사에는 4개의 설정 버튼이 있으며 각 버튼의 기능은 아래와 같습니다.

Button	Description
Add	TabPanel을 추가합니다.
Remove	선택한 TabPanel을 삭제합니다.
▲	선택된 TabPanel의 인덱스를 한 단계 앞으로 이동시킵니다.
▼	선택된 TabPanel의 인덱스를 한 단계 뒤로 이동시킵니다.

Table 마법사

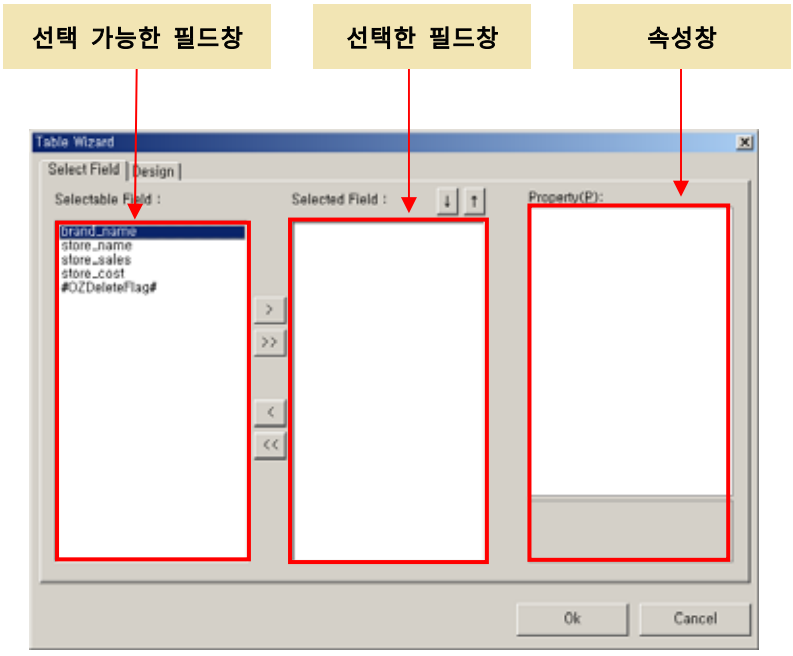
Table에 데이터를 표시하기 위해 Table 마법사를 이용합니다. Table 마법사를 실행하기 위해서는 먼저 Board에 Table을 추가하고, Table에 표시할 ODI 파일과 데이터 셋을 설정하여야 합니다.

새로운 프로젝트에 Ⅵ.장에서 만든 'Foodmart.odi' 파일과 Table을 추가하고 Table의 속성 중 'ODIKey'는 'Foodmart'로 'DataSet'은 'ProductSales'로 설정합니다. Table을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Wizard]를 클릭하여 Table 마법사를 실행합니다.

Table 마법사는 Table에 표시할 필드를 설정하는 "Select Field" 탭과 Table의 타이틀 셀과 데이터 셀을 디자인할 수 있는 "Design" 탭으로 구성되어 있습니다.

Select Field

Table에 표시할 DataSet의 필드를 추가하고, 각 컬럼의 속성을 설정합니다.
Select Field 탭은 아래 그림과 같이 Table에 연동된 DataSet의 필드를 리스트 형태로 나타낸 ''선택 가능한 필드창'', Table에 표시할 필드를 추가하는 "선택한 필드창", 각 컬럼의 속성을 설정하는 '속성창'으로 구성되어 있습니다.



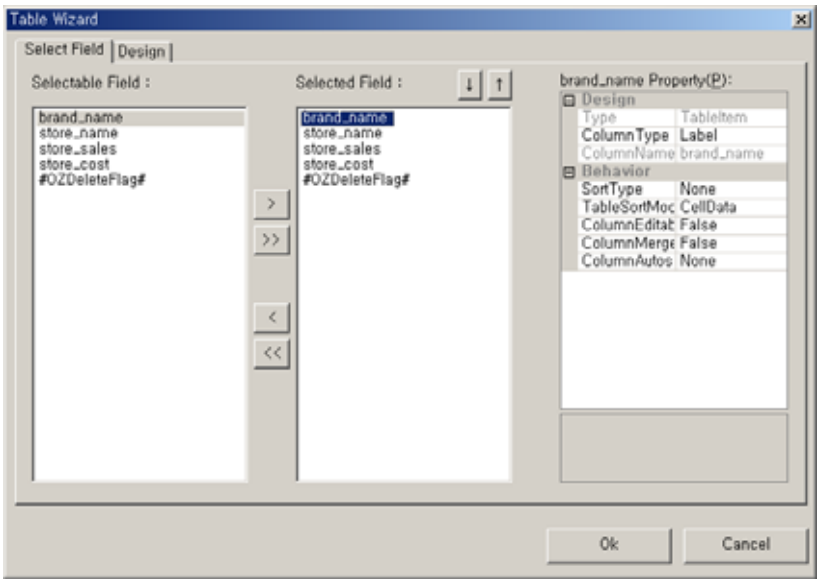
'선택 가능한 필드창''에서 '#OZDeleteFlag#'는 Table과 연동된 DataSet의 필드가 아니라 Table 마법사에서 기본적으로 추가해 주는 플래그로, 선택한 Row에 대해 데이터의 삭제 여부를 설정할 수 있습니다.

Table 마법사에는 6개의 설정 버튼이 있으며 각 버튼의 기능은 아래와 같습니다.

Button	Description
	'선택 가능한 필드창'의 모든 필드를 '테이블 컬럼창'으로 추가하여 Table에 표시할 필드를 설정합니다.
	'선택 가능한 필드창'에서 선택된 필드를 '테이블 컬럼창'으로 추가하여 Table에 표시할 필드를 설정합니다.
	'선택한 필드창'에서 선택된 필드를 삭제합니다. 즉, 선택한 필드를 Table에 표시하지 않습니다.
	'선택한 필드창'에 추가된 모든 필드를 삭제합니다. 즉, 모든 필드를 Table에 표시하지 않습니다.
	'선택한 필드창'에 선택된 필드를 한 단계 위로 올립니다. 즉, Table에 표시되는 필드의 순서를 한 단계 앞으로 이동시킵니다.
	'선택한 필드창'에 선택된 필드를 한 단계 아래로 내립니다. 즉, Table에 표시되는 필드의 순서를 한 단계 뒤로 이동시킵니다.

Table에 표시할 필드를 추가하는 방법은 설정 버튼을 이용하거나 설정할 데이터를 '테이블 컬럼창'에 Drag & Drop하여 추가할 수도 있습니다.

'선택한 필드창'에 표시할 필드를 추가하면 '속성창'이 활성화되어 해당 필드의 속성을 설정할 수 있습니다.



각 속성과 설명은 아래와 같습니다.

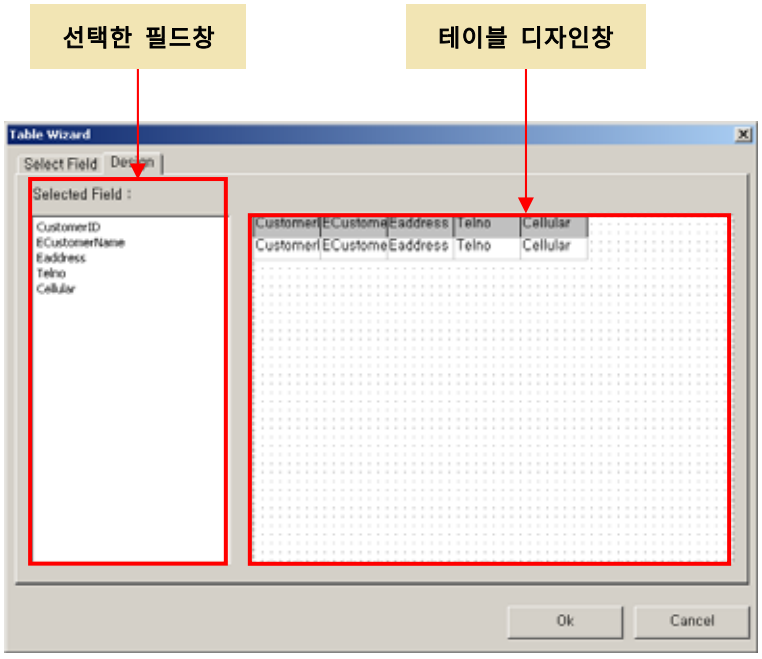
Property	Description
Type	아이템의 컴포넌트 타입입니다. Table 아이템의 컴포넌트 타입은 TableItem으로 설정됩니다.
ColumnType	컬럼의 타입 즉, 표시할 모양을 Label, PictureBox, Button, CheckBox, RadioButton, ComboBox, 사용자 정의형 등으로 설정합니다.
ColumnName	컬럼명을 표시합니다.
SortType	실행시 컬럼의 제목을 클릭하면 정렬하여 데이터를 표시할지 여부와 정렬 방식을 설정합니다. 정렬하지 않을 경우 'None'으로 설정하고, 정렬할 경우에는 'Text' 또는 'Numeric'로 설정합니다.
TableSortMode	테이블의 정렬 모드를 설정합니다. 'CellData'로 설정하면 Table의 Cell에 표시된 값을 기준으로 현재 Page에 대해 정렬하고, 'DataSet'으로 설정하면 Table에 설정된 DataSet을 기준으로 정렬합니다. ※ 제약사항 : 'DataSet'으로 설정 시에는 Table의 속성 중 'ApplyToDataSet' 속성 값과 'SynchronizeToDataSet' 속성 값을 "True"로 설정해야 적용됩니다.
ColumnEditable	실행시 데이터 편집 가능 여부를 설정합니다.
ColumnMerge	실행시 컬럼 데이터에 같은 값이 있을 경우 셀을 병합하여 표시할지 여부를 설정합니다.

ColumnAutosize	컬럼에 표시할 문자열의 길이가 디자인된 컬럼의 크기보다 클 경우 컬럼의 크기 조절 옵션을 설정합니다. 'None'은 컬럼 크기를 조정하지 않는 옵션이며, 'FixWidth'는 컬럼의 너비는 고정하고 높이를 문자열이 모두 표시되는 크기로 조절하는 옵션이며, 'Auto'는 컬럼의 너비를 문자열이 모두 표시되는 크기로 조절하는 옵션이며, 'FixWidthNotShrink'는 컬럼의 너비는 고정하고 높이를 문자열이 모두 표시되는 크기로 조절하되 높이에 약간의 여백을 두어 조절하는 옵션입니다.

Design

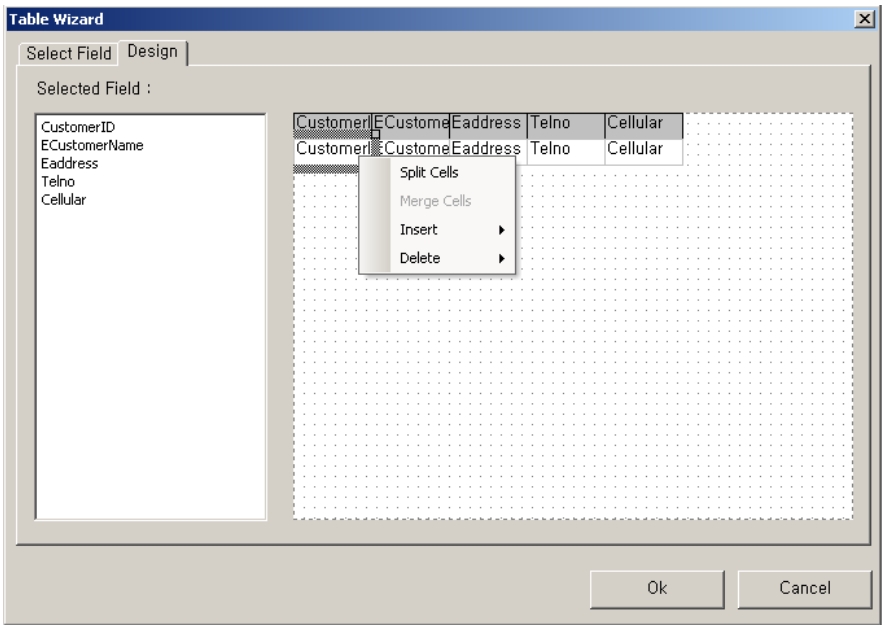
테이블의 타이틀 셀과 데이터 셀을 병합/ 분할할 수 있고, 행과 열을 추가/삭제하여 테이블을 디자인할 수 있습니다.

Design 탭은 아래 그림과 같이 Select Field 탭에서 선택한 필드를 표시하는 ''선택한 필드창'', Table의 셀을 디자인하는 창으로 구성되어 있습니다.



'선택한 필드창'에서 오른쪽 테이블을 디자인하는 영역으로 표시할 데이터를 Drag & Drop하여 설정할 수 있으며 셀을 선택한 후 마우스 오른쪽 버튼을 클릭하면 셀을 분할/병합하고 추가/삭제

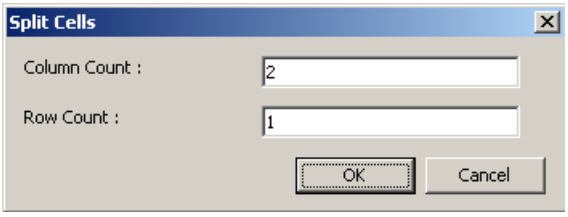
할 수 있는 팝업 메뉴가 나타납니다.



■ Split Cells

[Split Cells] 메뉴는 선택한 셀을 분할합니다. 테이블의 셀이 한 개만 선택되었을 때 활성화 됩니다.

[Split Cells] 메뉴를 클릭하면 아래와 같이 셀 분할 다이얼로그가 나타납니다.



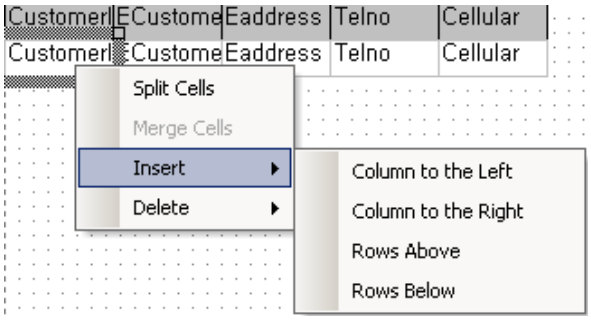
분할할 열 개수와 행 개수를 입력한 후 [OK] 버튼을 클릭하면 셀이 분할됩니다.

■ Merge Cells

[Merge Cells] 메뉴는 선택한 셀을 병합합니다. 테이블의 셀이 병합할 수 있는 형태 즉 행 방향 또는 열 방향으로 두 개 이상 선택된 상태에서만 활성화됩니다.

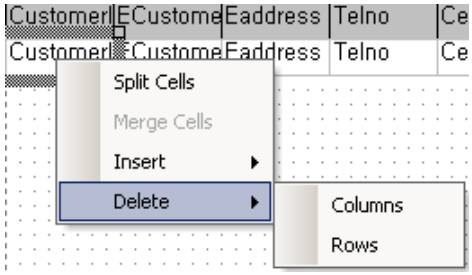
■ Insert

[Insert] 메뉴는 선택한 셀의 왼쪽/오른쪽/위쪽/아래쪽으로 열이나 행을 삽입합니다.



- Delete

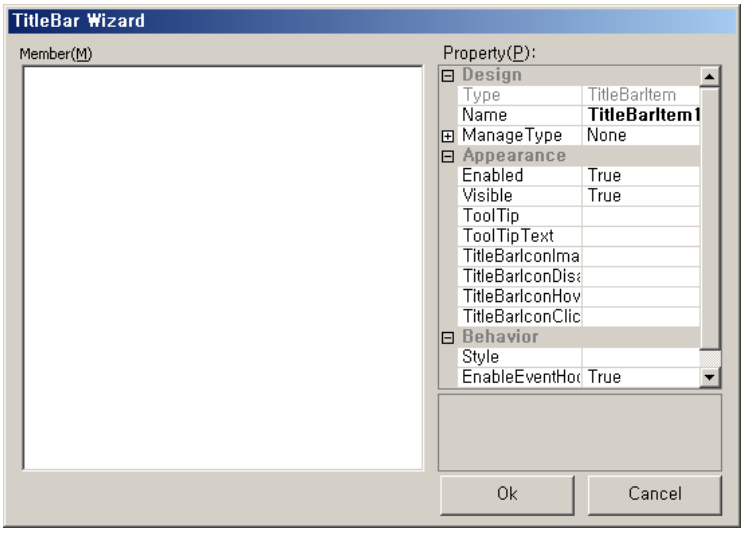
  [Delete] 메뉴는 선택한 셀의 열 단위 또는 행 단위로 행이나 열을 삭제합니다.



TitleBar 마법사

TitleBar의 타이틀바 아이템을 추가하고 표시하기 위해 TitleBar 마법사를 이용합니다. TitleBar 마법사를 실행하기 위해서는 먼저 Document에 TitleBar를 추가하고, 'Invisibles 트리창'에서 TitleBar를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Wizard]를 클릭하여 TitleBar 마법사를 실행합니다.

TitleBar 마법사는 아래 그림과 같이 TitleBar의 타이틀바 아이템을 추가하는 화면 왼쪽의 'Member창'과 각각 메뉴 아이템의 속성을 설정하는 화면 오른쪽의 '속성창'으로 구성되어 있습니다.



Member

TitleBar의 멤버 즉, 타이틀바 아이템을 입력하여 추가합니다.

Property

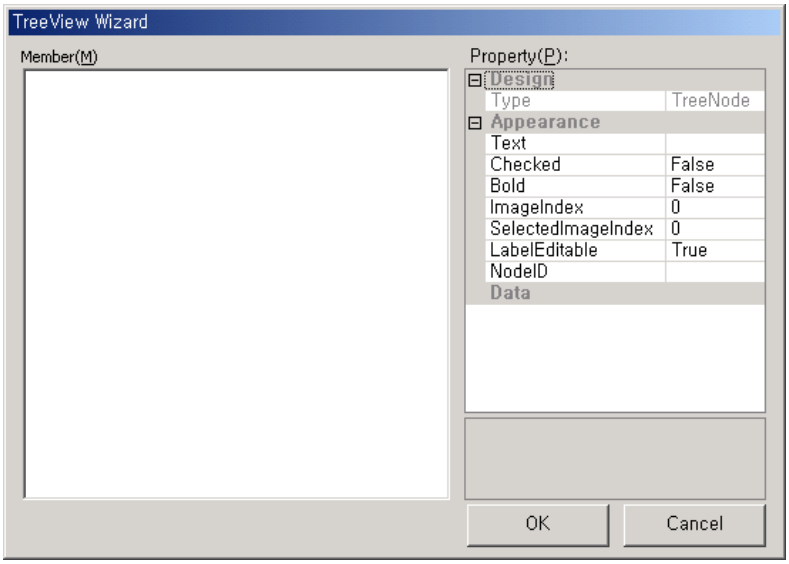
해당 아이템 즉, TitleBarItem의 속성을 설정합니다.

Property	Description
TitleBarIconImage	타이틀바 아이콘에 표시할 이미지 파일을 설정합니다.
TitleBarIconDisableImage	타이틀바 아이템의 Enabled 속성이 false인 경우 표시할 이미지 파일을 설정합니다.
TitleBarIconHoverImage	마우스 포인터가 타이틀바 아이템 위에 있을 경우 표시할 이미지 파일을 설정합니다.
TitleBarIconClickImage	타이틀바 아이템을 마우스로 클릭하였을 경우 표시할 이미지 파일을 설정합니다.

TreeView 마법사

TreeView에 아이템을 표시하기 위해 TreeView 마법사를 이용합니다. TreeView 마법사를 실행하기 위해서는 먼저 Board에 TreeView를 추가하고, TreeView를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Wizard]를 클릭하여 TreeView 마법사를 실행합니다.

TreeView 마법사는 아래 그림과 같이 TreeView의 아이템을 추가하는 화면 왼쪽의 'Member창'과 각각 아이템의 속성을 설정하는 화면 오른쪽의 '속성창'으로 구성되어 있습니다.



Member

TreeView의 멤버 즉, 아이템을 입력하여 추가합니다.
아이템 추가시에 하위 아이템을 추가할 경우에는 해당 아이템 문자열 앞에 [Tab]키를 입력하면 바로 위에 입력한 아이템의 하위 아이템으로 추가됩니다I

Property

해당 아이템, 즉 TreeNode의 속성을 설정합니다.

Property	Description
Type	아이템의 컴포넌트 타입입니다. TreeView 아이템의 컴포넌트 타입은 TreeNode로 설정됩니다.
Text	화면에 보이는 TreeNode의 텍스트를 설정합니다.
Checked	TreeNode 체크 여부를 설정합니다. 이 옵션은 체크 박스 형태로 나타낸 TreeView 즉, TreeView 속성 중 'CheckBoxes' 속성 값을 'True'로 설정한 경우에 적용됩니다.
Bold	TreeNode의 텍스트를 굵게 표시합니다.
ImageIndex	TreeNode 앞에 표시할 이미지의 인덱스를 설정합니다. 이 옵션은 TreeView에 ImageList 컴포넌트가 설정되어 있는 경우에 가능합니다.
SelectedImageIndex	TreeNode가 선택된 상태일 경우에 표시할 이미지의 인덱스를 설정합니다. 이 옵션은 TreeView에 ImageList 컴포넌트가 설정되어 있는 경우에 가능합니다.
LabelEditable	실행시에 TreeNode 텍스트의 변경 가능 여부를 설정합니다. TreeView의 속성 중 'LabelEdit'의 속성 값이 'True'로 설정된 경우 적용됩니다.
NodeID	TreeNode의 ID를 설정합니다.

Ⅷ. OZ Function Editor

- OZF(OZ Function) 개념
- OZ Function Editor 실행
- OZ Function Editor 화면 구성
- 메뉴바
- 툴바
- OZF 만들기
- OZF 사용하기

본 장에서는 OZF의 개념, OZ Function Editor의 사용 방법, OZF를 프로젝트에 추가하고 사용하는 방법 대해 설명합니다.

OZF(OZ Function) 개념

OZF는 공통적으로 사용되는 함수를 모듈화시킨 OZ Function 파일로 다수의 개발자간 정보 공유를 용이하게 하고 불필요한 코딩 시간을 줄여 개발 기간을 단축시키는 이점이 있습니다. OZF 파일은 OZ Application Designer의 프로젝트에 추가하여 사용하며, OZF에 코딩된 함수는 OZ Application Designer의 전역 함수처럼 사용 가능합니다. OZF 파일은 'OZ Function Editor'를 이용하여 만들거나 수정할 수 있습니다.

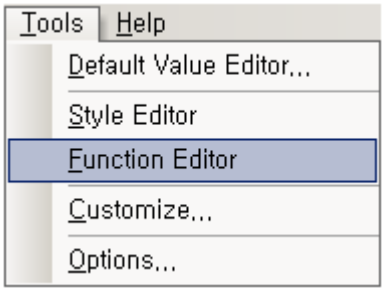
※ 참고사항 : 프로젝트에 추가된 OZF는 프로젝트에 OZF를 추가한 순서대로 초기화됩니다.

OZ Function Editor 실행

OZ Function Editor는 OZ Application Designer의 메뉴바를 통해 실행할 수 있습니다.

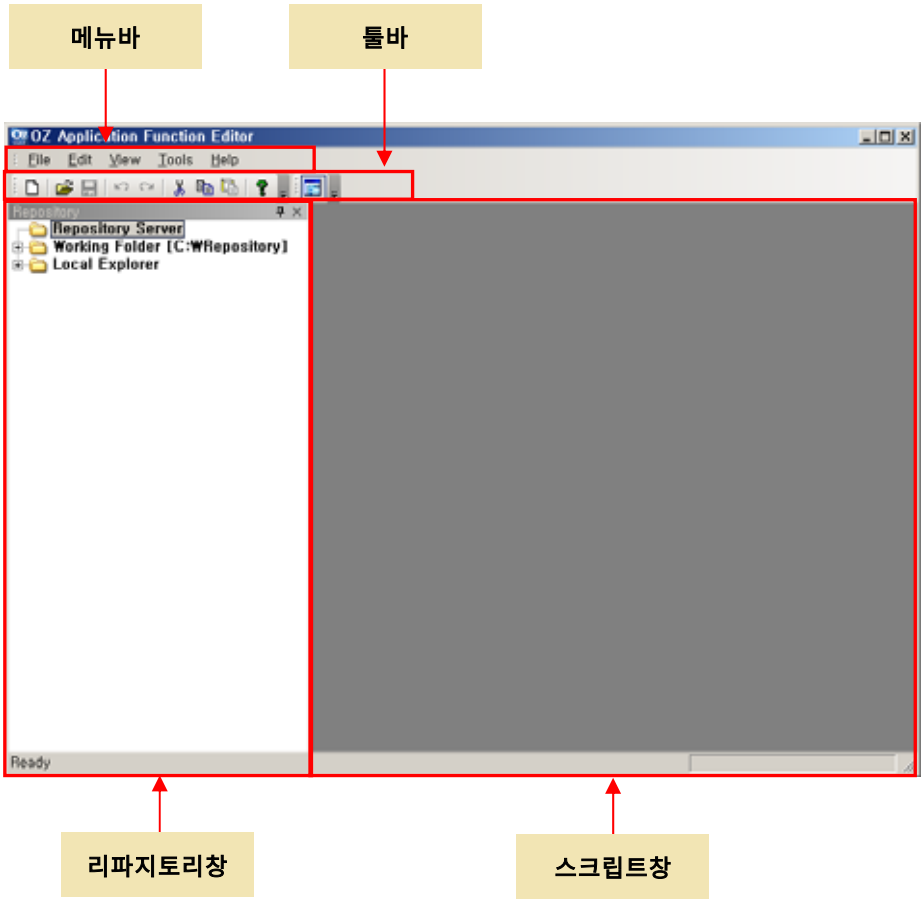
OZ Application Designer의 메뉴바 이용

'OZ Application Designer'을 실행한 후 [Tools] -> [Function Editor] 메뉴를 클릭하여 'OZ Function Editor'를 실행합니다.



OZ Function Editor 화면 구성

다음은 OZ Function Editor의 사용자 인터페이스 화면이며, 각 구성 요소의 이름과 기능은 아래와 같습니다.



Window	Description
메뉴바	메뉴바는 OZ Function Editor의 상단에 풀다운(pull down)형식으로 제공되며, 파일, 편집, 뷰, 도구, 도움말로 구성됩니다.
툴바	OZ Function Editor의 함수 열기, 함수 저장, 잘라내기, 복사하기, 붙여넣기, 실행 취소, 다시 실행, 리파지토리 창 보기 등을 쉽고 빠르게 설정할 수 있도록 아이콘 형태로 제공됩니다.

리파지토리창	로컬 및 서버의 리파지토리 내의 폴더 및 파일을 탐색기 형태로 나타냅니다.
스크립트창	OZF 파일에 포함할 함수를 코딩하거나 코딩된 내용을 표시합니다. 스크립트창은 새로 만들기를 하면 활성화 됩니다.

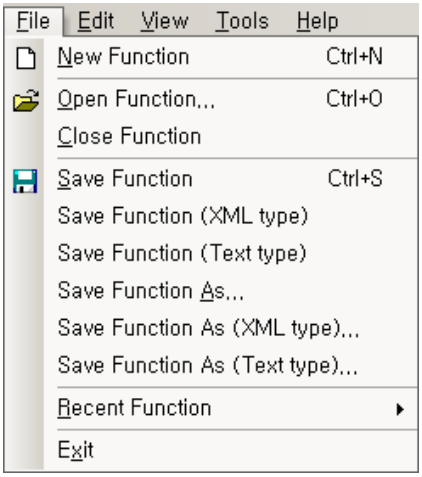
각 항목별 상세 기능과 사용법에 대해서는 다음에 기술되는 각 장에서 설명합니다.

메뉴바

메뉴바는 OZ Function Editor의 상단에 풀다운(pull down)형식으로 제공되며, 파일, 편집, 뷰, 도구, 도움말로 구성됩니다.

파일(File)

메뉴바의 [File] 메뉴를 선택하면 아래와 같은 파일 관련 메뉴가 나타납니다.



- New Function (Ctrl+N)
  새로운 OZF 파일을 만듭니다. [New Function] 메뉴를 클릭하면 스크립트창이 활성화됩니다.

- Open Function (Ctrl+O)
  저장된 OZF 파일을 읽어 옵니다.

- Close Function
  현재 작업 중인 OZF 파일을 닫습니다. 만일 변경된 내용이 있으면 저장 여부를 묻는 다이얼로그가 표시됩니다.

- Save Function (Ctrl+S)
  현재 포맷을 유지하면서 OZF 파일로 저장합니다. 기존에 저장된 파일의 형식이 XML 타입

인 경우 XML 방식으로 저장되고 텍스트 타입인 경우 텍스트 방식으로 저장됩니다. OZF의 기본 저장 타입은 XML 타입입니다.

※ 참고사항 : 현재 저장 중인 포맷 확인

현재 저장 중인 포맷은 OZ Function Editor의 타이틀바에서 확인할 수 있습니다.

만일 현재 저장 중인 포맷이 XML 타입일 경우 OZ Function Editor의 타이틀바에 "파일명(XML type)"로 표시되고,



텍스트 타입일 경우 "파일명(Text type)"으로 표시됩니다.



- Save Function (XML Type)

  현재 작업 중인 OZF 파일을 XML 타입으로 저장합니다.

- Save Function (Text Type)

  현재 작업 중인 OZF 파일을 텍스트 타입의 OZF 파일로 저장합니다. 텍스트 타입이 XML 타입보다 로딩 시간을 단축시키는 이점이 있습니다.

- Save Function As

  현재 포맷을 유지하면서 OZF 파일을 다른 이름으로 저장합니다. 기존에 저장된 파일 형식이 XML 타입인 경우 XML 타입의 다른 이름으로 저장되고 텍스트 타입인 경우 텍스트 타입의 다른 이름으로 저장됩니다. OZF의 기본 저장 타입은 XML 타입입니다.

- Save Functin As (XML Type)

  현재 작업 중인 OZF 파일을 XML 타입의 다른 이름으로 저장합니다.

- Save Functin As (Text Type)

  현재 작업 중인 OZF 파일을 텍스트 타입의 다른 이름으로 저장합니다. 텍스트 타입이 XML 타입보다 로딩 시간을 단축시키는 이점이 있습니다.

- Recent Function

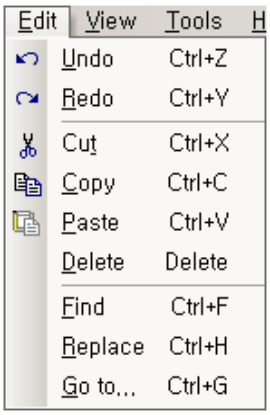
  최근에 열어본 OZF 파일을 선택합니다. 최근에 열어본 OZF 파일 중 최대 4개의 OZF 파일을 서브 메뉴로 표시합니다.

- Exit

  OZ Function Editor를 종료합니다.

편집(Edit)

메뉴바의 [Edit] 메뉴를 선택하면 아래와 같은 편집용 메뉴가 나타납니다.



- Undo (Ctrl+Z)

  바로 전에 입력한 스크립트를 취소하고 이전 상태로 되돌립니다.

- Redo (Ctrl+Y)

  실행 취소했던 작업을 다시 복구하며, 실행 취소한 횟수만큼 복구 가능합니다.

- Cut (Ctrl+X)

  선택한 스크립트를 클립보드로 복사하고 스크립트창에서는 삭제합니다.

- Copy (Ctrl+C)

  선택한 스크립트를 클립보드로 복사합니다.

- Paste (Ctrl+V)

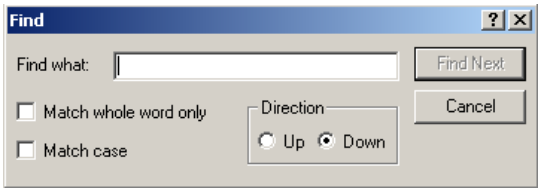
  클립보드의 내용을 스크립트창에 삽입합니다.

- Delete (Delete)

선택한 스크립트를 삭제합니다. [Undo] 메뉴를 클릭하여 삭제한 작업을 취소할 수 있습니다.

- Find (Ctrl+F)

  입력한 문자열을 찾습니다.

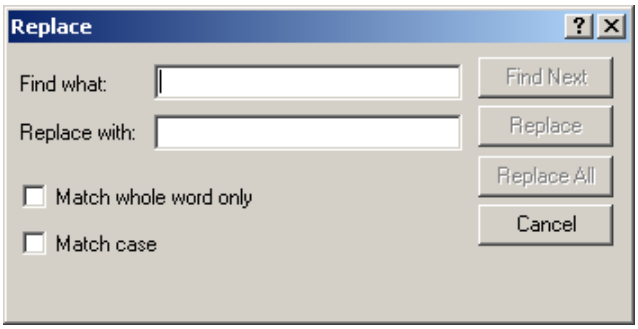


  텍스트 박스에 문자열을 입력하고 찾기 옵션을 설정한 후 [Find Next] 버튼을 클릭합니다. 각 옵션과 기능은 아래와 같습니다.

Option	Description
Match Whole World Only	전체 단어 단위로 찾을지를 설정합니다.
Match Case	대소문자를 구분하여 찾을지를 설정합니다.
Direction	찾을 방향을 위, 아래로 설정합니다.

- Relpace (Ctrl+H)

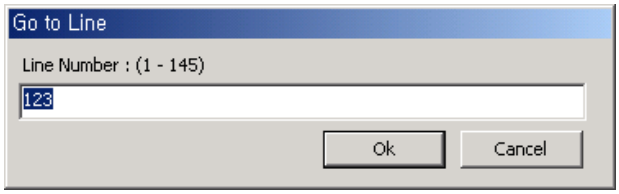
  스크립트 문자열을 변경합니다.



  찾을 내용과 바꿀 내용을 텍스트 박스에 입력하고 옵션을 설정한 후 [Find Next] 또는 [Replace] 또는 [Replace All] 버튼을 이용하여 변경합니다.

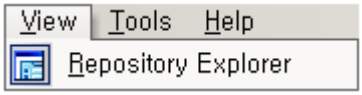
- Go To (Ctrl+G)

  스크립트창의 특정 줄 번호로 편집 커서를 이동합니다. 코딩한 스크립트의 전체 줄수가 다이얼로그에 표시되며 현재 편집 커서가 위치한 줄 번호가 자동으로 표시됩니다. 텍스트 박스에 이동할 줄 번호를 입력한 후 [Ok] 버튼을 클릭하여 편집 커서를 이동시킵니다.



뷰(View)

메뉴바의 [View] 메뉴를 선택하면 아래와 같은 메뉴가 나타납니다.

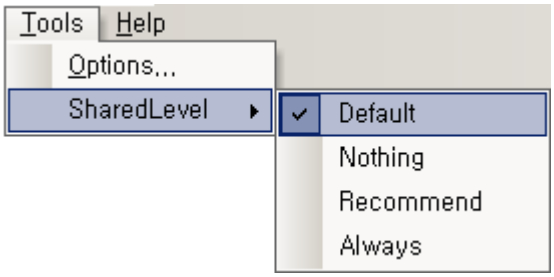


- Repository Explorer

  로컬 및 서버의 리파지토리 내의 폴더 및 파일을 탐색기 형태로 나타내는 리파지토리창을 OZ Function Editor 화면에 표시할지 여부를 설정합니다.

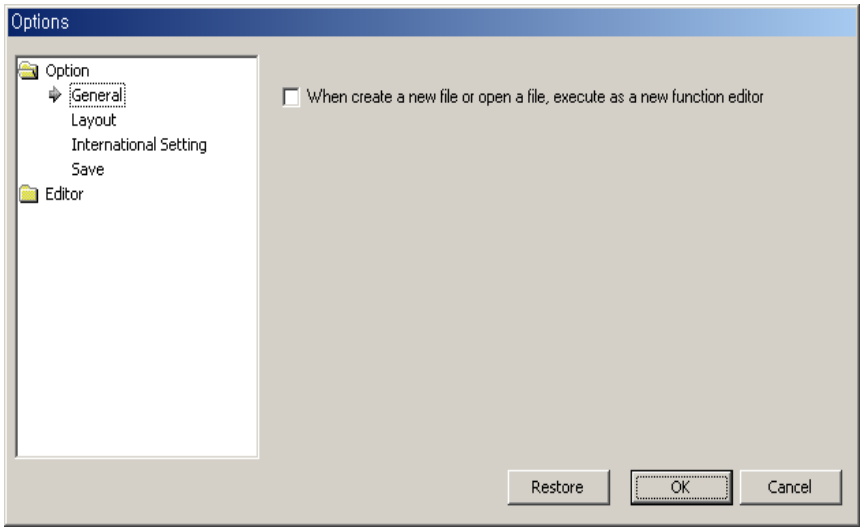
도구(Tools)

메뉴바의 [Tools] 메뉴를 선택하면 아래와 같은 메뉴가 나타납니다. XML 타입의 OZ Function Editor에서만 공유 레벨 설정 기능을 지원합니다.



- Options

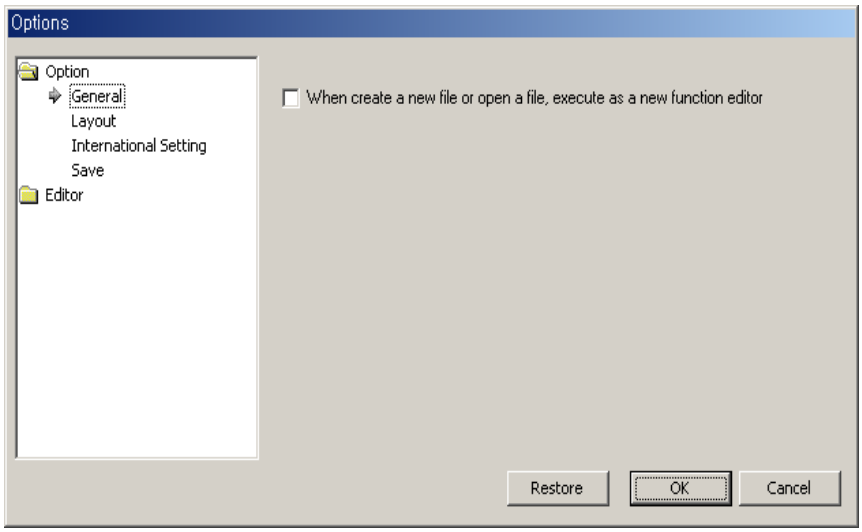
  '옵션 설정 다이얼로그'를 이용하여 OZ Function Editor의 레이아웃, 메뉴바의 언어, 암호화 저장 여부, 스크립트창의 폰트 등을 설정 등을 설정합니다.



변경한 옵션을 저장 및 적용하려면 [Ok] 버튼을 OZ Function Editor 초기에 설정된 기본 옵션으로 적용하려면 [Restore] 버튼을 클릭합니다.

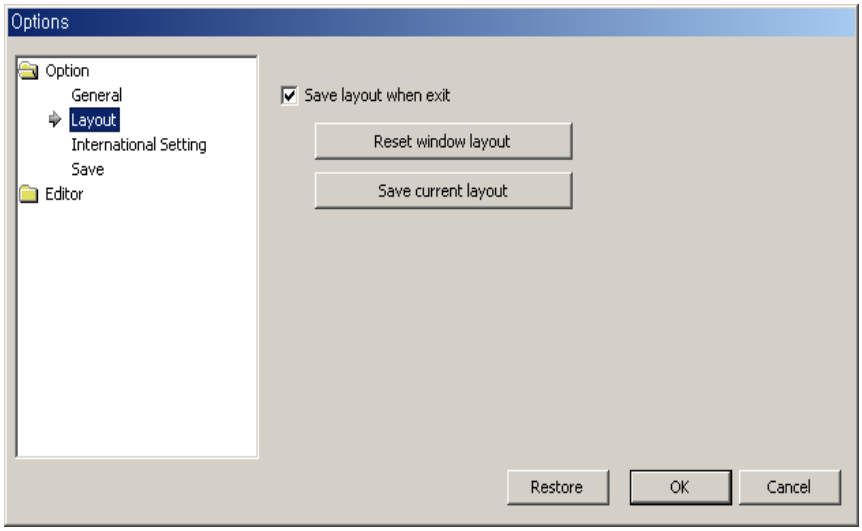
- Option
  - ➢ General

    새로 파일을 만들거나 리파지토리에서 파일을 열었을 때 새로운 OZ Function Editor에서 열지 여부를 설정합니다.



  - ➢ Layout

    OZ Function Editor의 레이아웃을 설정합니다.

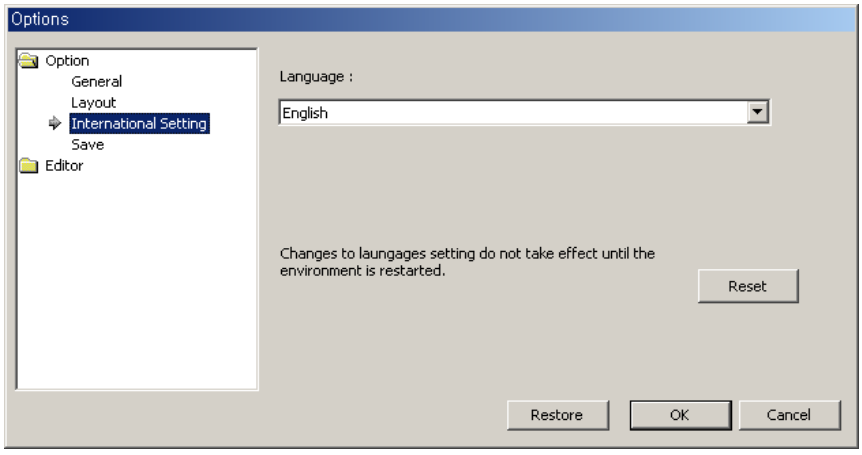


- Save layout when exit

  OZ Function Editor가 종료될 때 OZ Function Editor 레이아웃의 저장 여부를 설정합니다. OZ Function Editor 레이아웃을 기본 값으로 초기화 할 경우에는 [Reset window layout] 버튼을 클릭하고 현재 레이아웃을 저장할 경우에는 [Save current layout] 버튼을 클릭합니다.

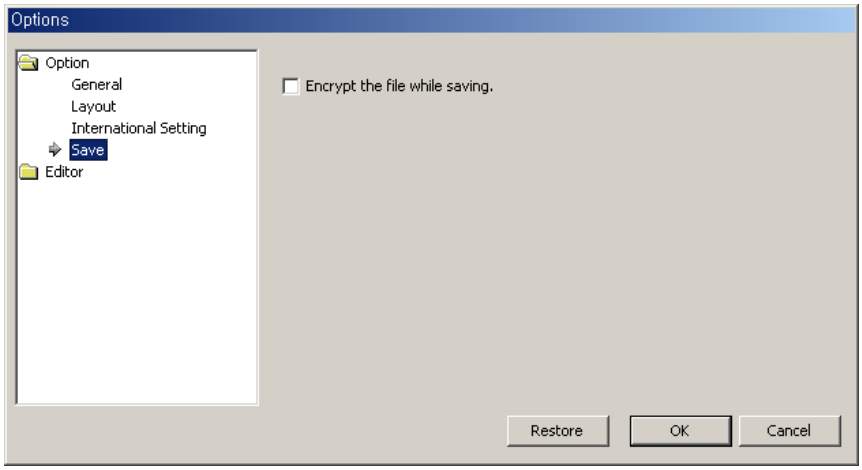
➢ International Setting

메뉴바에 표시할 언어를 설정합니다.



➢ Save

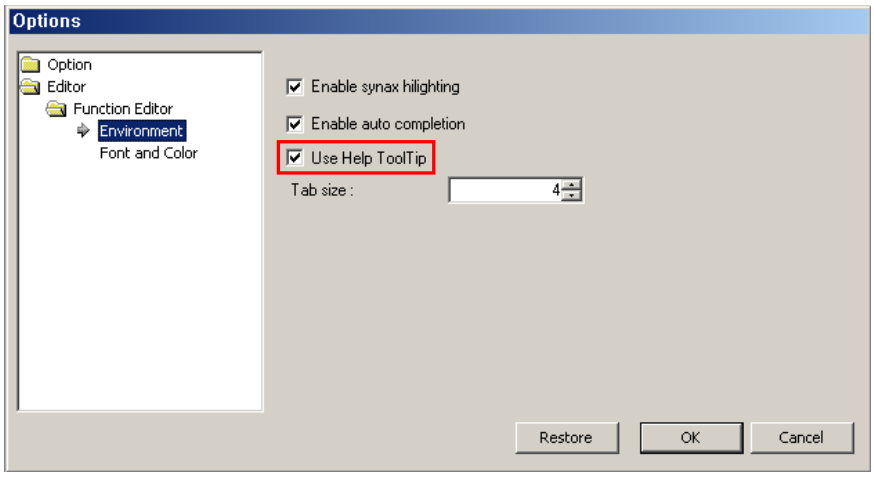
OZF 파일을 저장할 때 암호화하여 저장할지 여부를 설정합니다.



- Editor

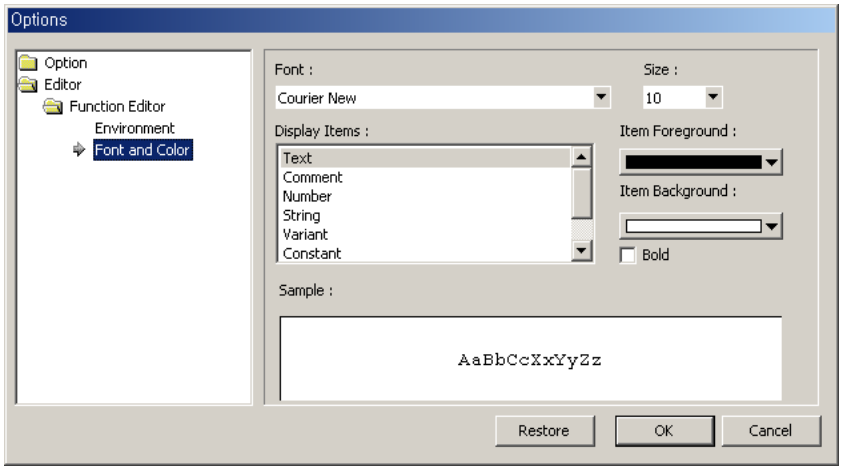
  ➢ Environment : Function Editor

  스크립트창에서 스크립트를 표시하는 텍스트에 대해 구문 강조 여부, 구문 자동 완성 여부, 탭 크기를 설정합니다.



  ➢ Font and Color : Function Editor

  스크립트창에서 스크립트를 표시하는 텍스트의 글꼴, 글꼴 크기, 전경색, 배경색을 설정합니다.



■ Options

OZ Function Editor의 공유 레벨을 설정합니다.

- SharedLevel

➢ Default

OZ Function Editor의 공유 레벨을 설정하지 않습니다. 공유 레벨이 "Default"인 OZ Function Editor를 애플리케이션 디자이너에서 추가할 경우 "SharedFunction" 속성 값이 "False"인 상태로 추가되며, 폼 개발자가 변경할 수 있도록 해당 속성이 활서화됩니다. (기본 값)

➢ Nothing

OZ Function Editor의 공유 레벨을 항상 공유하지 않도록 설정합니다. 공유 레벨이 "Nothing"인 OZ Function Editor를 애플리케이션 디자이너에서 추가할 경우 "SharedFunction" 속성 값이 "False"인 상태로 추가되며, 해당 속성은 변경할 수 없습니다.

➢ Recopmmend

OZ Function Editor의 공유 레벨을 공유함으로 설정합니다. 공유 레벨이 "Recommend"인 OZ Function Editor를 애플리케이션 디자이너에서 추가할 경우 "SharedFunction" 속성 값이 "True"인 상태로 추가되며, 폼 개발자가 변경할 수 있도록 해당 속성이 활성화됩니다.

➢ Always

OZ Function Editor의 공유 레벨을 항상 공유하도록 설정합니다. 공유 레벨이 "Always"인 OZ Function Editor를 애플리케이션 디자이너에서 추가할 경우

"SharedFunction" 속성 값이 "True"인 상태로 추가되며, 해당 속성은 변경할 수 없습니다.

도움말(Help)

메뉴바의 [Help] 메뉴를 선택하면 아래와 같은 메뉴가 나타납니다.



- About OZ Application Function Editor

  OZ Function Editor의 버전 정보를 표시합니다.

툴바

툴바는 OZ Function Editor의 새로 만들기, 열기, 저장, 잘라내기, 복사, 붙이기, 실행취소, 다시 실행, 리파지토리창 보기 등을 쉽고 빠르게 설정할 수 있도록 메뉴바 바로 아래에 아이콘 형태로 제공됩니다.

표준(Standard) 툴바

파일 관련 기능과 잘라내기, 복사 등 일반적인 기능을 실행합니다.



Icon	Description
	New Function, Open Function, Save Function
	Undo, Redo, Cut, Copy, Paste
	About OZ Application Function Editor

뷰(View) 툴바

리파지토리창의 표시 여부를 설정합니다.



Icon	Description
	View Repository

OZF 만들기

OZ Function Editor를 실행한 후 [File] 메뉴의 [New Function] 메뉴를 클릭하거나 툴바의 새로 만들기 아이콘()을 클릭하여 스크립트창을 활성화한 후 OZF 파일에 포함할 함수를 코딩한 후 저장합니다.

함수 선언, 스크립트 코딩

함수를 코딩하는 방법은 아래와 같으며, 전역 함수를 코딩하는 방법과 같습니다.

```
function 함수명 (Argument1, Argument2, ... ) {
  //Script 코딩
  ...
  ...
  //값을 리턴하는 경우
  return 리턴값;
}
```

하나의 OZF 파일은 경우에 따라 한 개의 함수로 구성할 수도 있고 여러 개의 함수로 구성할 수도 있습니다. 여러 개의 함수로 구성할 경우에는 아래와 같이 함수명을 달리하여 함수를 선언하면 됩니다.

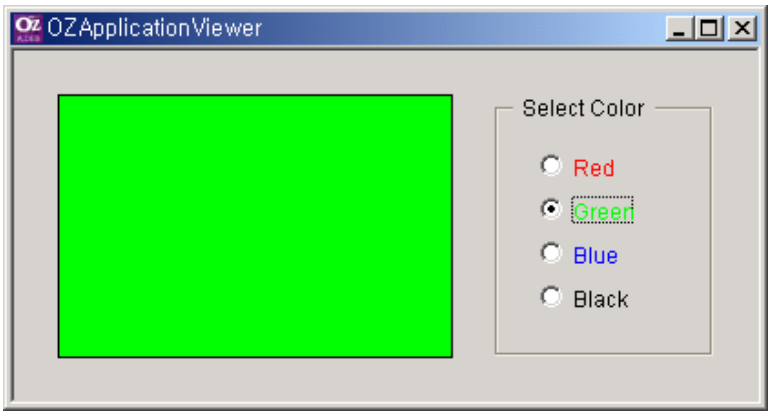
```
function changeBackColor(ComponentName, RGBColor) {
  //Script 코딩
  ...
  ...
}

function changeTextColor(ComponentName, RGBColor) {
  //Script 코딩
  ...
  ...
}
```

OZF 사용하기

OZ Function Editor에서 만든 OZF 파일을 OZ Application Designer에서 사용하는 방법을 다음과 같은 예제를 통해 설명합니다.

미리보기 화면



Step 1 OZF 만들기

'OZ Function Editor'를 이용하여 컴포넌트 이름과 RGB 색 정보를 인자값으로 받아 컴포넌트의 배경색을 변경하는 함수를 가진 OZF 파일을 만듭니다.

- 'OZ Function Editor'를 실행한 후 [File] 메뉴의 [New Function] 메뉴를 클릭하거나 툴바의 새로 만들기 아이콘( )을 클릭하여 스크립트창을 활성화합니다.

- 스크립트창에 아래와 같이 코딩합니다.

```
function changeBackColor(ComponentName, RGBColor) {
    var objComp = _GetCurrentBoard().GetChild(ComponentName);
    objComp.BackColor = RGBColor;
}
```

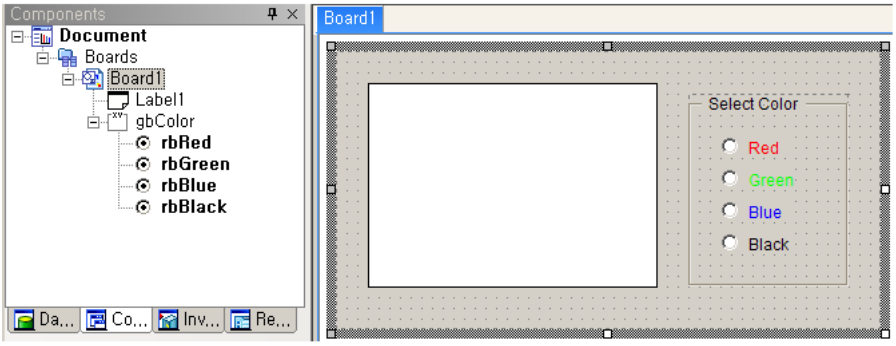
- [File] 메뉴의 [Save Function] 메뉴나 툴바의 저장하기 아이콘( )을 클릭하여

OZF 파일로 저장합니다. '파일 저장 다이얼로그'에서 리파지토리 루트 폴더에 'OZF_ChangeBackColor.ozf' 파일로 저장합니다. 본 매뉴얼에서 리파지토리 루트 폴더는 'C:\Repository' 입니다.

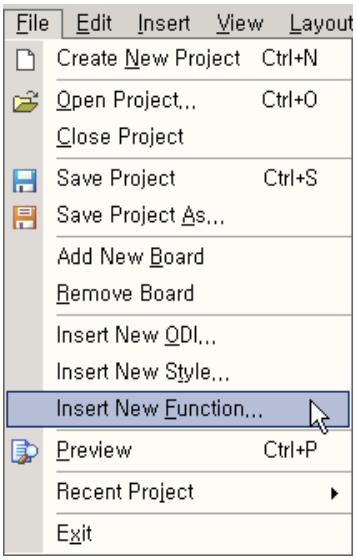
Step 2 OZF 추가

OZF 파일을 프로젝트에 추가합니다.

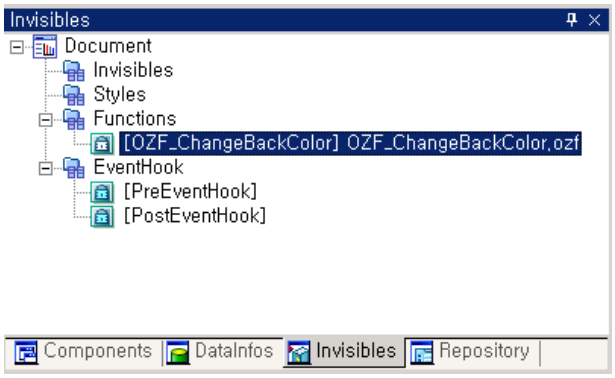
➢ 프로젝트를 새로 만들고 Board에 Label, GroupBox, RadioButton을 추가하고 아래 그림처럼 디자인합니다.



➢ [File] 메뉴의 [Insert New Function] 메뉴를 클릭하거나 리파지토리창에서 OZF 파일을 더블클릭하여 프로젝트에 추가합니다.



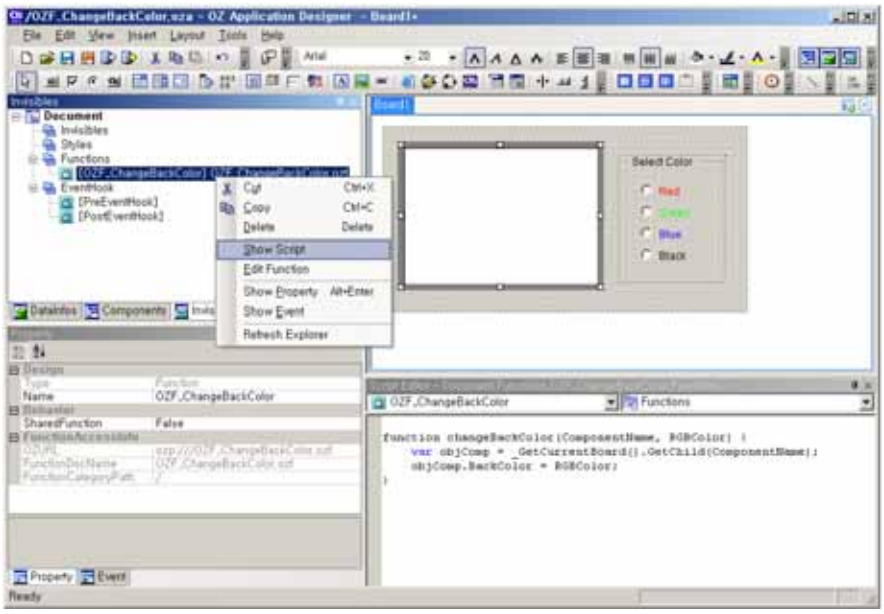
➢ 추가한 OZF 파일은 Invisibles 트리창에서 확인할 수 있습니다.



➢ 참고로, 추가한 OZF의 스크립트를 조회 또는 편집하거나 OZF 파일을 삭제하는 방법은 다음과 같습니다.

- OZF 스크립트 보기

  Invisibles 트리창에서 OZF 파일을 선택하고 마우스 오른쪽 버튼을 눌러 나타나는 팝업 메뉴에서 [Show Script] 메뉴를 클릭하여 스크립트 내용을 확인합니다.



- OZF 스크립트 편집

  Invisibles 트리창에서 편집할 OZF 파일을 선택하고 마우스 오른쪽 버튼을 눌러 나타나는 팝업 메뉴에서 [Edit Function] 메뉴를 클릭하면 자동으로 'OZ Function Editor'가 실행됩니다. 원하는 대로 스크립트를 편집하고 저장합니다.

- OZF 파일을 프로젝트에서 삭제

  Invisibles 트리창에서 삭제할 OZF 파일을 선택하고 마우스 오른쪽 버튼을 눌러 나타나는 팝업 메뉴에서 [Delete Function] 메뉴를 클릭하거나 키보드의 [Delete] 버튼을 클릭하여 해당 OZF 파일을 프로젝트에서 삭제합니다.

Step 3 스크립트 코딩

색을 나타내는 RadioButton을 클릭하면 Label의 배경색을 변경하는 OZF 파일의 함수를 호출하도록 코딩합니다.

➢ 스크립트창에서 'Red', 'Green', 'Blue', 'Black'를 나타내는 각각의 RadioButton의 'OnClick' 이벤트를 선택하여 OZF 파일의 함수를 호출하는 스크립트를 코딩합니다.

- rbRed의 OnClick 이벤트

```
changeBackColor("Label1", "255, 0, 0");
```

- rbGreen의 OnClick 이벤트

```
changeBackColor("Label1", "0, 255, 0");
```

- rbBlue의 OnClick 이벤트

```
changeBackColor("Label1", "0, 0, 255");
```

- rbBlack의 OnClick 이벤트

```
changeBackColor("Label1", "0, 0, 0");
```

Step 4 미리보기

[File] 메뉴의 [Preview]를 클릭하거나 툴바의 미리보기 아이콘()을 클릭하여 미리보기합니다.

색을 나타내는 RadioButton을 클릭하여 Label의 배경색을 변경합니다.

Ⅸ. OZ Style Editor

- OZS(OZ Style) 개념
- OZ Style Editor 실행
- OZ Style Editor 화면 구성
- 메뉴바
- 툴바
- OZS 만들기
- OZS 사용하기

본 장에서는 OZS의 개념, OZ Style Editor의 사용 방법, OZS를 프로젝트에 추가하고 사용하는 방법 대해 설명합니다.

OZS(OZ Style) 개념

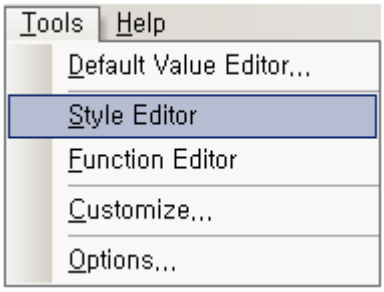
OZS는 다수의 속성과 속성 값을 하나의 스타일로 모듈화시킨 OZ Style 파일로, 하나의 OZS 파일은 다수의 스타일로 구성되고, 하나의 스타일은 다수의 속성으로 구성됩니다. 배경색, 글꼴, 패턴, 여백 등 컴포넌트에 반복적으로 적용되는 속성 값을 OZS 파일로 모듈화시켜 적용하면 컴포넌트를 일관성 있게 디자인 할수 있으며, 반복적이고 불필요한 디자인 시간을 줄여 개발 기간을 단축시키는 이점이 있습니다. OZS 파일은 'OZ Style Editor'를 사용하여 만들거나 수정할 수 있습니다.

OZ Style Editor 실행

OZ Style Editor는 OZ Application Designer의 메뉴바를 통해 실행할 수 있습니다.

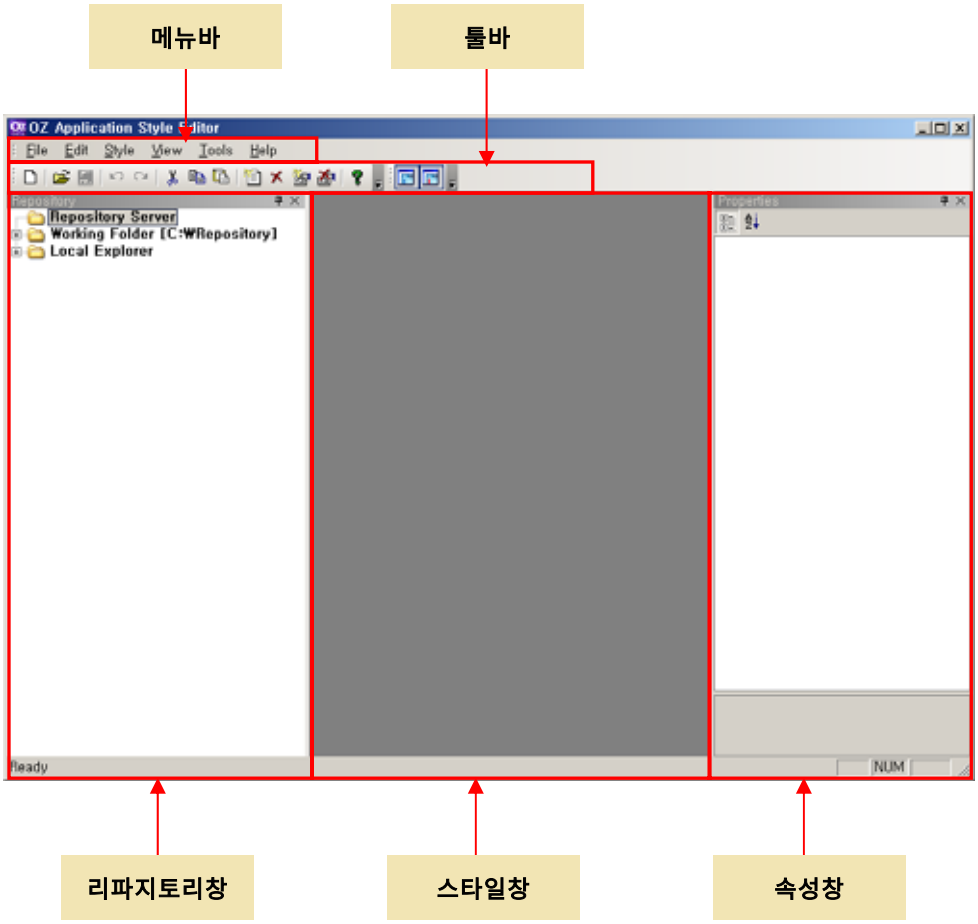
OZ Application Designer의 메뉴바 이용

'OZ Application Designer'을 실행한 후 [Tools] → [Style Editor] 메뉴를 클릭하여 'OZ Style Editor'를 실행합니다.



OZ Style Editor 화면 구성

다음은 OZ Style Editor의 사용자 인터페이스 화면이며, 각 구성 요소의 이름과 기능은 아래와 같습니다.



Window	Description
메뉴바	메뉴바는 OZ Style Editor의 상단에 풀다운(pull down)형식으로 제공되며, 파일, 편집, 스타일, 뷰, 도구, 도움말로 구성됩니다.
툴바	OZ Style Editor의 새로 만들기, 열기 등 파일 관련 기능과 잘라내기, 복사 등 편집 관련 기능, 스타일 추가 등 스타일 관련 기능, 윈도우 표시 등을 쉽고 빠르게 설정할 수 있도록 아이콘 형태로 제공됩니다.

리파지토리창	로컬 및 서버의 리파지토리 내의 폴더 및 파일을 탐색기 형태로 나타냅니다.
스타일창	OZS 파일에 추가된 스타일을 표시합니다. 스타일창은 새로 만들기를 하면 활성화 됩니다.
속성창	해당 스타일에 추가된 속성과 속성 값을 표시합니다.

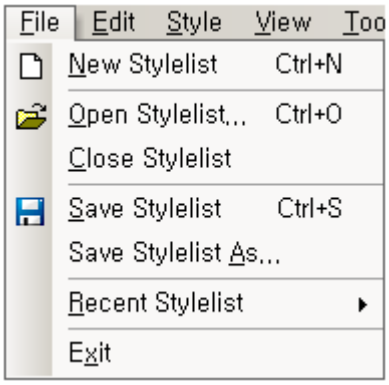
각 항목별 상세 기능과 사용법에 대해서는 다음에 기술되는 각 장에서 설명합니다.

메뉴바

메뉴바는 OZ Style Editor의 상단에 풀다운(pull down)형식으로 제공되며, 파일, 편집, 스타일, 뷰, 도구, 도움말로 구성됩니다.

파일(File)

메뉴바의 [File] 메뉴를 선택하면 아래와 같은 파일 관련 메뉴가 나타납니다.



- New Stylelist (Ctrl+N)
  새로운 OZS 파일을 만듭니다. [New Stylelist] 메뉴를 클릭하면 스타일창이 활성화되고, OZS에 하나의 스타일이 자동으로 추가됩니다.

- Open Stylelist (Ctrl+O)
  저장된 OZS 파일을 읽어 옵니다.

- Close Stylelist
  현재 작업 중인 OZS 파일을 닫습니다. 만일 변경된 내용이 있으면 저장 여부를 묻는 다이얼로그가 표시됩니다.

- Save Stylelist (Ctrl+S)
  현재 작업 중인 OZS 파일을 저장합니다.

- Save Stylelist As

현재 작업 중인 OZS 파일을 다른 이름으로 저장합니다.

- Recent Stylelist

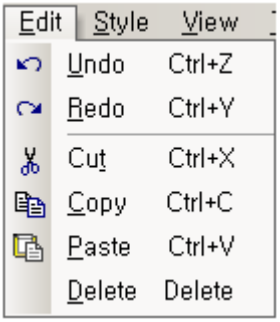
  최근에 열어본 OZS 파일을 선택합니다. 최근에 열어본 OZS 파일 중 최대 4개의 OZS 파일을 서브 메뉴로 표시합니다.

- Exit

  OZ Style Editor를 종료합니다.

편집(Edit)

메뉴바의 [Edit] 메뉴를 선택하면 아래와 같은 편집용 메뉴가 나타납니다.



- Undo (Ctrl+Z)

  바로 전에 실행한 작업을 취소하고 이전 상태로 되돌립니다.

- Redo (Ctrl+Y)

  실행 취소했던 작업을 다시 복구하며, 실행 취소한 횟수만큼 복구 가능합니다.

- Cut (Ctrl+X)

  선택한 스타일을 클립보드로 복사하고 스타일창에서는 삭제합니다.

- Copy (Ctrl+C)

  선택한 스타일을 클립보드로 복사합니다.

- Paste (Ctrl+V)

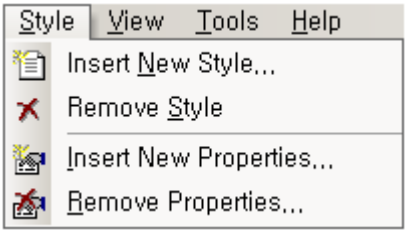
  클립보드에 복사된 스타일을 스타일창에 삽입합니다.

- Delete (Delete)

  선택한 스타일을 삭제합니다. [Undo] 메뉴를 클릭하여 삭제한 작업을 취소할 수 있습니다.

스타일(Style)

메뉴바의 [Style] 메뉴를 선택하면 아래와 같은 스타일 관련 메뉴가 나타납니다.



- Insert New Style

  스타일창에 새로운 스타일을 추가합니다. [Insert New Style] 메뉴를 클릭하면 아래 그림과 같이 스타일 이름을 입력하는 다이얼로그가 표시됩니다. 스타일 이름을 입력한 후 [OK] 버튼을 클릭하여 스타일을 추가합니다.

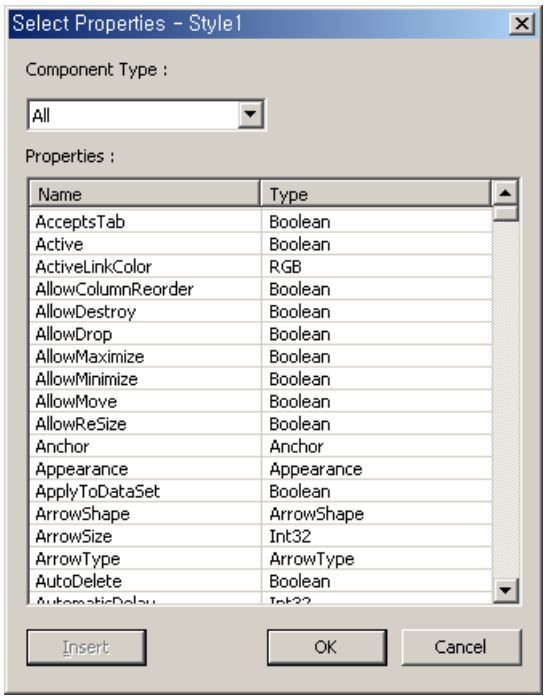


- Remove Style

  선택한 스타일을 삭제합니다.

- Insert New Properties

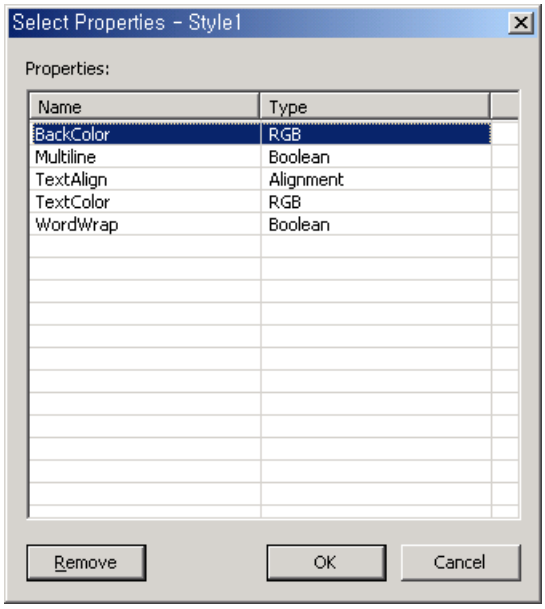
  선택한 스타일에 새로운 속성을 추가합니다. [Insert New Properties] 메뉴를 클릭하면 아래 그림과 같이 속성을 추가하는 다이얼로그가 표시됩니다.



Component Type에서 컴포넌트 타입을 선택한 후 추가할 속성을 선택하고 더블 클릭하거나 [Insert] 버튼을 클릭하여 스타일에 속성을 추가합니다. Component Type을 "All"로 선택하면 모든 컴포넌트가 가질 수 있는 속성을 표시합니다.

- Remove Properties

  선택한 스타일에 추가된 속성을 삭제합니다. [Remove Properties] 메뉴를 클릭하면 스타일에 추가된 속성이 표시되고 추가된 속성을 삭제하는 다이얼로그가 표시됩니다. 아래 그림은 'BackColor', 'Multiline', 'TextAlign', 'TextColor', 'WordWrap' 속성이 추가된 'Style1'이라는 이름의 스타일을 나타낸 그림입니다.



삭제할 속성을 선택하고 더블클릭하거나 [Remove] 버튼을 클릭하여 스타일에서 속성을 삭제합니다.

뷰(View)

메뉴바의 [View] 메뉴를 선택하면 아래와 같은 메뉴가 나타납니다.



- Repository

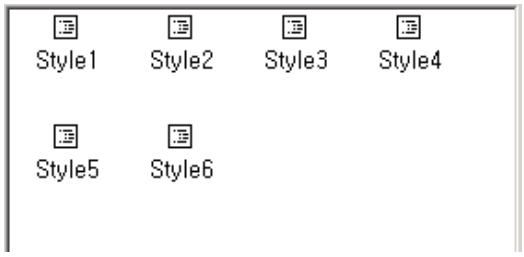
  로컬 및 서버의 리파지토리 내의 폴더 및 파일을 탐색기 형태로 나타내는 리파지토리창을 OZ Style Editor 화면에 표시할지 여부를 설정합니다.

- Properties

  해당 스타일에 추가된 속성과 속성 값을 나타내는 속성창을 OZ Style Editor 화면에 표시할지 여부를 설정합니다.

- Icons

  OZS 파일에 추가된 스타일을 아래 그림과 같이 스타일창에 아이콘 형태로 표시합니다.



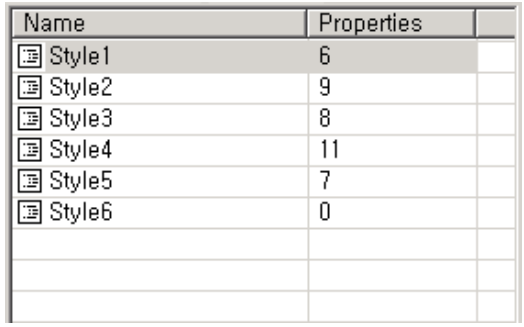
- List

  OZS 파일에 추가된 스타일을 아래 그림과 같이 스타일창에 이름만 표시하는 간단한 리스트 형태로 표시합니다.



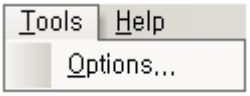
- Details

  OZS 파일에 추가된 스타일을 아래 그림과 같이 스타일창에 이름과 추가된 속성 갯수를 표시하는 자세한 리스트 형태로 표시합니다.



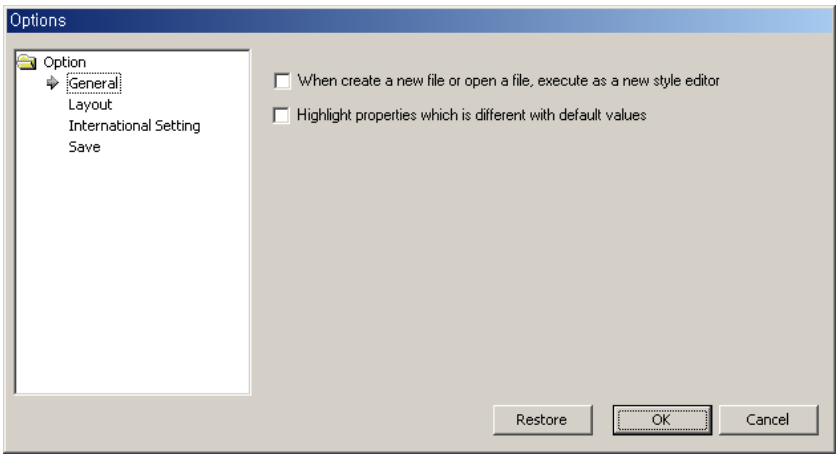
도구(Tools)

메뉴바의 [Tools] 메뉴를 선택하면 아래와 같은 메뉴가 나타납니다.



■ Options

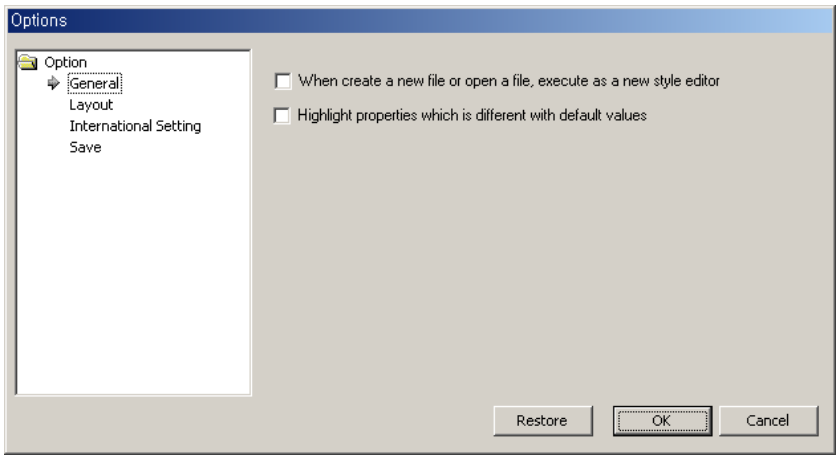
'옵션 설정 다이얼로그'를 이용하여 OZ Style Editor의 레이아웃, 메뉴바의 언어, 암호화 저장 여부 등을 설정 등을 설정합니다.



변경한 옵션을 저장 및 적용하려면 [Ok] 버튼을 OZ Style Editor 초기에 설정된 기본 옵션으로 적용하려면 [Restore] 버튼을 클릭합니다.

- Option
  - ➢ General

    새로 파일을 만들거나 리파지토리에서 파일을 열었을 때 새로운 OZ Style Editor에서 열지 여부를 설정합니다.

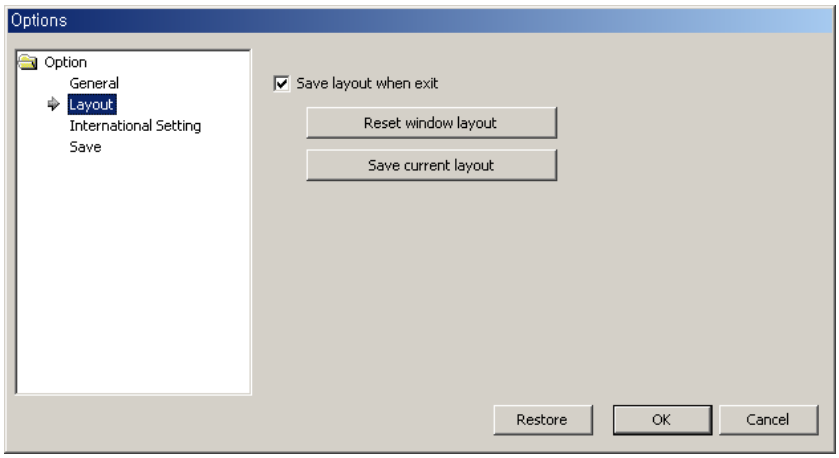


- When create a new file or open a file, execute as a new style editor
  새로 파일을 만들거나 리파지토리에서 파일을 열었을 때 새로운 OZ Style Editor에서 열지 여부를 설정합니다.

- Highlight properties which is different with default values
  속성 값이 기본 값과 다를 경우 속성 값의 글꼴을 굵게 표시할지 여부를 설정합니다.

➢ Layout

OZ Style Editor의 레이아웃을 설정합니다.

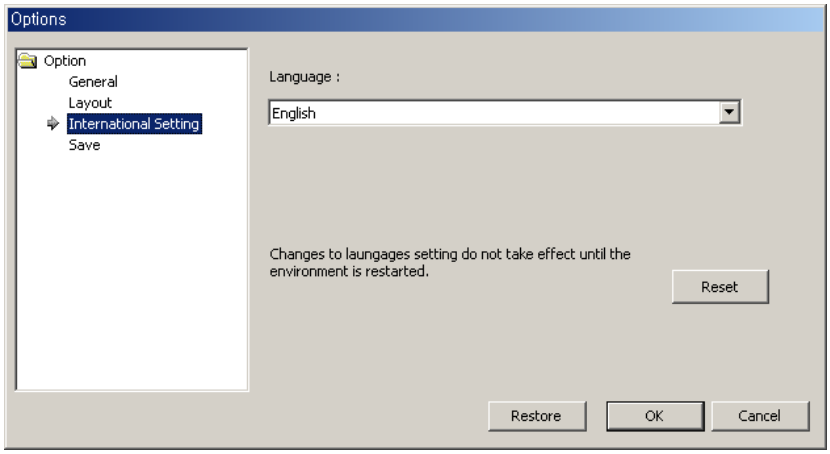


- Save layout when exit
  OZ Style Editor가 종료될 때 OZ Style Editor 레이아웃의 저장 여부를 설정합니다. OZ Style Editor 레이아웃을 기본 값으로 초기화 할 경우에는 [Reset

window layout] 버튼을 클릭하고 현재 레이아웃을 저장할 경우에는 [Save current layout] 버튼을 클릭합니다.

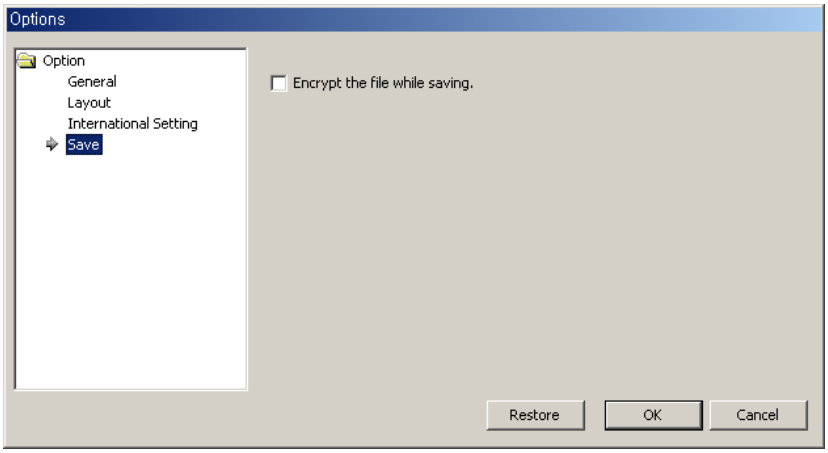
- International Setting

  메뉴바에 표시할 언어를 설정합니다.



- Save

  OZS 파일을 저장할 때 암호화하여 저장할지 여부를 설정합니다.



도움말(Help)

메뉴바의 [Help] 메뉴를 선택하면 아래와 같은 메뉴가 나타납니다.



- About OZ Application Style Editor

  OZ Style Editor의 버전 정보를 표시합니다.

툴바

툴바는 OZ Style Editor의 새로 만들기, 열기, 저장, 실행취소, 다시실행, 잘라내기, 복사, 붙이기, 스타일 추가, 스타일 삭제, 속성 추가, 속성 삭제, 리파지토리창 보기, 속성창 보기 등을 쉽고 빠르게 설정할 수 있도록 메뉴바 바로 아래에 아이콘 형태로 제공됩니다.

표준(Standard) 툴바

파일 관련 기능과 잘라내기, 복사 등 일반적인 기능을 실행합니다.



Icon	Description
	New Stylelist, Open Stylelist, Save Stylelist
	Undo, Redo, Cut, Copy, Paste
	Insert New Style, Remove Style Insert New Propeties, Remove Properties
	About OZ Application Style Editor

뷰(View) 툴바

리파지토리창과 속성창의 표시 여부를 설정합니다.



Icon	Description
	View Repository, View Properties

OZS 만들기

OZ Style Editor를 실행한 후 [File] 메뉴의 [New Stylelist] 메뉴를 클릭하거나 툴바의 새로 만들기 아이콘()을 클릭하여 스타일창을 활성화한 후 스타일을 추가하고, 스타일에 적용할 속성을 추가하여 OZS 파일을 만듭니다.

스타일 추가

■ 스타일창이 비활성화된 경우

[File] 메뉴의 [New Stylelist] 메뉴를 클릭하거나 툴바의 새로 만들기 아이콘()을 클릭하면 스타일창이 활성화되면서 스타일이 하나 추가됩니다.

■ 스타일창이 활성화된 경우

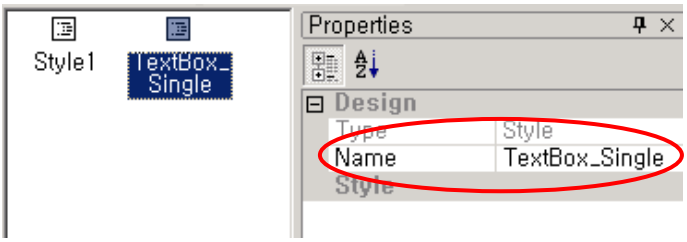
- 메뉴바에서 [Style] 메뉴의 [Insert New Style] 메뉴를 클릭하거나 툴바에서 스타일 추가 아이콘()을 클릭합니다.

- 스타일 이름 입력 다이얼로그에서 추가할 스타일 이름을 입력한 후 [OK] 버튼을 클릭하여 스타일을 추가합니다.



스타일 이름 변경

이름을 변경할 스타일을 선택한 후 속성창에서 'Name' 속성 값을 변경합니다. 아래 그림은 'Style2'의 스타일 이름을 'TextBox_Single'로 변경한 모습입니다.

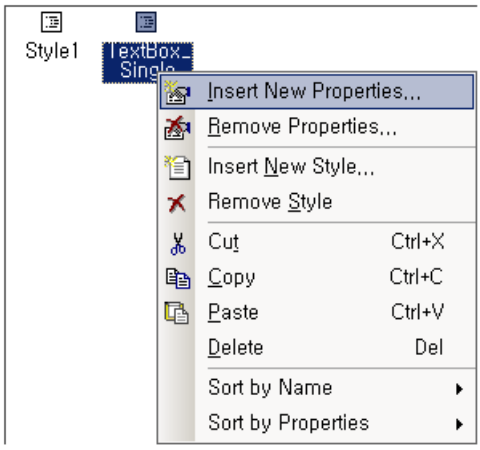


스타일에 속성 추가

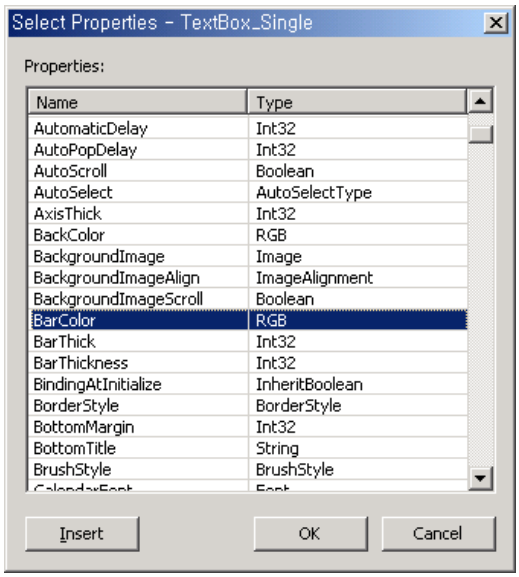
스타일에 적용할 속성을 추가하고 속성 값을 설정합니다.

■ 속성 추가

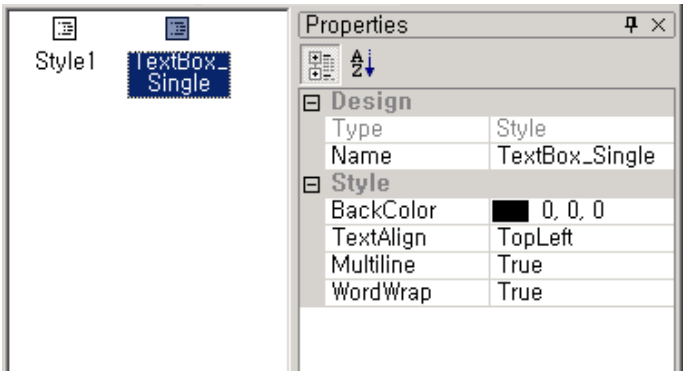
- 속성을 추가할 스타일을 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [Insert New Properties] 메뉴을 클릭하거나 툴바에서 속성 추가 아이콘( )을 클릭합니다.



- 스타일에 추가할 속성을 리스트 형태로 나타낸 다이얼로그가 나타납니다. 속성 선택 다이얼로그에서 추가할 속성을 선택하고 더블클릭하거나 [Insert] 버튼을 클릭하여 해당 스타일에 속성을 추가합니다.

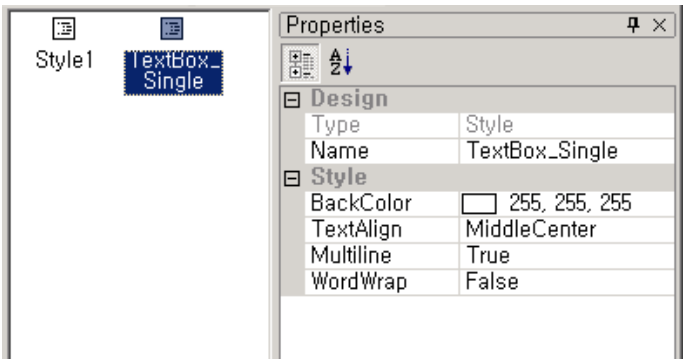


- 스타일에 속성이 추가된 것을 속성창에서 확인합니다. 아래 그림은 'TextBox_Single'이라는 스타일에 'BackColor', 'TextAlign', 'Multiline', 'WordWrap' 속성이 추가된 모습입니다.



■ 속성 값 변경

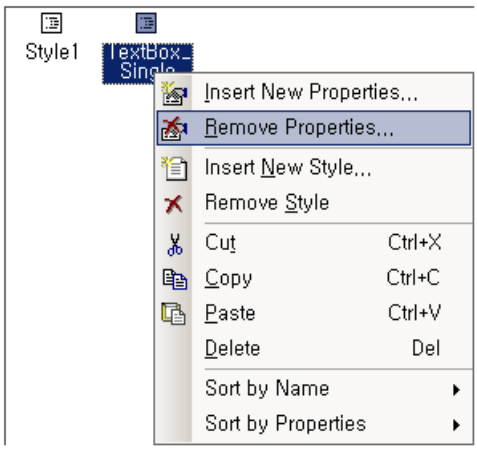
- 스타일을 선택하여 속성창에서 속성 값을 원하는 값으로 변경합니다. 아래 그림은 'BackColor'는 '255,255,255'로, 'TextAlign'은 'MiddleCenter'로, 'Multiline'은 'True'로, 'WordWrap'은 'False'로 설정한 모습입니다.



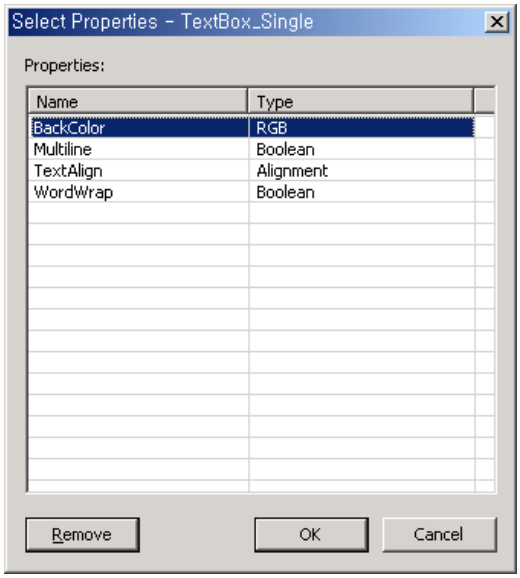
스타일에 추가된 속성 삭제

스타일에 추가된 속성을 삭제합니다.

- 속성을 삭제할 스타일을 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [Remove Properties] 메뉴을 클릭하거나 툴바에서 속성 삭제 아이콘( )을 클릭합니다.



- 선택한 스타일에 추가된 속성을 리스트 형태로 나타낸 다이얼로그가 나타납니다. 삭제할 속성을 선택하고 더블클릭하거나 [Remove] 버튼을 클릭하여 스타일에서 속성을 삭제합니다.

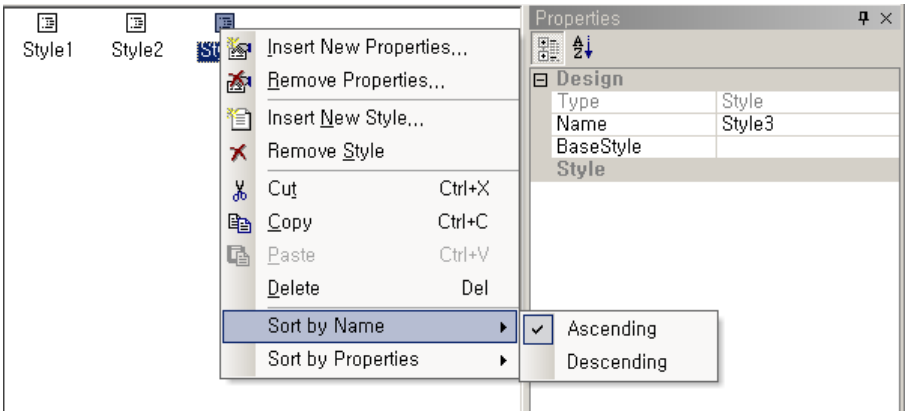


스타일 정렬 기능

스타일 창에서 마우스 오른쪽 버튼을 클릭하여 팝업 메뉴로 스타일을 정렬합니다.

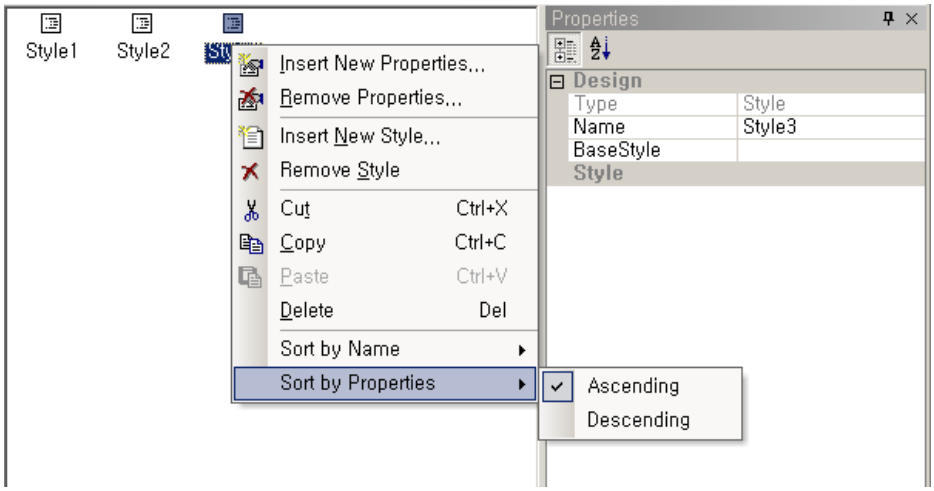
■ 이름별 정렬

[이름별 정렬] 하위 메뉴인 [오름차순] 또는 [내림차순]을 클릭하여 정렬합니다.



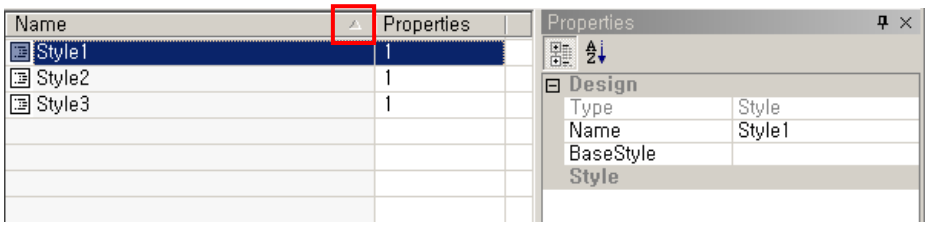
■ 속성 개수별 정렬

[속성 개수별 정렬] 하위 메뉴인 [오름차순] 또는 [내림차순]을 클릭하여 정렬합니다.



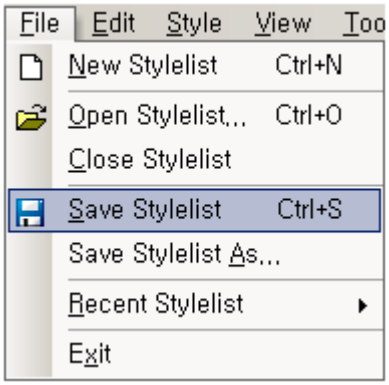
■ 타이틀 클릭하여 스타일 정렬

상세 보기 시 스타일 창에서 타이틀을 클릭하여 정렬합니다.



OZS 파일로 저장

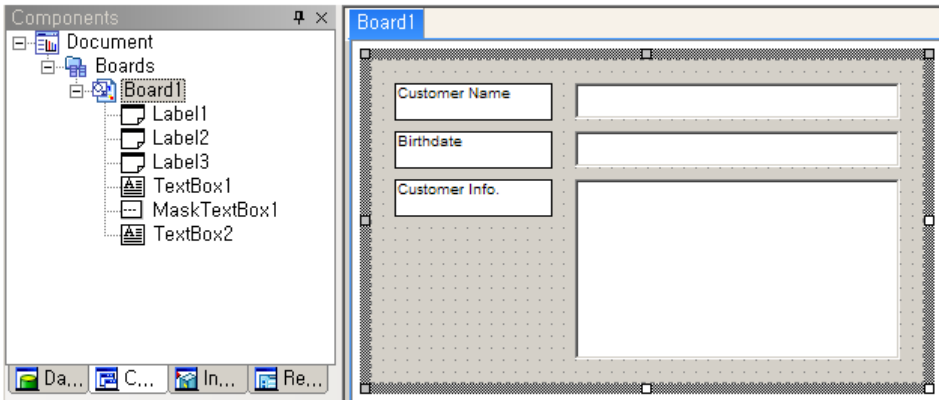
스타일과 속성 디자인이 모두 완료되면 [File] 메뉴의 [Save Stylelist] 메뉴나 툴바의 저장 아이콘()을 클릭하여 OZS 파일로 저장합니다.



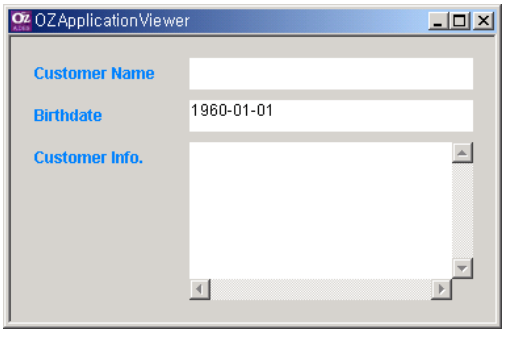
OZS 사용하기

OZ Style Editor에서 만든 OZF 파일을 OZ Application Designer에서 사용하는 방법을 다음과 같은 예제를 통해 설명합니다.

디자인 화면



미리보기 화면

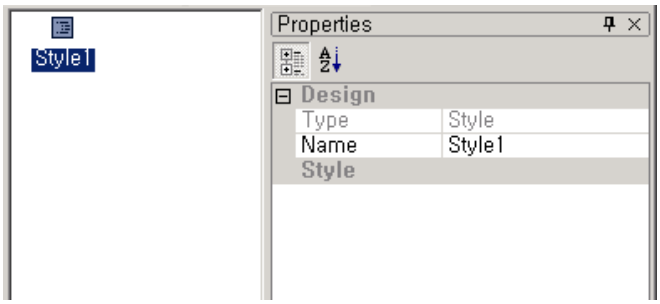


Step 1 OZS 만들기

OZ Style Editor를 이용하여 Label, TextBox, MaskTextBox의 스타일을 미리보기 화면처럼 표시되도록 하는 OZS 파일을 만듭니다.

➢ Label의 배경색, 테두리색, 글자색, 글꼴, 텍스트 정렬 방식 등을 설정하는 스타일을 만듭니다.

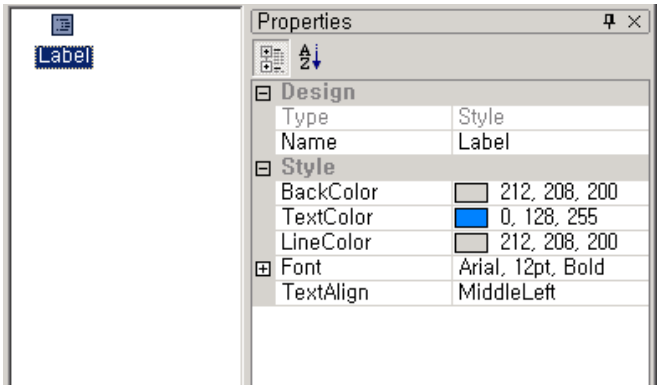
- OZ Style Editor를 실행하여 [File] 메뉴의 [New Stylelist] 메뉴를 클릭하여 새로운 OZS 파일을 만듭니다.



- 'Style1'을 선택하고 속성창에서 스타일의 이름을 'Label'로 변경합니다.

- Label을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Insert New Properties] 메뉴를 선택하여 속성 선택 다이얼로그를 나타냅니다.

- '속성 선택 다이얼로그'에서 BackColor, TextColor, LineColor, Font, TextAlign 속성을 추가하고, 각 속성 값을 아래와 같이 변경합니다.

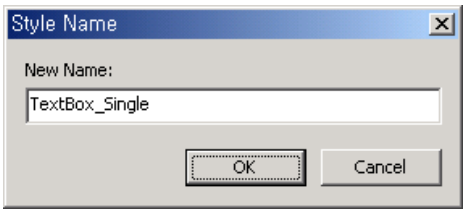


➢ TextBox 중 'Customer Name'을 입력하는 TextBox의 글꼴과 외곽선 스타일을 설정하는 스타일을 만듭니다.

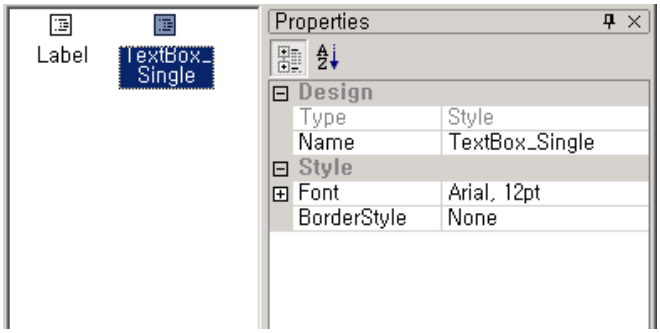
- 메뉴바에서 [Style] 메뉴의 [Insert New Style] 메뉴를 클릭하거나 툴바에서

스타일 추가 아이콘( )을 클릭합니다.

- 스타일 이름 다이얼로그에서 스타일 이름을 'TextBox_Single'로 입력한 후 [OK] 버튼을 클릭하여 스타일을 추가합니다.

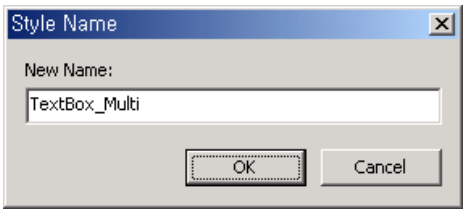


- 'TextBox_Single' 스타일에 'Font', 'BoardStyle' 속성을 추가한 후 속성 값을 아래 그림과 같이 변경합니다.

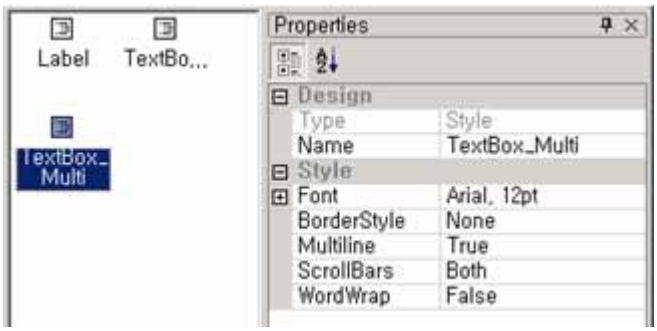


➢ TextBox 중 'Customer Info.'를 입력하는 TextBox의 글꼴, 외곽선 스타일, 여러 줄 옵션, 스크롤바 등을 설정하는 스타일을 만듭니다.

- 메뉴바에서 [Style] 메뉴의 [Insert New Style] 메뉴를 클릭하거나 툴바에서 스타일 추가 아이콘( )을 클릭하여, 스타일 이름 다이얼로그에서 스타일 이름을 'TextBox_Multi'로 입력한 후 [OK] 버튼을 클릭하여 스타일을 추가합니다.

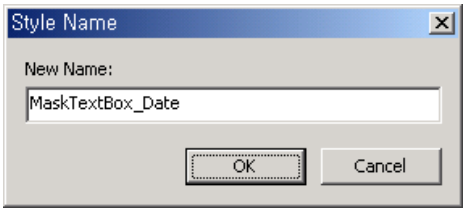


- 'TextBox_Multi' 스타일에 'Font', 'BoardStyle', 'Multiline', 'ScrollBars', 'WordWrap' 속성을 추가한 후 속성 값을 아래 그림과 같이 변경합니다.

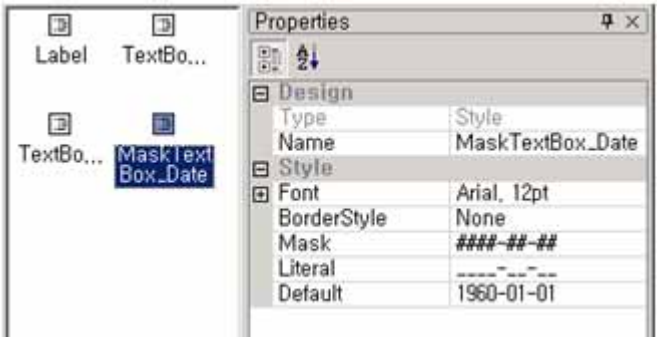


➢ 생년월일을 입력하는 MaskTextBox의 글꼴, 외곽선 스타일, 마스크, 리터럴, 기본 값 등을 설정하는 스타일을 만듭니다.

- 메뉴바에서 [Style] 메뉴의 [Insert New Style] 메뉴를 클릭하거나 툴바에서 스타일 추가 아이콘( )을 클릭하여, 스타일 이름 다이얼로그에서 스타일 이름을 'MaskTextBox_Date'로 입력한 후 [OK] 버튼을 클릭하여 스타일을 추가합니다.



- 'MaskTextBox_Date' 스타일에 'Font', 'BoardStyle', 'Mask', 'Literal', 'Default' 속성을 추가한 후 속성 값을 아래 그림과 같이 변경합니다.



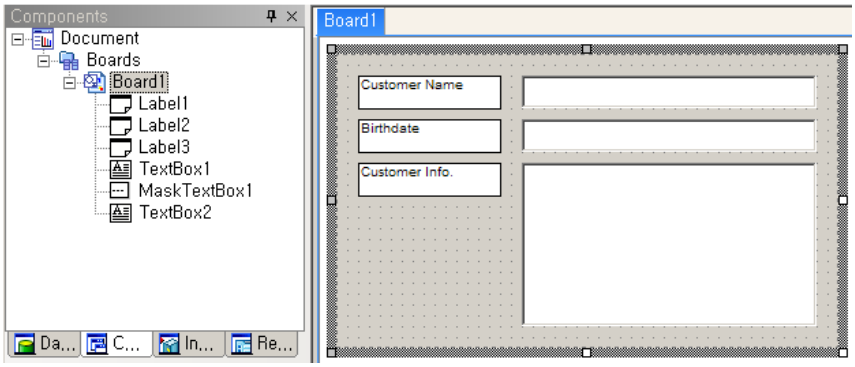
➢ [File] 메뉴의 [Save Stylelist] 메뉴나 툴바의 저장하기 아이콘( )을 클릭하여 OZS 파일로 저장합니다. '파일 저장 다이얼로그'에서 리파지토리 루트 폴더에 'OZS.ozs' 파일로 저장합니다. 본 매뉴얼에서 리파지토리 루트 폴더는 'C:\Repository' 입니다.

Step 2 OZS 추가

프로젝트의 메인 Board를 디자인하고 프로젝트에 OZS 파일을 추가합니다.

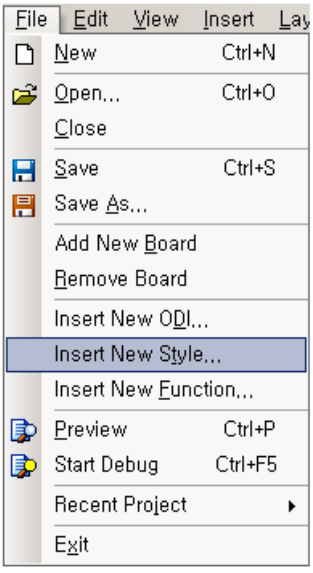
➢ 메인 Board 디자인

- 'OZ Application Designer'를 실행하여 프로젝트를 새로 만들고, Board에 Label, TextBox, MaskTextBox를 추가하고 예제의 디자인 화면처럼 디자인합니다.

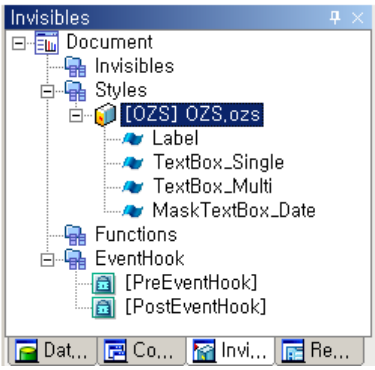


➢ OZS 파일 추가

- [File] 메뉴의 [Insert New Style] 메뉴를 클릭하거나 리파지토리창에서 OZS 파일을 더블클릭하여 프로젝트에 추가합니다.



- 추가한 OZS 파일은 Invisibles 트리창에서 확인할 수 있습니다.



Step 3 스타일 적용

컴포넌트에 스타일을 적용합니다.

- Label1, Label2, Label3을 다중 선택한 후 속성창에서 'Style' 속성 값을 'OZS.Label'로 설정하여 스타일을 적용합니다.

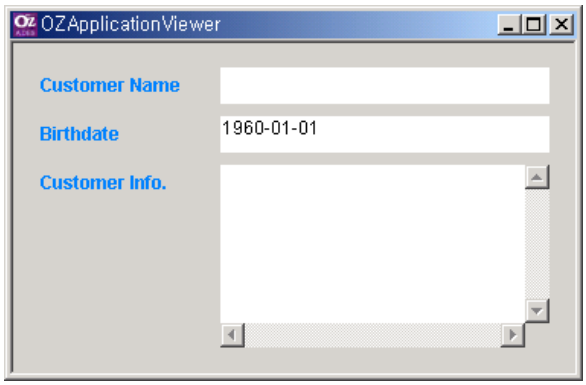
- TextBox1을 선택한 후 속성창에서 'Style' 속성 값을 'OZS.TextBox_Single'로 설정합니다.

- TextBox2를 선택한 후 속성창에서 'Style' 속성 값을 'OZS.TextBox_Multi'로 설정합니다.

- MaskTextBox1을 선택한 후 속성창에서 'Style' 속성 값을 'OZS.MaskTextBox_Date'로 설정합니다.

Step 4 미리보기

[File] 메뉴의 [Preview]를 클릭하거나 툴바의 미리보기 아이콘()을 클릭하여 실행하여 미리보기합니다.



X. 서버 리파지토리 사용법

- 서버 리파지토리 개요
- 서버 리파지토리 연결
- 서버로부터 파일 열기
- 파일 업로드, 다운로드
- 체크인, 체크아웃
- 히스토리 보기
- 아이템 찾기

본 장에서는 서버 리파지토리의 개념과 OZ Application Designer에서 서버 리파지토리를 이용하는 방법에 대해 설명합니다.

서버 리파지토리 개요

서버 리파지토리는 OZA, OZC, OZF, OZS, ODI 파일, 데이터 파일 등 각종 파일을 저장하는 원격 저장소로, 다수의 개발자가 원격으로 파일을 공유할 수 있으며, 각 사용자마다 접근 권한을 달리할 수 있습니다. 또한 파일 버전 관리 기능을 통해 개발의 효율을 높여줍니다.

서버 리파지토리 연결

다음은 OZ Application Designer에서 서버 리파지토리에 연결하고 연결을 해제하는 방법에 대해 설명합니다.

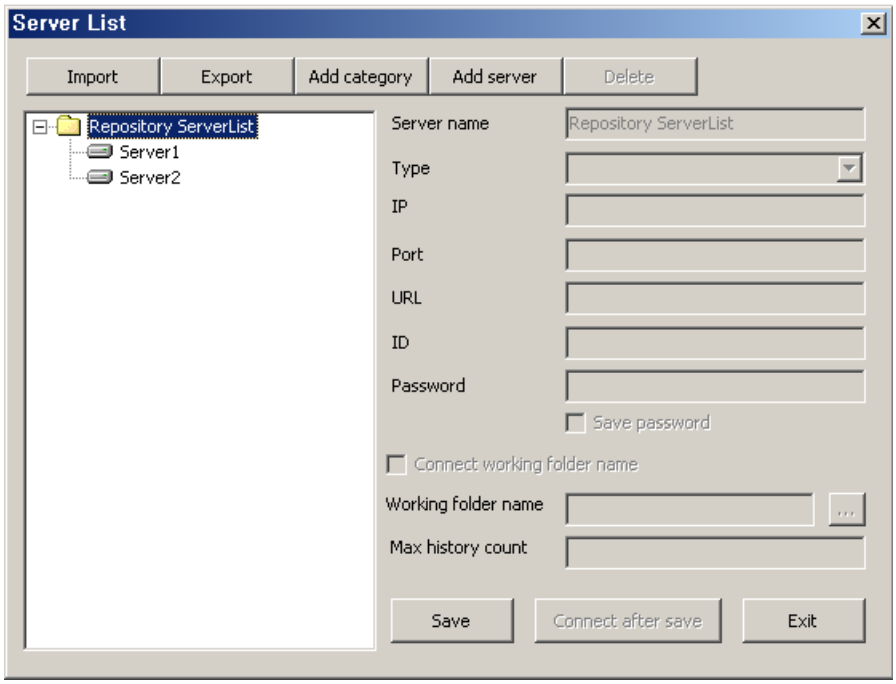
서버 리파지토리 연결

OZ Application Designer의 리파지토리창을 이용하여 서버 리파지토리로 연결합니다.

Step 1 OZ Application Designer의 리파지토리창에서 'RepositoryServer'를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Open Repository] 메뉴를 선택합니다.



Step 2 아래 그림과 같이 저장된 리파지토리 서버 리스트와 접속 정보를 나타내는 '서버 리스트 다이얼로그'가 표시됩니다.

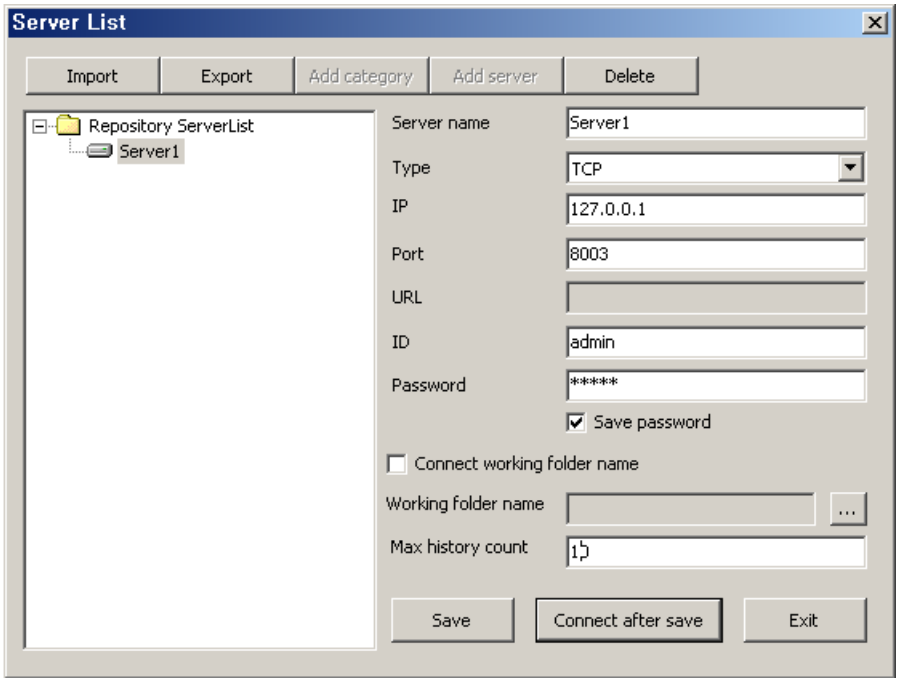


'서버 리스트 다이얼로그'의 설정 버튼 및 옵션의 기능은 아래와 같습니다.

Button	Description
Import	서버 리스트 정보를 저장하고 있는 XML 파일을 가져와서 서버 리스트를 표시합니다.
Export	현재 다이얼로그에 추가되어 있는 서버 리스트 정보를 XML 파일로 저장합니니다.
Add Category	서버 리파지토리에 카테고리를 추가합니다.
Add Server	새로운 서버 리파지토리를 추가합니다. [서버 추가] 버튼을 클릭하면 새로운 서버 리파지토리를 추가할 수 있도록 옵션 설정이 활성화됩니다.
Delete	선택된 서버 또는 카테고리를 삭제합니다.
...	서버 리파지토리에 연결할 작업 폴더 이름을 선택할 수 있도록 "작업 폴더 매니저 다이얼로그"를 실행합니다.
Save	현재 서버 리스트 정보를 저장합니다.
Connect after save	현재 서버 리스트 정보를 저장한 후 선택된 리파지토리 서버로 접속합니다.
Exit	서버 리스트 다이얼로그를 닫습니다.

Option	Description
Server name	접속할 서버 리파지토리의 이름을 입력합니다.
Type	연결할 OZ Server의 타입을 설정합니다. Daemon 타입일 경우 TCP로, Servlet 타입일 경우 HTTP로 설정합니다.
IP	Daemon 타입의 서버로 설정할 경우 OZ Server가 실행 중인 IP 주소를 입력합니다.
Port	Daemon 타입의 서버로 설정할 경우 OZ Server의 Port를 입력합니다.
URL	Servlet 타입의 서버로 설정할 경우 OZ Server의 URL을 입력합니다.
Name	서버 리파지토리에 등록된 사용자 계정을 입력합니다.
Password	해당 사용자 계정의 패스워드를 입력합니다.
Connect working folder name	서버 리파지토리 접속 시 작업 폴더를 작업 폴더 매니저에 등록된 작업 폴더 이름에 설정된 경로로 자동으로 변경할지 여부를 설정합니다. ※ 참고사항 : 작업 폴더 이름 연결에 따라 작업 폴더가 변경되는 규칙은 아래 "리파지토리 서버 접속 시 작업 폴더 설정 규칙" 참고 사항을 참조하시기 바랍니다.
Working folder name	서버 접속 시 연결할 작업 폴더 이름이 표시됩니다.
Save Password	비밀번호 저장 여부를 설정합니다.
Max history count	리파지토리 파일 다이얼로그에서 표시되는 폴더 히스토리의 개수 중 해당 서버의 폴더 히스토리 개수를 설정합니다.

Step 3 접속할 서버 리파지토리 정보를 추가한 후 접속합니다. 본 매뉴얼에서는 Daemon 타입의 OZ Server를 실행한 후 아래와 같이 접속 정보를 입력한 후 [Connect after save] 버튼을 클릭하여 서버 리파지토리를 연결하였습니다.



패스워드는 'admin'으로 입력하였습니다.

※ 참고사항 : 리파지토리 서버 접속 시 작업 폴더 설정 규칙

리파지토리 서버 접속 시 작업 폴더 매니저에 설정된 작업 폴더 이름과 연결하여 작업 폴더를 자동으로 변경할 수 있으며, 작업 폴더가 변경되는 규칙은 다음과 같습니다.

1) "작업 폴더 이름 연결" 옵션을 체크한 경우

① 작업 폴더 이름이 작업 폴더 매니저에 등록된 경우

리파지토리 서버 연결 시 작업 폴더 매니저에 등록된 경로로 작업 폴더가 자동으로 변경됩니다. 단, 작업 폴더 매니저에 등록된 경로가 로컬에 없을 경우 작업 폴더를 생성한 후 해당 경로로 작업 폴더가 변경됩니다.

② 작업 폴더 이름이 작업 폴더 매니저에 등록되지 않은 경우

리파지토리 서버 연결 시 아래와 같이 작업 폴더 이름을 작업 폴더 매니저에 등록할지 여부를 묻는 메시지 창이 표시됩니다.

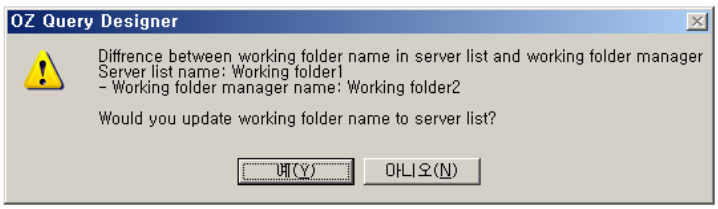


[예] 버튼을 클릭할 경우 작업 폴더 이름과 현재 작업 폴더 경로가 작업 폴더 매니저에 자동으로 등록되며 리파지토리 서버에 접속하고, (이 때, 작업 폴더 트리 제목에 등록된 작업 폴더 이름이 표시됩니다.) [아니오] 버튼을 클릭할 경우 작업 폴더 매니저에 등록하지 않고 리파지토리 서버에 접속하며, [취소] 버튼을 클릭할 경우 작업 폴더 매니저에 등록하지 않고 리파지토리 서버에 접속하지 않습니다.

2) "작업 폴더 이름 연결" 옵션을 체크하지 않은 경우
작업 폴더를 변경하지 않고 리파지토리 서버에 접속합니다.

※ 참고사항 : 작업 폴더 이름 변경 규칙

리파지토리 서버에 접속된 상태에서 작업 폴더 매니저 또는 리파지토리 창의 [작업 폴더 설정] 메뉴를 클릭하여 작업 폴더를 변경할 때 접속된 서버에 "작업 폴더 이름 연결" 옵션이 체크되어 있고, 작업 폴더 이름이 작업 폴더 매니저에 등록되어 있으며, 서버에 설정된 작업 폴더 이름과 선택한 작업 폴더 이름이 다를 경우 아래와 같이 서버에 설정된 작업 폴더 이름을 변경할지 여부를 묻는 메시지 창이 표시됩니다.



[예] 버튼을 클릭할 경우 선택한 작업 폴더 이름이 서버에 반영되고, [아니오] 버튼을 클릭할 경우 서버에 반영되지 않습니다.

Step 4 서버 리파지토리 연결이 완료되면 OZ Application Designer의 리파지토리창에서 서버 리파지토리의 폴더 및 파일을 확인합니다.



파일 앞에 있는 아이콘은 해당 파일의 종류와 체크인 상태(,)인지 체크아웃 상태(,)인지 표시하며, 체크인, 체크아웃에 대한 설명은 본 장의 체크인, 체크아웃 부분을 참조하십시오. 서버 리파지토리의 리파지토리 루트 폴더(/)는 오즈 서버의 "conf\repository.properties" 파일에 설정된 폴더이며, 자세한 설명은 "오즈 엔터프라이즈 서버 관리자 매뉴얼"을 참조하시기 바랍니다.

서버 리파지토리 연결 해제

OZ Application Designer의 리파지토리창을 이용하여 연결된 서버 리파지토리의 연결을 해제합니다. OZ Application Designer의 리파지토리창에서 연결된 서버 리파지토리를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [Close Repository] 메뉴를 선택하여 연결을 해제합니다.



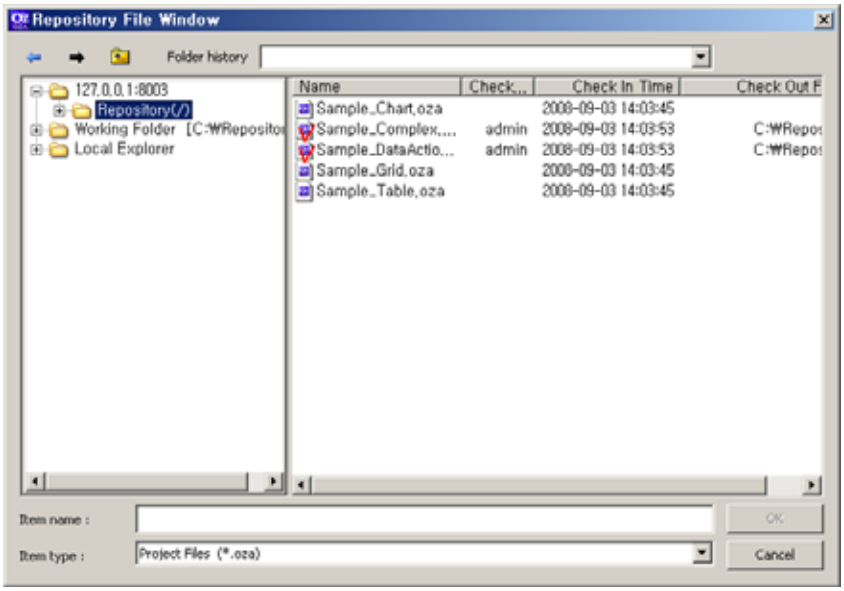
서버로부터 파일 열기

서버 리파지토리에 연결된 상태에서는 서버 리파지토리에 저장된 프로젝트 파일, OZC, OZF, OZS 파일 등을 직접 열어 편집할 수 있습니다.

서버로부터 파일 열기

[File] 메뉴의 [Open] 메뉴나 툴바의 열기 아이콘( )을 클릭하여 서버 리파지토리의 파일을 선택하거나 리파지토리창에서 해당 파일을 더블클릭하여 저장된 파일을 읽어 옵니다.

[File] 메뉴의 [Open] 메뉴나 툴바의 열기 아이콘( )을 클릭하면 아래 그림과 같은 리파지토리 파일 다이얼로그가 화면에 나타납니다.



Button	Description
	이전에 선택한 폴더로 이동합니다.
	다음에 선택한 폴더로 이동합니다.
	현재 선택한 폴더의 상위 폴더로 이동합니다.

Option	Description
Folder history	최근에 파일을 열었던 폴더의 순서로 목록이 표시됩니다. 로컬에서 파일을 열었던 폴더의 목록이 먼저 표시되고 현재 접속중인 서버 리파지토리에서 파일을 열었던 폴더의 목록이 그 아래 표시됩니다. 로컬 리파지토리 폴더의 경우에는 "ozp://" 형식으로, 서버 리파지토리 폴더의 경우에는 "ozpserver://" 형식으로 표시됩니다. 목록에 표시되는 히스토리 수의 기본 값은 로컬/서버 리파지토리 각 10개씩 즉, 20개입니다. ※ 주의사항 : 히스토리에서 선택한 파일 목록의 경로가 실제 리파지토리에 존재하지 않는 경로일 경우 현재 선택되어진 폴더가 표시됩니다. ※ 참고사항 • 폴더 히스토리에 표시되는 히스토리의 최대 개수는 서버 리스트 다이얼로그에서 설정한 "Max history count"와 로컬 리파지토리의 히스토리 수가 합쳐진 수만큼 목록에 표시됩니다. 예를들어 서버 리파지토리의 히스토리 개수가 10이고 로컬 리파지토리의 히스토리 개수가 10인 경우 리파지토리 파일 다이얼로그의 폴더 히스토리 목록에 표시되는 최대 개수는 20개입니다. • 폴더 히스토리의 개수는 conf/OZServerList.xml 파일에서 확인할 수 있습니다. • 서버 리파지토리의 히스토리 수는 해당 서버의 "MaxHistoryCount" 옵션 즉, "<SERVER Name=..."의 "MaxHistoryCount"에 설정되어있는 값이고, 로컬 리파지토리의 히스토리 수는 "<LOCAL MaxHistoryCount="에 설정되어있는 값입니다.

서버로부터 파일 열기의 주의사항

■ 서버로부터 열기한 파일의 편집한 내용을 서버 리파지토리에 반영하기 위해서는 반드시 체크아웃을 먼저하고 내용을 편집한 다음 체크인을 하는 과정을 거쳐야 합니다.

■ 서버로부터 열기한 파일이 체크아웃 상태에서 편집을 마친 후에는 반드시 체크인하거나 체크아웃을 취소하여야 합니다.

■ 파일을 서버로부터 열기할 때 다운로드하여 로컬의 작업 폴더에 저장한 경우에는 [File] 메뉴의 [Save Project] 메뉴나 툴바의 저장 아이콘()을 클릭하여 저장하는 등의 작업은 로

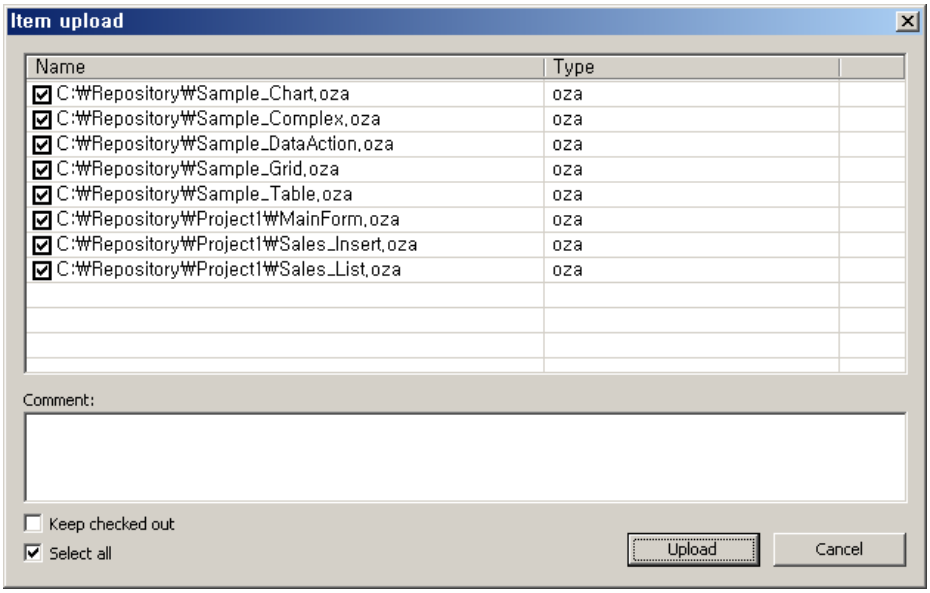
컬의 파일에 저장시키는 것입니다. 즉 다시 파일을 서버로부터 열기를 하면 서버 리파지토리에 있는 파일이 열리기 때문에 로컬의 작업폴더에 저장한 내용과 다를 수 있습니다.

파일 업로드, 다운로드

로컬 시스템에서 작업한 파일을 서버 리파지토리로 업로드 하거나 서버 리파지토리의 파일을 로컬 시스템으로 다운로드 합니다.

파일 업로드

OZ Application Designer의 리파지토리창에서 업로드할 파일을 선택한 후 서버 리파지토리내의 특정 폴더로 Drag & Drop하거나 업로드할 파일을 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [Upload to Repository] 메뉴를 클릭하면 아래 그림과 같이 '파일 업로드 다이얼로그'가 표시됩니다.

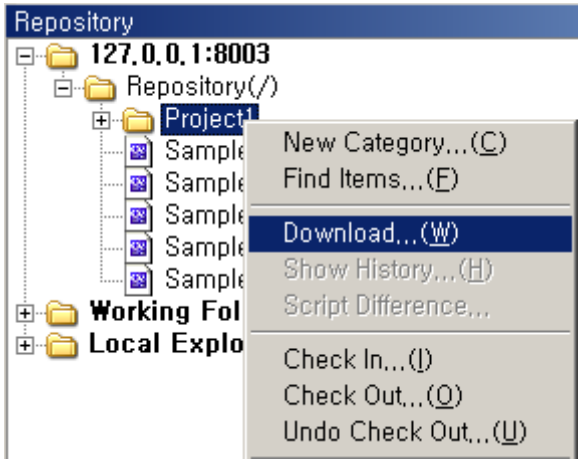


업로드할 파일을 체크한 후 [Upload] 버튼을 클릭하여 서버 리파지토리로 업로드 합니다.

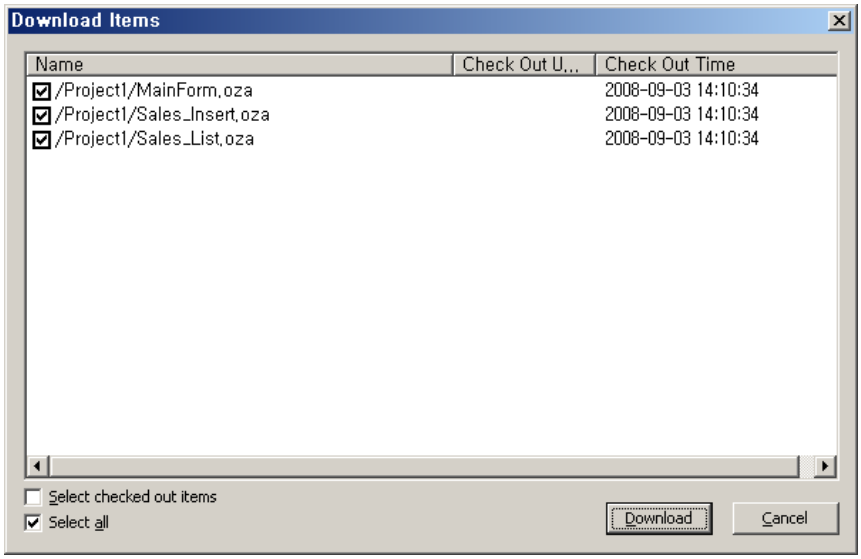
파일 다운로드

서버 리파지토리에서 다운로드할 파일을 선택한 후 마우스 오른쪽 버튼을 눌러 나타나는 팝업 메

뉴에서 [Download] 메뉴를 클릭합니다.



'파일 다운로드 다이얼로그'에서 다운로드할 파일을 체크한 후 [Download] 버튼을 클릭하여 로컬 시스템의 작업 폴더로 다운로드 합니다.



※ 주의사항 : 서버 리파지토리에서 다운되는 파일은 로컬 작업 폴더에 읽기 전용 파일로 저장됩니다.

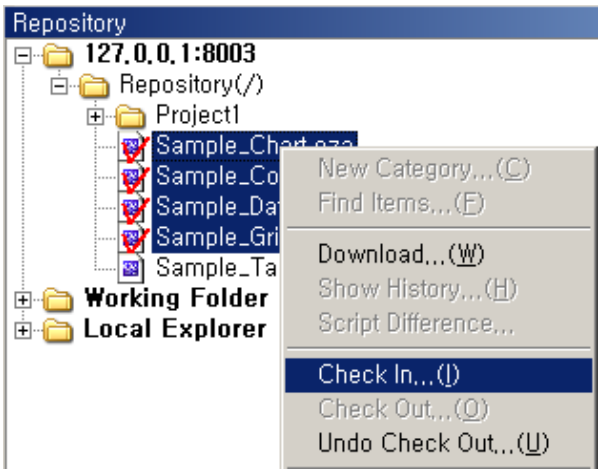
체크인, 체크아웃

오즈 리파지토리 서비스는 다중 사용자 개발 환경을 제공하며, 다중 개발자 환경에서 파일의 안전한 공유와 갱신을 위해 체크인, 체크아웃 기능을 제공합니다.
Admin 계정으로 'None' 타입의 서버에 접속한 경우에는 모든 파일에 대해 항상 체크아웃한 상태로 서버에 접속되며, 체크인, 체크아웃 기능을 정상적으로 사용하기 위해서는 오즈 서버의 리파지토리 타입이 BuiltIn 또는 RDB 방식이어야 합니다. 자세한 설명은 'OZ Enterprise Server Administrator's Guide'를 참조하십시오.

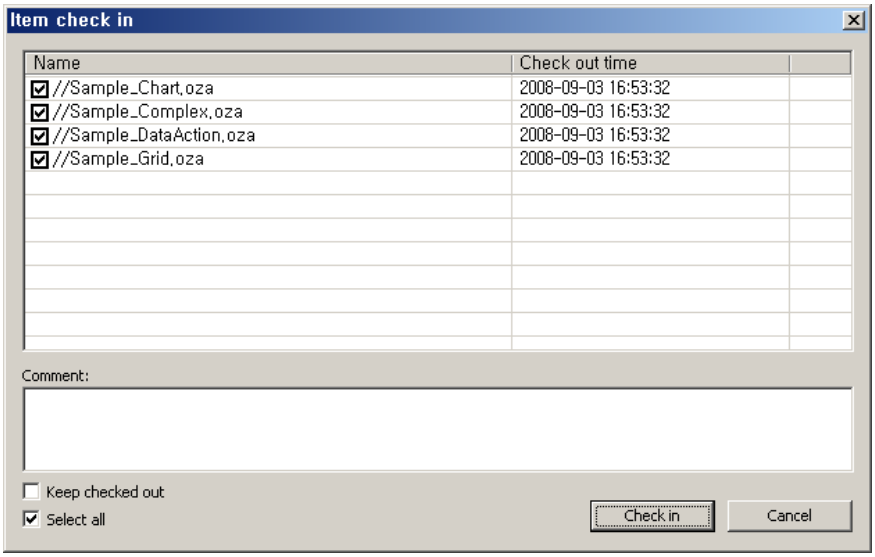
오즈 리파지토리 서비스는 파일의 안전한 공유와 갱신을 위해 하나의 파일이 한 명 이상의 개발자에 의해 동시에 수정되거나 갱신되는 것을 허용하지 않습니다. 이를 위해 리파지토리 서버는 어떤 순간에도 각각의 파일에 대한 수정 권한을 오직 한 명의 개발자에게만 부여하고, 리파지토리 서버가 관리하는 파일을 갱신하기 위해서는 반드시 리파지토리 서버로부터 갱신 권한을 받도록 제한합니다.
이렇게 리파지토리 서비스가 정의하는 원칙에 따라 리파지토리 서버의 파일을 수정하고 갱신할 수 있는 권한을 받는 과정을 체크아웃이라하고, 반대로 수정된 파일을 리파지토리에 등록하고 갱신 권한을 반납하는 과정을 체크인이라고 합니다.

체크인

체크인은 서버 리파지토리에서 체크아웃 후 수정된 내용을 서버 리파지토리에 저장하고 갱신 권한을 반납하는 기능입니다. 리파지토리창에서 체크인할 파일을 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [Check In] 메뉴를 클릭합니다.



'체크인 다이얼로그'에서 체크인할 파일을 체크한 후 [Check In] 버튼을 클릭하여 체크인 합니다.



만일 체크아웃 상태인 파일을 수정하였을 경우 수정된 내용을 서버 리파지토리에 저장하고 갱신 권한을 반납하지 않고자 할 경우에는 'Keep checked out' 옵션을 체크한 후 [Check In] 버튼을 클릭하십시오. 이 옵션을 이용하면 체크아웃 상태인 파일을 서버 리파지토리에 저장하고 갱신권한을 계속 갖고자 할 경우 체크인을 하고 다시 체크아웃을 해야 하는 번거러움을 줄일 수 있습니다.

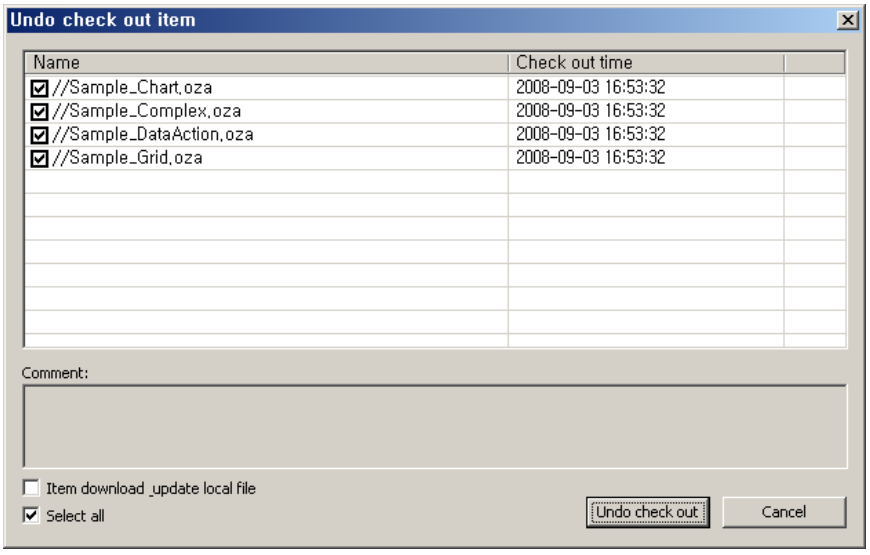
※ 주의사항 : 서버 리파지토리에서 파일을 체크인으로 변경한 상태에서 서버 리파지토리 연결을 해제한 후 다시 Admin 계정으로 'None' 타입의 서버 리파지토리를 연결할 경우 Admin 계정은 모든 파일에 대해 항상 체크아웃한 상태로 서버에 접속되지 때문에, 접속 전에 체크인으로 변경한 파일도 모두 체크아웃 상태로 자동 변경됩니다.

체크아웃

체크아웃은 서버 리파지토리에 저장된 파일의 수정을 위해 리파지토리 서버로부터 특정 파일에 대한 갱신 권한을 얻어오는 기능입니다.
리파지토리창에서 체크아웃할 파일을 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [Check Out] 메뉴를 클릭하고, '체크아웃 다이얼로그'에서 체크아웃할 파일을 체크한 후 [Check Out] 버튼을 클릭하여 체크아웃 합니다.

체크아웃 취소

체크아웃 취소는 체크아웃된 파일의 수정 내용을 서버 리파지토리에 반영하지 않고 단순히 갱신 권한을 반납하는 기능입니다.
리파지토리창에서 체크아웃 취소할 파일을 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [Undo Check Out] 메뉴를 클릭하고, '체크아웃 취소 다이얼로그'에서 체크아웃 취소할 파일을 체크한 후 [Undo Check Out] 버튼을 클릭하여 체크아웃을 취소합니다.



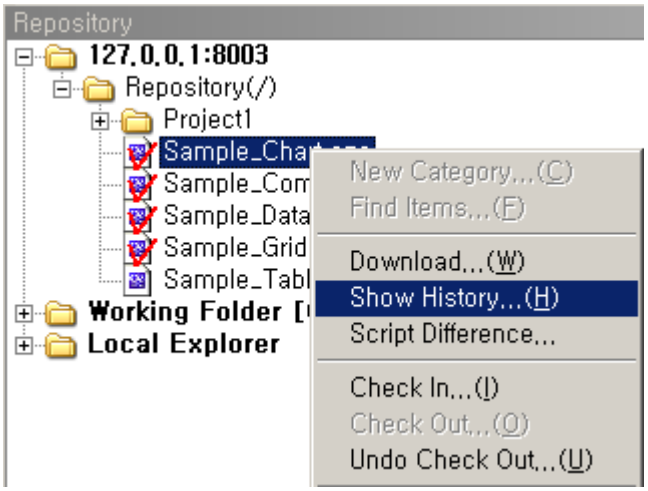
체크아웃을 취소할 때 만일 로컬 시스템의 작업 폴더에 선택한 파일을 다운로드하여 로컬 시스템의 작업 폴더의 파일을 최신 버전으로 갱신할 경우에는 'Download the items and update the local files' 옵션을 체크하십시오.

히스토리 보기

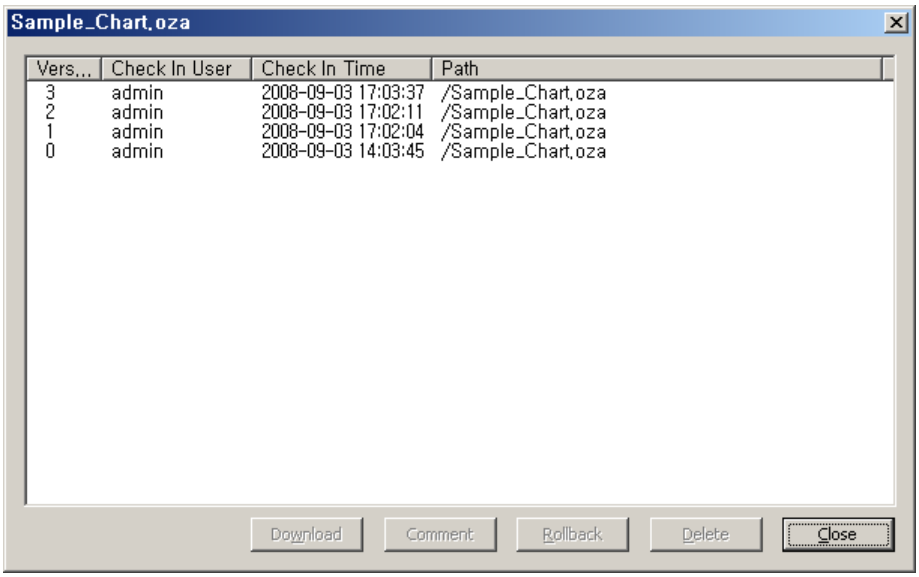
아이템을 체크인 또는 체크아웃하여 각 아이템별로 히스토리를 남길 경우 히스토리 보기 기능을 사용하여 아이템의 내역을 볼 수 있습니다.

아이템 히스토리 보기

서버 리파지토리에 있는 아이템을 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [Show History] 메뉴를 클릭합니다.



[Show History] 메뉴를 클릭하면 선택한 아이템의 히스토리를 나타낸 다이얼로그가 실행됩니다.



버전은 아이템의 히스토리 버전으로 아이템 체크인 시에 자동으로 부여되며, 낮은 숫자일수록 오래된 버전입니다.
특정 버전을 선택하여 다운로드, 주석 조회, 롤백, 히스토리 삭제 등의 기능을 실행할 수 있으며, 두 개의 버전을 선택하여 버전 간 스크립트를 비교할 수 있습니다.

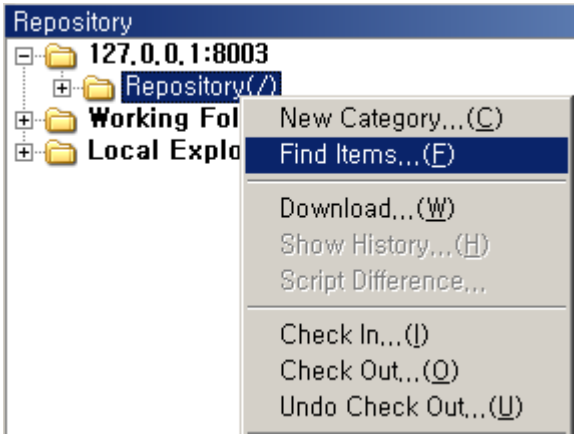
버전 간 스크립트를 비교하는 방법은 본 매뉴얼의 "IV. OZ Application Designer 사용 – 스크립트 비교" 부분을 참조하시기 바랍니다.

아이템 찾기

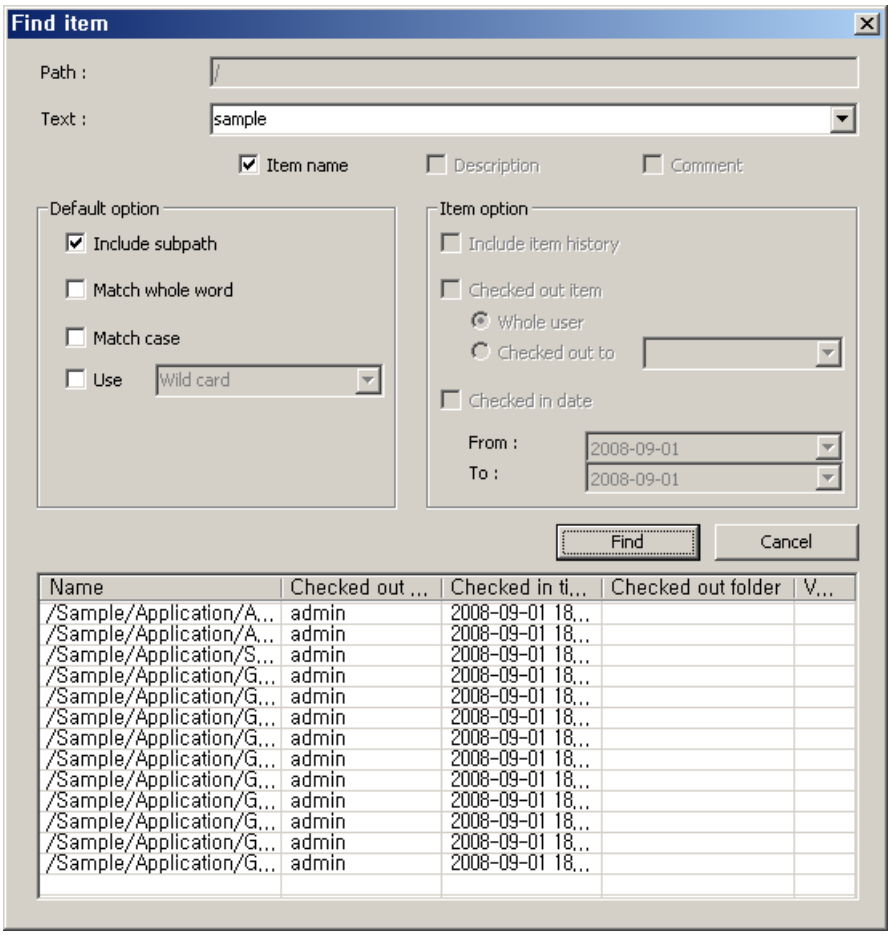
서버 리파지토리에 있는 아이템을 검색 조건을 지정하여 찾을 수 있습니다.

아이템 찾기

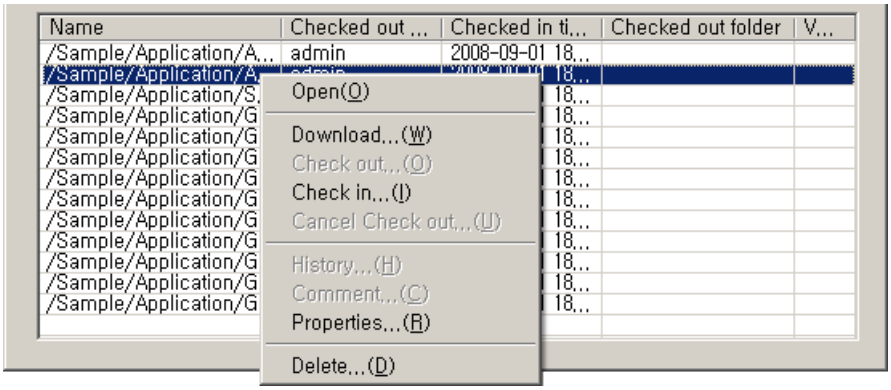
서버 리파지토리에서 검색할 카테고리를 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [Find Items] 메뉴를 클릭합니다.



[Find Items] 메뉴를 클릭하면 "아이템 찾기 다이얼로그"가 실행되고, 찾을 경로에는 선택한 카테고리가 자동으로 표시됩니다. 찾을 내용, 기본 옵션, 아이템 옵션 등을 설정한 후 [Find] 버튼을 클릭하면 검색된 아이템이 리스트 형태로 표시됩니다.



검색된 아이템을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴를 이용하여 체크인, 체크아웃, 체크아웃 취소 등의 기능을 실행할 수 있습니다.



아이템 찾기 관련 주의사항

- 아이템 찾기 기능은 서버 리파지토리에서만 사용 가능합니다. (로컬 작업 폴더에서는 사용할 수 없습니다.)

- 주석으로 찾을 경우 검색 옵션 중 "Include Item Hitem history" 옵션을 반드시 체크하시기 바랍니다.

- 이력 정보 기능을 사용하지 않는 일반적인 None 타입 서버의 경우 체크인, 체크아웃 등 히스토리 기능을 지원하지 않는 서버이므로, "아이템 찾기 다이얼로그"에서 "Item Option"이 모두 비활성화되어 표시됩니다.

- 버전 정보는 "주석", "아이템 히스토리 포함" 옵션이 체크된 경우에만 버전 정보를 가져와서 표시하며, 두 가지 옵션이 모두 체크되지 않은 경우에는 버전 정보를 표시하지 않습니다. 버전의 숫자가 낮을수록 오래된 버전이며, 버전 정보가 있는 히스토리 아이템의 경우 해당 아이템의 팝업 메뉴를 띄우면 [Download] 메뉴와 [Comment] 메뉴만 활성화됩니다.