OZ Report Training Book

Professional Enterprise Reporting Solution OZ Report

(주)포시에스

목 차

Chapter1. 오즈 소개

1. 오즈 소개 **6**
1.1 오즈 보고서 개발 및 운영 절차 6
1.2 오즈 제품 구성 7

2. 오즈 환경 **8**
2.1 운영 흐름도 8
2.2 오즈 서버 타입 9
2.3 시스템 요구사항 10

3. 디자이너 설치 및 실행 **11**
3.1 설치 11
3.2 실행 12
3.3 라이선스 적용 및 버전 확인 13

Chapter2. 오즈 Overview

1. 교재 샘플 DB 설정 **16**
2. 쿼리 작성 **17**
3. 리포트 디자인 **19**

Chapter3. 쿼리 디자이너

1. 사용 환경 **24**
1.1 쿼리 디자이너 화면 구성 24
1.2 주요 옵션 설정 25
1.3 작업 폴더 26
1.4 오즈 데이터 트리 27

2. 데이터 소스 **28**
2.1 데이터 스토어, 데이터 셋, ODI 정의 28
2.2 데이터베이스 스토어 추가 29
2.3 초기화 SQL과 종료 SQL 30
2.4 사용자 정의 데이터베이스 연결 31
2.5 동적 데이터베이스 연결 32
2.6 테이블 정보 33

3. 데이터 셋 **34**
3.1 쿼리 작성 34
3.2 마스터 셋 35
3.3 사용자 지정 패러미터 36

4. 스크립트 **38**
4.1 스크립트 소개 38
4.2 스크립트 쿼리 39
4.3 수식 필드 40

목 차

Chapter4. 리포트 디자이너

1. 사용 환경 **42**
1.1 리포트 디자이너 화면 구성 42
1.2 환경 설정 44

2. 리포트 **46**
2.1 리포트 템플릿 46
2.2 리포트 컴포넌트 48
2.3 백그라운드·포그라운드 모드 49
2.4 컴포넌트 선택, 이동 51
2.5 조회 툴바 52

3. 데이터 셋 연결 **53**
3.1 ODI 추가 53

4. 밴드 **54**
4.1 밴드 개요 54
4.2 타이틀, 페이지 헤더·풋터, 테일 밴드 55
4.3 데이터 밴드와 관련 종속 밴드 56
4.4 써머리 밴드, 더미 밴드 63
4.5 서브 리포트 밴드 64
4.6 사이드 밴드, 데이터 익스팬더 밴드 65

5. 라벨 **66**
5.1 라벨 개요 66
5.2 라벨 동적 속성 67
5.3 라벨 서식 72
5.4 목차 73
5.5 그룹 라벨 75
5.6 요약 라벨 77
5.7 수식 라벨 78
5.8 시스템 라벨 79
5.9 패러미터 라벨 80
5.10 이미지 라벨 82
5.11 바코드 라벨 84
5.12 HTML 라벨 85
5.13 노트 라벨 86

6. 멀티플렉스 **87**
6.1 멀티플렉스 개요 87
6.2 테이블 88
6.3 고정 테이블 91
6.4 크로스탭 93
6.5 차트 97
6.6 리전 107

7. 스크립트 **109**
7.1 스크립트 소개 109
7.1 스크립트 활용 113

목 차

Chapter5. 보고서 유형별 실습

1. 리스트 보고서 **116**
1.1 목표 보고서 정의 116
1.2 보고서 작성 가이드 117

2. 리스트 보고서(변형) **118**
2.1 목표 보고서 정의 118
2.2 보고서 작성 가이드 119

3. 고정 폼 보고서 **121**
3.1 목표 보고서 정의 121
3.2 보고서 작성 가이드 122

4. 다단 보고서 **124**
4.1 목표 보고서 정의 124
4.2 보고서 작성 가이드 125

5. 통계표 보고서(수직) **126**
5.1 목표 보고서 정의 126
5.2 보고서 작성 가이드 127

6. 통계표 보고서(복합) **128**
6.1 목표 보고서 정의 128
6.2 보고서 작성 가이드 129

7. 차트 보고서(막대) **130**
7.1 목표 보고서 정의 130
7.2 보고서 작성 가이드 131

8. 차트 보고서(사용자) **132**
8.1 목표 보고서 정의 132
8.2 보고서 작성 가이드 133

9. 차트 보고서(누적) **135**
9.1 목표 보고서 정의 135
9.2 보고서 작성 가이드 136

10. 드릴다운 보고서 **138**
10.1 목표 보고서 정의 138
10.2 보고서 작성 가이드 140

11. 그룹 보고서 **141**
11.1 목표 보고서 정의 141
11.2 보고서 작성 가이드 142

12. 마스터디테일 보고서 **143**
12.1 목표 보고서 정의 143
12.2 보고서 작성 가이드 144

목 차

Chapter6. 보고서 배포 및 운영

1. 보고서 업로드 **146**
1.1 리파지토리 서버 접속 146
1.2 보고서 업로드·다운로드 147
1.3 보고서 체크인·체크아웃 148

2. DB 커넥션 풀 생성 **149**
2.1 오즈 서버의 DB 연결 정보 설정 149

3. 보고서 출력 **151**
3.1 배포 마법사 151
3.2 뷰어 기능 152
3.3 뷰어 기본 옵션 153
3.4 뷰어 고급 옵션 157
3.5 UI 연동 160

Chapter7. 기타 데이터 처리

1. 서버 바인딩 데이터 **162**
1.1 HTTP 스토어 162
1.2 XML 스토어 164
1.3 사용자 데이터 스토어 166

2. 뷰어 바인딩 데이터 **168**
2.1 데이터 서비스 168
2.2 FX 데이터 172

Chapter8. 오즈 e-Form

1. OZ e-Form 소개 **176**
2. OZ e-Form 기능 **177**
3. OZ e-Form Exercise **178**

※ 교재 샘플 소개 **182**

Chapter1. 오즈 소개

1. 오즈 소개
2. 오즈 환경
3. 디자이너 설치 및 실행

가. 보고서 개발 절차

1단계 쿼리 디자인

- 데이터 스토어 추가 (데이터베이스 연결)
- 데이터 셋 추가 (쿼리 작성)
- 데이터 정보 파일 저장 (ODI 파일)

2단계 리포트 디자인

- 데이터 정보 파일 추가 (ODI 파일)
- 보고서 폼 디자인
- 각 컴포넌트의 속성 설정
- 컴포넌트와 데이터 정보 연결
- 리포트 파일 저장 (OZR 파일)

3단계 보고서 미리보기

- 개발한 보고서 확인

나. 보고서 운영 절차

1단계 오즈 서버 구동

- 데이터베이스 연결 정보 추가
- 오즈 서버 실행

2단계 서버로 ODI, OZR 전송

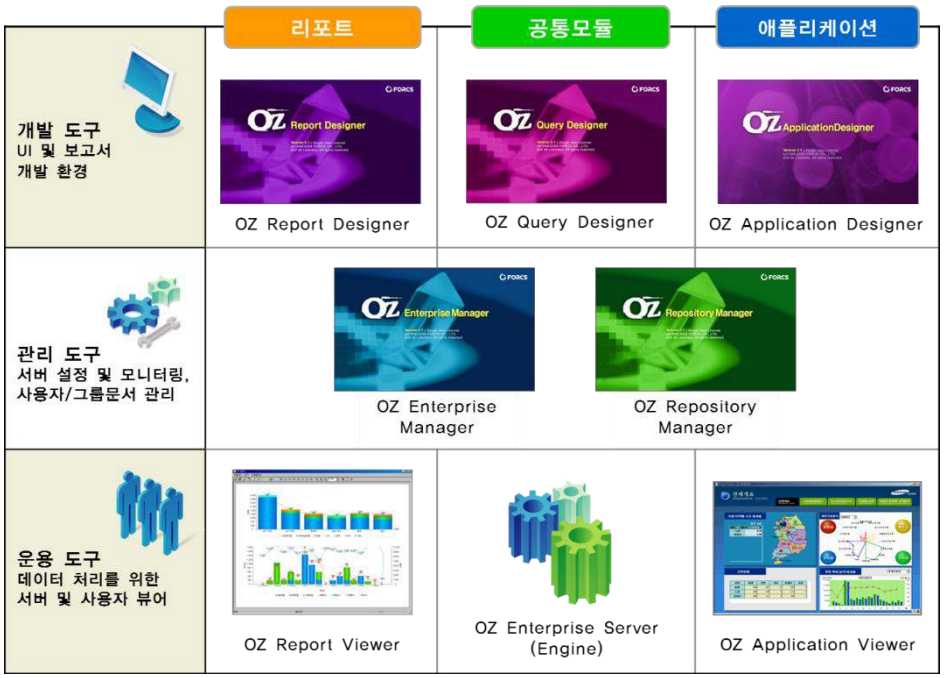
- 오즈 서버 리파지토리 연결
- 보고서 파일 업로드

3단계 오즈 뷰어 호출 소스 작성

- 오즈 뷰어 설치 태그 작성
- 보고서 호출 태그 작성

4단계 보고서 실행

- 완성된 보고서 확인 및 인쇄, 다른 포맷으로 저장



[오즈 제품 구성도]

- **OZ Report Designer**
  보고서 폼을 디자인하고 쿼리 디자이너에서 정의한 데이터 셋과 결합하여 보고서 작성

- **OZ Query Designer**
  리포트 디자이너에서 사용할 데이터 셋 정의 및 관리

- **OZ Application Designer**
  화면 폼을 디자인하고 스크립트로 로직을 작성하여 클라이언트의 UI(User Interface: 사용자 인터페이스) 구현

- **OZ Enterprise Manager**
  - 스케줄러 서버가 수행할 태스크 등록, 관리
  - 복수개의 오즈 서버, 오즈 스케줄러 서버에 대한 중앙 집중적 관리 기능 제공

- **OZ Repository Manager**
  오즈 서버에 저장되어 있는 보고서 폼 파일(*.OZR)과 데이터 셋 파일(*.ODI)을 관리

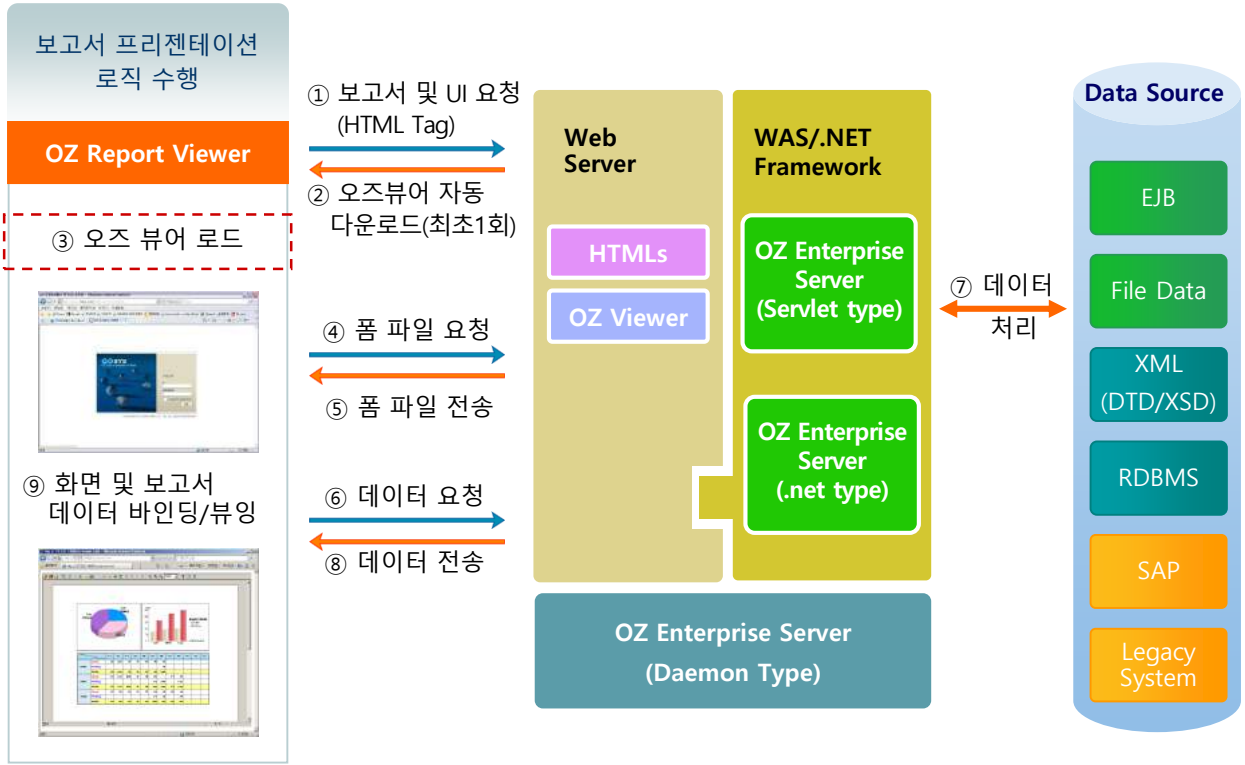
- **OZ Report Viewer**
  - ActiveX, Applet, Flash, HTML5, .Net, iOS, Android Viewer 제공 (뷰어 종류별 라이선스 별도임)
  - PDF, HTML 등의 포맷으로 보고서를 보여주는 서버 바인딩(자바/.NET) 제품 제공
  - 서버에서 받은 보고서 폼과 데이터 모듈을 결합하여 출력 양식으로 변환
  - 출력 양식의 미리보기 및 파일 저장, 프린트 출력 기능 제공

- **OZ Application Viewer**
  - ActiveX 형태로 웹 서버를 통해 자동 다운로드
  - UI 형태로 사용자의 제어에 따라 데이터 CRUD 처리

- **OZ Server**
  - 오즈 보고서 폼을 저장, 관리하고 이것을 클라이언트에 전송
  - 데이터베이스 쿼리 결과에 대한 데이터 셋을 생성하고 이것을 클라이언트에 전송
  - 오즈 리파지토리 관리
  - 독립 구동형인 Daemon Type과 WAS 연동형인 Servlet Type, .NET Framework 기반으로 구동하는 .NET Type 으로 구분

- **OZ Scheduler Server**
  주기를 지정하여 정해진 시간에 자동으로 보고서 생성, 배포

2.1 운영 흐름도



[오즈 운영 흐름도]

■ **흐름도**

① 웹 브라우저에서 보고서 설치·실행 HTML Tag가 호출됩니다.

② 웹 서버에 있는 오즈 뷰어가 사용자 PC에 다운로드 되어 자동 설치됩니다.

③ 설치된 오즈 뷰어가 실행됩니다.

④ 오즈 뷰어는 오즈 서버에 보고서 폼 파일을 요청합니다.

⑤ 오즈 서버는 폼 파일을 오즈 뷰어에게 전송합니다.

⑥ 오즈 뷰어는 오즈 서버에 데이터를 요청합니다.

⑦ 오즈 서버는 데이터 소스와의 연동을 통해 데이터 모듈을 생성합니다.

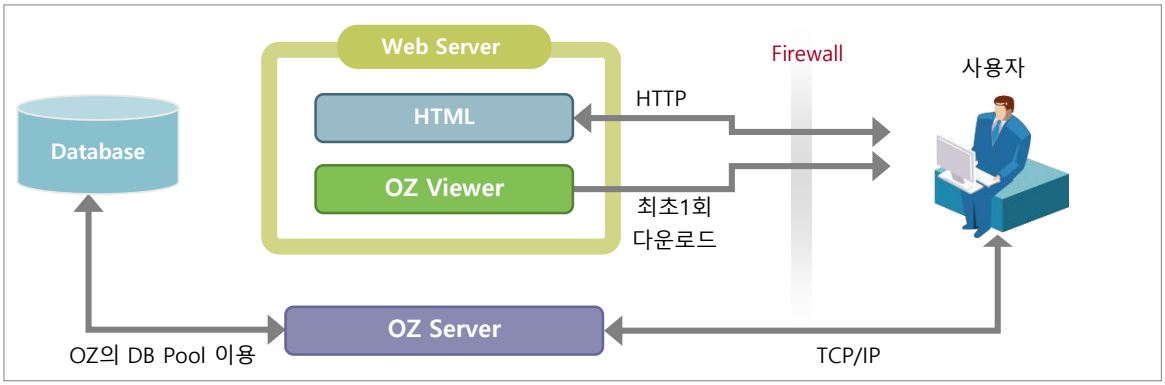
⑧ 오즈 서버는 데이터 모듈을 오즈 뷰어에게 전송합니다.

⑨ 오즈 뷰어는 폼 파일과 데이터를 바인딩 해 보고서를 보여줍니다.

■ **오즈 서버 타입**

- 데몬 : Stand-alone 형태의 독립서버 구조
- 서블릿 : WAS(Web Application Server)와 연동되는 구조
- 닷넷 : .NET Framework와 연동되는 구조

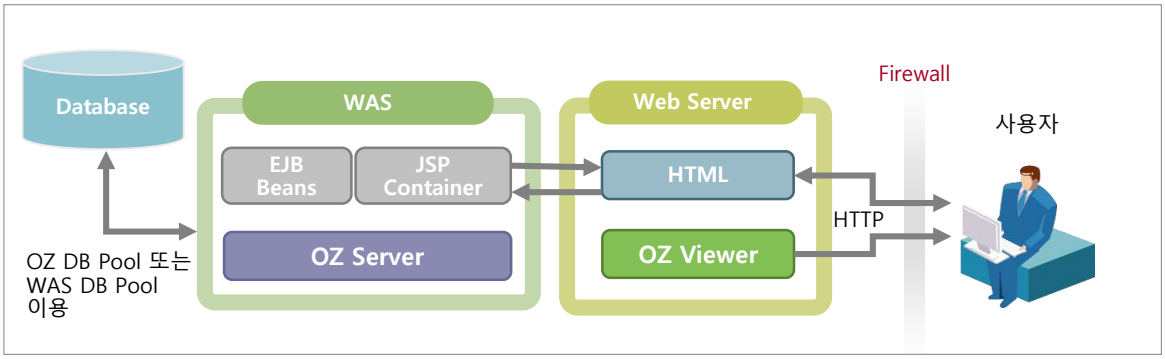
가. 데몬 타입



[Standalone 형태의 독립 서버 구조]

- 오즈 서버는 독립적으로 구동되어, 별도 포트를 사용합니다.
- 오즈 뷰어는 웹 서버 포트를 공유하여 다운로드 됩니다.
- 오즈 서버의 자체 DB Pool을 이용합니다.

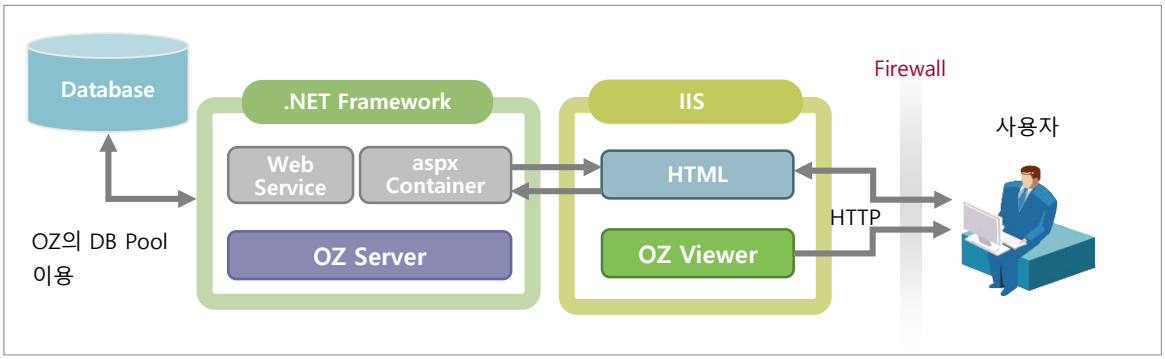
나. 서블릿 타입



[Web Application Server와 연동되는 구조]

- WAS(Web Application Server)의 포트를 공유하여 오즈 뷰어를 다운로드 하거나 오즈 서버에 연결합니다.
- 오즈 서버의 자체 DB Pool 또는 WAS의 DB Pool을 이용합니다.

다. 닷넷 타입



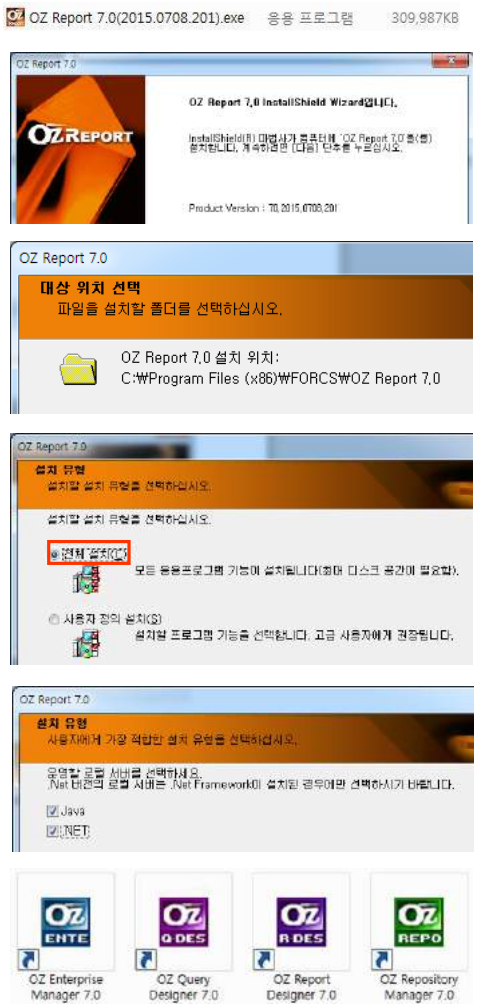
[.NET Framework와 연동되는 구조]

- IIS가 사용하는 포트를 공유하여 오즈 뷰어를 다운로드 하거나 오즈 서버에 연결합니다
- 오즈 서버의 자체 DB Pool을 이용합니다.

❖ **제품별 시스템 요구사항**

제품	요구사항	
OZ Server	자바 (Servlet /Daemon)	JRE(Java Runtime Environment) 1.5.X 이상을 지원하는 모든 운영 체제 (Windows, Linux, HP-UX, Solaris, AIX, Unixware, Mainframe 등)
	.Net	(Windows, Linux, HP-UX, Solaris, AIX, Unixware, Mainframe 등) .NET Framework 2.0 이상을 지원하는 모든 운영체제
Report Viewer	ActiveX	• Windows XP 이상에 설치된 Internet Explorer 8.0 이상 • Windows XP 이상에 Mozilla Firefox 3.0.1 이상 • Windows XP 이상에 Apple Safari 5.0 이상 • Windows XP 이상에 Google Chrome 6.0 이상 • Windows XP 이상에 Opera 10 이상
	Applet	JRE 1.6 이상을 지원하는 운영 체제 및 브라우저
	Flash	Adobe Flash Player 11.0을 지원하는 운영 체제 및 브라우저 (상세내용 참고 : http://www.adobe.com/kr/products/flashplayer/systemreqs/)
	HTML5	HTML5 표준을 지원하는 운영체제 및 웹 브라우저
	.Net	.NET Framework 3.5 Service Pack 1 이상 지원하는 환경 (상세내용 참고 : http://msdn.microsoft.com/en-us/library/bb882520(v=VS.90) .aspx)
	서버 바인딩	• 자바 서버의 경우 JRE 버전이 1.5 이상인 서블릿 환경 • .NET 서버의 경우 Microsoft .NET Framework 3.5 이상을 설치할 수 있는 환경
Query Designer / Report Designer	• OS : Windows Vista 이상 • JRE 1.5 이상, .Net Framework 2.0 이상 • CPU : Dual Core 2.80GHz 이상 • Main Memory : 2GB 이상 • HDD : 최소 500MB 이상의 빈 공간 필요 • 모니터 : 컬러 XGA (1920 x 1080 이상의 해상도)	

가. 설치 방법



① 오즈 디자이너 설치 파일 실행

② '다음' 버튼 클릭하고 '사용권 계약' 동의

③ 설치 경로 확인

④ '전체 설치' 선택

⑤ 로컬 서버 타입 선택

※ 로컬 서버 타입은 오즈 서버와 같은 타입을 선택합니다.
오즈 서버 타입을 모르는 경우 'Java', '.Net' 모두 선택합니다.

⑥ 설치 완료되면 바탕화면과 시작 메뉴에 아이콘 생성됨

나. 기본 설치 경로 및 주요 파일

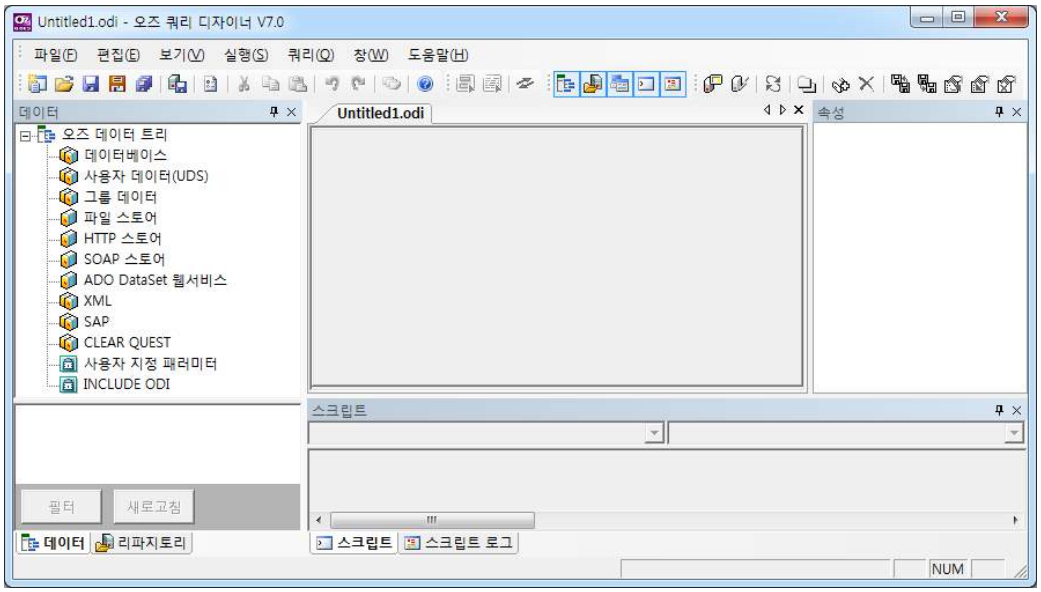
제품	기본 설치 경로
OZ Report Designer	%ProgramFiles%\FORCS\OZ Report 7.0\OZ Report Designer 7.0 (이하%OZD_HOME%)
OZ Query Designer	%ProgramFiles%\FORCS\OZ Report 7.0\OZ Query Designer 7.0 (이하%OZD_HOME%)

폴더	설명
%OZD_HOME%	- dbconfig.xml, dbconfig.xsd : DB 벤더 연결 정보 설정 파일
%OZD_HOME%\lib	XML파서, HTTPS/SSL관련 J2EE, LOG, SOAP, 메일, 압축, 오즈 서버 라이브러리 파일
%OZD_HOME%\license	- ozlicense.xml : 디자이너 라이선스 파일
%OZD_HOME%\UniversalDriver	벤더 별 JDBC Driver
%TEMP%₩OZ₩임시폴더	- Console.txt : 디자이너 내부 서버 로그 파일
내 문서₩OZ₩임시폴더	- QDConfig.ini : 쿼리 디자이너 설정 파일 - OZReportDesigner.ini : 리포트 디자이너 설정 파일

가. 쿼리 디자이너 실행

바탕 화면의 오즈 쿼리 디자이너 아이콘 을 실행합니다.

- 실행 파일 경로 : %ProgramFiles%\FORCS\OZ Report 7.0\OZ Query Designer 7.0\bin\OZQueryDesigner.exe

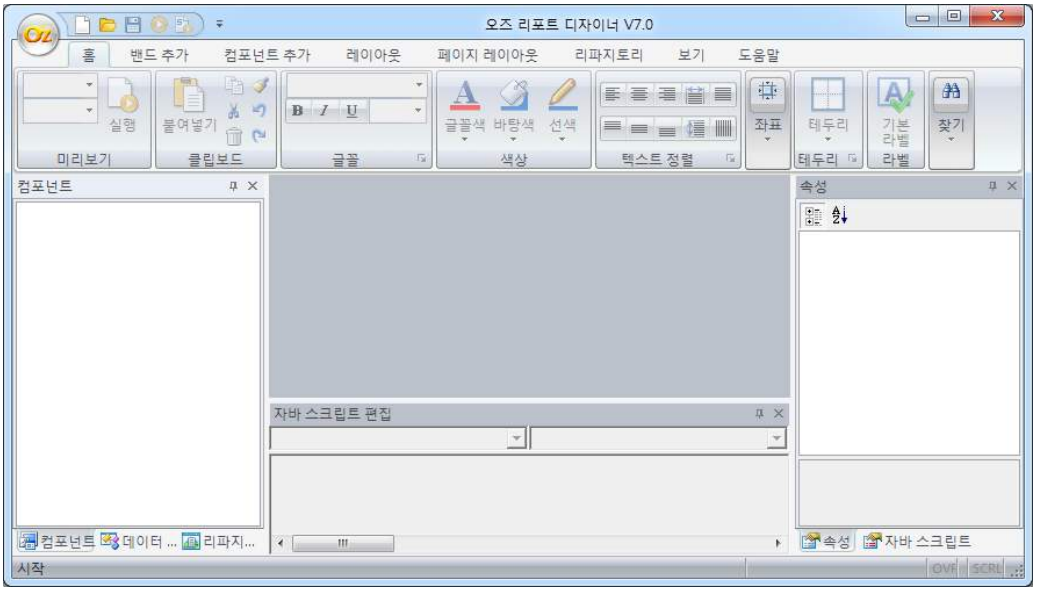


[쿼리 디자이너 실행 초기화면]

나. 리포트 디자이너 실행

바탕 화면의 오즈 리포트 디자이너 아이콘 을 실행합니다.

- 실행 파일 경로 : %ProgramFiles%\FORCS\OZ Report 7.0\OZ Report Designer 7.0\bin\OZReportDesigner.exe



[리포트 디자이너 실행 초기화면]

가. 라이선스 정책

디자이너 최초 설치 후 라이선스 파일 없이 3개월간 사용할 수 있습니다.

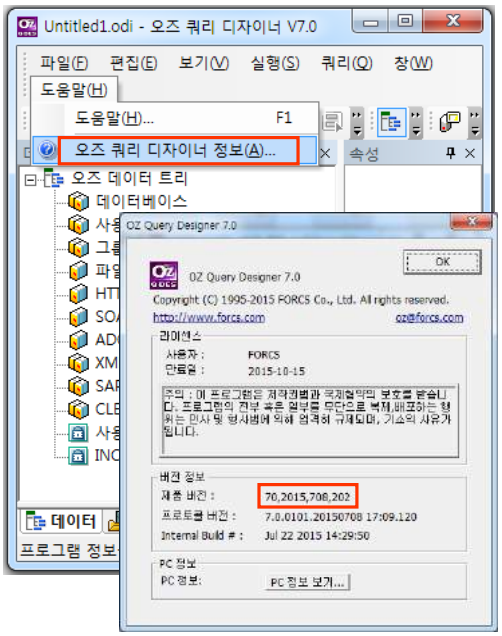
라이선스 파일을 적용하려면 설치되어 있는 각 디자이너 경로의 license 폴더에 ozlicense.xml 파일명으로 복사합니다.

- %ProgramFiles%\FORCS\OZ Report 7.0\OZ Query Designer 7.0\license\ozlicense.xml
- %ProgramFiles%\FORCS\OZ Report 7.0\OZ Report Designer 7.0\license\ozlicense.xml

나. 디자이너 버전 확인

쿼리디자이너 : 디자이너 실행 → '도움말' → '오즈 쿼리 디자이너 정보' 메뉴 선택

리포트디자이너 : 디자이너 실행 → 'OZ' 버튼 → '오즈 리포트 디자이너 정보' 메뉴 선택



[쿼리 디자이너 버전 확인]

[리포트 디자이너 버전 확인]

Chapter2. 오즈 Overview

1. 교재 샘플 DB 설정
2. 쿼리 작성
3. 리포트 디자인

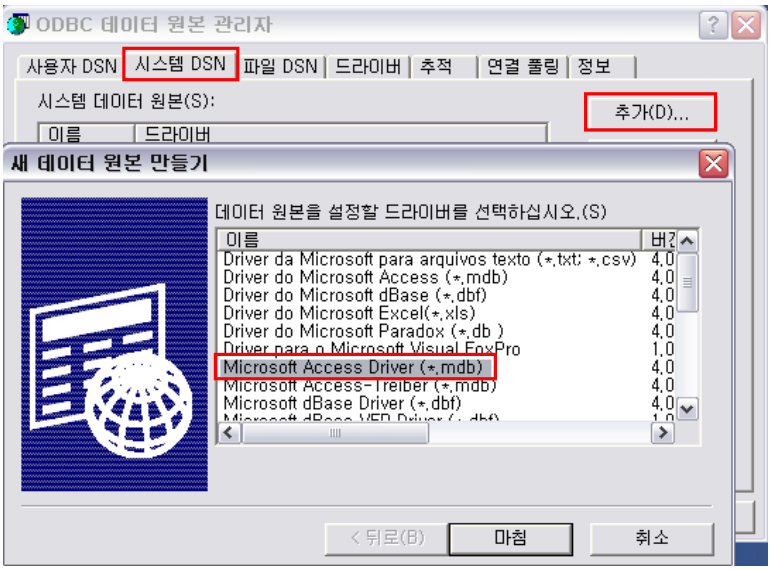
1. 교재 샘플 DB 설정

(1) 교재 첨부 파일 복사

교재 첨부 파일에서 'OZwork' 폴더를 '내 문서\OZRepository\OZwork' 폴더로 복사합니다.
교재에서 사용할 샘플 DB는 '내 문서\OZRepository\OZwork\data\OZwork.mdb' 파일입니다.

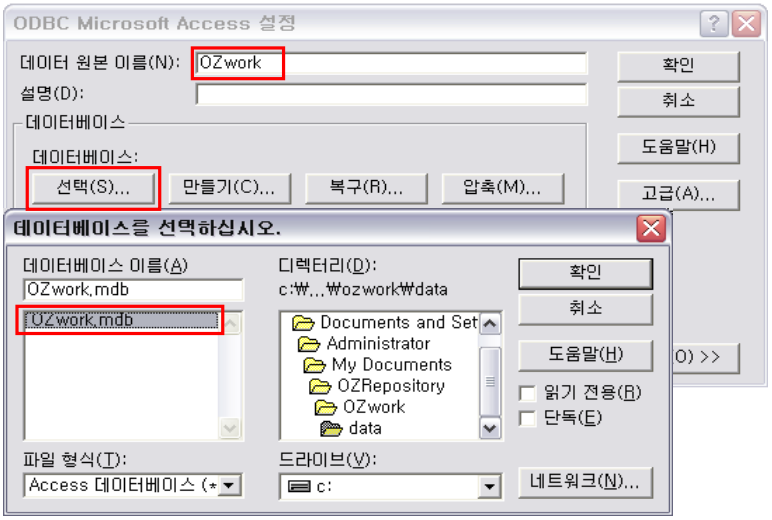
(2) MS Windows 에서 ODBC 데이터 원본 설정

다음은 교재의 샘플 DB를 ODBC로 이용하기 위해 설정하는 방법입니다.
64bit PC에서는 32bit용 데이터 원본(c:₩Windows₩SysWOW64₩odbcad32.exe)을 실행하셔야 합니다.



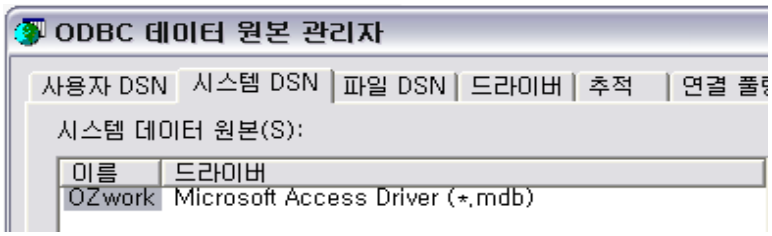
① '시작' → '설정' → '제어판' → '관리 도구' → '데이터 원본 (ODBC)' 실행

② '시스템 DSN' 탭 → '추가' 버튼 클릭 → 'Microsoft Access Driver(*.mdb)' 선택 → '마침' 버튼 클릭



③ '데이터 원본 이름'에 'OZwork' 입력 → '선택' 버튼 클릭 → 'OZwork.mdb' 파일 선택

④ '확인' 버튼 클릭

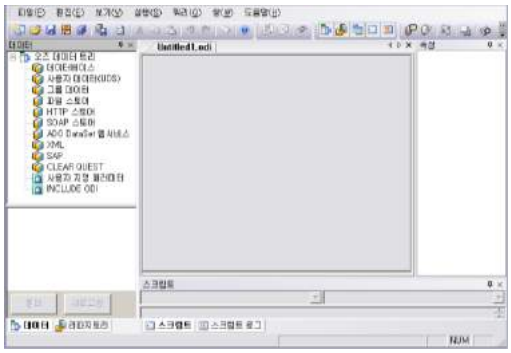


⑤ 목록에서 생성된 'OZwork' 확인

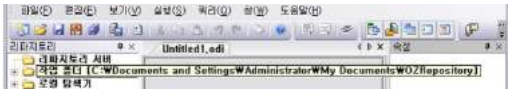
※ (2)의 설정은 mdb 에만 필요한 설정입니다. 타 DB에는 필요하지 않은 설정입니다.

2. 쿼리 작성

(1) 데이터베이스 연결

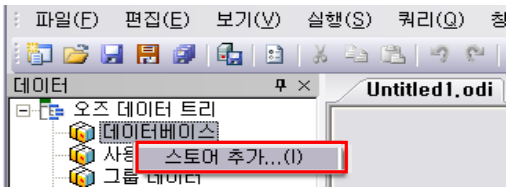


① '쿼리 디자이너' 실행

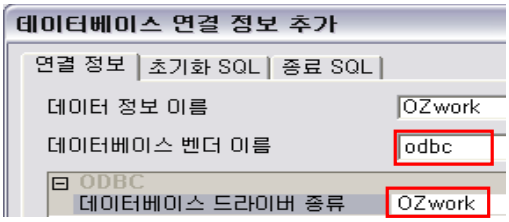


② '리파지토리' 창의 '작업 폴더' 확인

※ 작업 폴더는 ODI 파일 저장 위치가 됩니다.



③ '데이터' 창의 '데이터베이스' 팝업 메뉴에서 '스토어 추가' 선택

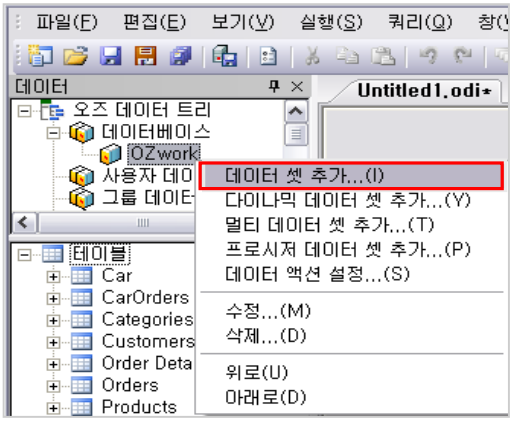


④ '데이터베이스 연결 정보 추가' 창의 '데이터베이스 벤더 이름'에 'odbc' 선택

⑤ '데이터베이스 드라이버 종류'에 'OZwork' 입력

⑥ '확인' 버튼 클릭

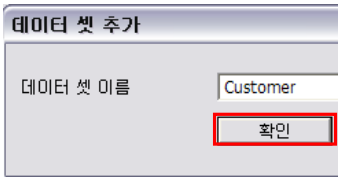
(2) 데이터 셋 추가



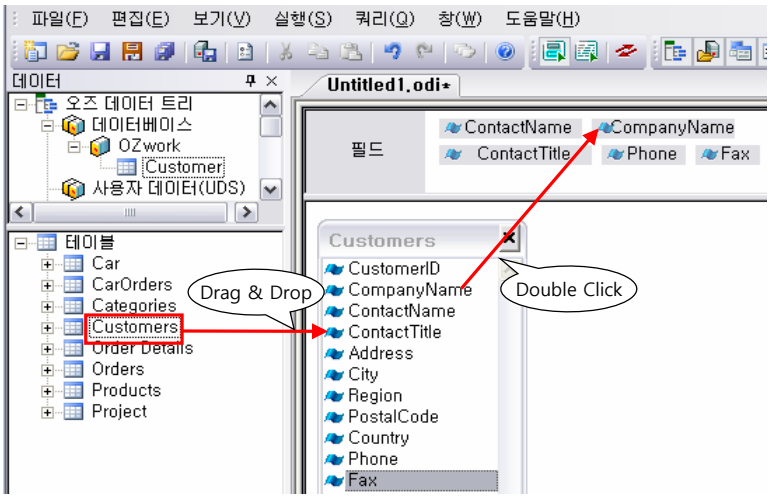
① 추가된 스토어의 팝업 메뉴에서 '데이터 셋 추가' 선택

② '데이터 셋 추가' 창의 '데이터 셋 이름'에 'Customer' 입력

③ '확인' 버튼 클릭

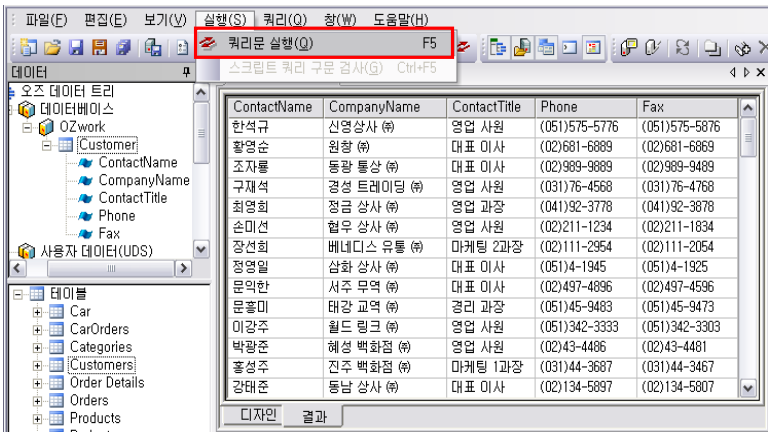


(3) 쿼리문 작성



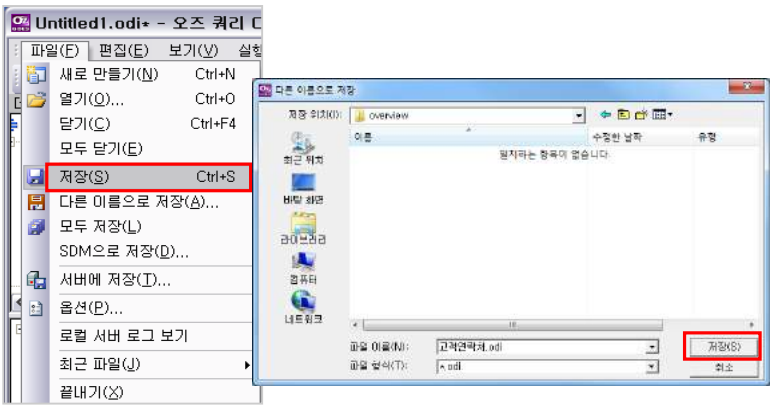
① 'Customers' 테이블을 디자인 영역에 드래그&드롭으로 추가

② 테이블의 필드를 더블 클릭으로 '필드' 창에 추가



③ '실행' → '쿼리문 실행' 메뉴 선택

(4) 쿼리 저장 (ODI 저장)

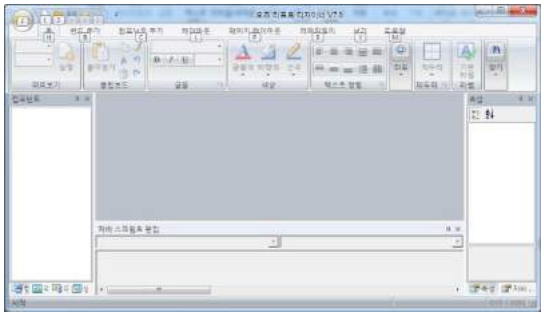


① '파일' → '저장' 메뉴 선택

② '파일 이름'에 '고객연락처.odi' 입력

- 저장 경로 : 내 문서\OZRepository \OZwork\overview\고객연락처.odi

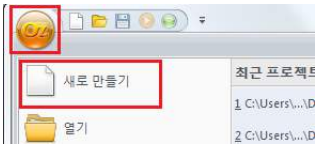
(1) 리포트 생성



① '리포트 디자이너' 실행

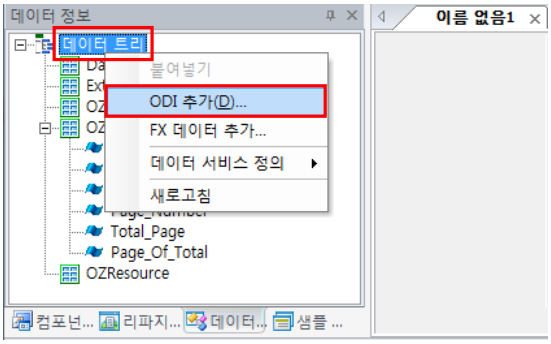
② '리파지토리' 창의 '작업 폴더' 확인

※ 작업 폴더는 OZR 파일 저장 위치가 됩니다.

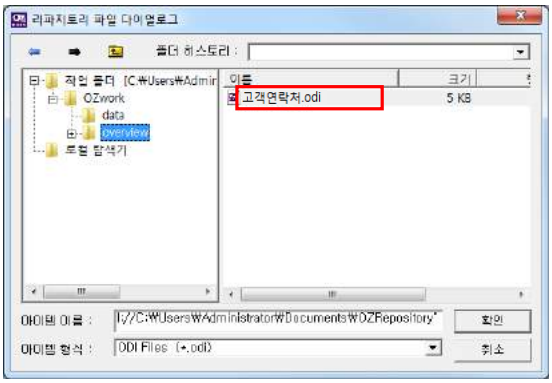


③ 'OZ'버튼 → '새로 만들기' 메뉴 선택

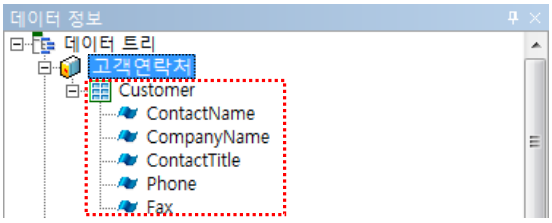
(2) ODI 추가 (데이터소스 연결)



① '데이터 정보' 창의 '데이터 트리' 팝업 메뉴에서 'ODI 추가' 선택

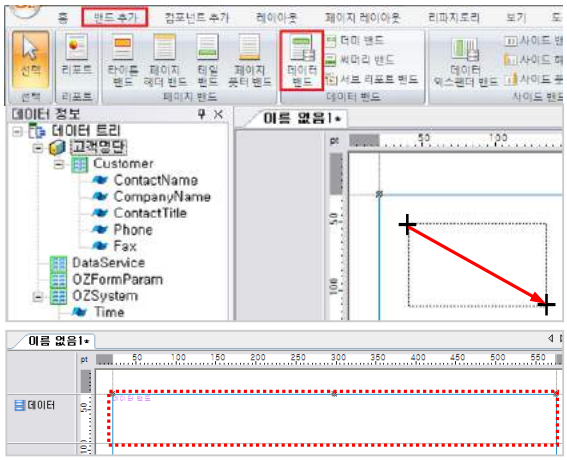


② '고객연락처.odi' 선택 후 '확인' 버튼 클릭



③ '데이터 정보' 창에 추가된 ODI의 데이터 셋이 표시됨

(3) 데이터 밴드 추가

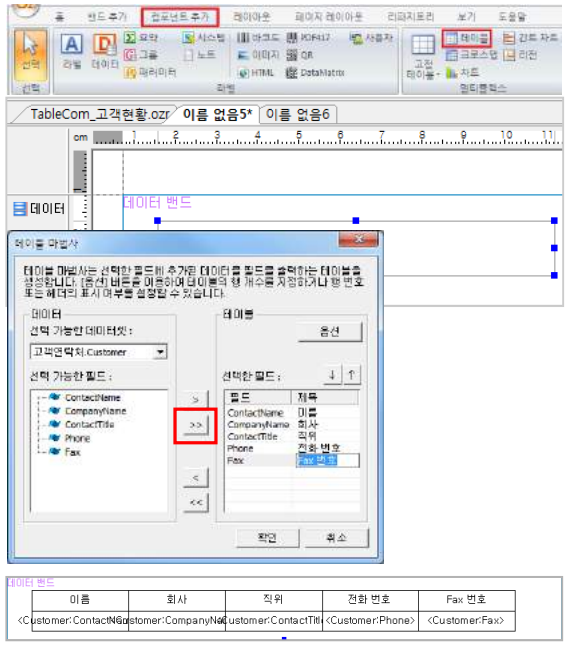


① 툴바의 '데이터 밴드' 아이콘 클릭

② 오른쪽 디자인 화면에 대각선으로 그림

③ 대각선 높이만큼 데이터 밴드가 그려짐

(4) 테이블 추가



① 툴바의 '테이블' 아이콘 클릭

② 데이터 밴드 위에 대각선 방향으로 그리면 '테이블 마법사'가 실행됨

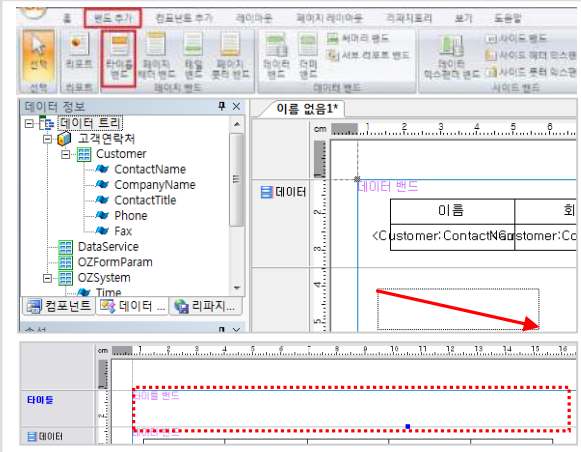
③ '테이블 마법사' 창에서 '선택 가능한 필드'의 모든 필드를 '선택한 필드'로 이동

④ '선택한 필드'의 '제목'을 더블 클릭한 후 한글 제목으로 변경

⑤ '확인' 버튼 클릭

⑥ 데이터 밴드 위에 테이블 그려짐

(5) 타이틀 밴드 추가

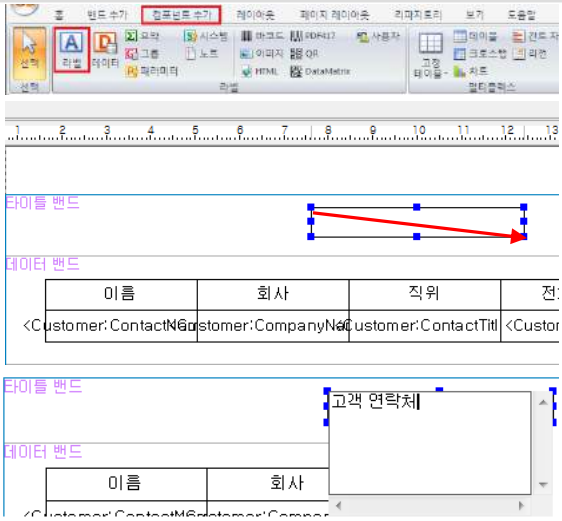


① 툴바의 '타이틀 밴드' 아이콘 클릭

② '데이터 밴드' 아래 대각선으로 그림

③ 대각선 높이 만큼 '타이틀 밴드' 추가됨

(6) 라벨 추가

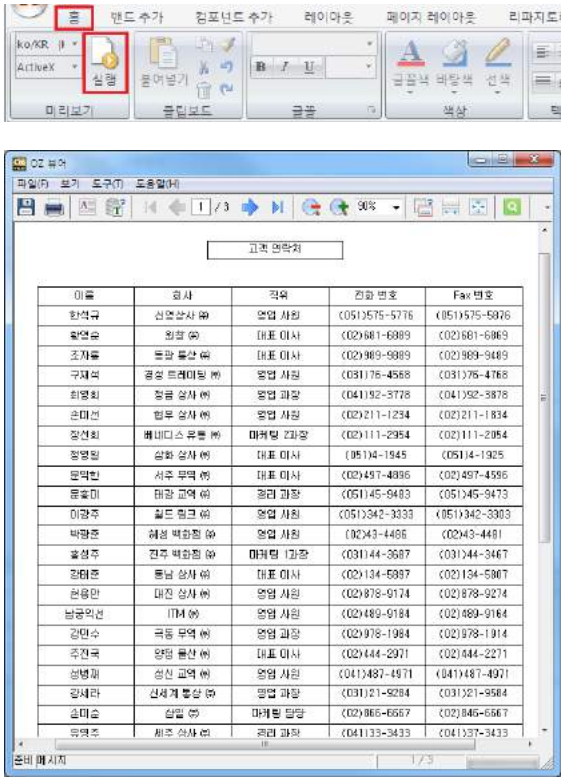


① 툴바의 '라벨' 아이콘 A 클릭

② '타이틀 밴드' 위에 대각선 방향으로 그리면 라벨 추가됨

③ 라벨을 더블 클릭한 후 '고객 연락처'라고 텍스트 입력

(7) 보고서 미리보기



① 툴바의 '실행' 아이콘 클릭

② 완성된 보고서 결과 확인

(8) 리포트 저장 (OZR 저장)

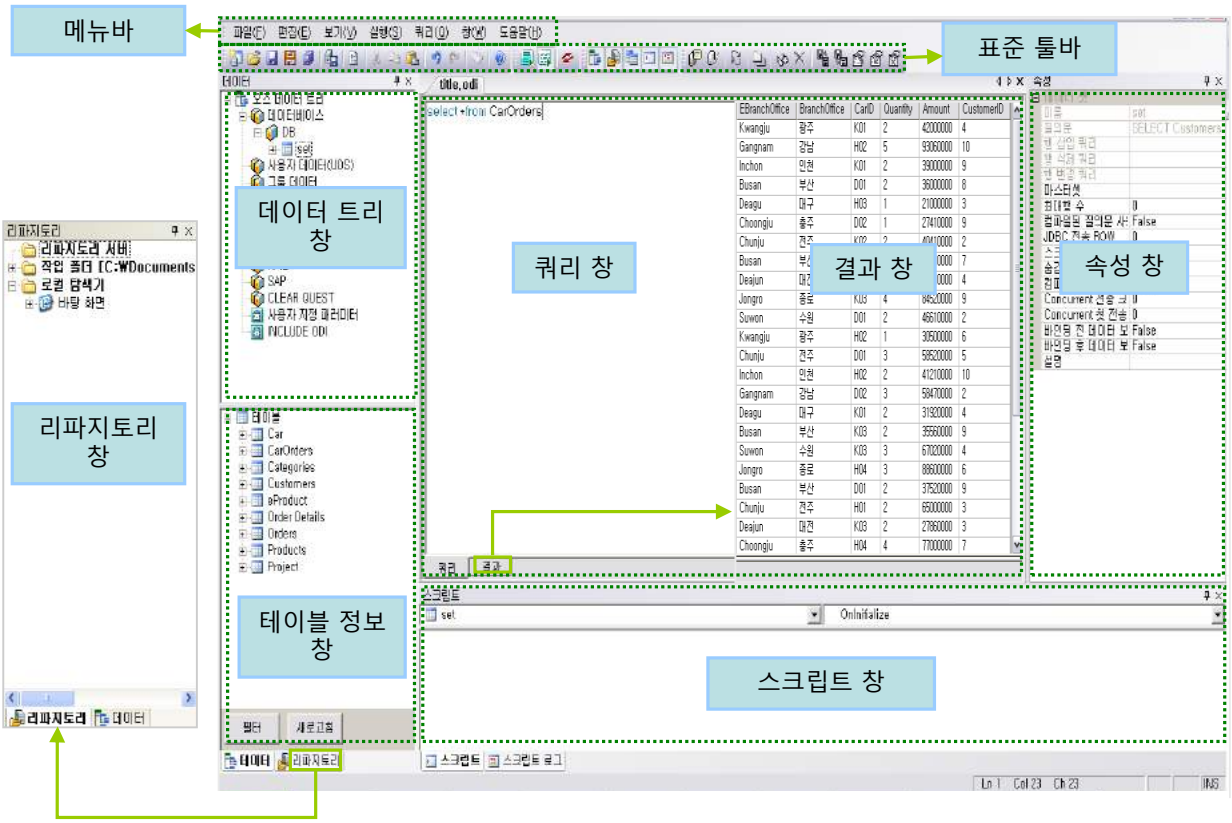
① 리포트 디자이너의 'OZ 버튼' → '저장' 메뉴 선택

② '파일 이름'에 '고객연락처.ozr' 입력

- 저장 경로 : 내 문서\OZRepository\OZwork \overview\고객연락처.ozr

Chapter3. 쿼리 디자이너

1. 사용 환경
2. 데이터 소스
3. 데이터 셋
4. 스크립트



[쿼리 디자이너 화면]

- **화면 설명**

a. 데이터 트리 창 : 사용자가 작성한 데이터베이스 정보 및 데이터 셋 정보 표시

b. 리파지토리 창 : 로컬 및 서버 리파지토리의 폴더 및 파일을 탐색기 형태로 표시

c. 쿼리 창 : SQL문 작성

d. 결과 창 : SQL문 실행 결과 표시

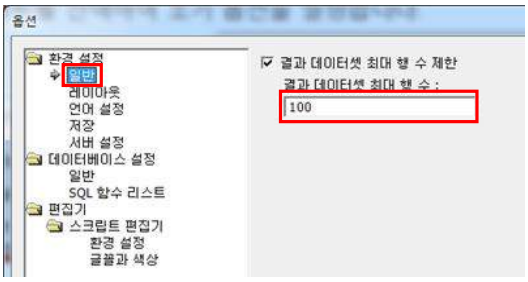
e. 속성 창 : 데이터베이스, 데이터 셋 등의 속성 표시

f. 테이블 정보 창 : 현재 연결된 데이터베이스의 조회 가능한 테이블 정보 표시

g. 스크립트 창 : 오즈 데이터 객체의 각 이벤트에 대한 스크립트 작성

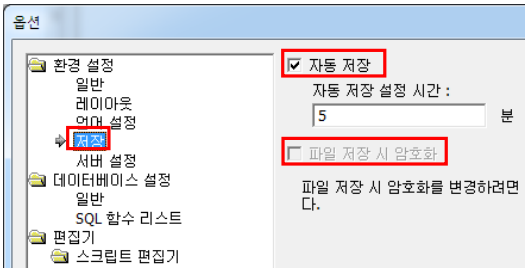
쿼리 디자이너의 '파일' → '옵션(P)' 메뉴를 선택하여 초기 옵션을 설정합니다.

가. 환경설정 - 일반



- **결과 데이터 셋 최대 행 수 제한** :
대용량 데이터를 가져올 때 속도 문제로 쿼리 결과를 보여주는 행 수를 제한합니다.
(쿼리 디자이너에서 실행할 때만 적용됨)

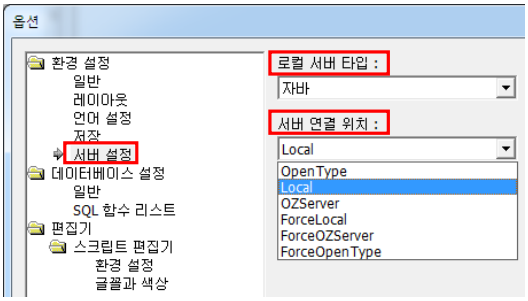
나. 환경설정 - 저장



- **자동저장** :
작업 중인 ODI 파일을 백업하는 옵션으로 지정된 시간마다 백업 파일을 생성합니다.

- **파일 저장 시 암호화** :
파일 저장 시 암호화하여 저장할지 여부를 설정합니다.

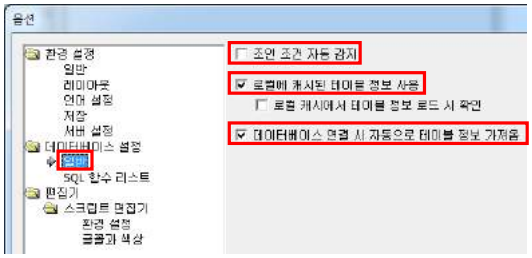
다. 환경설정 - 서버설정



- **로컬 서버 타입** :
디자이너 내부 서버 타입을 설정합니다. 오즈 쿼리 디자이너의 로컬 서버 타입은 자바, 닷넷 중 하나로 설정할 수 있습니다.

- **서버 연결 위치** :
쿼리문 실행 시 데이터를 바인딩 할 서버의 위치를 설정합니다. 예를 들어 OZServer로 설정하면 현재 연결된 리파지토리 서버를 통해 데이터를 가져옵니다.

라. 데이터베이스 설정 - 일반



- **조인 조건 자동 감지** :
데이터베이스에 설정된 테이블 간의 조인을 자동으로 감지하도록 설정합니다.

- **로컬에 캐시된 테이블 정보 사용** :
최초 데이터베이스 접속 시 가져온 테이블 정보를 로컬 캐시에 저장하고 다음 접속 시부터는 로컬에 저장된 테이블 정보를 사용할 것인지 여부를 설정합니다.

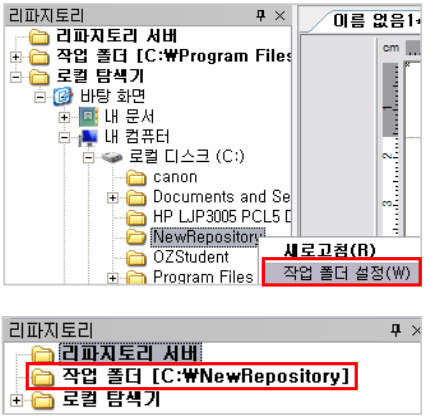
- **데이터베이스 연결 시 자동으로 테이블 정보 가져옴** :
데이터베이스 접속 시 테이블 정보 창에 테이블 목록과 데이터 필드 정보를 자동으로 가져올지 여부를 설정합니다.

가. 작업 폴더 정의

'작업 폴더'란 로컬에서 작업중인 보고서 파일을 저장하는 루트 파일 경로를 의미합니다.
디자이너의 기본 작업 폴더는 '내 문서\OZRepository' 입니다.

리포트에 ODI 파일이나 이미지 파일을 추가했을 때, 추가된 파일의 경로는 'ozp://카테고리명/파일명' 형식으로 설정됩니다. ozp 경로는 작업 폴더를 루트로 하는 상대 경로입니다.

나. 작업 폴더 변경

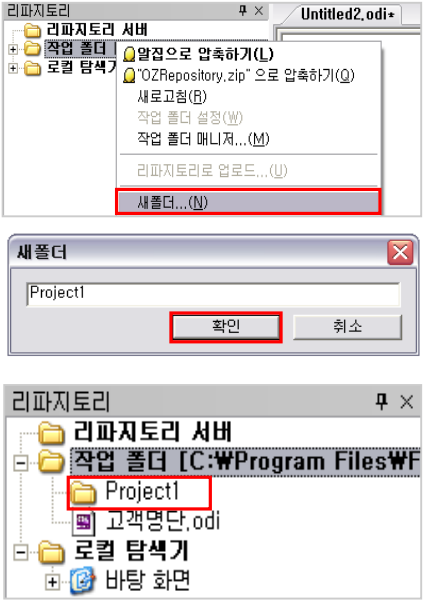


① '리파지토리' 창의 '로컬 탐색기' 트리에서 작업 폴더로 설정하고자 하는 폴더 선택

② 선택된 폴더의 팝업 메뉴에서 '작업 폴더 설정' 선택

다. 카테고리 생성

작업 폴더의 하위 폴더를 '카테고리'라고 합니다.

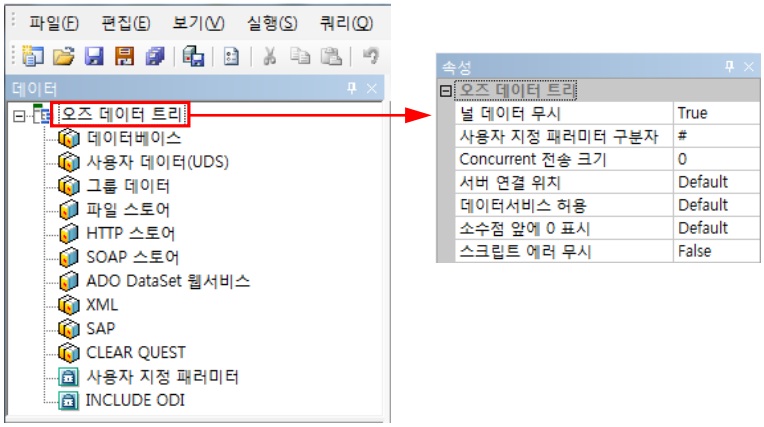


① 작업 폴더나 작업 폴더의 하위 폴더에 대한 팝업 메뉴에서 '새폴더' 선택

② '새폴더' 창에서 생성할 폴더명을 입력하고 '확인' 버튼 클릭

'데이터' 창의 '오즈 데이터 트리'는 오즈 쿼리 디자이너에서 지원하는 다양한 데이터 스토어들과 사용자 지정 패러미터 등을 트리 형태로 나타냅니다.

'오즈 데이터 트리'를 선택하여 아래 속성을 설정할 수 있습니다.



가. 널 데이터 무시

데이터 필드 값이 널 데이터일 때 처리하는 방법을 설정합니다.

- True : 데이터 필드 타입이 문자형일 경우는 ""(빈 문자)로, 숫자 형일 경우는 0으로 처리합니다.
- False : 데이터 값을 널 데이터(데이터 값이 없음을 나타냄)로 그대로 취급하며, 스크립트에서 null 상수와의 비교를 통해 널 데이터를 판단할 수 있습니다.

TIP

쿼리 디자이너에서 설정한 널 데이터 무시 속성은 데이터 셋이 생성되는 과정에서만 적용됩니다.
즉, 오즈 서버에서 데이터 셋을 생성하기 위한 스크립트 처리 등에서 적용되며, 오즈 뷰어 클라이언트에서 보고서를 바인딩할 때에는 적용되지 않습니다. 이 때에는 리포트 디자이너에서 설정한 널 데이터 무시 속성이 적용됩니다.

나. 사용자 지정 패러미터 구분자

패러미터 구분자를 설정합니다.

구분자가 '#'으로 설정된 경우 패러미터는 #OZParam.패러미터이름# 형식으로 사용합니다.

다. Concurrent 전송 크기

클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다.

설정된 값 만큼의 버퍼 단위로 데이터를 내보냅니다.

속성값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.

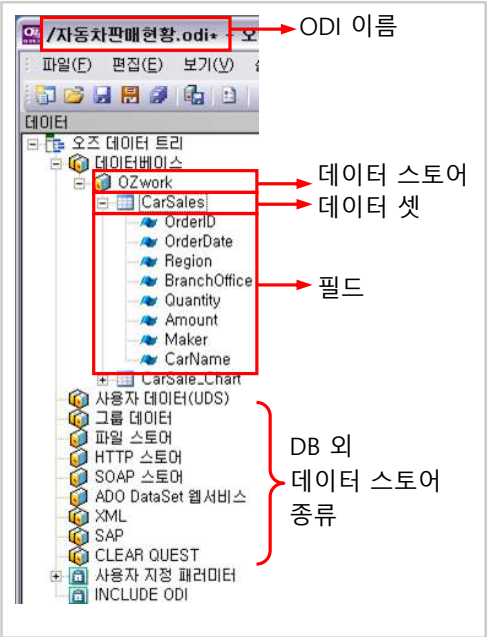
라. 서버 연결 위치

데이터 바인딩 시 연결할 서버를 설정합니다.

디자이너에 내장된 로컬 서버를 기본으로 사용합니다.

가. 데이터 스토어

데이터베이스를 비롯한 여러 가지 형태의 데이터 소스들을 데이터 스토어(Data Store)라고 합니다.



나. 데이터 셋

데이터 스토어로부터 필요한 데이터 필드를 선택하여 정의한 데이터 필드의 집합으로, 전체 데이터에 대한 정의와 각 필드들에 대한 정보를 포함하고 있습니다. 데이터 스토어의 종류에 따라 데이터 셋을 정의하는 방법이 다르며, 데이터베이스 스토어의 경우 쿼리를 사용하여 데이터 셋을 정의합니다.

다. ODI

쿼리 디자이너에서 작업한 내용 즉, 데이터 소스에 대한 연결 정보, 작성한 쿼리에 대한 정보, 데이터 소스에서 추출한 필드 정보 등을 저장하는 XML 형태의 파일입니다. 오즈 리포트 디자이너는 개발자가 디자인한 폼에 데이터를 매핑하기 위하여 ODI 파일을 사용합니다.

라. '데이터 스토어' – '데이터 셋' – '필드' 계층 구조

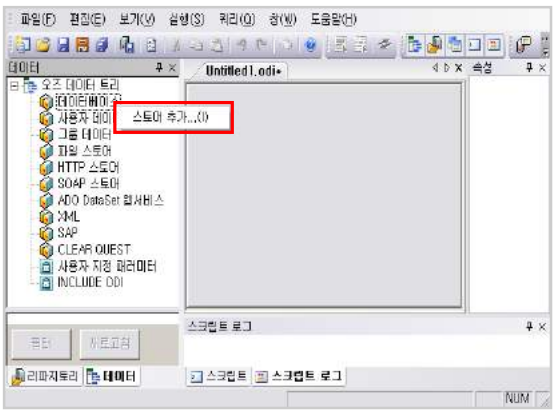
하나의 ODI 파일 안에 여러 개의 데이터 스토어를 가질 수 있으며, 하나의 데이터 스토어 역시 여러 개의 데이터 셋을 가질 수 있습니다.

마. 데이터 소스 종류별 기능 및 용도

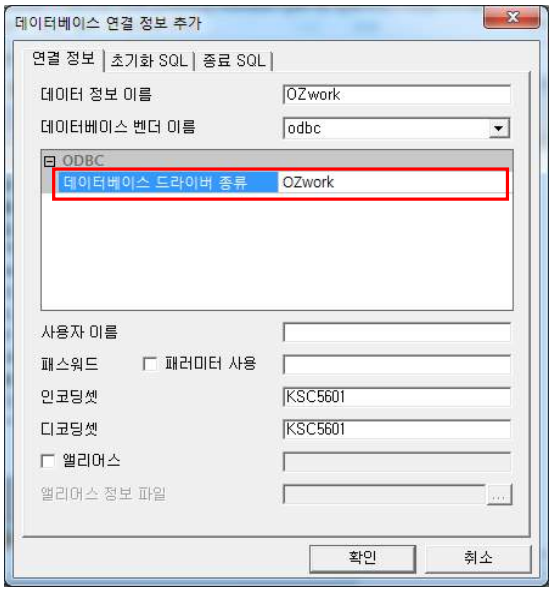
데이터 소스	설명
데이터베이스	JDBC 또는 ODBC/OLEDB를 지원하는 RDBMS와 연동하고, SQL을 이용해 데이터 셋을 생성합니다.
사용자 데이터 (UDS)	일반 파일 데이터(텍스트 파일, CSV, XML) 및 사용자 프로그램 데이터(EJB, Servlet, ASP, JSP, Stored Procedure)와 같은 다양한 형태의 외부 데이터 소스와 연동하기 위한 인터페이스입니다. J2EE기반의 WAS 및 .NET환경을 지원합니다.
그룹 데이터	데이터베이스, UDS 등의 다른 데이터 소스로부터 생성한 데이터 셋을 특정 필드들을 기준으로 분할하여 새로운 데이터 셋 들을 만들 수 있는 데이터 스토어입니다. 일반적으로 그룹 데이터 스토어는 하나의 데이터 셋을 마스터/디테일 데이터 셋으로 재구성할 때 사용됩니다.
파일 스토어	서버의 로컬 파일 경로에 있는 XML(DTD가 없는 일정한 형식의 XML 파일), CSV 형식의 파일 데이터에 대한 표준화된 연동 방식입니다.
HTTP 스토어	URL로 접근 가능한 XML(DTD가 없는 일정한 형식의 XML 파일), CSV 형식의 파일 데이터에 대한 표준화된 연동 방식입니다.
XML	DTD/XSD를 이용하는 XML 형식 파일 데이터를 데이터 소스로 사용하는 연동 방식입니다.
SAP	SAP R/3에 정의된 RFC 실행 결과 데이터를 데이터 소스로 사용하는 연동 방식입니다.
CLEAR QUEST	형상관리 툴인 IBM Rational Clear Quest 프로그램에서 생성한 DB를 데이터 소스로 사용하는 연동 방식입니다.
SOAP	웹 서비스를 통하여 얻어온 데이터를 데이터 소스로 사용하는 연동 방식입니다.

2.2 데이터베이스 스토어 추가

여러 종류의 데이터 소스 중에서 다음은 데이터베이스 스토어를 추가하는 방식입니다.
데이터베이스 연결은 JDBC/ODBC 드라이버를 이용합니다.

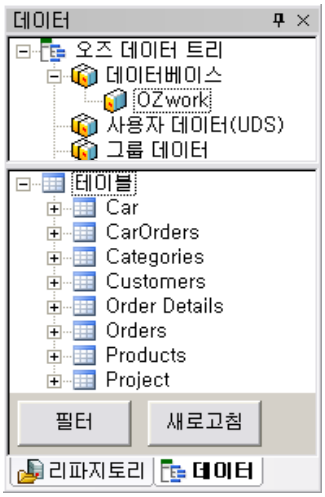


① '데이터'창의 '데이터베이스' 팝업 메뉴에서 '스토어 추가' 클릭



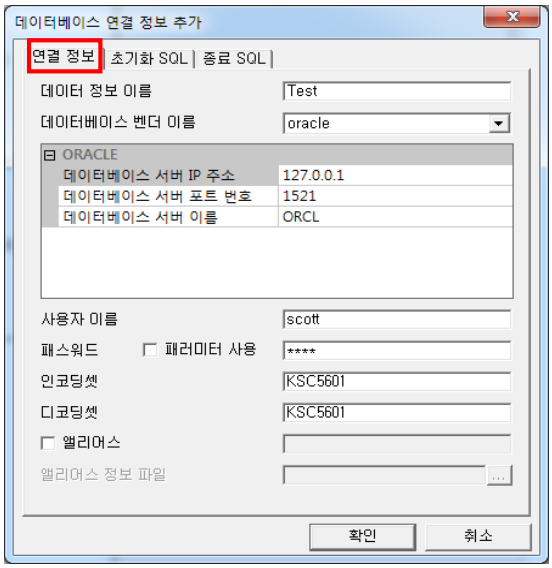
② '데이터베이스 연결 정보 추가'창의 '데이터베이스 벤더 이름'에 'oracle/mssql/db2/odbc/informix/sybase/unisql /mysql/tibero/mariadb' 등 연결할 데이터베이스 선택

③ 나머지 연결 정보 입력 후 '확인' 버튼 클릭



④ '데이터' 창 에 '데이터 정보 이름'으로 스토어가 추가되며, 연결된 데이터베이스의 테이블 목록이 '테이블 정보' 창에 표시됨

가. 데이터베이스 연결 정보



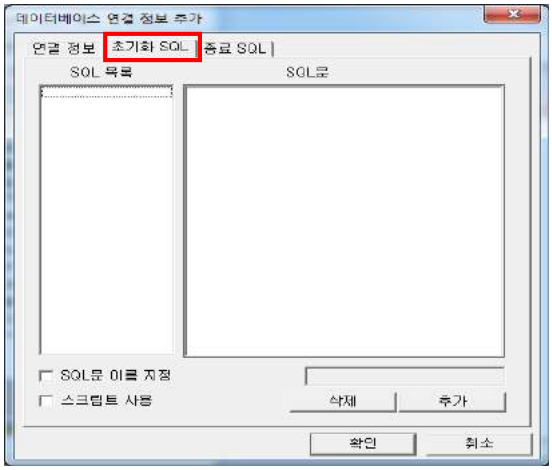
■ **연결 정보 탭**

데이터베이스 연결을 위한 정보를 입력하는 화면입니다. DBMS 종류 별로 입력하는 정보는 달라질 수 있습니다.

- 데이터 정보 이름 : 데이터 스토어 이름으로 사용
- 인코딩 셋 : 데이터베이스의 문자 셋 설정
- 디코딩 셋 : 가져올 데이터의 문자 셋 설정

※ 인코딩셋, 디코딩셋에 설정할 수 있는 문자셋에 대해서는 '도움말' → 'Reference' → '문자셋과 유효한 앨리어스' 참조

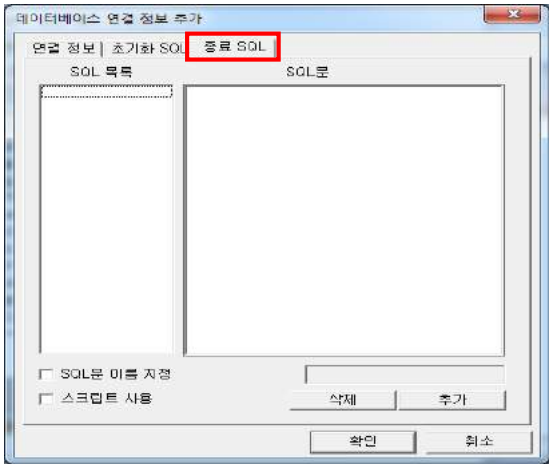
나. 초기화 SQL



■ **초기화 SQL 탭**

DB 연결 후 보고서 데이터 셋 생성 전 초기에 실행할 SQL문을 설정합니다.

다. 종료 SQL



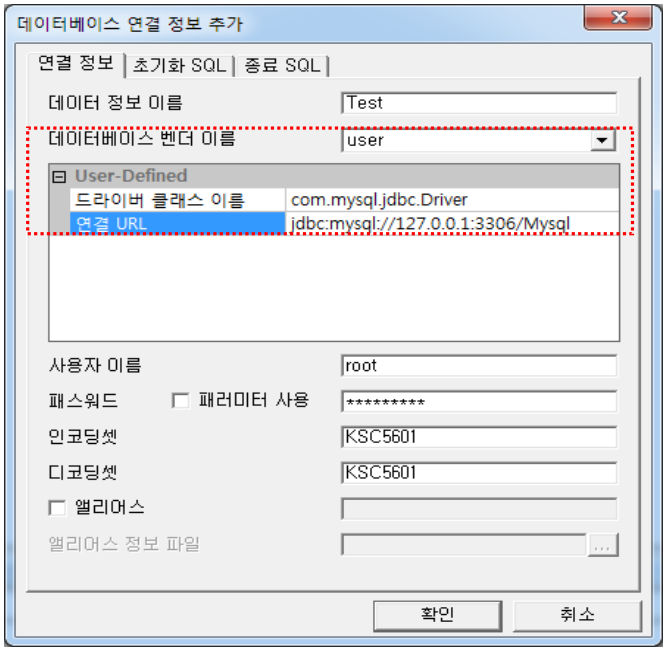
■ **종료 SQL 탭**

보고서 생성 후 DB 연결 종료 전에 실행할 SQL문을 설정합니다.

가. 사용자 정의 연결 정보

쿼리 디자이너의 데이터베이스 벤더 리스트에 등록되어 있지는 않지만, JDBC를 제공하는 DB인 경우 직접 드라이버 정보를 입력하여 연결하는 방법입니다.

➢ **MySQL 드라이버 정보를 직접 입력하여 연결하는 예**



① '데이터베이스 연결 정보 추가' 창에서 '데이터베이스 벤더 이름'을 'user'로 선택합니다.

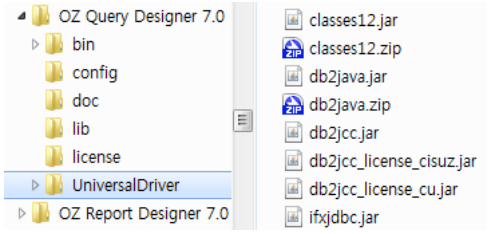
② 연결할 DB의 JDBC '드라이버 클래스 이름'과 '연결 URL'을 입력합니다.

③ '사용자 이름'과 '패스워드'를 입력합니다.

④ '확인' 버튼을 클릭하면 DB에 연결됩니다.

나. JDBC 드라이버 설정

사용자 정의 연결을 하려면 디자이너를 시작하기 전에 해당 JDBC Driver 파일이 클래스 패스에 반드시 존재해야 합니다.



① '%OZD_HOME%\UniversalDriver' 폴더에 JDBC Driver 파일(.jar)을 복사합니다.

```
CLASSPATH = .₩lib₩ozsfw70.jar;...
;.₩UniversalDriver₩JDBC드라이버명.jar
JRE_PATH = ..₩jre15₩bin₩client₩jvm.dll
```

② '%OZD_HOME%\config\launch.cfg' 파일을 편집기로 열어 CLASSPATH 에 Driver 파일 경로를 추가합니다.

※ OZ Report Designer 도 동일하게 설정합니다.

※ 운영 시점에는 오즈 서버의 lib 폴더에 JDBC Driver 파일이 있어야 하고, 클래스 패스로 인식되어야 합니다.

가. DB 앨리어스를 이용한 동적 데이터베이스 연결 (권장 사항)

데이터베이스 연결 정보를 쿼리 디자이너에서 직접 입력하지 않고 별도 파일로 관리할 수 있습니다.

```
#
# sample configuration for Oracle
#
OracleAlias1.vendor=oracle
OracleAlias1.serverAddress=127.0.0.1
OracleAlias1.portNo=1521
OracleAlias1.sid=ORCL
OracleAlias1.user=oz
OracleAlias1.password=oz
OracleAlias1.maxconns=20
OracleAlias1.initconns=5
OracleAlias1.timeout=5
OracleAlias1.doConnectionCheck=true
OracleAlias1.testQueryString=select 1 from dual
```

앨리어스 명

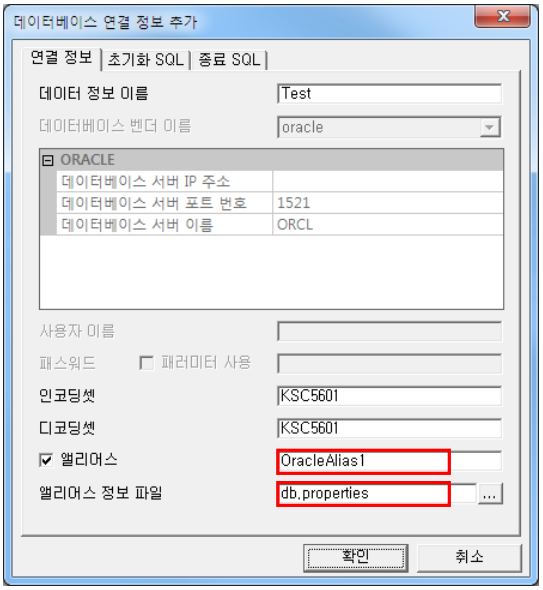
① 쿼리 디자이너가 설치된 경로에 연결 정보 파일을 만듭니다.

%ProgramFiles%\FORCS\OZ Report 7.0\OZ Query Designer 7.0\db.properties

② '.' 앞의 이름 'OracleAlias1'가 앨리어스 명입니다.
임의의 값으로 입력할 수 있으나, 오즈 서버의 연결 정보 파일에서도 동일한 이름을 사용해야 합니다.

※ 오즈 서버의 연결 정보 파일 위치 :
[오즈 서버 홈]\conf\db.properties

※ ODI에서 앨리어스 사용 여부와 상관없이 오즈 서버에서는 연결되는 DB 연결 정보가 모두 db.properties 파일에 설정되어 있어야 합니다.



③ '연결 정보' 탭의 '앨리어스'를 체크하고 앨리어스 명을 입력합니다.

④ '앨리어스 정보 파일' 에는 'db.properties'을 입력합니다.

※ '앨리어스 정보 파일'에 절대 경로를 입력할 수도 있습니다. 파일명만 입력할 때는 리포트 디자이너 폴더에도 동일한 db.properties 파일이 존재해야 합니다.

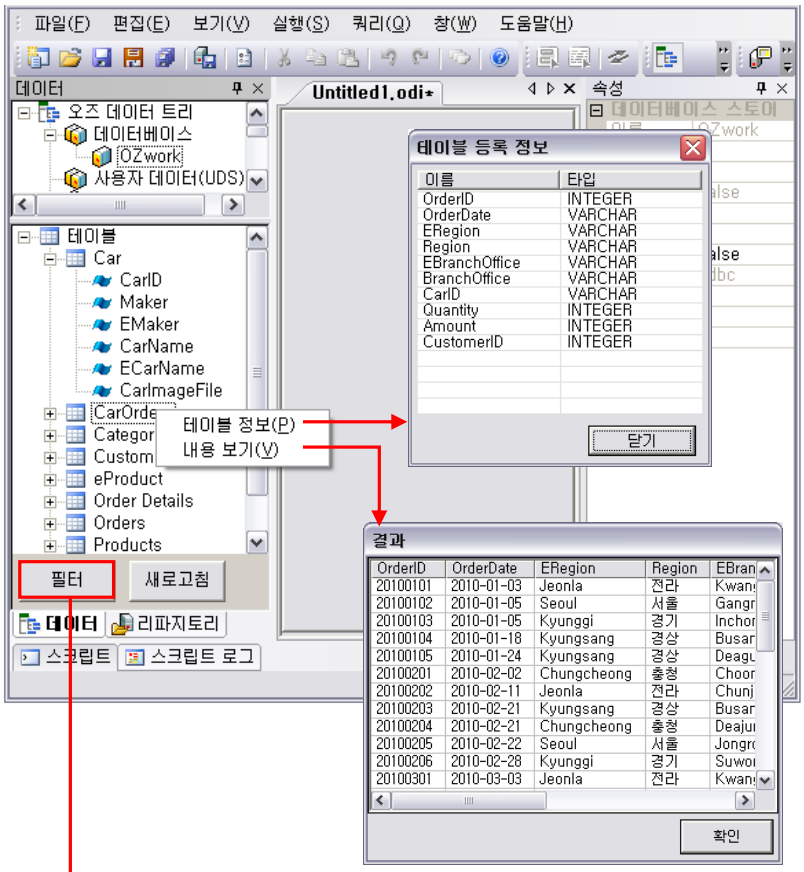
나. 동적 데이타베이스 연결의 장점

- 향후 DB 연결 정보가 변경될 경우 일일이 ODI 파일을 편집하지 않아도 됩니다.
  연결 정보를 수정해야 할 경우 db.properties 파일만 수정하면 됩니다.

- ODI 파일에 DB 연결 정보가 노출되지 않으므로 안전하게 DB 보안을 유지할 수 있습니다.

2.6 테이블 정보

데이터베이스에 연결 되면 '테이블 정보' 창에 테이블 목록이 표시되고 '필터' 와 '새로고침' 기능을 사용할 수 있습니다.

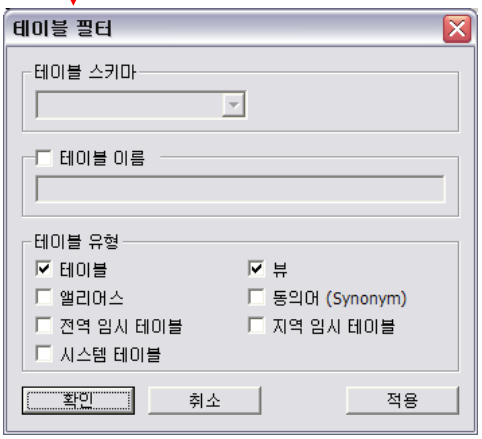


■ **테이블 정보**

테이블의 팝업 메뉴에서 '테이블 정보'를 클릭하면 테이블의 필드명과 타입 정보를 확인할 수 있습니다.

■ **테이블 데이터**

테이블의 팝업 메뉴에서 '내용 보기'를 클릭하면 테이블의 데이터를 확인할 수 있습니다.



■ **필터**

'테이블 정보' 창에 표시되는 테이블 정보를 스키마, 테이블 이름, 테이블 유형별로 필터링 해 보여줄 수 있습니다.

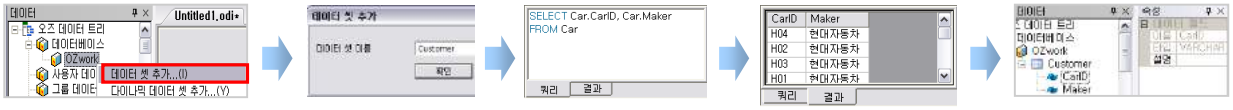
■ **새로고침**

데이터베이스에서 테이블 정보를 새로 가져온 후 '테이블 정보' 창의 정보를 갱신합니다.

가. 데이터베이스 스토어의 데이터 셋

데이터베이스 스토어의 데이터 셋은 쿼리를 통해 가져온 필드 정보와 데이터가 포함된 결과 셋 입니다.

■ 데이터 셋 만드는 과정

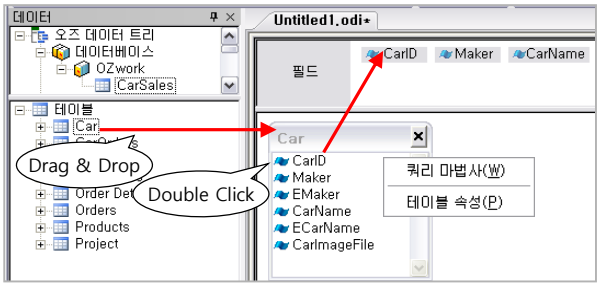


① 스토어의 팝업 메뉴에서 '데이터 셋 추가' 선택 ② '데이터 셋 이름' 입력 ③ 쿼리문 작성 ④ 쿼리 실행(F5) ⑤ 필드 생성됨

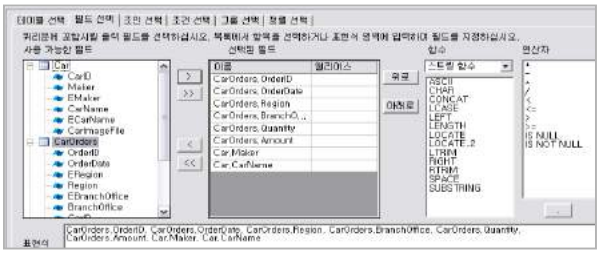
쿼리문은 디자인 모드에 따라 작성 방법이 달라집니다. 디자인 모드에는 '디자인 뷰'와 '매뉴얼'이 있습니다. '쿼리' → '디자인 모드' 메뉴에서 선택할 수 있습니다.

나. '디자인 뷰' 모드에서 쿼리문 작성

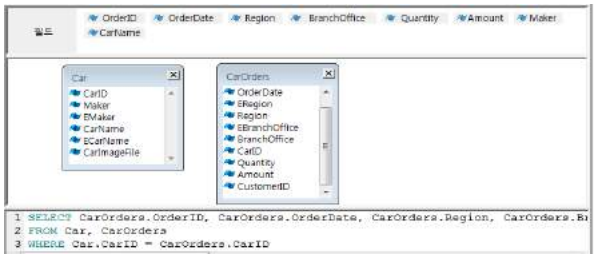
'디자인 뷰' 모드에서는 테이블을 드래그&드롭으로 추가할 수 있으며, 쿼리 마법사를 이용하여 쿼리를 만듭니다.



- '테이블 정보' 창에서 테이블을 디자인 화면으로 드래그&드롭으로 추가합니다.
- 테이블의 필드를 더블 클릭하여 필드 창에 추가합니다.



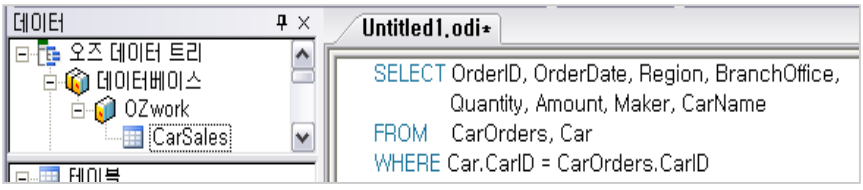
- 쿼리 마법사를 이용하여 테이블, 필드, 조인, 조회 조건 등을 지정하여 쉽게 쿼리를 작성할 수 있습니다.



- 마법사를 완료하면 하단에 자동으로 생성된 쿼리문이 보여집니다.

다. '매뉴얼' 모드에서 쿼리문 작성

'매뉴얼' 모드에서는 쿼리문을 직접 키보드로 입력해서 만듭니다.



- 직접 쿼리를 입력하거나, 다른 곳에서 쿼리를 복사한 후 붙여넣기 해서 작성할 수 있습니다.

가. 마스터 셋 정의

두 개의 데이터 셋을 '마스터 셋' – '디테일 셋' 관계로 설정할 수 있습니다.

디테일 셋은 마스터 셋의 레코드 수 만큼 쿼리문이 수행되며, 현재 마스터 레코드의 값을 디테일 셋 쿼리문에서 참조할 수 있습니다.

나. 마스터 셋 설정

2단 마스터-디테일 관계를 설정하는 방법을 예로 설명합니다.

```
SELECT Car.Maker, Car.EMaker
FROM Car
GROUP BY Car.Maker, Car.EMaker
ORDER BY Car.Maker ASC
```

① 'SQL_Maker' 데이터 셋의 쿼리문을 작성합니다.

```
SELECT Car.CarID, Car.CarName
FROM Car
WHERE Car.Maker = '#SQL_Maker.Maker#'
ORDER BY Car.CarID ASC
```

② 'SQL_Car' 데이터 셋의 쿼리문을 작성합니다.
조건절에 'SQL_Maker' 데이터 셋의 'Maker' 필드를 설정합니다.

```
SELECT CarOrders.OrderDate,
CarOrders.Region, CarOrders.Amount
FROM CarOrders
WHERE CarOrders.CarID = '#SQL_Car.CarID#'
ORDER BY CarOrders.OrderDate ASC
```

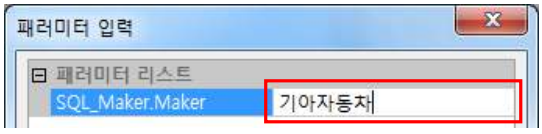
③ 'SQL_Orders' 데이터 셋의 쿼리문을 작성합니다.
조건절에 'SQL_Car' 데이터 셋의 'CarID'필드를 설정합니다.

마스터셋	SQL_Maker

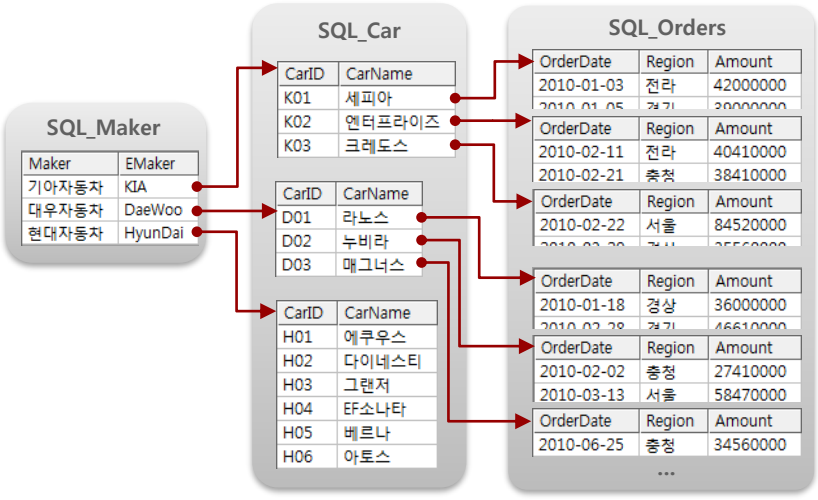
마스터셋	SQL_Car

④ 'SQL_Car' 데이터 셋의 '마스터셋' 속성 값을 'SQL_Maker'로 변경합니다.

⑤ 'SQL_Orders' 데이터 셋의 '마스터셋' 속성 값을 'SQL_Car'로 변경합니다.



⑥ 디테일 셋 쿼리문을 실행하면 마스터 셋 필드 값을 입력하는 '패러미터 입력' 창이 나옵니다. 임의의 값을 입력합니다. 보고서 출력할 때는 마스터 셋의 데이터가 자동 매핑됩니다.



⑦ 보고서 출력할 때 실제로 가져오는 데이터 구조 입니다.

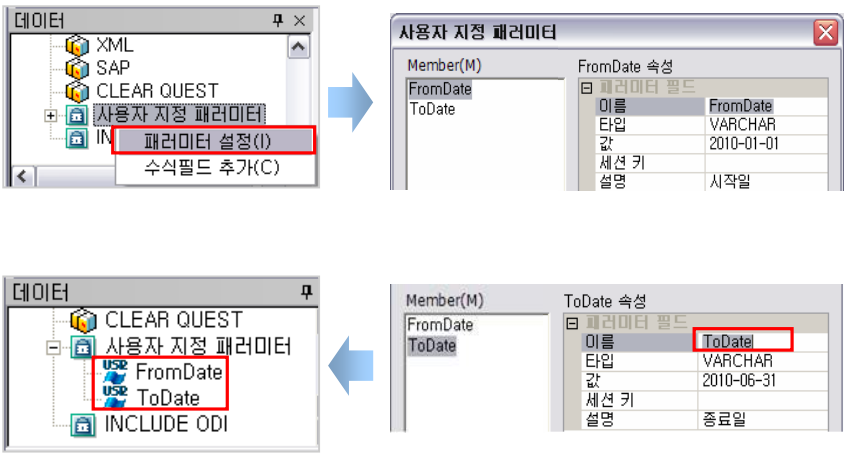
마스터 셋의 레코드 수만큼 디테일 셋이 만들어 집니다.

가. 사용자 지정 패러미터 정의

사용자 로그인 정보, 데이터베이스 연결 정보, SQL 조회 조건 등을 운영 시점(런타임)에 패러미터로 입력받아 처리하려면 '사용자 지정 패러미터'를 이용할 수 있습니다. 사용자 지정 패러미터를 이용하여 최종 사용자가 뷰어로 보고서를 출력할 때 조회 조건을 임의로 바꾸어 조회할 수 있습니다.

나. 사용자 지정 패러미터 생성

런타임에 조회 조건을 입력 받기 위해서 사용자 지정 패러미터를 다음과 같이 생성합니다.



① '데이터' 창의 '사용자 지정 패러미터' 팝업 메뉴에서 '패러미터 설정' 선택

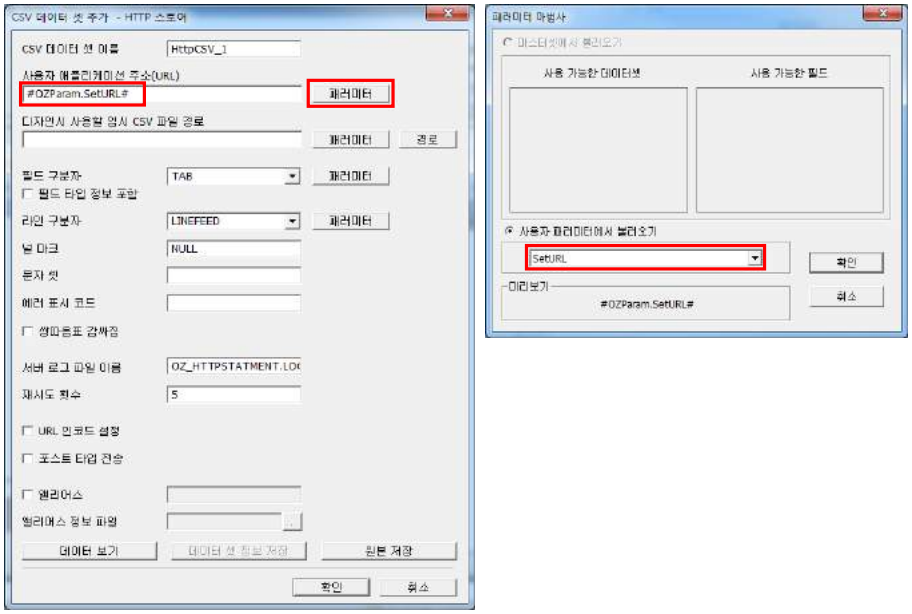
② '사용자 지정 패러미터' 창에 사용할 패러미터 입력 (엔터키로 추가)

③ '패러미터 필드' 속성 창의 '값'에 디자인 타임에 사용할 값 입력

④ '확인' 버튼 클릭 후 패러미터 생성됨

다. 스토어 창에 패러미터 사용

스토어 창의 연결 정보를 패러미터로 입력 받을 수 있습니다. 패러미터는 #OZParam.패러미터이름# 형식으로 입력됩니다. 패러미터 마법사를 이용하거나 직접 작성할 수 있습니다.



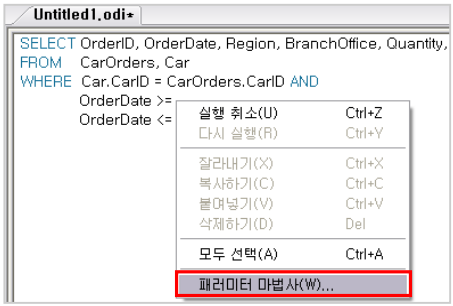
① 스토어 창의 '패러미터' 버튼 클릭

② '패러미터 마법사' 창의 '사용자 패러미터에서 불러오기'에서 사용할 패러미터 선택

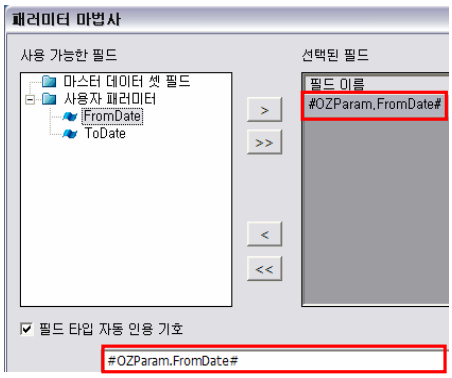
③ '확인' 버튼 클릭하면 입력란에 #OZParam.패러미터이름# 형식으로 패러미터 입력됨

라. 쿼리문에 패러미터 사용

쿼리문의 조회 조건을 패러미터로 입력 받을 수 있습니다. 패러미터는 #OZParam.패러미터이름# 형식으로 입력됩니다. 패러미터 마법사를 이용하거나 직접 작성할 수 있습니다.

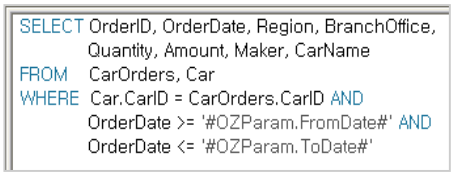


① 매뉴얼 모드인 경우 쿼리문이 입력된 화면의 팝업 메뉴에서 '패러미터 마법사' 선택

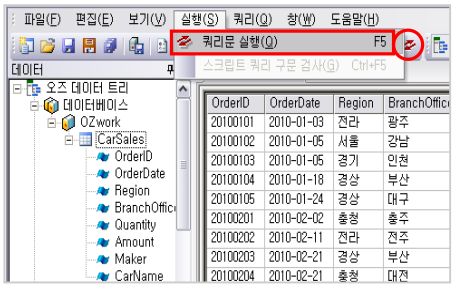


② 필요한 사용자 패러미터 선택

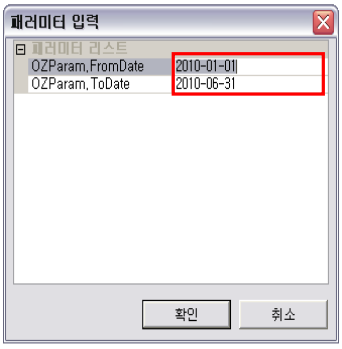
③ '확인' 버튼 클릭



④ 쿼리문에 패러미터가 #OZParam.패러미터이름# 형식으로 입력됨



⑤ '실행' → '쿼리문 실행'(F5) 메뉴 선택



⑥ '패러미터 입력' 창에 임의의 패러미터 값 입력

⑦ '확인' 버튼 클릭 후 결과 확인

※ '패러미터 입력' 창에는 쿼리문에 사용된 패러미터 목록만 나옵니다. 이 창에 입력된 값은 단지 쿼리 디자이너에서 쿼리 결과를 확인할 때만 사용됩니다.

가. 스크립트 용도

쿼리 디자이너에서 다음과 같은 경우에 스크립트를 사용할 수 있습니다.

- 조건에 따라 다른 쿼리문을 작성해야 할 경우
- 데이터 셋에 가상 데이터 필드(수식 필드)를 추가해야 할 경우
- 데이터베이스에 연결된 직후나 연결 해지 직전에 어떤 로직을 수행하고자 할 경우
- 데이터 스토어나 데이터 셋 등의 속성을 동적으로 변경할 경우

나. 스크립트 종류

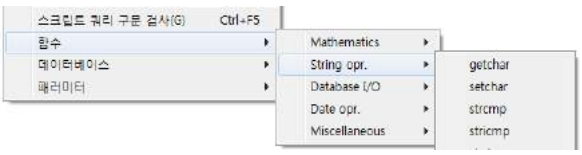
쿼리 디자이너는 '오즈 스크립트', '자바 스크립트' 두 가지 스크립트 언어를 지원합니다.
오즈 스크립트는 오즈에서 제공하는 자체 스크립트 언어로 쿼리문, 실행문, 수식 필드 작성에 이용됩니다.
자바 스크립트는 쿼리 디자이너의 모든 기능에 이용됩니다.

다. 오즈 스크립트

■ 문법

- 자료형 : int, double, string, Date
- 함수 : 수학 함수, 문자 함수, 날짜 함수, 데이터 셋 접근 함수 등 제공
- 변수 선언 및 초기화 예 : int i; i = 10; (주의 : 선언문에서 초기값을 지정하면 에러 발생)
- 연산자 : +, -, *, /, %, =, ==, !=, <, <=, >, >=, &, |, ^, ~, &&, ||, !
- 제어문 : if-else문, while 문
- 필드, 패러미터 정의 : #데이터셋명.필드명# 또는 #필드명#, #OZParam.패러미터명#

■ 팝업 메뉴



- 스크립트 편집 창의 팝업 메뉴에서 제공되는 함수, 데이터 필드, 패러미터 목록을 선택할 수 있습니다.

라. 자바 스크립트

■ 문법

- 자료형 : var 로 선언하며 변수의 내용에 따라 데이터 형이 결정됨
- 함수 : 자바 스크립트에서 제공하는 함수, 쿼리 디자이너에서 제공하는 함수, 사용자 정의 함수
- 연산자 : +, -, *, /, %, =, ==, !=, <, <=, >, >=, &, |, ^, ~, &&, ||, !
- 제어문 : if-else문, while 문, do while 문, for 문

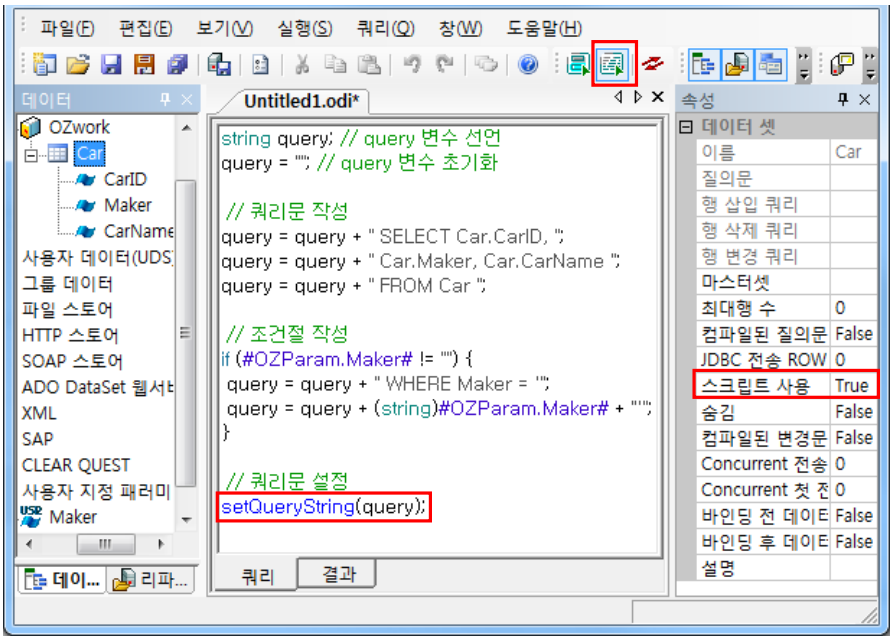
■ 주요 이벤트

- Functions : 모든 바인딩 및 데이터 액션 전에 실행되는 이벤트로 함수 선언, 전역 변수 선언에 이용
- OnInitialize : 바인딩 요청 시 데이터 스토어 초기화 직전에 발생하는 이벤트
- OnPostInitialize : 바인딩 요청 시 데이터 스토어 초기화 직후에 발생하는 이벤트
- OnFinalize : 바인딩 요청 시 데이터 스토어 삭제 직전에 발생하는 이벤트
- OnPreBind : 바인딩 실행 직전에 발생하는 이벤트
- OnPostBind : 바인딩 실행 직후에 발생하는 이벤트

※ 자세한 내용은 쿼리 디자이너의 도움말(F1)을 참고하시기 바랍니다.

가. 오즈 스크립트를 이용한 쿼리문 작성

조건에 따라 다른 쿼리문을 작성하기 위해 오즈 스크립트를 아래와 같이 작성합니다.



① 툴바의 '매뉴얼' 클릭

② 데이터 셋의 '스크립트 사용' 속성을 'True'로 변경

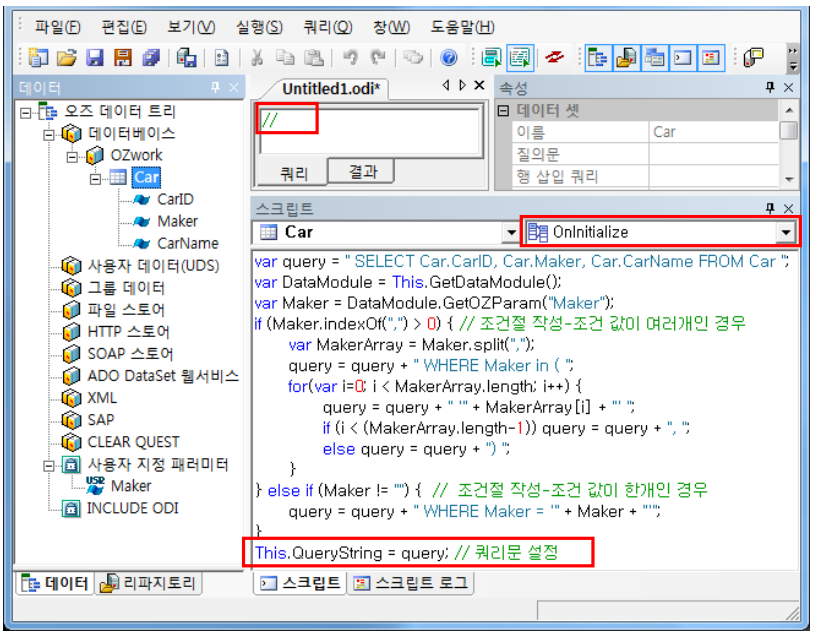
③ 쿼리 창에 스크립트로 쿼리문 작성

④ 마지막에 'setQueryString' 함수로 쿼리문 설정

⑤ 쿼리문 실행(F5)하면 데이터 필드 생성됨

나. 자바 스크립트를 이용한 쿼리문 작성

데이터 셋의 OnInitialize 이벤트에 자바 스크립트로 쿼리문을 작성할 수 있습니다.



① 데이터 셋의 쿼리 창에 임의의 문자 입력

② 데이터 셋의 OnInitialize 이벤트에 스크립트로 쿼리문 작성

③ 마지막에 'This.QueryString' 속성으로 쿼리문 설정

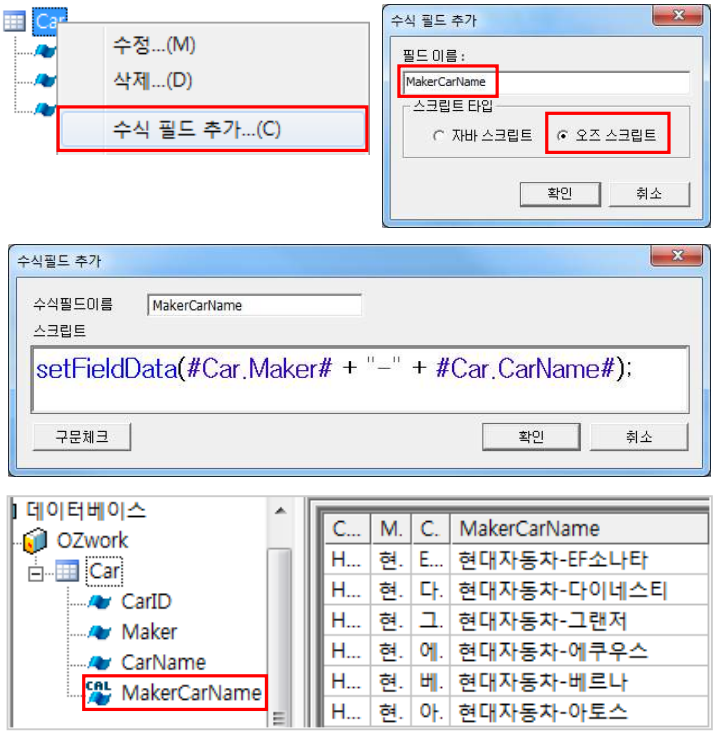
④ 쿼리문 실행(F5)하면 데이터 필드 생성됨

※ '편집' → '스크립트 미리보기' 메뉴에서 전체 자바 스크립트를 확인할 수 있습니다.

가. 수식 필드 정의

수식 필드는 데이터 셋에 추가되는 가상 데이터 필드입니다. 데이터 필드는 오즈 서버가 데이터 소스로부터 조회한 필드인 반면, 수식 필드는 오즈 서버가 자체적으로 스크립트를 수행하여 생성한 필드 입니다.
수식 필드와 데이터 필드의 생성 과정은 다르나 리포트 디자이너에서 이용 방법은 동일합니다.

나. 오즈 스크립트를 이용한 수식 필드 생성



① 데이터 셋의 팝업 메뉴에서 '수식 필드 추가' 선택

② '필드 이름' 입력

③ '오즈 스크립트' 선택 후 '확인' 버튼 클릭

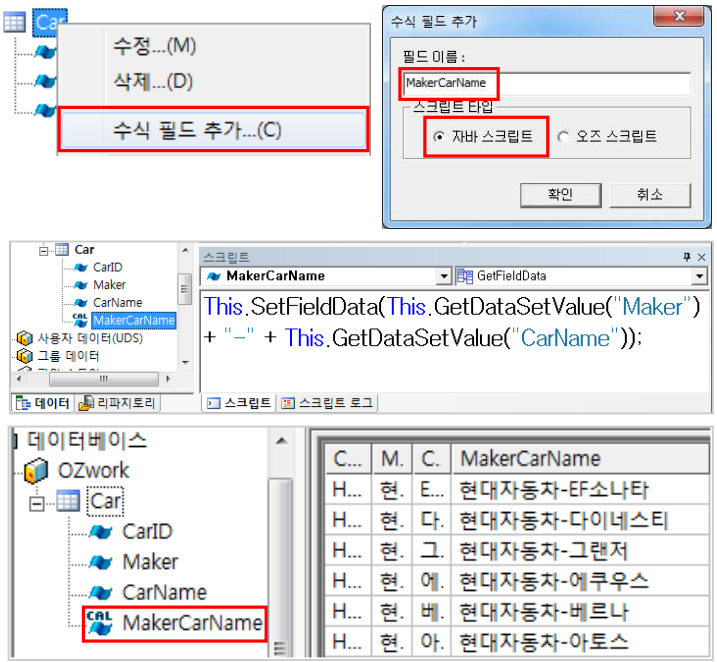
④ 스크립트 작성

⑤ 마지막에 'setFieldData' 함수로 수식 필드 값 설정

⑥ '확인' 버튼 클릭하면 수식 필드 생성됨

⑦ 쿼리문 실행(F5)으로 결과 확인

다. 자바 스크립트를 이용한 수식 필드 생성



① 데이터 셋의 팝업 메뉴에서 '수식 필드 추가' 선택

② '필드 이름' 입력

③ '자바 스크립트' 선택 후 '확인' 버튼 클릭 하면 수식 필드 생성됨

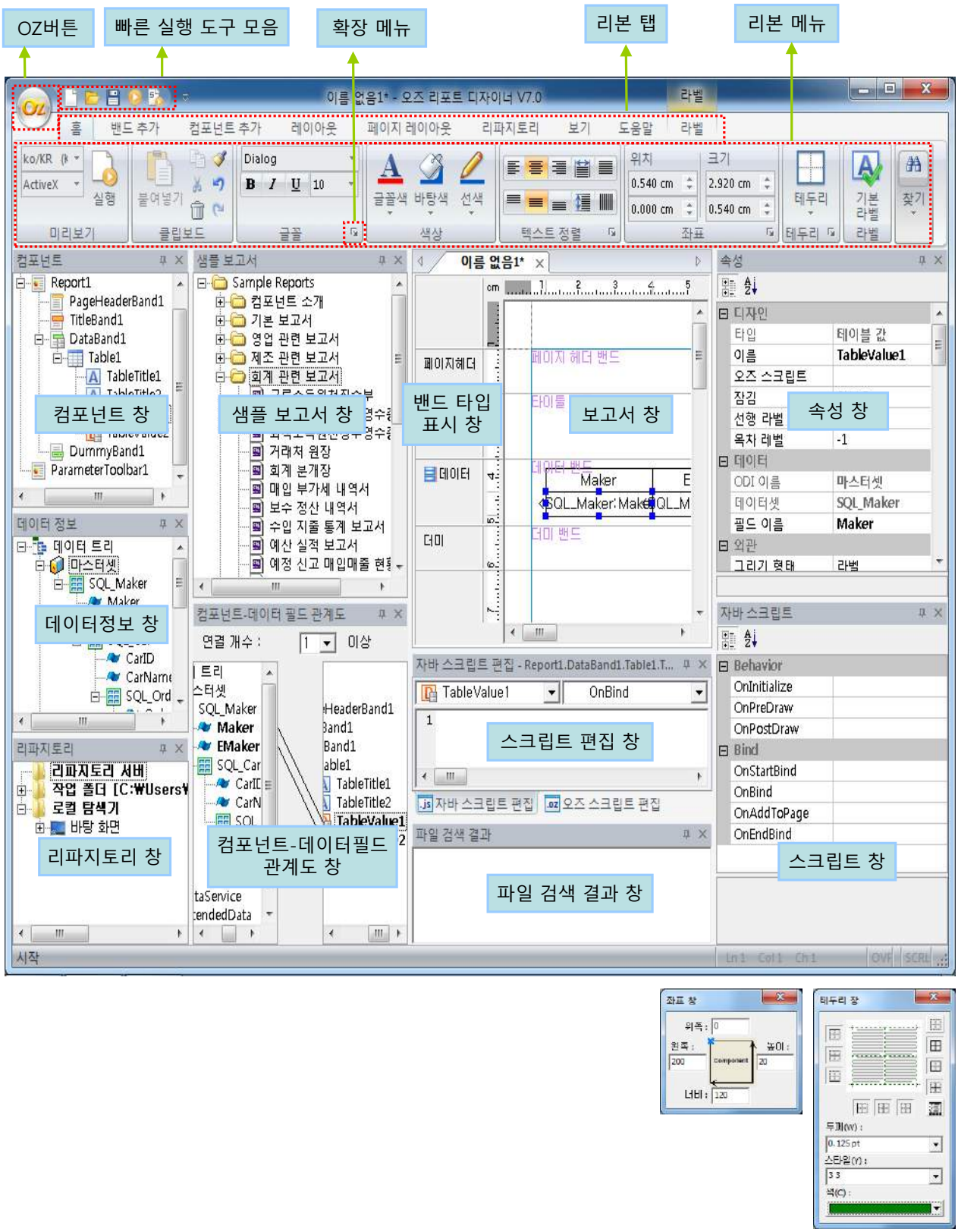
④ 추가된 수식 필드의 'GetFieldData' 이벤트에 스크립트 작성

⑤ 마지막에 'This.SetFieldData' 함수로 수식 필드 값 설정

⑥ 쿼리문 실행(F5)으로 결과 확인

Chapter4. 리포트 디자이너

1. 사용 환경
2. 리포트
3. 데이터 셋 연결
4. 밴드
5. 라벨
6. 멀티플렉스
7. 스크립트



[리포트 디자이너 화면]

리포트 디자이너의 각 창은 '보기' 탭에서 선택하여 볼 수 있습니다.

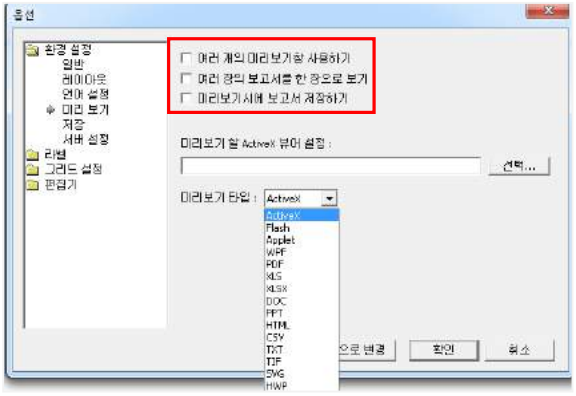
❖ **리포트 디자이너 화면 설명**

인터페이스	설명
OZ 버튼	새로 만들기, 열기, 닫기, 저장, 미리보기 등의 메뉴를 사용할 수 있습니다.
빠른 실행 도구 모음	독립적으로 실행 가능한 도구 모음입니다. 빠른 실행 도구 모음에 명령어 추가, 삭제가 가능합니다.
리본 탭	홈, 밴드 추가, 컴포넌트 추가, 레이아웃, 페이지 레이아웃, 리파지토리, 보기, 도움말 탭으로 구성됩니다. 각 탭을 클릭하면 리본 메뉴를 확인할 수 있습니다. 라벨, 차트 등 컴포넌트를 클릭하면 자주 설정하는 속성을 빠르게 설정할 수 있는 조작 탭이 추가됩니다. 조작 탭 종류로는 라벨, 고정 테이블, 차트, 이미지, 바코드, PDF 417, QR 바코드, HTML, 선, 화살표, 사각형, 원입니다 .
리본 메뉴	각 리본 탭을 클릭하면 리본 메뉴가 나열되며 리본 메뉴의 아이콘을 클릭하여 사용합니다. 예를 들어 컴포넌트 추가 탭을 클릭하면 라벨, 멀티플렉스, 도형을 사용할 수 있습니다.
컴포넌트 창	보고서에 추가된 컴포넌트, 컴포넌트가 그려지는 순서 등을 트리 형태로 표시하는 창입니다.
데이터 정보 창	추가한 ODI 파일의 데이터 셋, 사용자 지정 패러미터 정보와 폼 패러미터, 시스템 패러미터 등을 트리 형태로 표시하는 창입니다.
리파지토리 창	로컬 및 서버 리파지토리의 폴더 및 파일을 탐색기 형태로 표시하는 창으로 작업 폴더를 설정할 수 있고 오즈 서버와 연결하여 개발한 odi, ozr 파일을 업로드 및 다운로드 할 수 있습니다.
샘플 보고서 창	컴포넌트 별로 구현한 보고서와 자주 사용되는 보고서 샘플을 제공하는 창입니다.
속성 창	보고서에 추가된 컴포넌트의 속성을 설정하고 표시하는 창입니다.
밴드 타입 표시 창	밴드 타입 표시 창은 보고서에 추가되어 있는 밴드를 표시하는 창으로, 밴드 타입명에서 "밴드"를 제외한 이름이 표시되며, 데이터 밴드와 그룹 헤더/풋터 밴드의 경우 들여쓰기를 하여 밴드 간의 포함 관계를 알기 쉽게 표시하는 창입니다.
보고서 창	보고서 창은 보고서를 디자인하는 창으로 열려 있는 모든 보고서를 탭 형태로 표시하는 창입니다.
자바 스크립트 창	자바 스크립트 창은 자바 스크립트에서 지원하는 이벤트와 각 이벤트 별로 스크립트를 편집하는 창입니다.
오즈 스크립트 창	오즈 스크립트 창은 오즈 스크립트를 편집할 수 있는 창입니다.
컴포넌트-데이터필드 관계도 창	컴포넌트-데이터필드 관계도 창은 보고서에 추가된 컴포넌트와 데이터 필드간의 관계를 나타내는 창입니다.
좌표 창	컴포넌트의 위치와 크기를 세밀하게 조정할 수 있는 창입니다.
테두리 창	컴포넌트의 테두리와 두께, 스타일, 색상을 설정할 수 있는 창입니다.

가. 옵션 설정

리포트 디자이너의 'OZ버튼' → '옵션' 메뉴 에서 기본 환경을 설정합니다.

1) 환경설정 – 미리 보기

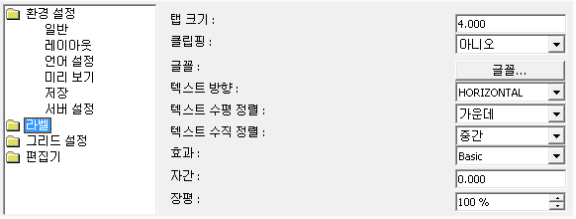


- **여러 개의 미리보기 창 사용하기** :
미리보기할 때마다 새로운 뷰어 창이 생김

- **여러 장의 보고서를 한 장으로 보기** :
한 개의 리포트가 여러 장으로 나누어질 때 사용

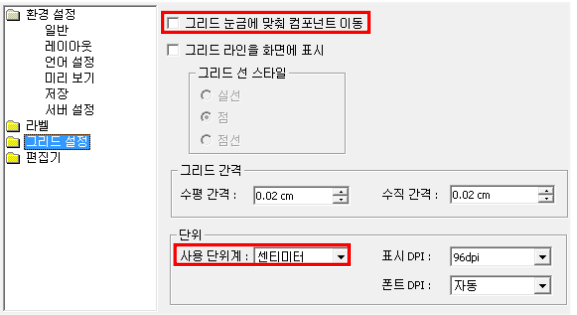
- **미리보기시에 보고서 저장하기** :
미리보기할 때마다 보고서 저장

2) 라벨



새로 디자인하는 라벨의 기본 속성값을 설정합니다.
디자이너 재 시작한 후에 반영됩니다.

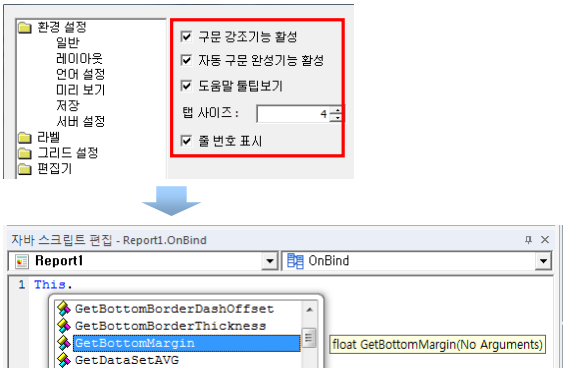
3) 그리드 설정



- **그리드 눈금에 맞춰 컴포넌트 이동** :
밴드에 추가된 컴포넌트의 위치나 크기를 그리드의 눈금에 맞춰 조절할지 여부를 설정합니다.

- **사용 단위계** :
용지 눈금 단위를 설정합니다.

4) 편집기

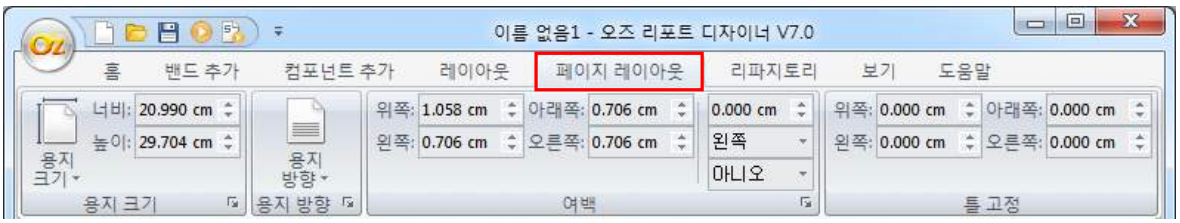


- **구문 강조기능 활성** :
스크립트 함수 사용시 강조 기능을 제공합니다.

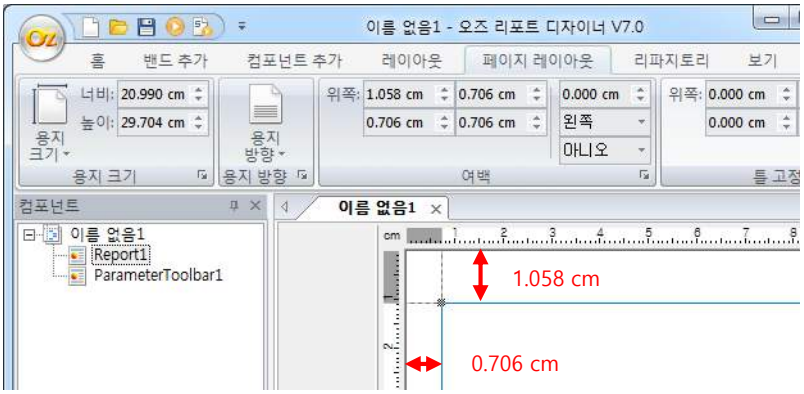
- **자동 구문 완성기능 활성** :
자바 스크립트의 함수 사용시 일부분 만을 입력해도 자동으로 완성될 수 있는 기능을 제공합니다.

나. 용지 설정

'페이지 레이아웃' 탭에서 리포트 용지 크기, 방향, 여백 등을 설정합니다.



용지 설정은 뷰어에서 미리보기 되는 페이지와 인쇄할 용지에 모두 반영됩니다.

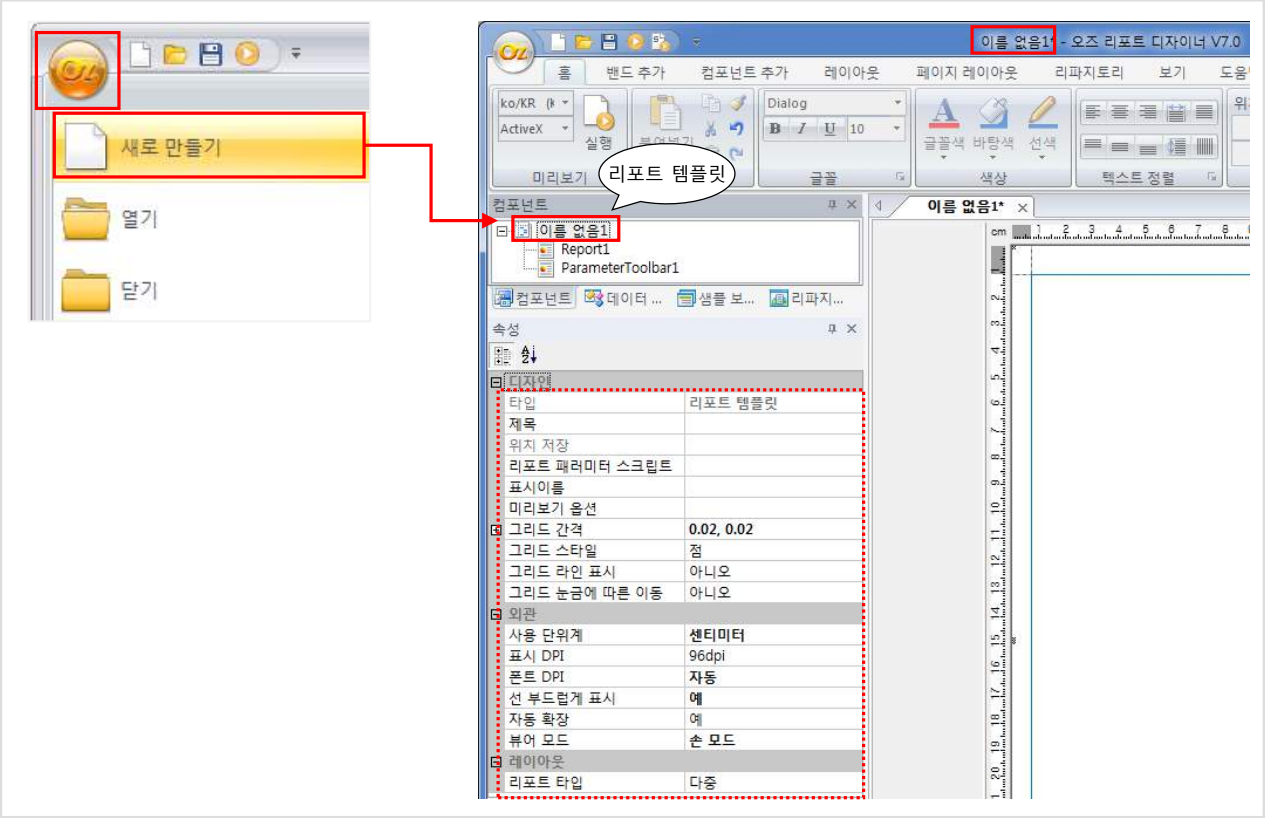


용지 여백을 위쪽 '1.058cm', 왼쪽 '0.706cm' 으로 설정했을 경우 리포트 디자이너에 적용된 화면입니다.

가. 리포트 템플릿 정의

리포트 템플릿은 보고서에 속해 있는 모든 컴포넌트의 정보를 담고 있는 최상위 컴포넌트입니다.
한 개의 리포트 템플릿은 여러 개의 리포트 컴포넌트를 포함할 수 있습니다.
OZR 파일에는 리포트 템플릿이 한 개만 존재합니다.

OZ 아이콘( )클릭 → '새로 만들기' 를 선택하면 '이름 없음1' 이라는 기본 리포트 템플릿이 만들어 집니다.



나. 리포트 템플릿 속성

1) 리포트 패러미터 스크립트
초기화 스크립트이며 보고서가 생성되기 바로 직전에 실행됩니다.

2) 표시이름
뷰어 창의 보고서 트리에 표시할 보고서 이름을 설정합니다.

3) 사용 단위계
보고서에서 컴포넌트의 크기나 위치 등을 설정할 때 사용하는 기본 단위를 설정합니다.
'센티미터', '포인트'(픽셀), '인치' 중에서 선택할 수 있습니다.

4) 폰트 DPI
DPI 값에 따라 글자 간격과 글자 너비를 설정할 수 있도록 폰트 DPI 속성을 설정합니다. '자동'을 권장합니다.

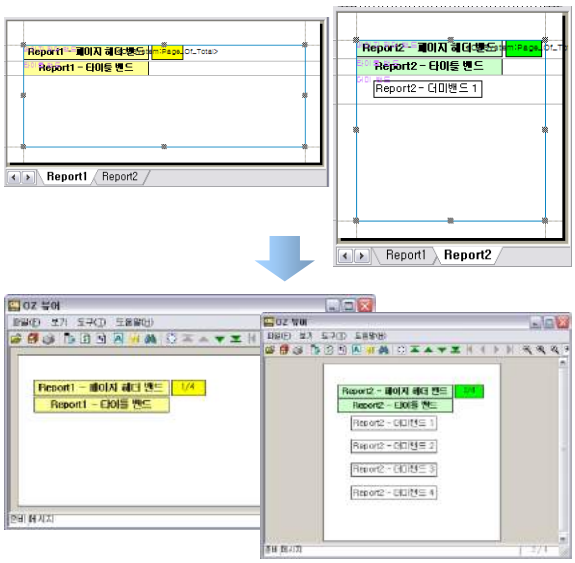
5) 선 부드럽게 표시
사각형, 원, 선, 화살표와 같은 도형 컴포넌트와 차트의 선, 표식을 부드럽게 표시할지 여부를 설정합니다.

6) 자동 확장
리포트 디자인 공간의 크기가 리포트 템플릿의 크기를 벗어났을 때 리포트 템플릿의 크기를 자동으로 확장시킬지 여부를 설정합니다.

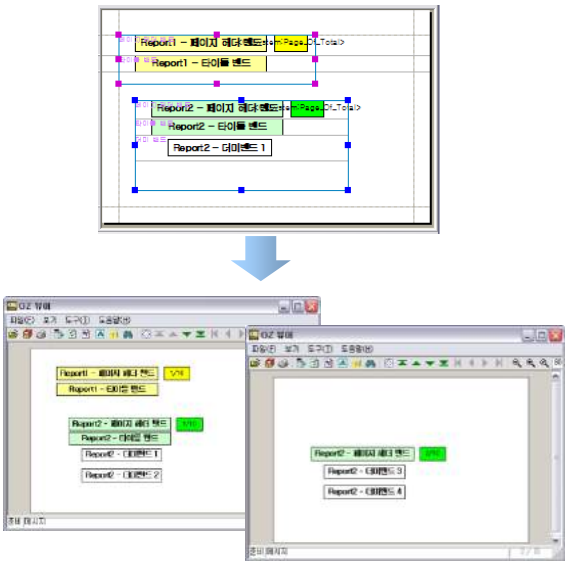
6) 리포트 타입

한 개의 리포트 템플릿은 두 개 이상의 리포트 컴포넌트를 가질 수 있으며 그 구성 형태에 따라 '다중 보고서' 또는 '복합 보고서'라고 합니다.

- **다중 보고서** : 여러 개의 리포트 컴포넌트가 페이지로 연결되어 하나의 보고서로 인식되는 구조



- **복합 보고서** : 여러 개의 리포트 컴포넌트가 한 페이지에 디자인되며, 각각 독립된 보고서로 인식되는 구조

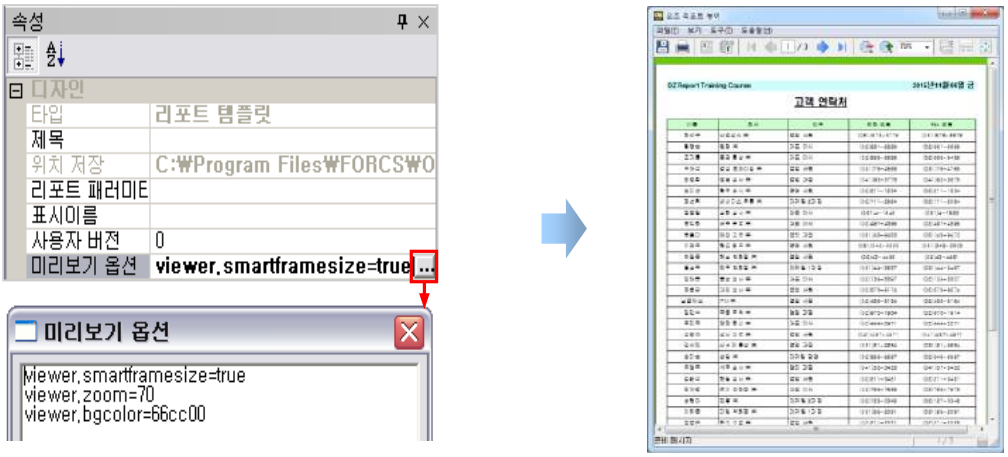


※ 참고로 뷰어 패러미터를 이용하여 한 뷰어에 여러 보고서를 보여주는 구조를 '멀티 보고서'라고 합니다.

7) 미리보기 옵션 (뷰어 패러미터 설정)

미리보기 옵션에 뷰어 패러미터를 지정하여 디자이너에서 미리보기를 할 수 있습니다. 서버에 보고서를 업로드 한 후에는 미리보기 옵션의 뷰어 패러미터는 반영되지 않습니다.

➢ **미리보기 옵션 예**



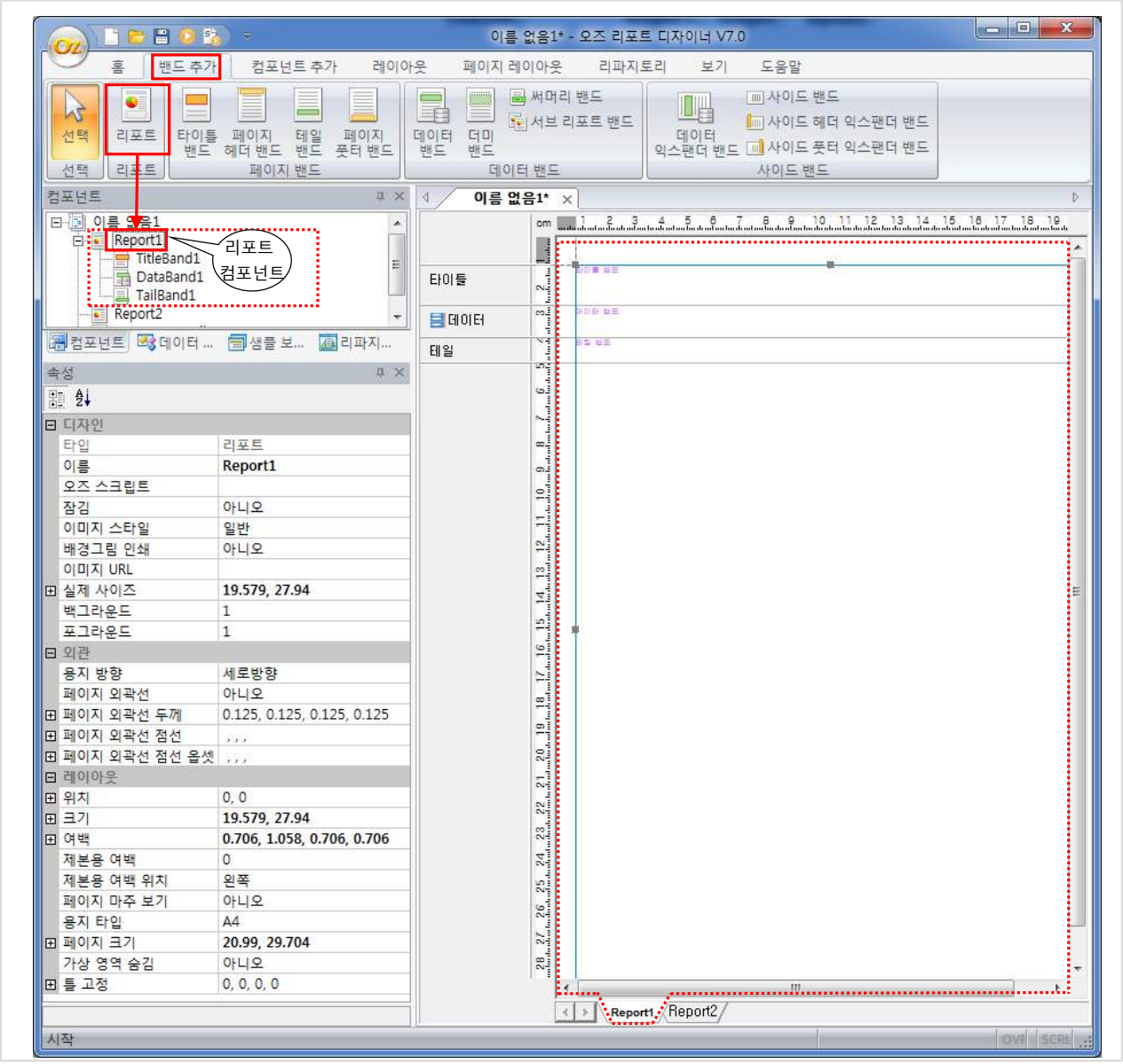
- viewer.smartsize=true : 뷰어가 별도 창으로 실행되는 경우에 뷰어 창의 크기를 보고서 크기에 자동으로 맞춤
- viewer.zoom=70 : 뷰어의 확대 축소 비율 (20% 이상)
- viewer.bgcolor : 뷰어 창의 배경색 지정

TIP

ReportTemplate의 OnStartUp이벤트에 SetReportOption("viewer.zoom", "120"); 와 같이 지정하면 배포 후에도 패러미터가 반영되며 뷰어를 호출하는 웹 페이지의 패러미터 보다 우선 순위가 높습니다.

가. 리포트 컴포넌트 정의

리포트 컴포넌트는 보고서를 구현하기 위해 필요한 각종 밴드 및 컴포넌트들을 계층 구조로 포함합니다.
즉, 하나의 리포트 컴포넌트는 여러 개의 밴드를 포함할 수 있고, 하나의 밴드는 여러 개의 컴포넌트를 포함할 수 있습니다. 또한 각 리포트 컴포넌트마다 독립적인 보고서 디자인 영역을 갖고 있습니다.



나. 리포트 컴포넌트 속성

1) 배경 이미지 관련 속성

'이미지 스타일', '배경그림 인쇄', '이미지 URL' 속성으로 보고서의 배경 이미지를 설정합니다.

2) 디자인 용지 관련 속성

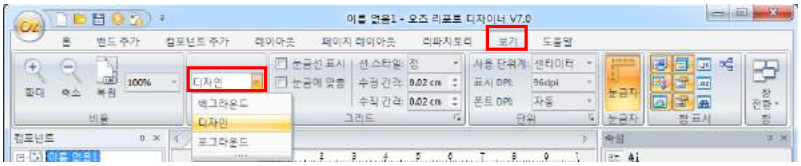
'용지 방향', '용지 타입', '페이지 크기', '여백' 속성으로 보고서의 페이지와 용지를 설정합니다.

3) 디자인 영역 관련 속성

'백그라운드', '포그라운드' 속성으로 리포트의 아래, 위로 겹쳐 보이는 밴드 개수를 설정합니다.

2.3 백그라운드·포그라운드 모드

리포트의 디자인 영역은 '디자인 모드', '백그라운드 모드', '포그라운드 모드'로 구성됩니다.



'백그라운드 모드'에서는 '백그라운드 밴드'가 자동 추가되며 리포트 배경을 디자인할 수 있습니다.
백그라운드 밴드의 내용이 '디자인 모드'에서 개발된 모든 페이지 하단에 겹쳐 보입니다.

'포그라운드 모드'에서는 '포그라운드 밴드'가 자동 추가됩니다.
포그라운드 밴드의 내용이 '디자인 모드'에서 개발된 모든 페이지 상단에 겹쳐 보입니다.

보고서가 만들어지는 순서는 백그라운드 모드 → 디자인 모드 → 포그라운드 모드 순입니다.

[디자인 화면]

[실행 결과]

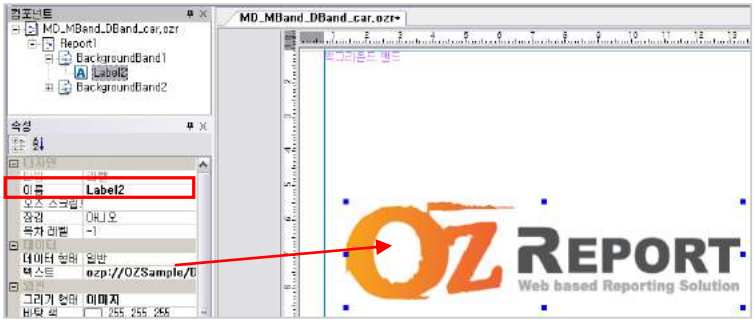
➢ 백그라운드 모드 사용 예

① 리포트의 '백그라운드' 속성을 '2'로 입력하면 백그라운드 밴드가 2개 생성됩니다.

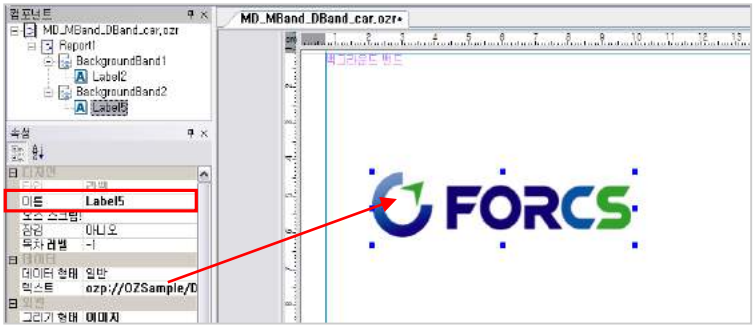
② '보기' → '백그라운드 모드' 메뉴를 선택합니다.

③ BackgroundBand1 위에 '라벨' 추가 후 '그리기 형태' 속성을 '이미지'로 변경합니다.

④ 라벨의 팝업 메뉴에서 '이미지 URL 설정'을 선택합니다.



④ 이미지 파일 선택 후 '텍스트' 속성의 이미지 경로를 확인합니다.



⑤ BackgroundBand2에도 이와 같이 다른 이미지를 보여주는 라벨을 추가합니다.

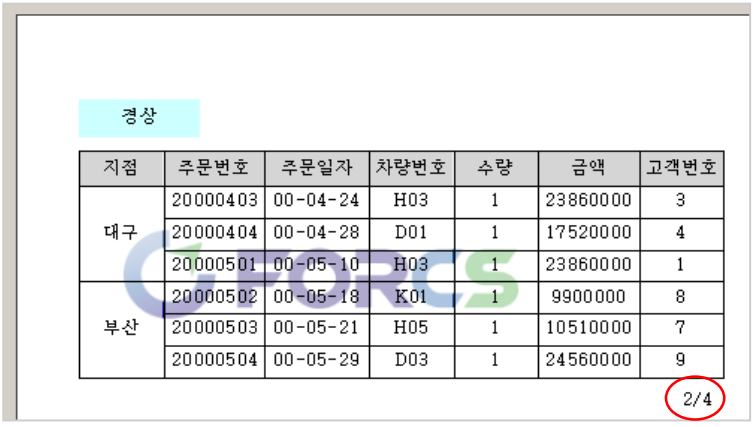
지역별 판매 실적

경기

지점	주문번호	주문일자	차량번호	수량	금액	고객번호
인천	20000301	00-03-05	K01	1	9900000	9
	20000302	00-03-09	H01	1	40610000	10
	20000303	00-03-25	D01	1	17520000	6
수원	20000304	00-03-31	H01	1	40610000	2
	20000402	00-04-12	H02	4	30000000	6

1/4

⑥ 미리보기 화면입니다. 백그라운드의 이미지가 디자인 화면 아래 겹쳐 보입니다. 단, 디자인 화면에서 라벨들의 '투명' 속성이 '예' 일 때 백그라운드의 이미지가 보입니다.



경상

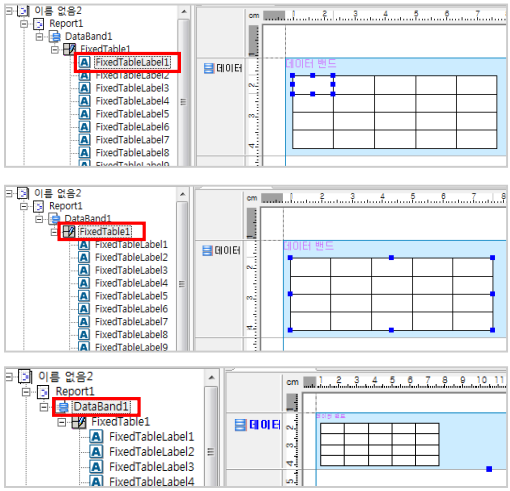
지점	주문번호	주문일자	차량번호	수량	금액	고객번호
대구	20000403	00-04-24	H03	1	23860000	3
	20000404	00-04-28	D01	1	17520000	4
	20000501	00-05-10	H03	1	23860000	1
부산	20000502	00-05-18	K01	1	9900000	8
	20000503	00-05-21	H05	1	10510000	7
	20000504	00-05-29	D03	1	24560000	9

2/4

⑦ 백그라운드 밴드가 2페이지씩 보고서에 반복적으로 그려집니다.

가. 컴포넌트 선택

1) 상위 컴포넌트 선택하기



디자인 화면이나 컴포넌트 창에서 컴포넌트를 선택하고 'esc' 키를 누르면 상위 컴포넌트가 선택됩니다.

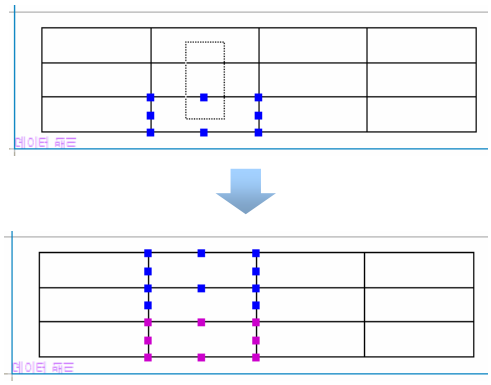
➢ **고정 테이블 예**

- 고정 테이블의 셀에서 'esc'키 입력 → 고정 테이블 선택됨
- 고정 테이블에서 'esc'키 입력 → 데이터 밴드 선택됨
- 데이터 밴드에서 'esc'키 입력 → 리포트 선택됨
- 리포트에서 'esc' 키 입력 → 리포트 템플릿 선택됨

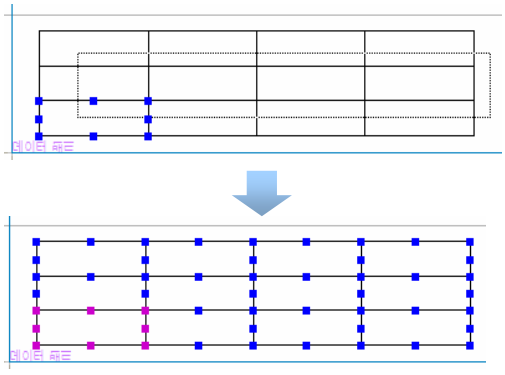
2) 멀티 라벨 선택하기

테이블, 고정 테이블, 크로스탭 컴포넌트에서 Ctrl키나 Shift키를 누른 상태에서 라벨을 선택하면 여러 라벨을 선택할 수 있습니다. 또는 아래와 같이 선택하고자 하는 라벨을 포함하는 영역을 마우스로 드래그하여 선택할 수 있습니다.

일부 선택



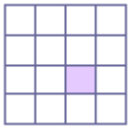
모든 라벨 선택



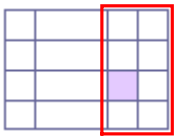
나. 컴포넌트 이동

컴포넌트를 선택한 상태에서 왼쪽 마우스로 드래그 하면 컴포넌트가 이동됩니다.
컴포넌트를 1포인트씩 이동하려면 Alt 키를 누른 상태에서 방향 키를 누릅니다.

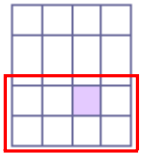
테이블, 고정 테이블, 크로스탭 컴포넌트의 경우에는 오른쪽 마우스로 드래그 해야 이동됩니다.
또한 컴포넌트 내의 특정 라벨을 선택하고 Alt 키를 누른 상태에서 방향 키를 누르면 선택한 라벨을 기준으로 오른쪽, 아래쪽 방향의 라벨들만 이동됩니다.



. 원래 위치



. 오른쪽 방향으로 라벨 이동 후

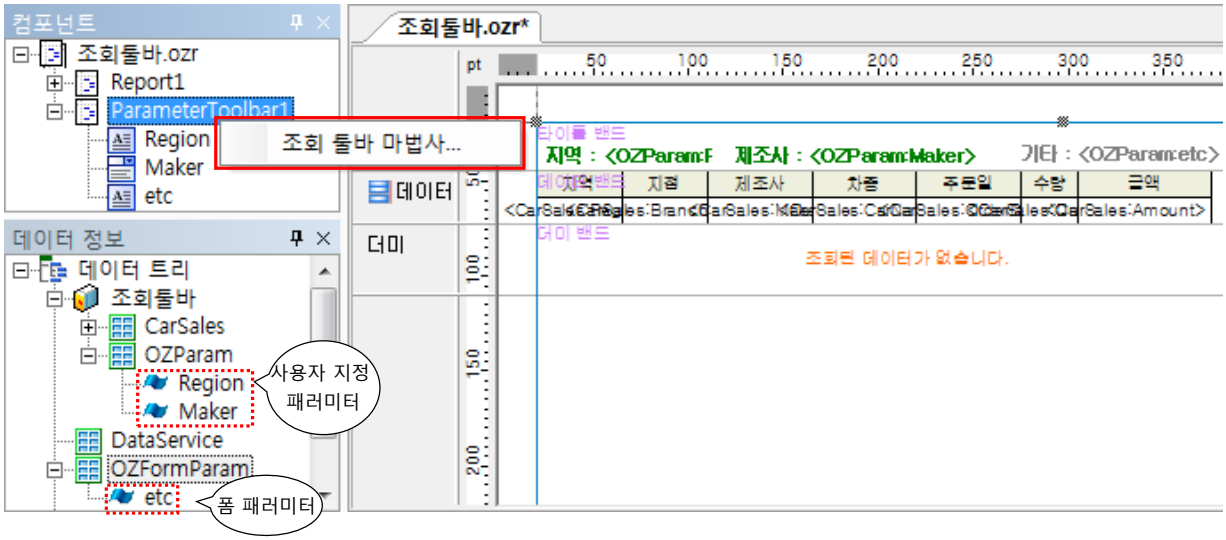


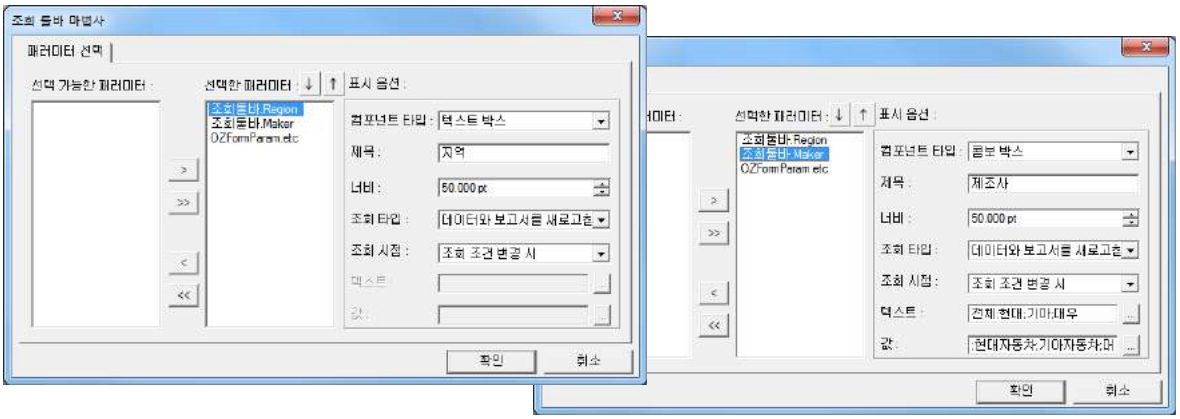
. 아래쪽 방향으로 라벨 이동 후

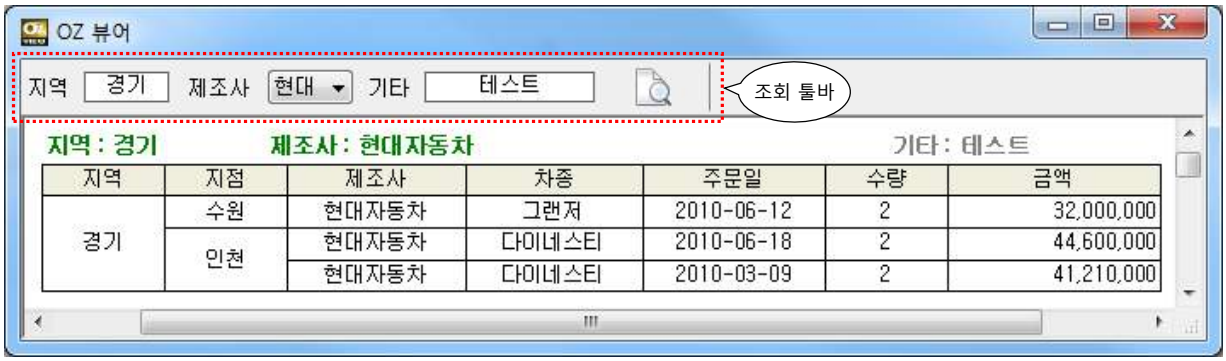
조회 툴바는 검색 조건을 입력 받는 UI를 뷰어 툴바에 보여주는 기능입니다.
조회 툴바는 텍스트, 텍스트 박스, 콤보 박스, 조회 버튼으로 구성됩니다.
조회 툴바를 이용하려면 사전에 '사용자 지정 패러미터'나 '폼 패러미터'를 만들어 둬야 합니다.
'조회 타입' 옵션은 '데이터와 보고서 새로고침', '보고서만 새로고침', '라벨의 텍스트만 새로고침' 중에서 선택할 수 있습니다.

➢ 조회 툴바 사용 예

컴포넌트 창의 'ParameterToolbar1'의 팝업 메뉴에서 '조회 툴바 마법사' 를 실행합니다. 마법사에서 뷰어 툴바에 보여줄 패러미터를 '선택한 패러미터'로 이동한 후 각 패러미터에 대한 표시 옵션을 지정합니다. 미리보기 한 후 조회 툴바의 값을 변경하면 보고서의 데이터가 갱신됩니다.

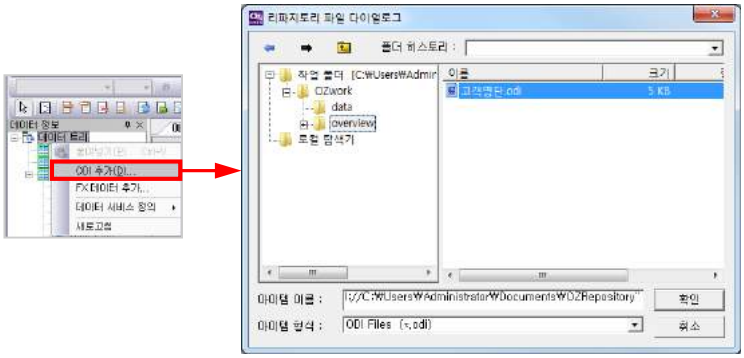






보고서에 데이터 라벨이나 테이블, 차트, 크로스탭과 같이 데이터 필드를 설정하는 마법사를 이용하려면 먼저 데이터 정보가 정의된 ODI 파일을 보고서에 추가해야 합니다.

가. 보고서에 ODI 추가



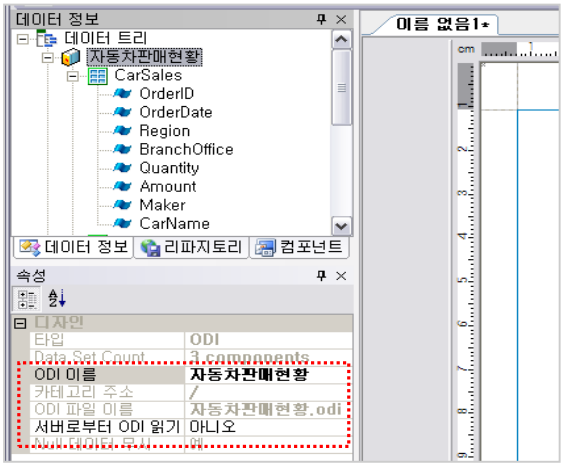
① '데이터 정보' 창의 '데이터 트리' 팝업 메뉴에서 'ODI 추가' 선택, 또는 '리파지토리' 창에서 ODI 파일 더블 클릭

② ODI 파일 선택

③ '확인' 버튼 클릭

나. ODI 속성

추가된 ODI는 '데이터 정보' 창의 '데이터 트리' 아래 생성됩니다.
ODI 선택 후 '속성' 창에서 다음 속성을 확인할 수 있습니다.



- **ODI 이름** :
ODI 파일명으로 설정되며 동일 ODI를 여러 번 추가하면 이름 뒤에 순번이 추가됩니다.

- **카테고리 주소** :
작업 폴더를 기준으로 하는 ODI 상대 경로를 표시합니다. 이 카테고리 주소는 오즈 서버의 리파지토리를 기준으로 하는 ODI 파일 경로와 일치해야 합니다.

※ 리포트 파일(.ozr)을 디자이너로 열었을 때, 추가된 ODI 파일이 '작업 폴더' 아래 '카테고리 주소' 폴더에 없으면 인식되지 않습니다. 이 때 작업폴더 경로를 변경하거나 ODI 파일 위치를 변경하여 경로를 맞춰주어야 합니다.

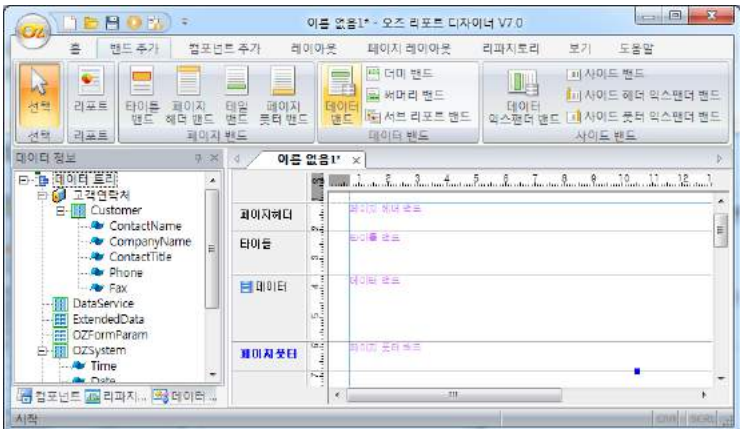
- **서버로부터 ODI 읽기** :
'서버로부터 ODI 읽기' 속성을 '예'로 하면 연결되어 있는 리파지토리 서버의 odi 파일을 인식하여 오즈 서버가 데이터 조회를 처리하게 됩니다. 디자이너에서 데이터 조회를 할 수 없는 경우 유용하게 사용됩니다.

4. 밴드

4.1 밴드 개요

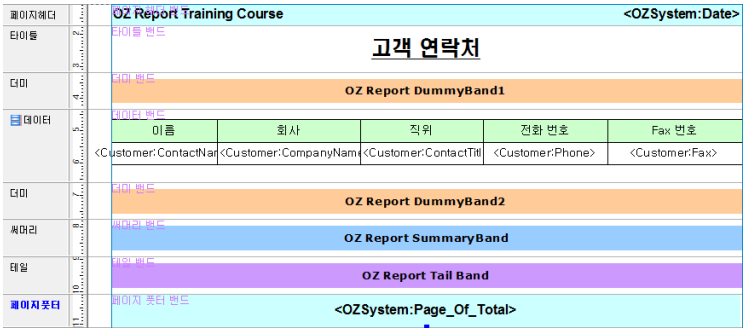
리포트 디자인은 밴드 단위로 이루어 집니다. 라벨, 멀티플렉스, 도형, 입력 컴포넌트는 밴드 위에 디자인 되며 밴드 고유의 출력 특성에 따라 그 위에 배치된 컴포넌트의 보여지는 결과가 정해집니다.

리포트는 여러 개의 밴드로 구성되고 각각의 밴드는 배치되는 순서가 정해져 있으며, 만들어지는 순간 정해진 위치로 자동 정렬됩니다.

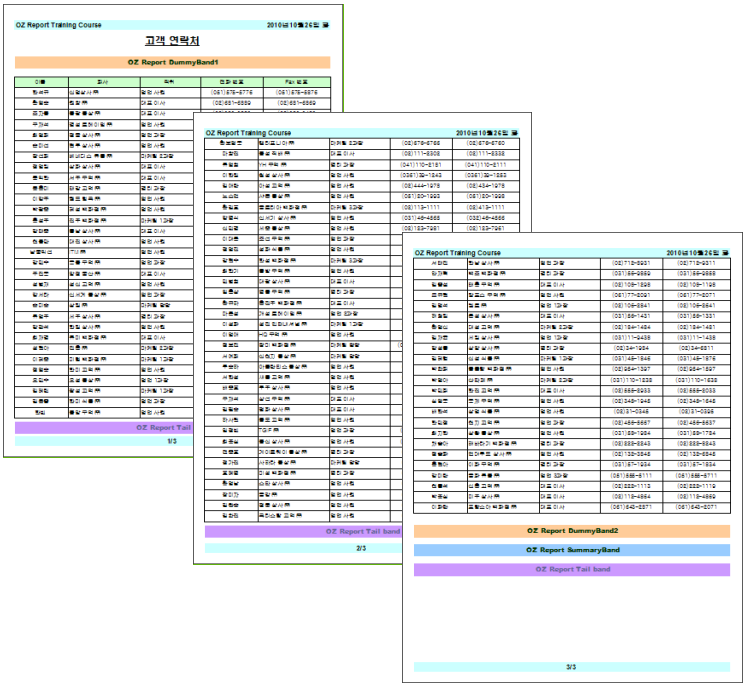


① '밴드추가' 탭에서 사용할 밴드를 선택 합니다.

② 리포트 위에 마우스를 대각선으로 그리면 밴드가 추가됩니다.



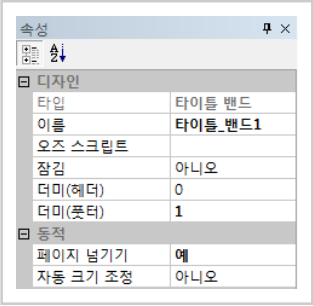
③ 밴드 위에 컴포넌트를 그려 디자인 합니다.



④ 미리보기 결과입니다.

가. 타이틀 밴드

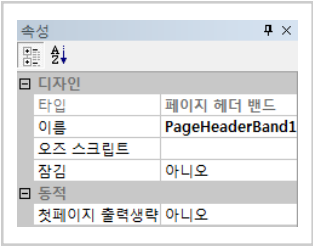
타이틀 밴드는 보고서 첫 페이지 상단에 한번 출력되는 밴드입니다.
보고서의 제목을 그려주는 용도에 주로 사용됩니다.



속성	설명
더미(헤더)	타이틀 밴드 위로 설정한 개수만큼 더미 밴드 추가
더미(풋터)	타이틀 밴드 아래로 설정한 개수만큼 더미 밴드 추가
페이지 넘기기	타이틀 밴드 이후 밴드는 다음 페이지에 그림
자동 크기 조정	타이틀 밴드 위의 컴포넌트 높이에 맞추어 밴드 높이 자동 조정

나. 페이지 헤더 밴드

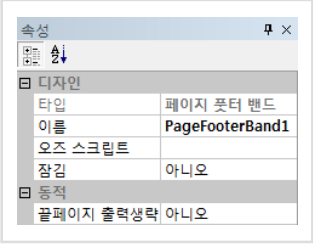
페이지 헤더 밴드는 문서의 머리글처럼 매 페이지 상단에 반복 출력되는 밴드로 타이틀 밴드보다 위에 그려집니다.
보고서의 제목이나 페이지 번호, 회사 로고 등을 그려주는 용도에 사용됩니다.



속성	설명
첫 페이지 출력 생략	보고서의 첫 페이지에 페이지 헤더 밴드의 출력 생략 여부

다. 페이지 풋터 밴드

페이지 풋터 밴드는 문서의 바닥글처럼 매 페이지 하단 특정 위치에 반복 출력되는 밴드입니다.
보고서의 페이지 번호, 회사 로고 등을 그려주는 용도에 사용됩니다.



속성	설명
끝 페이지 출력 생략	보고서의 끝 페이지에 페이지 풋터 밴드의 출력 생략 여부

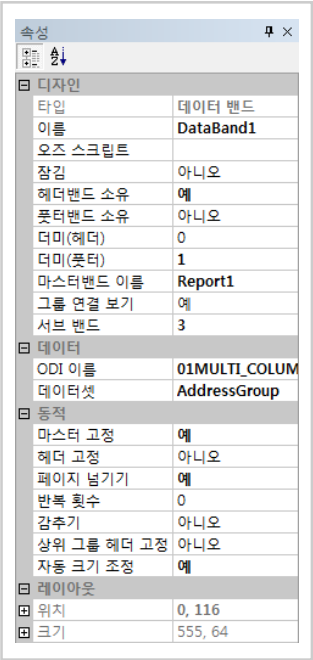
라. 테일 밴드

테일 밴드는 매 페이지 하단에 반복 출력되는 밴드로 페이지 풋터 밴드를 제외한 다른 밴드 아래 이어져 그려집니다.
보고서의 페이지 별 소계를 그려주는 용도에 주로 사용됩니다.

속성	설명
감추기	테일 밴드 보여줄지 여부 (화면에 테일 밴드를 출력하지 않아도 스크립트는 실행됨)

가. 데이터 밴드

데이터 밴드는 데이터 셋의 데이터를 보여주기 위해 사용되는 밴드입니다.

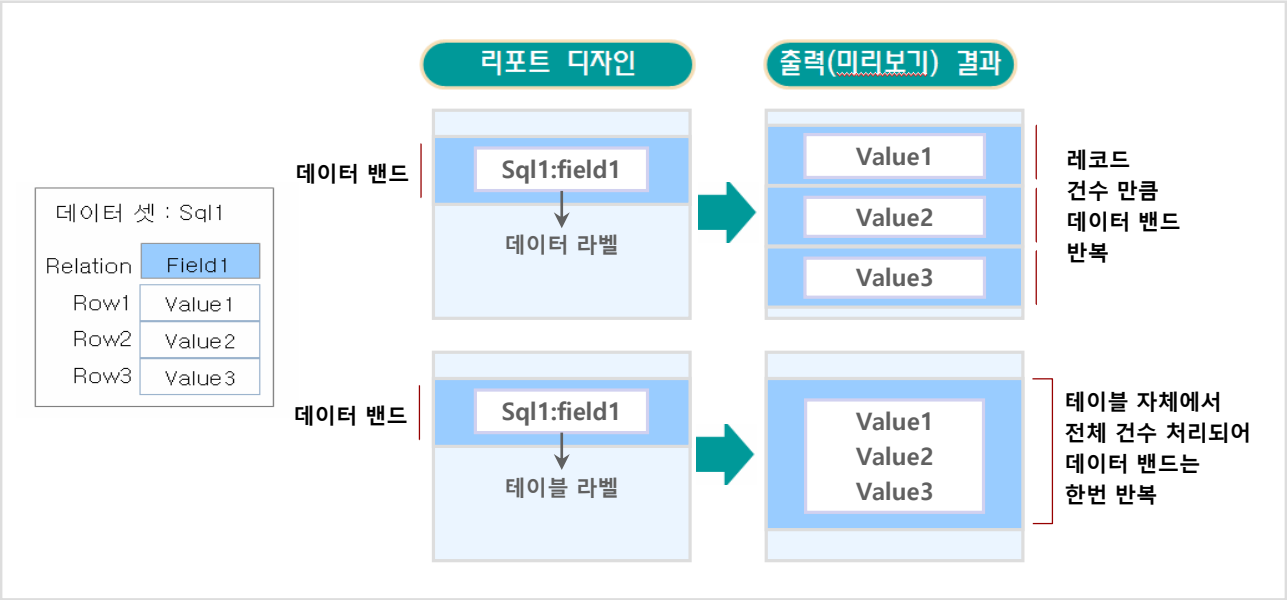


속성	설명
마스터밴드 이름	마스터 밴드로 설정할 다른 데이터 밴드 설정 마스터 밴드가 없을 경우 리포트가 자동 설정됨
그룹 연결 보기	그룹 라벨간의 연결 관계를 화살표로 표시할지 여부를 설정
ODI 이름	데이터 밴드에 보여줄 ODI 이름 설정 (필수) '데이터 정보' 창에 추가된 ODI 선택 가능
데이터셋	데이터 밴드에 보여줄 데이터 셋 이름 설정 (필수)
마스터 고정	마스터 밴드 내용을 데이터 밴드가 그려지는 매 페이지 반복 출력하도록 설정
헤더 고정	데이터 헤더 밴드를 데이터 밴드가 그려지는 매 페이지 반복 출력하도록 설정
반복 횟수	반복 횟수가 데이터 셋의 레코드 건수보다 클 때만 유효한 속성으로, '반복 횟수 - 레코드 건수' 만큼 데이터 밴드가 반복됨
상위 그룹 헤더 고정	상위 그룹 헤더를 데이터 밴드가 그려지는 매 페이지 반복 출력하도록 설정

데이터 밴드 위에 그려지는 컴포넌트 타입에 따라 데이터 밴드의 반복되는 횟수가 달라집니다.

'고정테이블', '라벨' 컴포넌트를 데이터 밴드에 그리면 데이터 셋의 레코드 건수만큼 수직 방향으로 데이터 밴드가 반복되면서 현재 행의 데이터가 데이터 라벨에 보여집니다.

'테이블', '차트', '크로스탭' 과 같이 컴포넌트 자체에서 모든 레코드의 데이터를 보여주는 컴포넌트를 데이터 밴드에 그리면 데이터 밴드는 한번만 출력됩니다.



[컴포넌트 타입 별 데이터 밴드 반복 횟수]

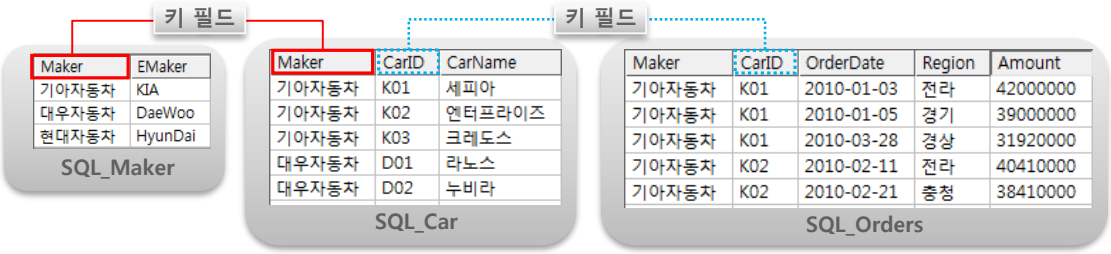
나. 마스터 밴드

데이터 밴드는 다른 데이터 밴드를 마스터 밴드로 설정하여 밴드간의 마스터-디테일 관계를 형성할 수 있습니다.

1) 일반 데이터 셋을 설정한 데이터 밴드의 마스터 밴드 설정 방법

동일한 필드 정보를 가진 데이터 셋을 설정한 데이터 밴드 간의 마스터–디테일 관계를 설정하여 그룹핑 보고서를 만드는 방법입니다.

① 제조사 별, 차종 별 주문 내역을 보여주는 보고서를 만들고자 할 때, 쿼리 디자이너에서 조회된 데이터 셋 결과를 다음과 같이 가정합니다.



Maker	EMaker
기아자동차	KIA
대우자동차	DaeWoo
현대자동차	HyunDai

Maker	CarID	CarName
기아자동차	K01	세피아
기아자동차	K02	엔터프라이즈
기아자동차	K03	크레도스
대우자동차	D01	라노스
대우자동차	D02	누비라

Maker	CarID	OrderDate	Region	Amount
기아자동차	K01	2010-01-03	전라	42000000
기아자동차	K01	2010-01-05	경기	39000000
기아자동차	K01	2010-03-28	경상	31920000
기아자동차	K02	2010-02-11	전라	40410000
기아자동차	K02	2010-02-21	충청	38410000

'제조사 정보'인 SQL_Maker 데이터 셋이 마스터 밴드에 보여지게 됩니다.
'차종 정보'인 SQL_Car 데이터 셋은 디테일 밴드에 보여지게 됩니다.
'주문 내역'인 SQL_Orders 데이터 셋은 '차종 정보'를 마스터로 하는 디테일 밴드에 보여지게 됩니다.

제조사 별로 차종 정보가 보여지려면 SQL_Maker, SQL_Car 데이터 셋의 제조사명이 같은 데이터 끼리 그룹핑 되어야 하기 때문에 'Maker' 필드가 '키 필드'가 됩니다.

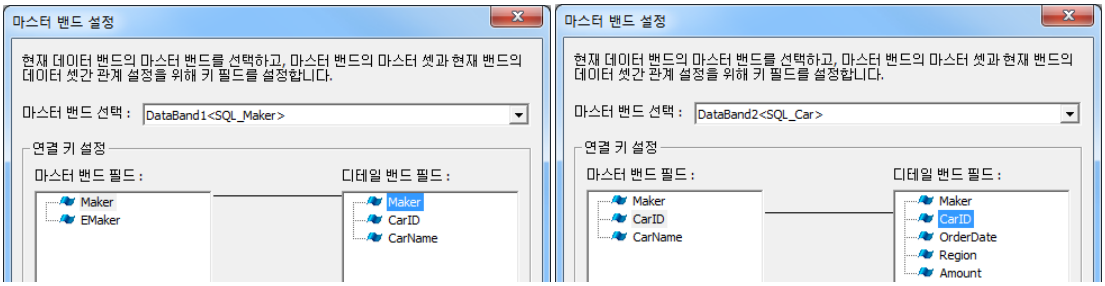
마찬가지로 차종 별 주문 내역을 보려면 SQL_Car, SQL_Orders 데이터셋의 차종 ID가 같은 데이터 끼리 그룹핑되어야 하기 때문에 'CarID' 필드가 '키 필드'가 됩니다.

이 때, 키 필드의 데이터는 순차적으로 비교되기 때문에 반드시 키 필드로 정렬된 결과를 가져와야 합니다.

② 리포트 위에 마스터-디테일 관계를 표현하고자 하는 데이터 셋 개수 만큼 데이터 밴드를 3개 그립니다.

③ 데이터 밴드의 '데이터셋' 속성 값을 각각 'SQL_Maker', 'SQL_Car', 'SQL_Orders'로 변경합니다.

④ 데이터 밴드의 '마스터밴드 이름' 속성 값을 '...' 아이콘을 눌러 변경합니다. 마스터밴드 이름 DataBand1 ...
'마스터 밴드 설정' 창의 '마스터 밴드 선택' 에서 마스터 밴드를 선택합니다.

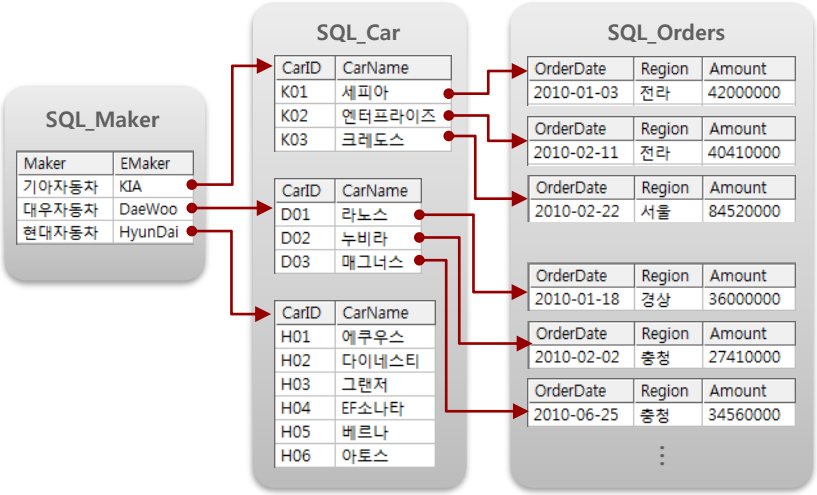


⑤ '마스터 밴드 설정' 창에서 '마스터 밴드 필드'와 '디테일 밴드 필드'의 키 필드를 드래그&드롭으로 연결합니다.

2) 마스터-디테일 셋을 설정한 데이터 밴드의 마스터 밴드 설정 방법

쿼리 디자이너에서 데이터 셋을 생성할 때, 데이터 셋간의 마스터-디테일 관계가 설정되어 있으면, 리포트 디자이너에서 데이터 밴드의 데이터 셋 설정할 때 마스터 밴드는 자동 설정됩니다. 또한 별도로 키 필드를 지정할 필요가 없습니다.

① 쿼리 디자이너에서 조회된 데이터 셋 결과가 다음과 같다고 가정합니다. (마스터 셋 설정 방법은 35 page 참고)



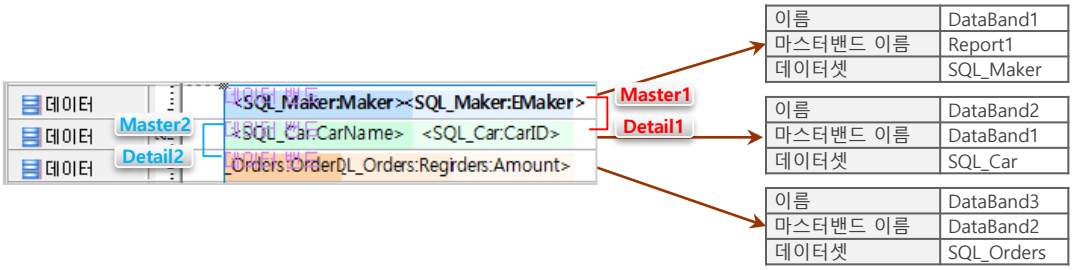
② 데이터 셋 개수 만큼 데이터 밴드를 3개 그립니다.

③ 데이터 밴드의 '데이터셋' 속성 값을 각각 'SQL_Maker', 'SQL_Car', 'SQL_Orders'로 변경합니다.

④ 데이터 밴드의 '마스터밴드 이름' 속성 값은 자동으로 마스터 데이터 셋이 설정된 데이터 밴드로 변경됩니다.

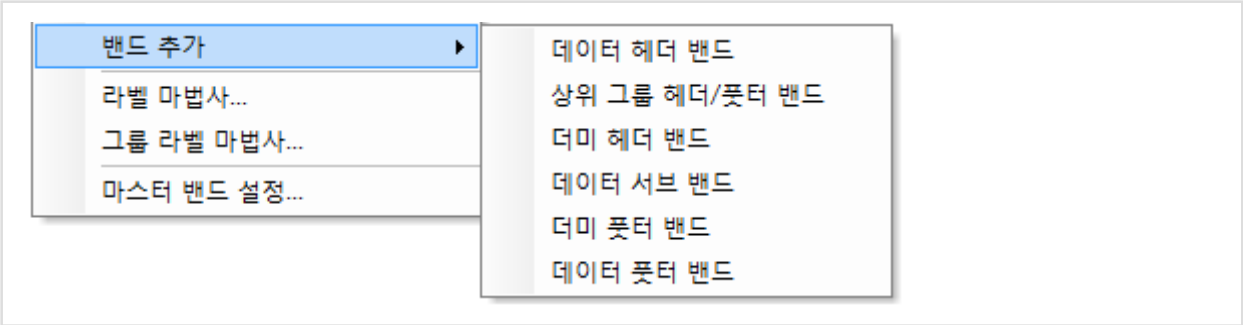
3) 마스터 밴드 설정 결과

앞의 1), 2) 번에서 디자인한 화면은 다음과 같습니다.



보고서 실행 결과 다음과 같이 마스터 밴드 사이 사이에 키 값이 일치하는 디테일 밴드가 그려집니다.

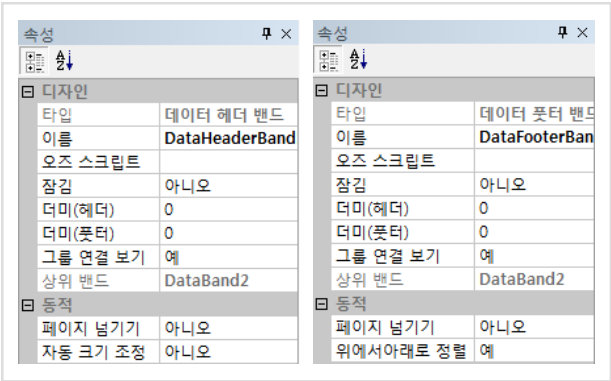
데이터 밴드의 팝업 메뉴에서 다음과 같은 밴드를 추가할 수 있습니다.



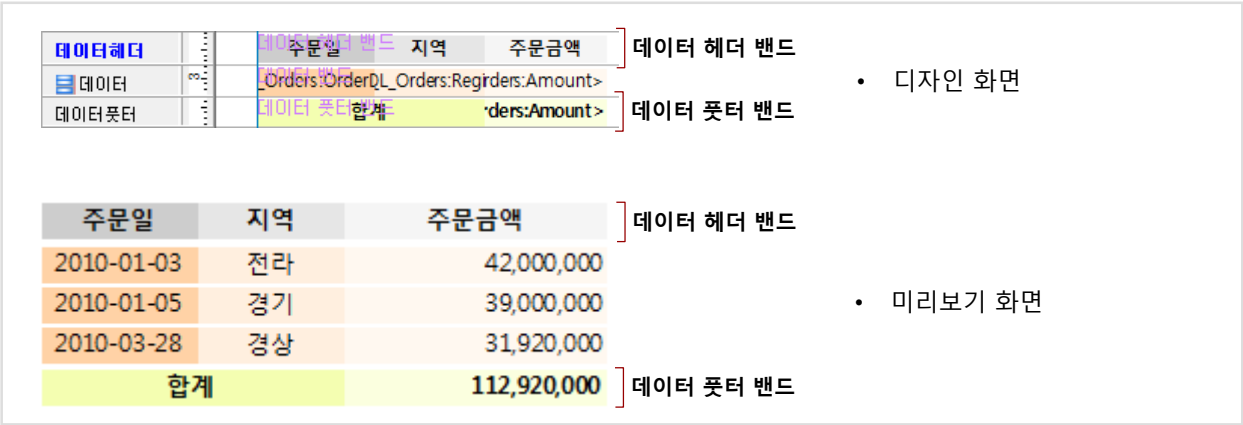
다. 데이터 헤더 밴드, 데이터 풋터 밴드

데이터 헤더 밴드는 데이터 밴드 직전에 한번 출력되는 밴드입니다.
데이터 밴드에 대한 제목이나 데이터 건수 등을 그려주는 용도에 사용됩니다.

데이터 풋터 밴드는 데이터 밴드 직후에 한번 그려지는 밴드입니다.
데이터 밴드에 대한 합계 등을 그려주는 용도에 사용됩니다.



속성	설명
상위 밴드	데이터 헤더 밴드를 추가한 데이터 밴드
위에서 아래로 정렬	'아니오' 인 경우 '데이터 풋터 밴드'가 페이지 하단에 위치하게 됨



라. 상위 그룹 헤더/풋터 밴드

데이터 셋의 특정 필드 값으로 그룹핑된 보고서를 그리기 위해서 '상위 그룹 헤더/풋터' 밴드를 이용합니다.

① 쿼리 디자이너에서 조회된 데이터 셋 결과를 다음과 같이 가정합니다.

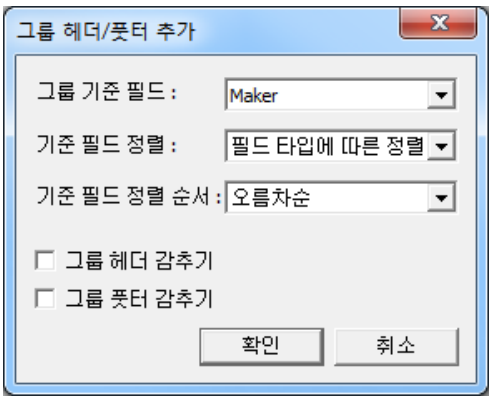
SQL_CarOrderInfo

Maker	EMaker	CarID	CarName	OrderDate	Region	Amount
현대자동차	HyunDai	H04	EF소나타	2010-04-15	서울	88600000
현대자동차	HyunDai	H04	EF소나타	2010-05-02	충청	77000000
현대자동차	HyunDai	H02	다이네스티	2010-01-05	서울	93060000
현대자동차	HyunDai	H02	다이네스티	2010-03-03	전라	30500000
현대자동차	HyunDai	H02	다이네스티	2010-03-09	경기	41210000
현대자동차	HyunDai	H02	다이네스티	2010-06-05	경상	32950000
현대자동차	HyunDai	H02	다이네스티	2010-06-18	경기	44600000
현대자동차	HyunDai	H03	그랜저	2010-01-24	경상	21000000
현대자동차	HyunDai	H03	그랜저	2010-05-10	경상	23800000
현대자동차	HyunDai	H03	그랜저	2010-06-12	경기	32000000
현대자동차	HyunDai	H01	에쿠우스	2010-04-21	전라	65000000
현대자동차	HyunDai	H05	베르나	2010-02-21	경상	32510000
대우자동차	DaeWoo	D01	라노스	2010-01-18	경상	36000000
대우자동차	DaeWoo	D01	라노스	2010-02-28	경기	46610000
대우자동차	DaeWoo	D01	라노스	2010-03-04	전라	58520000
대우자동차	DaeWoo	D01	라노스	2010-04-21	경상	37520000
대우자동차	DaeWoo	D01	라노스	2010-05-25	경기	32520000
대우자동차	DaeWoo	D02	누비라	2010-02-02	충청	27410000
대우자동차	DaeWoo	D02	누비라	2010-03-13	서울	58470000
대우자동차	DaeWoo	D02	누비라	2010-06-28	서울	48060000
대우자동차	DaeWoo	D03	매그너스	2010-06-25	충청	34560000
기아자동차	KIA	K01	세피아	2010-01-03	전라	42000000
기아자동차	KIA	K01	세피아	2010-01-05	경기	39000000
기아자동차	KIA	K01	세피아	2010-03-28	경상	31920000
기아자동차	KIA	K02	엔터프라이즈	2010-02-11	전라	40410000
기아자동차	KIA	K02	엔터프라이즈	2010-02-21	충청	38410000
기아자동차	KIA	K02	엔터프라이즈	2010-06-11	전라	37300000
기아자동차	KIA	K03	크레도스	2010-02-22	서울	84520000
기아자동차	KIA	K03	크레도스	2010-03-29	경상	35560000
기아자동차	KIA	K03	크레도스	2010-04-06	경기	67020000
기아자동차	KIA	K03	크레도스	2010-04-30	충청	27860000

② 데이터 밴드를 하나 그린 후 '데이터셋' 속성 값을 'SQL_CarOrderInfo'로 변경합니다.

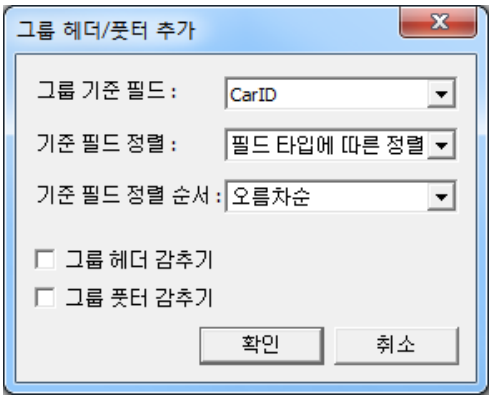
③ 데이터 밴드의 팝업 메뉴에서 '밴드 추가' → '상위 그룹 헤더/풋터 밴드'를 선택합니다.

④ '그룹 헤더/풋터 추가' 창의 '그룹 기준 필드'에 'Maker'를 선택하고, '확인' 버튼을 클릭합니다.



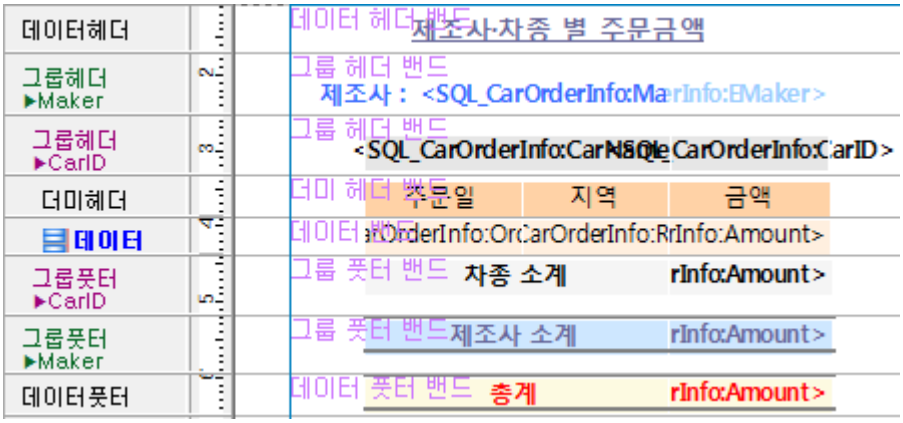
- '그룹 기준 필드'에 설정한 필드 값으로 그룹핑된 데이터를 '그룹 헤더·풋터 밴드'에 보여줄 수 있습니다.
- '기준 필드 정렬'을 선택하여 데이터 셋 결과를 다시 정렬한 후 그룹 기준 필드 별로 보여줄 수 있습니다.

⑤ 데이터 밴드에서 다시 '그룹 헤더/풋터 추가' 창을 띄운 후 '그룹 기준 필드'에 'CarID'를 선택합니다.



- '그룹 헤더 감추기'를 선택하면 '그룹 헤더 밴드'는 추가되지 않습니다.
- '그룹 풋터 감추기'를 선택하면 '그룹 풋터 밴드'는 추가되지 않습니다.

⑥ 추가된 그룹헤더·풋터 밴드를 이용하여 다음과 같이 디자인할 수 있습니다.
그룹 헤더 밴드에는 그룹핑된 '기준 필드'의 데이터를 보여주고, 그룹 풋터 밴드에는 소계를 보여주는 형태로 디자인 할 수 있습니다.



⑦ 보고서 실행 결과 다음과 같이 하나의 데이터 셋이 제조사 별, 차종 별로 그룹핑 되어 보여지게 됩니다.

제조사·차종 별 주문금액

제조사 : 기아자동차 KIA

세피아		K01
주문일	지역	금액
2010-01-03	전라	42,000,000
2010-01-05	경기	39,000,000
2010-03-28	경상	31,920,000
차종 소계		112,920,000

엔터프라이즈		K02
주문일	지역	금액
2010-02-11	전라	40,410,000
2010-02-21	충청	38,410,000
2010-06-11	전라	37,300,000
차종 소계		116,120,000

크레도스		K03
주문일	지역	금액
2010-02-22	서울	84,520,000
2010-03-29	경상	35,560,000
2010-04-06	경기	67,020,000
2010-04-30	충청	27,860,000
2010-05-20	서울	82,560,000
차종 소계		297,520,000

제조사 소계	526,560,000

제조사 : 대우자동차 DaeWoo

라노스		D01
주문일	지역	금액
2010-01-18	경상	36,000,000
2010-02-28	경기	46,610,000
2010-03-04	전라	58,520,000
2010-04-21	경상	37,520,000
2010-05-25	경기	32,520,000
차종 소계		211,170,000

누비라		D02
주문일	지역	금액
2010-02-02	충청	27,410,000
2010-03-13	서울	58,470,000
2010-06-28	서울	48,060,000
차종 소계		133,940,000

매그너스		D03
주문일	지역	금액
2010-06-25	충청	34,560,000
차종 소계		34,560,000

제조사 소계	379,670,000

제조사 : 현대자동차 HyunDai

에쿠우스		H01
주문일	지역	금액
2010-04-21	전라	65,000,000
차종 소계		65,000,000

다이네스티		H02
주문일	지역	금액
2010-01-05	서울	93,060,000
2010-03-03	전라	30,500,000
2010-03-09	경기	41,210,000
2010-06-05	경상	32,950,000
2010-06-18	경기	44,600,000
차종 소계		242,320,000

그랜저		H03
주문일	지역	금액
2010-01-24	경상	21,000,000
2010-05-10	경상	23,800,000
2010-06-12	경기	32,000,000
차종 소계		76,800,000

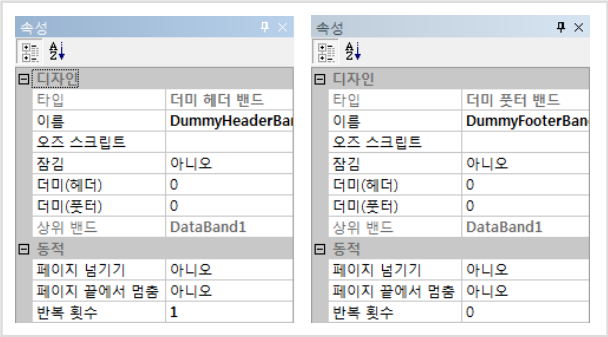
EF소나타		H04
주문일	지역	금액
2010-04-15	서울	88,600,000
2010-05-02	충청	77,000,000
차종 소계		165,600,000

베르나		H05
주문일	지역	금액
2010-02-21	경상	32,510,000
차종 소계		32,510,000

제조사 소계	582,230,000
총계	1,488,460,000

마. 더미 헤더 밴드, 더미 풋터 밴드

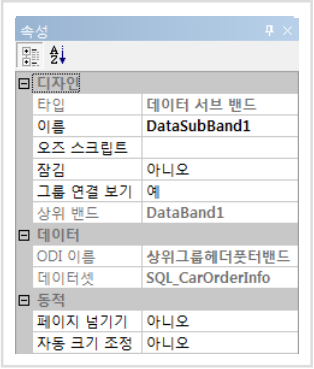
더미 헤더 밴드는 데이터 밴드 직전에 출력되는 밴드로 데이터 헤더 밴드와 데이터 밴드 사이에 그려집니다.
더미 풋터 밴드는 데이터 밴드 직후에 출력되는 밴드로 데이터 밴드와 데이터 풋터 밴드 사이에 그려집니다.
더미 헤더 밴드나 더미 풋터 밴드에 데이터 필드를 추가하려면 요약 라벨로 추가해야 합니다.



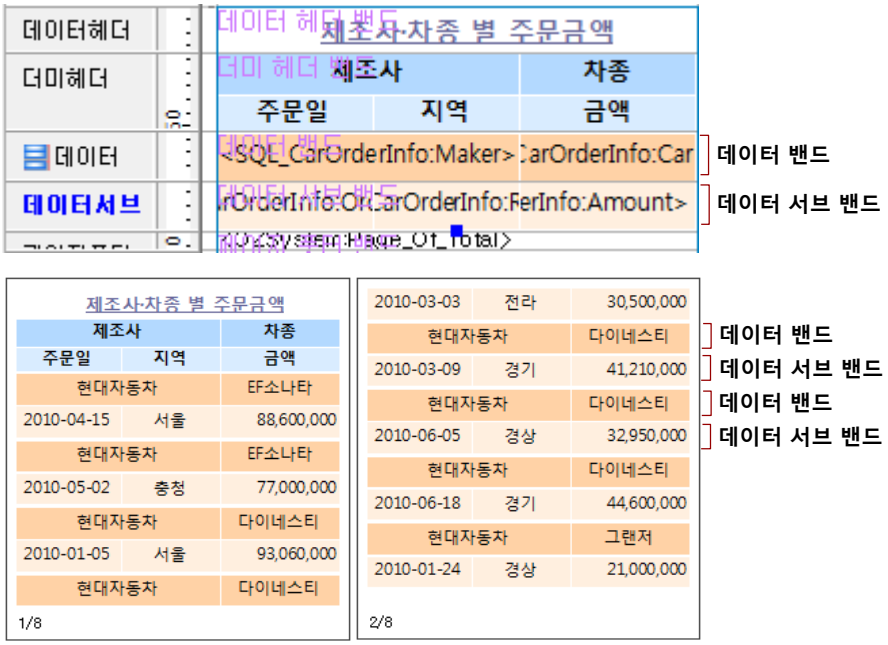
속성	설명
더미(헤더)	더미 밴드 직전에 설정한 개수 만큼 새로운 더미 밴드가 생성됨
더미(풋터)	더미 밴드 아래 설정한 개수 만큼 더미 밴드가 생성됨
반복 횟수	더미 밴드를 몇 번 출력할 지 설정. 기본값은 0으로 더미 밴드가 출력되지 않음

바. 데이터 서브 밴드

데이터 서브 밴드는 데이터 밴드 하단에 그려지며 데이터 밴드가 n건 반복되면, 데이터 서브 밴드도 n건 반복됩니다.
보고서를 보여줄 때 데이터 밴드가 페이지 남은 공간에 들어가지 못하면 밴드 자체가 다음 페이지로 넘어가기 때문에 밴드 내용을 서브 밴드에 분리하여 그릴 수 있습니다.



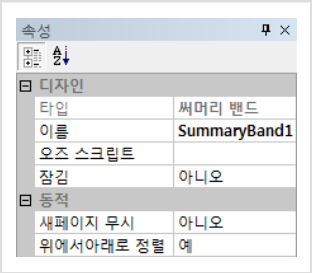
속성	설명
상위 밴드	데이터 서브 밴드를 추가한 데이터 밴드
ODI 이름	데이터 밴드에 설정된 ODI 이름
데이터셋	데이터 밴드에 설정된 데이터셋



- 데이터 밴드 팝업 메뉴에서 '밴드 추가' → '데이터 서브 밴드'를 선택하면 데이터 밴드 바로 아래 서브 밴드가 그려집니다.
- 서브 밴드에 디자인된 것은 데이터 밴드와 함께 반복됩니다.

가. 써머리 밴드

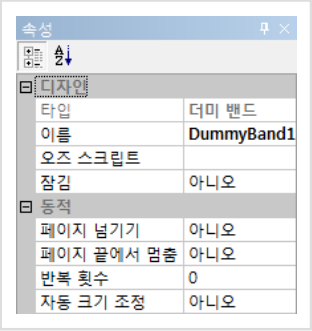
써머리 밴드는 전체 보고서에 대한 요약 정보를 보여주는 밴드입니다.
보고서의 마지막 페이지에 한 번만 출력됩니다



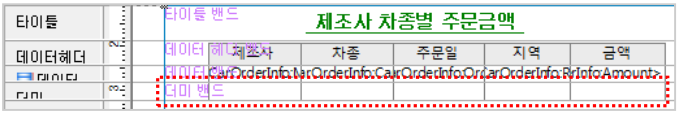
속성	설명
새페이지 무시	써머리 밴드 직전 밴드의 '페이지 넘기기' 속성 값이 '예'인 경우 '아니오'로 처리되어 직전 밴드에 붙어 출력됨

나. 더미 밴드

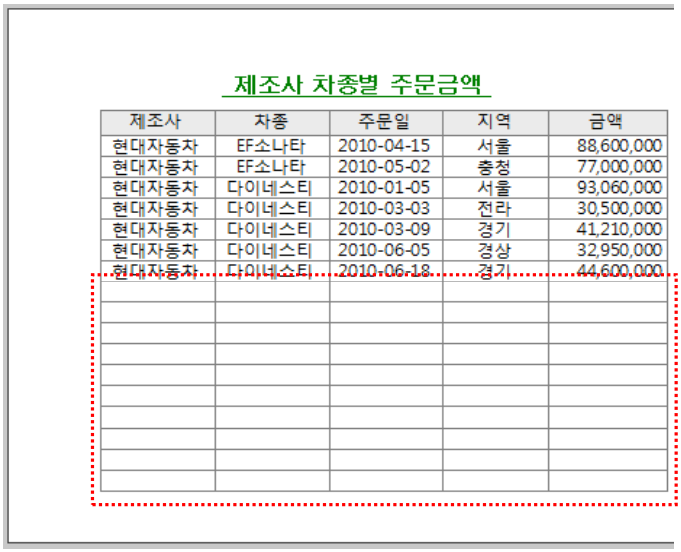
데이터 셋 연결 없이 임의의 위치에 밴드를 추가하고자 하는 경우 사용되는 밴드입니다.
보고서의 남은 공간을 빈칸으로 채울 때 주로 사용됩니다.



속성	설명
페이지 끝에서 멈춤	더미 밴드의 반복 횟수 만큼 출력되다가 현재 페이지 이후로는 더 이상 출력하지 않음
반복 횟수	반복 횟수 만큼 더미 밴드가 그려짐. 기본값은 0으로 더미 밴드가 그려지지 않음
자동 크기 조정	더미 밴드 위의 컴포넌트 높이에 맞추어 밴드 높이 자동 조정



페이지 끝에서 멈춤	예
반복 횟수	50



제조사 차종별 주문금액

제조사	차종	주문일	지역	금액
현대자동차	EF소나타	2010-04-15	서울	88,600,000
현대자동차	EF소나타	2010-05-02	충청	77,000,000
현대자동차	다이네스티	2010-01-05	서울	93,060,000
현대자동차	다이네스티	2010-03-03	전라	30,500,000
현대자동차	다이네스티	2010-03-09	경기	41,210,000
현대자동차	다이네스티	2010-06-05	경상	32,950,000
현대자동차	다이네스티	2010-06-18	경기	44,600,000

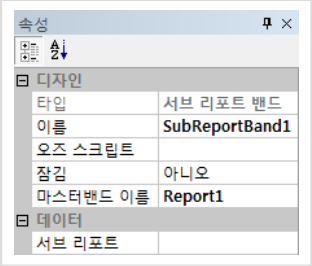
① 목록의 남은 공간을 더미 밴드로 채우기 위해서는 데이터 밴드 아래 '더미 밴드'를 추가합니다.

② '더미 밴드'의 '페이지 끝에서 멈춤'을 '예'로 변경하고, '반복 횟수' 속성을 한 페이지에 나올 수 있는 목록 개수 이상으로 설정합니다.

③ 보고서 출력된 결과입니다. 목록의 남은 공간이 더미 밴드에 그려진 빈 라벨로 채워집니다.

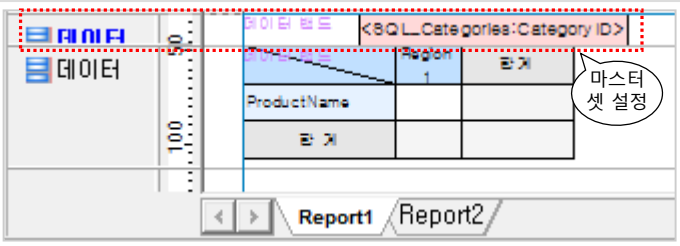
4.5 서브 리포트 밴드

서로 다른 용지 정보(크기, 방향)를 가진 리포트를 한 보고서로 표현하기 위해서는 다중 보고서로 구현해야 하는데, 리포트 간의 마스터-디테일 관계를 설정해야 할 때, 서브 리포트 밴드를 이용할 수 있습니다.



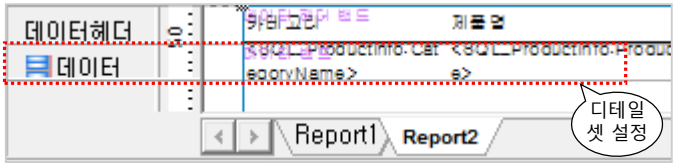
속성	설명
마스터밴드 이름	서브 리포트의 마스터 밴드 설정 마스터 밴드가 반복될 때 마다 서브 리포트가 보여짐
서브 리포트	서브 리포트 밴드 위치에 보여줄 리포트 컴포넌트 이름 설정 서브 리포트 밴드 앞 뒤로 자동 페이지 넘김이 됨

'제품 분류' 데이터 셋을 마스터 셋으로 하는 '판매 정보' 디테일 셋과 '제품 재고 정보' 디테일 셋이 있다고 가정할 때, 'Report1' 컴포넌트에는 '제품 분류' 를 보여주는 데이터 밴드(마스터 밴드)와 '판매 정보'를 보여주는 데이터 밴드(디테일 밴드1)를 그리고, 'Report2' 컴포넌트에는 '제품 재고 정보' 를 보여주는 데이터 밴드(디테일 밴드2)를 그릴 수 있습니다.

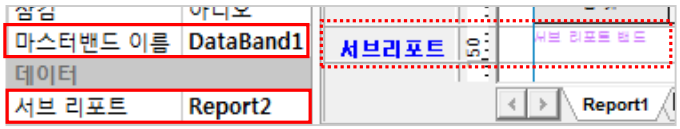


① 리포트 컴포넌트를 2개 추가하여 다중 보고서를 만듭니다.

② 첫 번째 리포트에 마스터 셋을 설정한 데이터 밴드를 추가합니다.

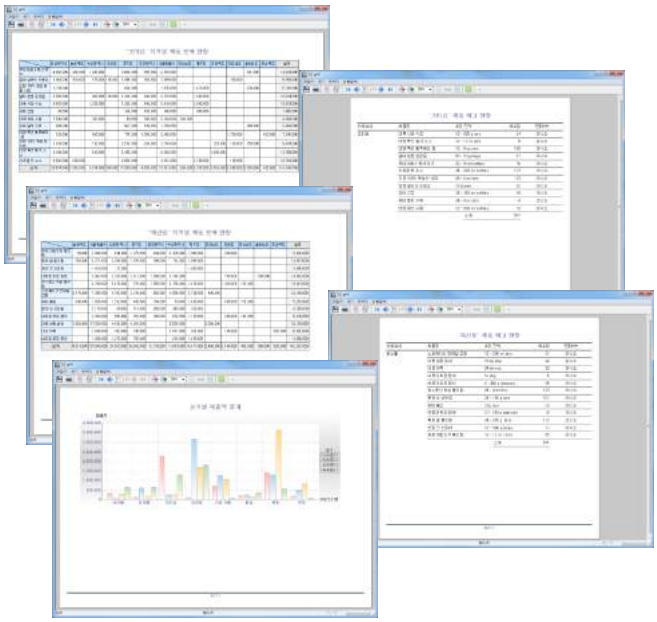


③ 두 번째 리포트에 디테일 셋을 설정한 데이터 밴드를 추가합니다.



④ 첫 번째 리포트에 서브 리포트 밴드를 추가합니다.

⑤ 서브 리포트 밴드의 '서브 리포트' 속성을 'Report2' 로 변경하고, '마스터밴드 이름' 속성을 ②의 마스터 밴드 이름으로 변경합니다.

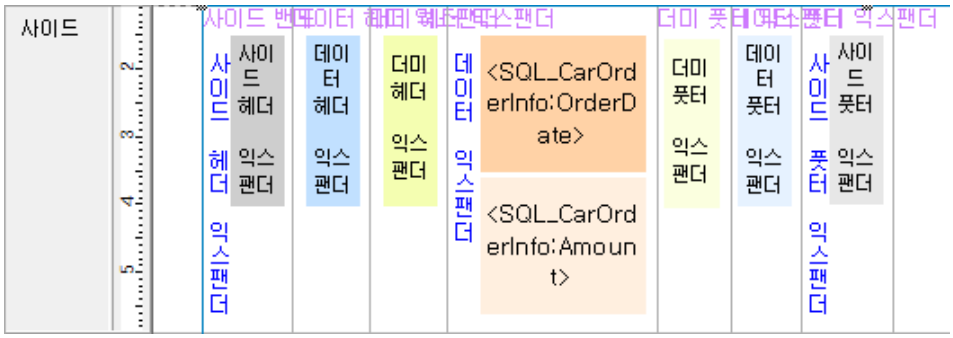


⑥ 보고서를 출력하면 'Report1' → 'Report2' → 'Report1' → 'Report2' ... 구조로 마스터-디테일 형태로 그려집니다.

※ 서브 리포트 밴드 위에는 컴포넌트를 그릴 수 없습니다.

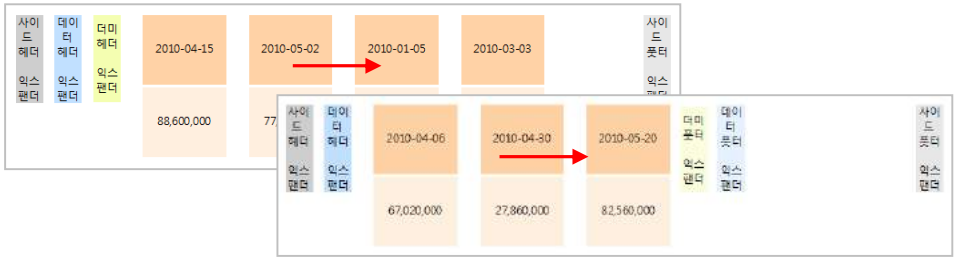
가. 사이드 밴드

앞에서 나온 밴드들은 수직 방향으로 출력되는 데 비해 사이드 밴드 위에 그려지는 밴드들은 수평 방향으로 출력됩니다.



수평 방향으로 출력되는 보고서를 구현하기 위해 먼저 사이드 밴드를 리포트에 추가합니다.

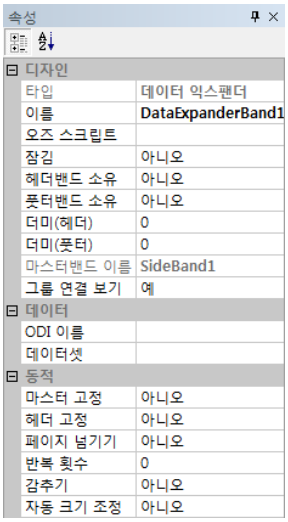
- 사이드 밴드 위에 '사이드 헤더 익스팬더 밴드'를 그리면 '페이지 헤더 밴드'와 같이 매 페이지에 출력되며, 페이지 왼쪽 부분에 그려집니다.
- 사이드 밴드 위에 '사이드 풋터 익스팬더 밴드'를 그리면 '페이지 풋터 밴드'와 같이 매 페이지에 출력되며, 페이지 오른쪽 부분에 그려집니다.
- 사이드 밴드 위에 '데이터 익스팬더 밴드'를 그리면 '데이터 밴드'와 같이 출력되며, 수평 방향으로 밴드가 반복됩니다.



※ 사이트 밴드 위에는 '~ 익스팬더 밴드' 이외의 밴드는 그릴 수 없습니다.

나. 데이터 익스팬더 밴드

데이터 익스팬더 밴드는 데이터 밴드처럼 데이터 셋 정보를 보여주는 용도로 사용됩니다.
데이터 헤더 익스팬더나 더미 헤더 익스팬더에 제목을 그릴 수 있습니다.
데이터 풋터 익스팬더에 합계를 그릴 수 있습니다.



속성	설명
헤더밴드 소유	데이터 헤더 익스팬더를 데이터 익스팬더 직전에 한번 출력
풋터밴드 소유	데이터 풋터 익스팬더를 데이터 익스팬더 직후에 한번 출력
더미(헤더)	더미 헤더 익스팬더를 데이터 헤더 익스팬더와 데이터 익스팬더 사이에 출력
더미(풋터)	더미 풋터 익스팬더를 데이터 익스팬더와 데이터 풋터 익스팬더 사이에 출력
ODI 이름	데이터 익스팬더 밴드에 보여줄 ODI 이름 설정 (필수) '데이터 정보'창에 추가된 ODI 선택 가능
데이터셋	데이터 익스팬더 밴드에 보여줄 데이터 셋 이름 설정 (필수)
헤더 고정	데이터 헤더 익스팬더 밴드를 데이터 익스팬더 밴드가 그려지는 매 페이지 반복 출력하도록 설정

가. 라벨 컴포넌트

라벨 컴포넌트는 텍스트, 데이터 필드 값, 시스템 날짜와 시간, 이미지 등을 표현하는 컴포넌트 입니다.
라벨에 보여지는 데이터의 형태와 그려지는 방법에 따라 다양한 종류의 라벨을 지원합니다.

나. 라벨 종류

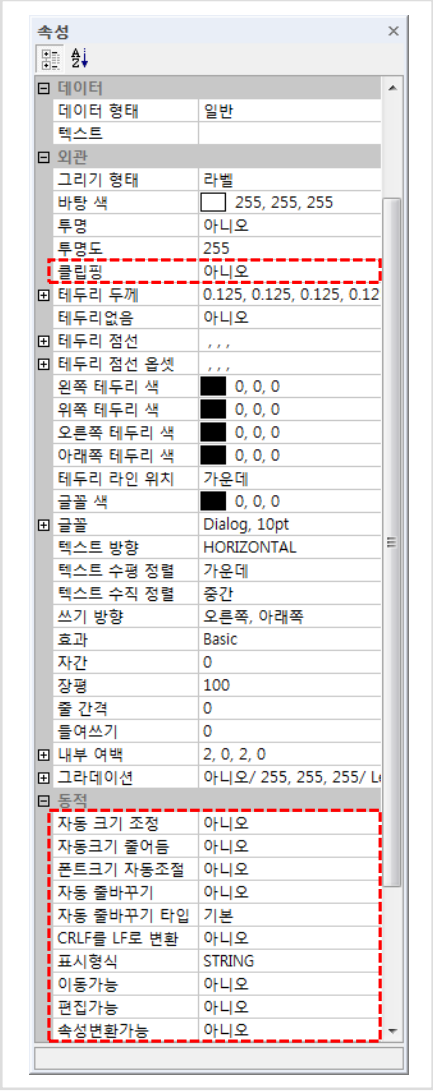
라벨의 '데이터 형태' 와 '그리기 형태' 속성 값을 변경하여 아래와 같이 라벨 종류를 변경할 수 있습니다.

구분	라벨 이름	관련 속성		설명
'데이터 형태' 에 따른 구분	일반 라벨	데이터 형태	일반	- 텍스트 속성 값 표시 - 모든 밴드에 추가 가능
		텍스트	고객 주소록	
	데이터 라벨	데이터 형태	데이터	- ODI 파일에 정의된 데이터 셋의 필드 데이터 표시 - 데이터 밴드(헤더·풋터, 서브, 상위 그룹 헤더·풋터), 데이터 익스팬더 밴드(헤더·풋터)에 추가 가능
		ODI 이름	상품재고현황	
		데이터셋	StockList	
		필드 이름	ProductName	
		NullType	공백	
	그룹 라벨	데이터 형태	그룹	- 동일한 데이터를 그룹핑해 표시 - 데이터 밴드(헤더·풋터, 서브, 상위 그룹 헤더·풋터), 데이터 익스팬더 밴드(헤더·풋터)에 추가 가능
		ODI 이름	상품재고현황	
		데이터셋	StockList	
		필드 이름	CategoryName	
		그룹핑 규칙	병합	
	요약 라벨	데이터 형태	요약	- 필드 데이터의 합계, 평균, 최대, 최소, 개수 표시 - 모든 밴드에 추가 가능
		ODI 이름	상품재고현황	
		데이터셋	StockList	
		필드 이름	UnitsInStock	
		요약 타입	Sum	
	수식 라벨	데이터 형태	수식	- 표현식에 입력한 연산 결과를 표시 - 모든 밴드에 추가 가능
		표현식	Car.Amount.sum/1000	
		수식 ODI 이름	자동차판매현황	
	시스템 라벨	데이터 형태	시스템	- 사용자 PC의 날짜, 시간과 보고서 페이지 번호 표시 - 모든 밴드에 추가 가능
		시스템 필드 이름	Page_Of_Total	
		구분자	/	
	패러미터 라벨	데이터 형태	패러미터	- 쿼리 디자이너의 사용자 지정 패러미터와 리포트 디자이너의 폼 패러미터 값 표시 - 모든 밴드에 추가 가능
		패러미터 셋 이름	자동차판매현황	
		패러미터 이름	FromDate	
'그리기 형태' 에 따른 구분	이미지 라벨	그리기 형태	이미지	- 이미지 표시 - 모든 밴드에 추가 가능
		이미지 스타일	타일	
		텍스트	ozp://ozreport.gif	
	바코드 라벨	그리기 형태	바코드	- CODE 39등 수십 종 1차원 바코드 표시 - PDF417, QR, DataMatrix 바코드 표시 - 모든 밴드에 추가 가능
		바코드 타입	CODE 39	
	HTML 라벨	그리기 형태	HTML	- HTML 태그를 사용하여 텍스트 표시 - 모든 밴드에 추가 가능
		텍스트	<B>예제</B>	
	사용자 컴포넌트	그리기 형태	사용자 컴포넌트	- 외부 사용자 컴포넌트를 연동한 데이터 표시 - 모든 밴드에 추가 가능
		사용자 클래스	tagfree	
	노트 라벨	타입	노트	- 대용량 텍스트 표시 - 테일 밴드, 페이지 풋터 밴드를 제외한 밴드에 추가 가능
		텍스트	대용량텍스트데이터	

라벨의 동적 속성으로 라벨의 크기 조정, 편집 가능 여부, 표시 여부 등을 설정할 수 있습니다.

가. 라벨 크기 관련 속성

아래 속성으로 라벨 컴포넌트에 표시할 데이터 양에 따라 라벨 크기를 자동 조정할 수 있습니다.



속성	설명
자동 크기 조정	표시할 데이터 양이 라벨 컴포넌트의 크기보다 큰 경우 컴포넌트 크기를 자동으로 확대
자동크기 줄어듬	표시할 데이터 양이 라벨 컴포넌트의 크기보다 작은 경우 컴포넌트의 크기를 자동으로 축소. 단, '자동크기 조정' 속성이 '예'로 설정된 경우에만 유효함
폰트크기 자동조절	라벨 컴포넌트 크기에 맞게 폰트 크기를 자동 조정
자동 줄바꾸기	텍스트가 긴 경우 라벨 컴포넌트 너비에 맞게 자동으로 여러 라인으로 잘라서 출력
클립핑	표시할 데이터 양이 라벨 컴포넌트의 크기보다 큰 경우, 컴포넌트 안에 들어가는 내용만 표시

- **속성 적용 우선 순위**

➢ 라벨의 자동 크기 조정 예

- ✓ 자동 크기 조정 : 아니오
- ✓ 자동크기 줄어듬 : 아니오
- ✓ 폰트크기 자동조절 : 아니오
- ✓ 자동 줄바꾸기 : 아니오
- ✓ 클립핑 : 아니오

안녕하세요,
오즈 리포트 디자이너 입니다.

- ✓ **자동 크기 조정 : 예**
- ✓ 자동크기 줄어듬 : 아니오
- ✓ 폰트크기 자동조절 : 아니오
- ✓ 자동 줄바꾸기 : 아니오
- ✓ 클립핑 : 예/아니오

안녕하세요,
오즈 리포트 디자이너 입니다.

※ '자동 크기 조정'은 '폰트크기 자동조절'이 '아니오'인 경우에만 적용됩니다.

※ '자동 줄바꾸기'가 '아니오' 일 경우 '자동 크기 조정' 속성이 너비와 높이에 모두 적용됩니다.

- ✓ **자동 크기 조정 : 예**
- ✓ **자동크기 줄어듬 : 예**
- ✓ 폰트크기 자동조절 : 아니오
- ✓ 자동 줄바꾸기 : 아니오
- ✓ 클립핑 : 아니오

안녕하세요,
오즈 리포트 디자이너 입니다.

※ '자동크기 줄어듬'은 '자동크기조정'이 '예' 이고 '폰트크기 자동조절'이 '아니오'인 경우에만 적용됩니다.

※ '자동 줄바꾸기' 가 '아니오' 일 경우 '자동크기 줄어듬' 속성이 너비와 높이에 모두 적용됩니다.

- ✓ **자동 크기 조정 : 예**
- ✓ 자동크기 줄어듬 : 예/아니오
- ✓ **폰트크기 자동조절 : 예**
- ✓ 자동 줄바꾸기 : 아니오
- ✓ 클립핑 : 아니오

안녕하세요,
오즈 리포트 디자이너 입니다.

※ '폰트크기 자동조절'은 '자동크기 조정'이 '예' 일 경우에만 적용됩니다.

- ✓ **자동 크기 조정 : 예**
- ✓ **자동크기 줄어듬 : 예**
- ✓ 폰트크기 자동조절 : 아니오
- ✓ **자동 줄바꾸기 : 예**
- ✓ 클립핑 : 아니오

안녕하세요,
오즈 리포트 디자이
너 입니다.

※ '자동 줄바꾸기'가 '예' 일 경우 '자동크기 조정'과 '자동크기 줄어듬'은 높이에만 적용됩니다

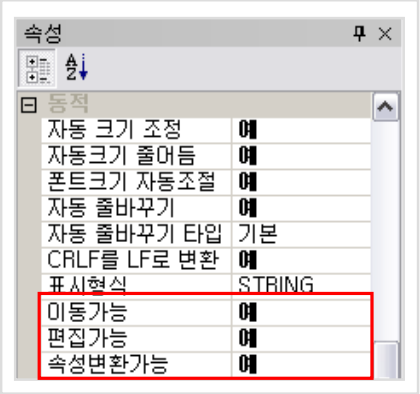
- ✓ 자동 크기 조정 : 아니오
- ✓ 자동크기 줄어듬 : 예/아니오
- ✓ 폰트크기 자동조절 : 예/아니오
- ✓ 자동 줄바꾸기 : 아니오
- ✓ **클립핑 : 예**

안녕하세요,
오즈 리포트 디자이너

※ '클립핑'은 '자동크기 조정'이 '아니오' 일 경우에만 적용됩니다.

나. 라벨 편집 관련 속성

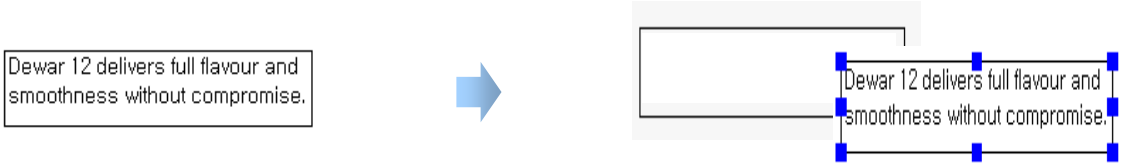
오즈 뷰어로 보고서가 출력된 이후 사용자가 직접 라벨 컴포넌트의 텍스트와 속성을 편집할 수 있습니다.
편집된 내용은 인쇄하거나 다른 포맷의 파일로 저장할 때도 반영됩니다.



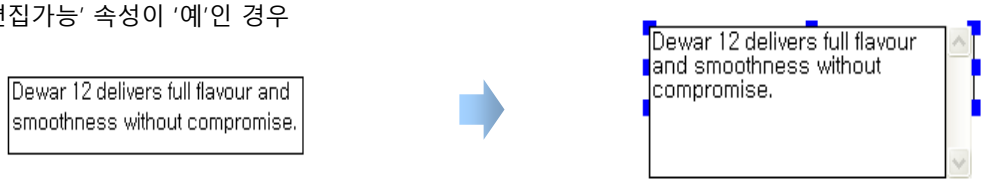
속성	설명
이동가능	오즈 뷰어에서 라벨 컴포넌트의 이동 허용 여부
편집가능	오즈 뷰어에서 라벨 컴포넌트의 텍스트 수정 허용 여부
속성변환가능	오즈 뷰어에서 라벨 컴포넌트의 속성 변경 허용 여부

➢ 라벨의 편집 속성 적용 예

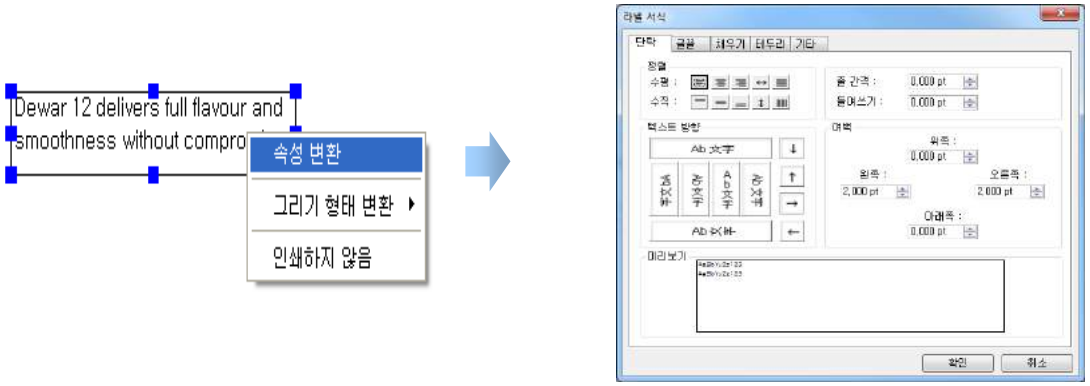
- '이동가능' 속성이 '예'인 경우



- '편집가능' 속성이 '예'인 경우

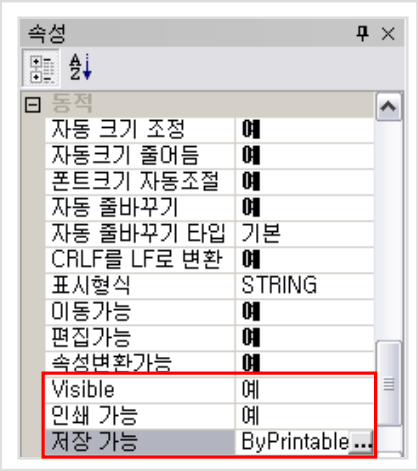


- '속성변환가능' 속성이 '예'인 경우



다. 라벨 표시 관련 속성

보고서를 미리보기, 인쇄, 다른 파일로 변환할 때 라벨 컴포넌트의 표시 여부를 설정하는 속성입니다.



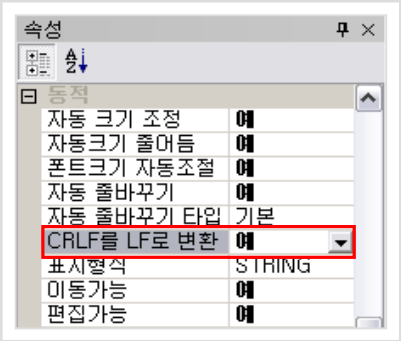
속성	설명
Visible	라벨 컴포넌트를 오즈 뷰어 화면에 표시 여부 설정 'Visible' 속성이 '아니오'로 설정된 경우에도 '인쇄 가능' 속성이 '예'인 경우 프린터로는 인쇄됨
인쇄 가능	라벨 컴포넌트를 프린터로 인쇄 여부 설정
저장 가능	보고서를 다른 포맷의 파일로 저장할 때 라벨 컴포넌트를 포함하여 저장 여부 설정

- **라벨의 '저장 가능' 속성 값**

속성 값	설명
All	모든 포맷으로 저장
Nothing	모든 포맷으로 저장 불가
ByPrintable	인쇄가능 속성을 따름
PDF,XLS,PPT ...	체크한 포맷으로 저장

라. 'CRLF를 LF로 변환' 속성

운영 환경이나 프로그래밍 언어의 차이로 인해 발생하는 줄바꿈 문자의 인식 문제를 해결하기 위해 'CRLF를 LF로 변환' 속성이 제공됩니다.

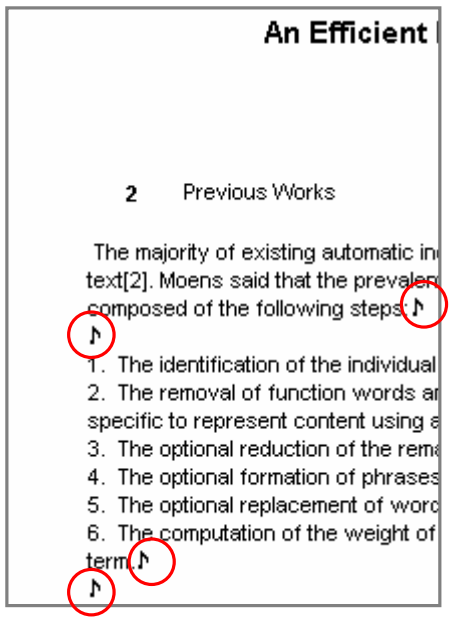


속성	설명
CRLF를 LF로 변환	라벨 컴포넌트의 데이터에서 Windows의 줄바꿈 문자인 'CRLF'를 UNIX와 JAVA의 줄바꿈 문자인 'LF'로 변환

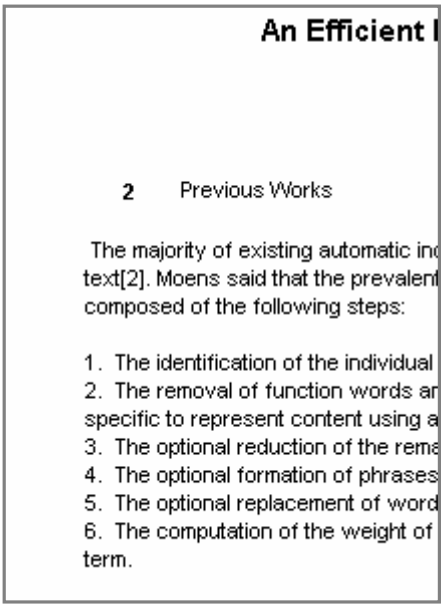
- **CRLF** : Windows의 줄 바꾸기 문자
- **LF** : UNIX, JAVA의 줄 바꾸기 문자

➢ **라벨의 'CRLF를 LF로 변환' 속성 적용 예**

데이터의 줄 바꿈 부분에 ♪ 표시가 보일 때 'CRLF를 LF로 변환' 속성 값을 '예' 로 변경하여 ♪ 표시를 없앨 수 있습니다.

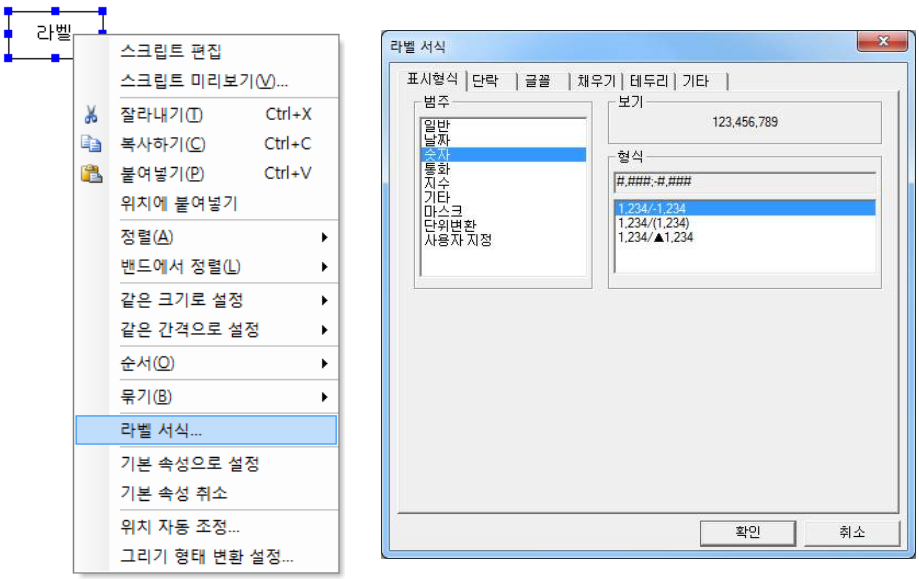


- 'CRLF를 LF로 변환' = '아니오'



- 'CRLF를 LF로 변환' = '예'

'라벨 서식' 창에서 표시형식, 단락, 글꼴, 테두리 등의 라벨 속성을 변경할 수 있습니다.



• 라벨 컴포넌트의 팝업 메뉴에서 '라벨 서식'을 선택하면 '라벨 서식' 창이 나옵니다.

탭 구분	설명
표시형식	날짜, 숫자 등의 표시 형식 설정
단락	텍스트 정렬, 텍스트 방향, 줄 간격, 들여쓰기, 여백 설정
글꼴	폰트, 글자 크기, 글자색, 자간, 장평, 텍스트 효과 등 설정
채우기	배경색, 투명도, 그라데이션 설정
테두리	테두리 선에 대한 두께, 스타일, 색상 설정

➢ 라벨의 '숫자' 표시형식 적용 예

- 숫자 자리수 표시 : '#'
- 자리수 구분 : 콤마 ','
- 음수·양수 구분 : 세미콜론 ';'
- 빈 자리 0으로 대체 : '0'

[데이터]		[표시형식]		[결과]
12345678 -12345678.9	+	#,###;-#,### #,###.00;-#,###.00	=	12,345,678 -12,345,678.90

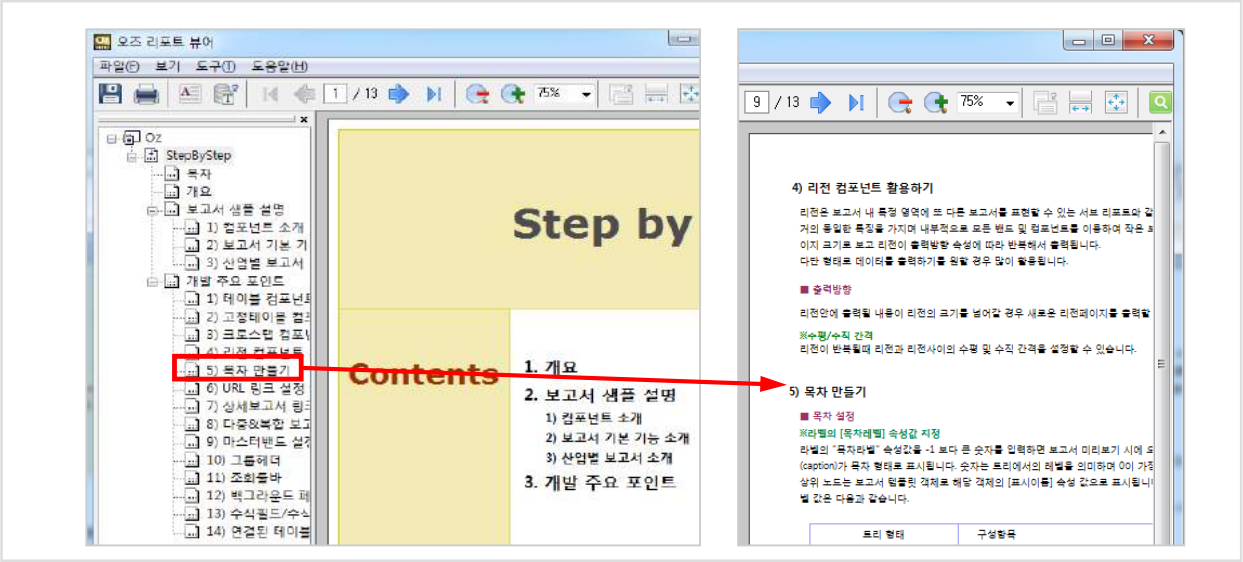
➢ 라벨의 '마스크' 표시형식 적용 예

- 마스크 패턴 형식 : Mask_[PromptChar]_[User define mask]

[데이터]		[표시형식]		[결과]
01122221234	+	mask_#_(###)####-####	=	(011)2222-1234

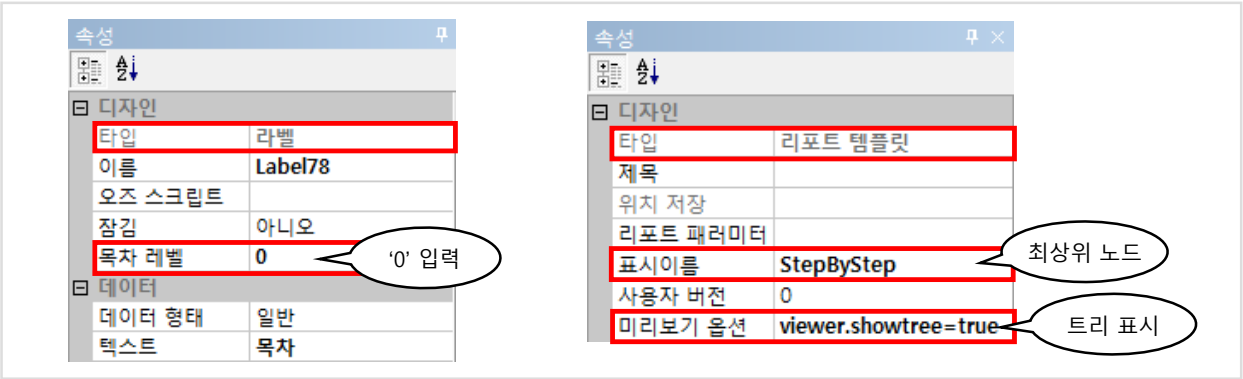
가. 목차 보고서

오즈 뷰어의 트리 창에 목차를 표시할 수 있습니다. 트리의 항목을 클릭하면 목차를 지정한 라벨이 있는 페이지로 이동됩니다.



나. 목차 구현

라벨의 '목차 레벨' 속성 값에 -1 보다 큰 숫자를 입력하면, 리포트 뷰어의 트리에 해당 라벨의 텍스트가 목차 형태로 표시됩니다. 숫자는 트리에서의 레벨을 의미하며, 0은 최상위 노드 바로 아래 레벨을 나타냅니다.
최상위 노드는 '리포트 템플릿' 의 '표시이름' 속성 값으로 표시됩니다.
미리보기할 때 뷰어에 트리를 보여주려면 '리포트 템플릿'의 '미리보기 옵션'에 'viewer.showtree=true'를 입력합니다.



위의 그림에서 목차 트리와 각 라벨의 '목차 레벨' 값은 다음과 같습니다.

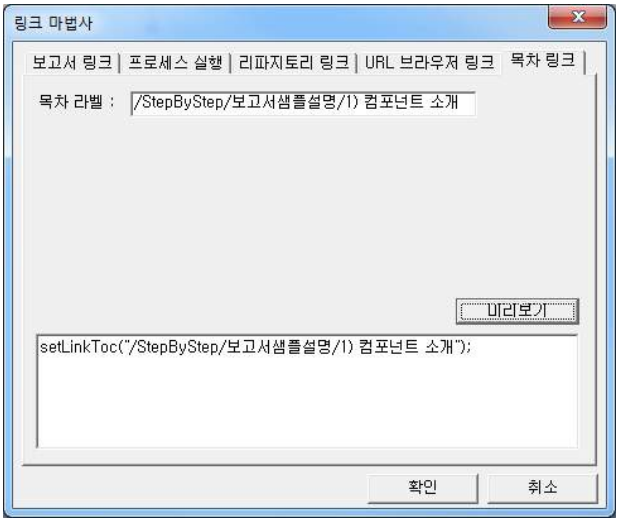
트리	컴포넌트 타입		속성 값
- StepbyStep	리포트 템플릿	-------	표시이름 : StepbyStep
ㄴ 목차	라벨	-------	목차레벨 : 0
ㄴ 개요	라벨	-------	목차레벨 : 0
ㄴ 보고서 샘플 설명	라벨	-------	목차레벨 : 0
ㄴ 1) 컴포넌트 소개	라벨	-------	목차레벨 : 1
ㄴ 2) 보고서 기본기능 소개	라벨	-------	목차레벨 : 1
ㄴ 3) 산업별 보고서 소개	라벨	-------	목차레벨 : 1

다. 링크 구현

아래의 setLinkToc 함수를 스크립트에 작성한 라벨을 클릭하면 '목차 레벨'을 설정한 라벨로 링크 됩니다.
'TocPath' 패러미터로 '목차 레벨'을 '0' 이상 값으로 설정한 라벨의 목차 단계를 지정합니다.
링크될 때 해당 목차 항목으로 페이지가 이동됩니다.

➢ **오즈 스크립트 함수**

void setLinkToc (string tocPath)



- '오즈 스크립트 편집' 창의 팝업 메뉴에서 '링크 마법사'를 선택합니다. '목차 링크' 탭에서 '목차 라벨' 칸에 목차 단계를 입력하면 함수가 생성됩니다.

➢ **자바 스크립트 함수**

void SetLinkToc (string TocPath)

- 목차 단계(TocPath) : /표시이름/상위 레벨 목차 이름/목차 이름/

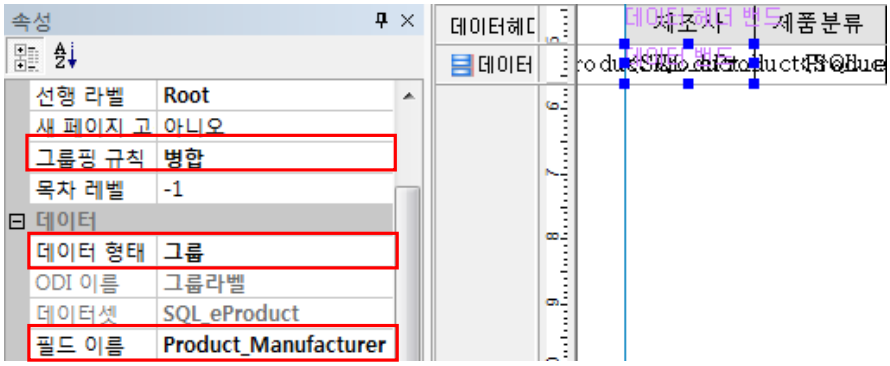
앞 페이지 그림에서 'Contents' 페이지에 있는 라벨의 오즈 스크립트는 다음과 같이 작성됩니다.
Contents의 각 라벨을 클릭하면 TocPath에 지정된 목차가 설정된 라벨이 있는 페이지로 이동합니다.

```
setLinkToc("/StepByStep/개요");
setLinkToc("/StepByStep/보고서샘플설명");
setLinkToc("/StepByStep/보고서샘플설명/1) 컴포넌트 소개");
setLinkToc("/StepByStep/보고서샘플설명/2) 보고서 기본기능 소개");
setLinkToc("/StepByStep/보고서샘플설명/3) 산업별 보고서 소개");
setLinkToc("/StepByStep/개발 주요 포인트");
```

가. 그룹 라벨 정의

데이터 셋의 필드 값을 레코드 수만큼 라벨에 보여줄 때 특정 필드 열의 같은 값들을 그룹핑하여 한 라벨로 표현하려면 그룹 라벨을 사용합니다.

'라벨'이나 '고정테이블 라벨' 컴포넌트만 그룹 라벨로 설정할 수 있습니다.

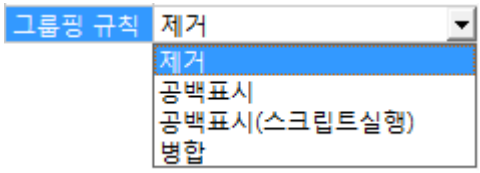


① 라벨의 '데이터 형태' 속성을 '그룹'으로 설정하면 그룹 라벨이 됩니다.

② 라벨의 '그룹핑 규칙' 속성을 선택합니다.

③ 라벨의 '필드 이름'을 선택합니다.

나. 그룹핑 규칙



라벨의 '그룹핑 규칙' 속성을 이용하여 라벨의 표시 형태를 설정할 수 있습니다.

1) 제거

제조사	제품분류
SAMSUNG	TV
	VTR
	AUDIO
LG	CAMERA
	TV
	AUDIO
SONY	TV
	AUDIO
	VTR

2) 병합

제조사	제품분류
SAMSUNG	TV
	VTR
	AUDIO
LG	CAMERA
	TV
	AUDIO
SONY	TV
	AUDIO
	VTR

3) 공백표시

제조사	제품분류
SAMSUNG	TV
	VTR
	AUDIO
LG	CAMERA
	TV
	AUDIO
SONY	TV
	AUDIO
	VTR

4) 공백표시(스크립트실행)

제조사	제품분류
SAMSUNG	TV
"	VTR
"	AUDIO
LG	CAMERA
"	TV
"	AUDIO
SONY	TV
"	AUDIO
"	VTR

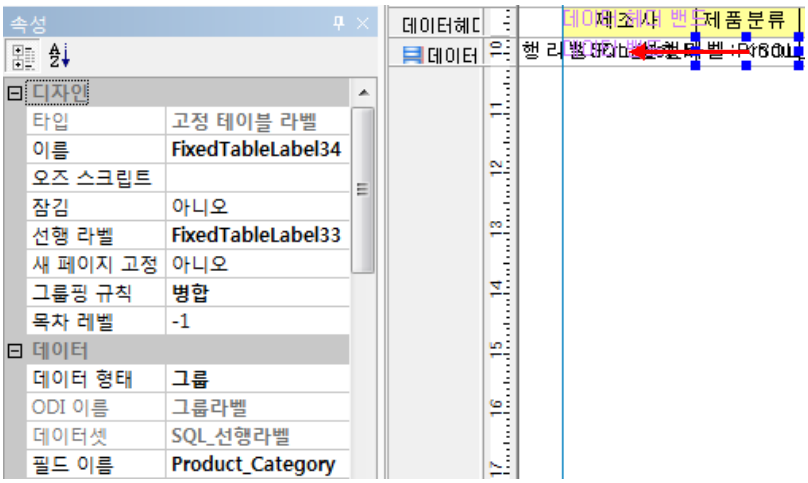
- '그룹핑 규칙'이 '병합'일 때에는 반복되는 라벨 사이의 공백이나 다른 라벨이 없어야 합니다.
- '그룹핑 규칙'이 '제거', '병합', '공백표시'일 때에는 값이 중복된 라벨의 스크립트는 반복 실행되지 않습니다.

➢ **'공백표시(스크립트실행)'로 설정된 그룹 라벨의 오즈 스크립트 사용 예**

```
if (getattr("caption") == "") {
    setattr("caption", " 〃 ");
}
```

다. 선행 라벨

그룹 라벨은 같은 밴드 내의 다른 그룹 라벨을 선행 라벨로 지정하여 선행 라벨에 따라 그룹핑이 되도록 할 수 있습니다.



① '제조사', '제품분류' 데이터 필드를 지정한 라벨을 각각 그룹 라벨로 설정합니다.

② '제품분류' 데이터가 '제조사' 별로 그룹핑이 되게 하려면 '제품분류' 라벨의 '선행 라벨' 속성 값을 '제조사' 라벨로 변경합니다.

③ 선행 라벨 방향으로 빨간색 화살표가 생깁니다.

- **선행라벨 지정하기 전 결과**

제조사	제품분류	판매월	판매수량
SAMSUNG	TV	2010.01	110
		2010.02	89
LG		2010.01	130
		2010.02	100

- **선행라벨 지정 후 결과**

제조사	제품분류	판매월	판매수량
SAMSUNG	TV	2010.01	110
		2010.02	89
LG	TV	2010.01	130
		2010.02	100

TIP

테이블 마법사로 만들어진 '테이블 값' 타입인 라벨은 그룹 라벨과 같은 기능으로 '그룹핑' 속성을 사용합니다. '테이블 값' 타입 라벨은 같은 테이블의 다른 '그룹핑' 속성이 설정된 라벨을 선행 라벨로 지정할 수 있습니다.

라. 새 페이지 고정

- **새 페이지 고정 : 아니오**

제조사	제품분류
SAMSUNG	TV
	VTR
	AUDIO
LG	CAMERA
	TV

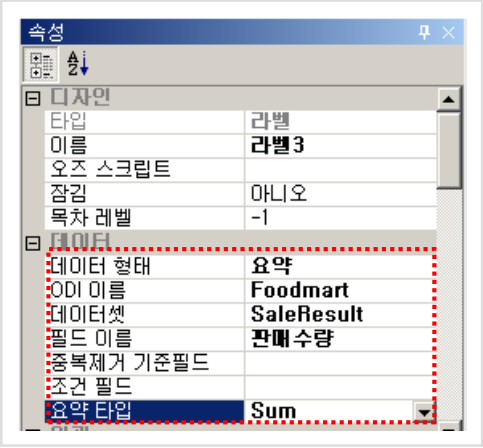
- **새 페이지 고정 : 예**

제조사	제품분류
SAMSUNG	TV
	VTR
SAMSUNG	AUDIO
LG	CAMERA
	TV

그룹 라벨의 '새 페이지 고정' 속성을 이용하여 그룹 라벨의 텍스트가 페이지가 바뀔 때 다시 보여지도록 설정할 수 있습니다.

가. 요약 라벨 정의

데이터 필드 값에 대한 합계, 평균, 최대, 최소, 개수 값을 보여주고자 할 때 요약 라벨을 사용합니다.
라벨의 '데이터 형태' 속성을 '요약'으로 설정하면 요약 라벨이 됩니다.

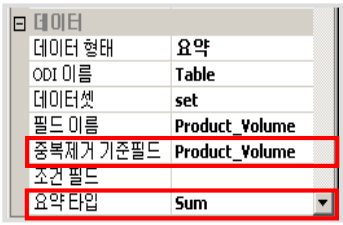


요약 타입	설명
None	데이터 셋의 마지막 행의 값 표시
Sum	데이터 필드에 대한 합계 값 표시
Avg	데이터 필드에 대한 평균 값 표시
Max	데이터 필드에 대한 최대 값 표시
Min	데이터 필드에 대한 최소 값 표시
Cnt	데이터 필드에 대한 개수 표시

나. 요약 라벨의 활용

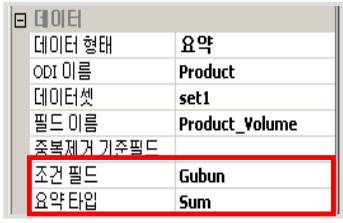
SONY	TV	SY-246-GS	5
	AUDIO	SY-235-KQ	3
	VTR	SY-233-DE	5
합계			13

• 요약 타입을 'Sum'으로 설정하면 전체 데이터의 합계가 표시됩니다.



SONY	TV	SY-246-GS	5
	AUDIO	SY-235-KQ	3
	VTR	SY-233-DE	5
중복 제거 합계			8

• 중복되는 값은 제외하고 계산할 경우 '중복제거 기준필드' 속성에 중복되는 값을 가진 필드를 설정합니다.



SONY	TV	true	5
	AUDIO	false	3
	VTR	true	5
조건 합계			10

• 특정 필드의 값이 'true'인 경우에만 계산할 경우 '조건 필드' 속성에 'true' 또는 'false' 값을 가져오는 필드를 설정합니다.

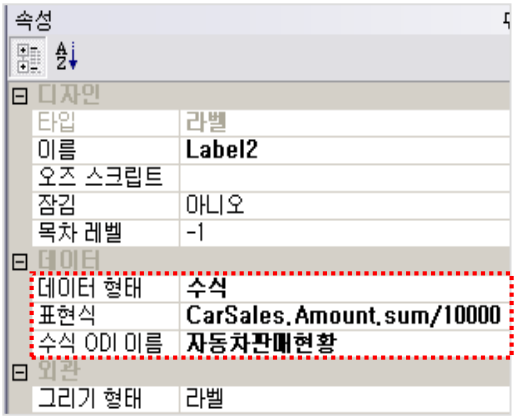
※ '데이터 정보' 창에서 데이터 필드를 데이터 셋이 다른 데이터 밴드로 드래그&드롭 할 경우 데이터 필드가 '데이터 라벨'로 추가되지 않고 '요약 라벨'로 추가됩니다. 이 때, '요약 타입'은 'None'으로 추가되나 데이터 셋의 첫 번째 행의 값이 표시됩니다.

가. 수식 라벨 정의

수식 라벨은 표현식에 입력한 연산 결과를 보여주는 라벨입니다.
라벨의 '데이터 형태' 속성을 '수식'으로 설정하면 수식 라벨이 됩니다.

- 표현식에는 텍스트, 데이터 필드, 패러미터, 함수, 사칙연산을 이용할 수 있습니다.

  예) MyDataset.VolumeField.**sum** /10000



표현식 구분	사용 방법
텍스트	문자가 그대로 라벨에 보여짐
데이터 필드	DataSet.FieldName
ODI 패러미터	OZParam.ParameterName
폼 패러미터	OZFormParam.ParameterName
합계 함수	DataSet.FieldName.sum
평균 함수	DataSet.FieldName.avg
최대값 함수	DataSet.FieldName.max
최소값 함수	DataSet.FieldName.min
사칙연산	+, -, *, /

나. 집계함수 스크립트 사용

요약 라벨, 수식 라벨로 구할 수 있는 데이터 필드의 합계, 평균 등의 값을 구하는 집계 함수는 스크립트에서 다음과 같이 지원됩니다.

오즈 스크립트 집계함수	자바 스크립트 집계함수	설명
dbsum	GetDataSetSUM	데이터 필드의 합계 반환
dbavg	GetDataSetAVG	데이터 필드의 평균값 반환
dbmax	GetDataSetMAX	데이터 필드의 최대값 반환
dbmin	GetDataSetMIN	데이터 필드의 최소값 반환
dbfreq	GetDataSetFREQ	데이터 필드의 특정 값의 개수 반환
totalRowCount	GetDataSetCNT	데이터 필드의 레코드 개수 반환

➢ **오즈 스크립트 사용 예**

```
double sum;
sum = dbsum("MyODI.TestDataSet.TestField"); // 패러미터 : ODI이름.데이터셋이름.필드이름
setattr("caption",(string)sum);
```

➢ **자바 스크립트 사용 예**

```
var sum = This.GetDataSetSUM("MyODI.TestDataSet.TestField");
This.SetText(sum);
```

가. 시스템 라벨 정의

날짜, 시간, 페이지 번호, 전체 페이지 수를 보고서에 표시할 수 있도록 시스템 필드가 제공됩니다.
시스템 라벨은 이 시스템 필드 값을 보여주는 컴포넌트이며, 모든 밴드에서 사용 가능합니다.

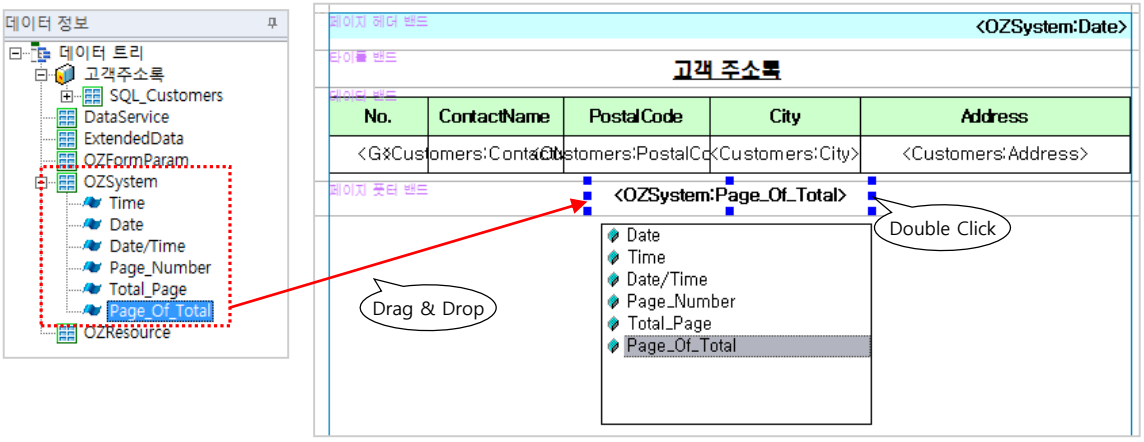
종류	설명	예
Date	보고서가 작성된 날짜	10시43분22초 오전
Time	보고서가 작성된 시간	2011.02.24 (날짜 서식 적용)
Date/Time	보고서가 작성된 날짜와 시간	2011년02월24일 목 10시43분22초 오전
Page_Number	현재 페이지 번호	1
Total_Page	전체 페이지 수	3
Page_Of_Total	현재 페이지 번호와 전체 페이지 수 표시	1/3

※ 날짜, 시간은 보고서를 실행한 로컬 PC 기준입니다.

나. 시스템 라벨 설정

① '데이터 정보' 창의 'OZSystem'에 있는 시스템 필드를 선택한 후 밴드 위로 끌어다 놓으면 시스템 라벨이 생성됩니다.

② 시스템 라벨을 더블 클릭하면 다른 시스템 필드로 변경 가능하도록 목록이 보여집니다.



다. 스크립트 사용

➢ **자바 스크립트로 페이지 수 표시 예**

일반 라벨의 자바 스크립트에 다음과 같이 작성하여 페이지의 보여지는 형태를 변경할 수 있습니다.

```
This.SetText( This.GetDataSetValue("OZSystem.Page_Number") + " 페이지"); // 결과 : n 페이지
```

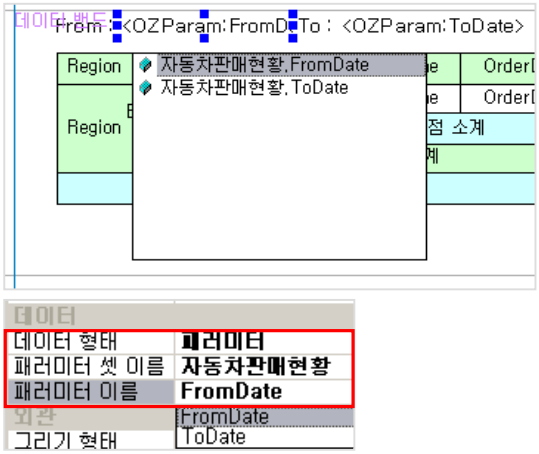
가. 패러미터 라벨 정의

오즈 리포트에서 사용하는 패러미터는 쿼리 디자이너에서 만드는 '사용자 지정 패러미터'와 리포트 디자이너에서 만드는 '폼 패러미터' 두 가지가 있습니다.
패러미터 라벨은 이 두 가지 패러미터의 값을 보여주는 컴포넌트입니다.

나. 사용자 지정 패러미터 사용

1) 패러미터 라벨 설정 방법

쿼리 디자이너에서 만든 사용자 지정 패러미터를 다음과 같이 패러미터 라벨을 사용하여 보고서에 표시합니다.



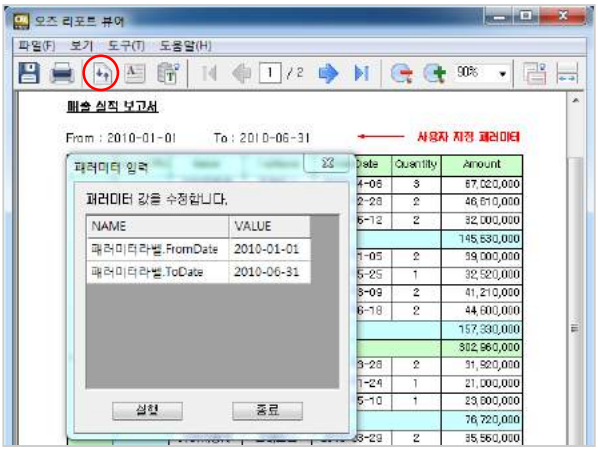
① 일반 라벨을 추가하여 '데이터 형태' 속성을 '패러미터'로 설정합니다.

② 라벨을 더블 클릭하여 패러미터 이름을 선택합니다.

③ 라벨의 '패러미터 셋 이름'속성에 ODI 이름과 '패러미터 이름'이 설정됩니다.

2) 뷰어에서 사용자 지정 패러미터 변경 방법

보고서가 뷰어에 보여진 후에 사용자가 직접 패러미터 값을 변경하여 데이터를 다시 조회할 수 있습니다.



① 뷰어 툴바의 '패러미터 입력' 아이콘 클릭

② 'VALUE' 변경

③ '실행' 버튼 클릭하면 새로운 조건의 데이터로 리포트가 다시 생성됨

※ '패러미터 입력' 아이콘이 뷰어 툴바에 안보이면 '리포트 템플릿'의 '미리보기 옵션' 속성에 'toolbar.inputparameter=true'를 입력합니다.

3) 운영시 사용자 지정 패러미터 값 설정 방법

➢ **HTML에서 ODI 패러미터 전달 예**

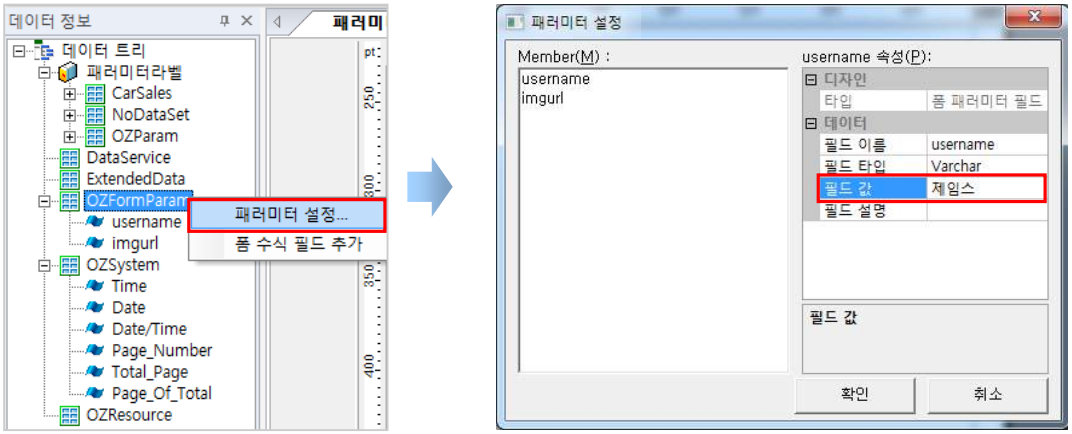
```
<param name="odi.odinames" value="ODI명"> <!-- ODI 개수가 여러 개일 때 , 로 구분 -->
<param name="odi.ODI명.pcount" value="2"> <!-- 패러미터 개수 지정 -->
<param name="odi.ODI명.args1" value="fromdate=20071030"> <!-- 첫번째 패러미터 설정 -->
<param name="odi.ODI명.args2" value="todate=20071201"> <!-- 두번째 패러미터 설정 -->
```

다. 폼 패러미터 사용

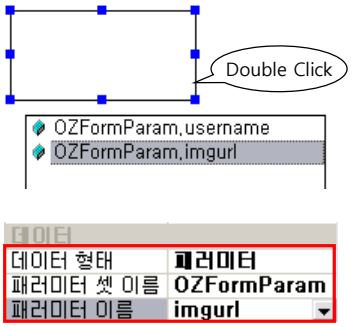
'사용자 지정 패러미터'는 데이터 조회 조건에 이용되는데 비해 '폼 패러미터'는 리포트 디자이너에서 폼을 그릴 때 필요한 정보로만 사용됩니다. 가령 이미지 파일 경로가 URL로 설정되는 경우 도메인 정보가 가변적일 때, 폼 패러미터로 도메인 정보를 넘길 수 있습니다.

1) 폼 패러미터 생성

① '데이터 정보' 창의 'OZFormParam'의 팝업 메뉴에서 '패러미터 설정' 클릭
② '패러미터 설정' 창의 Member 항목에 패러미터 이름 입력 (엔터 키로 여러 개 입력)
③ '필드 값' 속성에 디자인시에 사용할 값 설정



2) 패러미터 라벨 설정 방법



① 일반 라벨을 추가하여 '데이터 형태' 속성을 '패러미터'로 설정

② 라벨을 더블 클릭하여 폼 패러미터 이름 선택

③ 라벨의 '패러미터 셋 이름'속성에 'OZFormParam'과 선택한 '패러미터 이름'이 설정됨

3) 운영시 폼 패러미터 값 설정 방법

폼 패러미터 만들 때 설정한 '필드 값'은 디자이너에서만 사용되는 값입니다.
실제 운영 환경에서는 뷰어를 호출할 때 HTML 태그 값으로 폼 패러미터 값이 전달됩니다.

➢ **HTML에서 폼 패러미터 전달 예**

```
<param name="connection.pcount" value="2"> <!-- 폼 패러미터 개수 지정 -->
<param name="connection.args1" value="username=홍길동">
<param name="connection.args2" value="imgurl=http://127.0.0.1/images/logo.gif">
```

가. 이미지 라벨 정의

이미지 라벨은 이미지를 보고서에 표시하는 컴포넌트입니다.
라벨의 '그리기 형태' 속성을 '이미지'로 설정하면 이미지를 보여줄 수 있습니다.

나. 이미지 경로

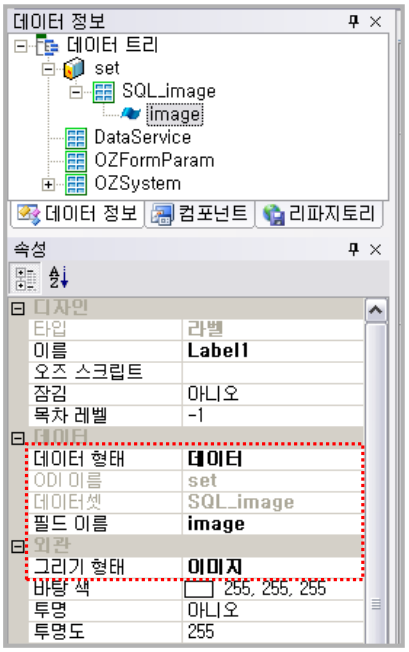
라벨의 '데이터 형태' 속성 값에 따라 이미지 소스는 달라집니다.

데이터 형태	이미지 정보 설정 방법	이미지 정보의 형태
데이터	'필드 이름' 속성에 데이터 셋의 필드명 지정	BLOB 데이터, 이미지 파일 경로
일반	'텍스트' 속성에 이미지 경로 지정	이미지 파일 경로
패러미터	패러미터 값에 이미지 경로 지정	이미지 파일 경로

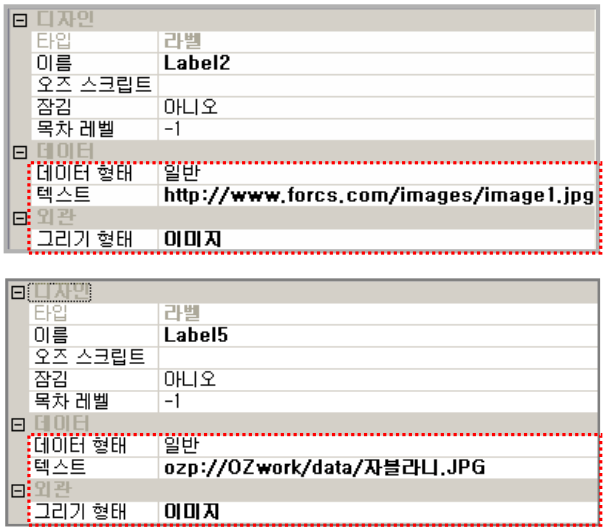
■ **이미지 파일 경로**

- URL : http://ip:port/경로/이미지파일명
- 오즈 리파지토리 경로 : ozp://카테고리명/이미지파일명

• 라벨의 '데이터 형태' 속성이 '데이터'인 경우



• 라벨의 '데이터 형태' 속성이 '일반'인 경우



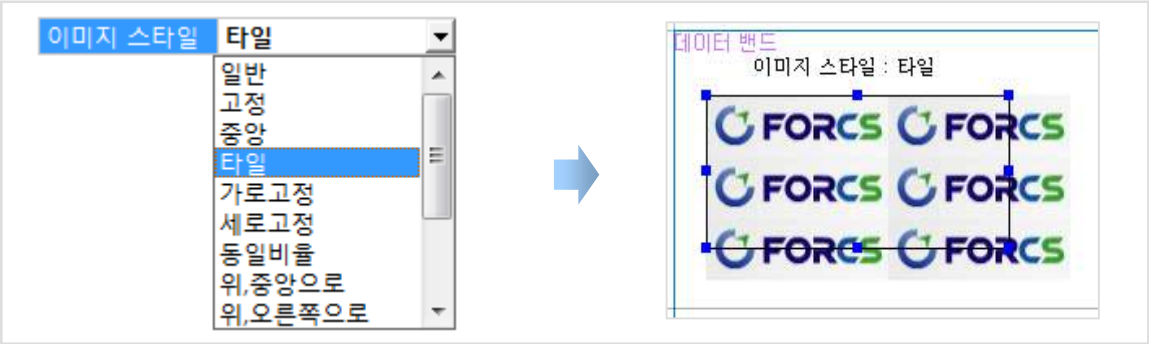
➢ **자바 스크립트 사용 예**

폼 패러미터 'imgurl'의 값이 'http://127.0.0.1/images/'이고 데이터 필드 값이 'test.jpg'라고 가정할 때, 스크립트로 이미지 라벨의 이미지 경로를 아래와 같이 설정할 수 있습니다.

```
var formparam_imgurl = This.GetDataSetValue("OZFormParam.imgurl");
var datafield_imgname = This.GetDataSetValue("ODI명.Dataset명.Field명");
This.SetText( formparam_imgurl + datafield_imgname );
```

다. 이미지 스타일

이미지 라벨의 '이미지 스타일' 속성으로 라벨에 대한 이미지의 위치와 크기, 표시 방식을 설정할 수 있습니다.



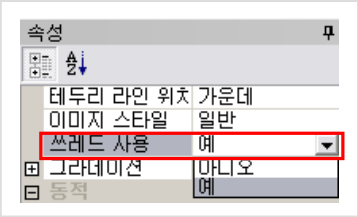
이미지 스타일	설명
일반	컴포넌트의 왼쪽 상단에서 시작하여 원래의 이미지 크기대로 표시
중앙	컴포넌트의 중앙에 원래의 이미지 크기대로 표시
고정	컴포넌트 크기에 맞추어 이미지 크기를 확대 또는 축소하여 표시
타일	이미지를 타일을 배열하듯 연속해서 표시
가로고정	이미지를 컴포넌트의 너비에 맞춤 높이는 너비가 축소/확대된 만큼 축소/확대 됨
세로고정	이미지를 컴포넌트의 높이에 맞춤 너비는 높이가 축소/확대된 만큼 축소/확대 됨
동일비율	이미지의 가로와 컴포넌트의 가로 비율, 이미지의 세로와 컴포넌트의 세로 비율 중 작은 값을 기준으로 축소/확대 됨
위, 중앙으로	이미지를 위 중앙에 맞춰 표시
위, 오른쪽으로	이미지를 위 오른쪽에 맞춰 표시
중간, 왼쪽으로	이미지를 중간 왼쪽에 맞춰 표시
중간, 오른쪽으로	이미지를 중간 오른쪽에 맞춰 표시
아래, 왼쪽으로	이미지를 아래 왼쪽에 맞춰 표시
아래, 중앙으로	이미지를 아래 중앙에 맞춰 표시
아래, 오른쪽으로	이미지를 아래 오른쪽에 맞춰 표시

라. 쓰레드 사용

이미지 파일을 별도의 쓰레드를 이용해서 가져올지 여부를 라벨의 '쓰레드 사용' 속성으로 설정할 수 있습니다.

쓰레드를 사용하면 이미지 파일의 로딩이 끝나지 않더라도 보고서를 먼저 띄울 수 있습니다.

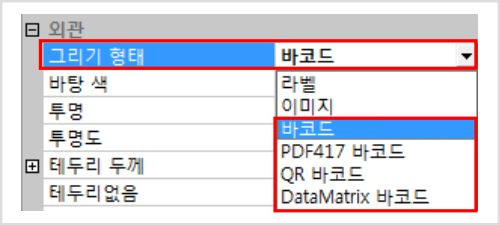
쓰레드를 사용하지 않을 때는 이미지 파일을 로딩할 때까지 뷰어에 보고서가 보여지지 않습니다.



가. 바코드 라벨 정의

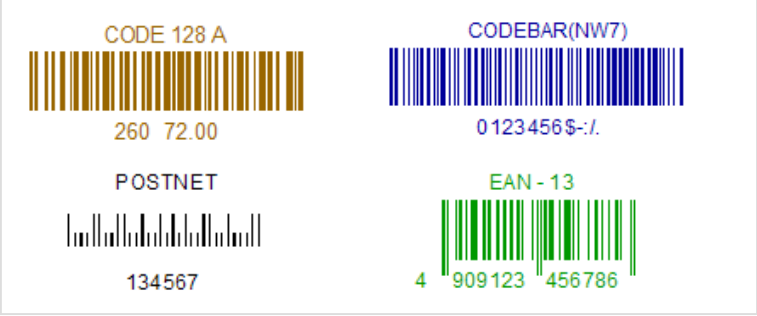
바코드 라벨은 1차원, 2차원 바코드를 보고서에 표시하는 컴포넌트 입니다.

라벨 컴포넌트를 추가한 후 '그리기 형태' 속성을 변경하여 사용합니다.

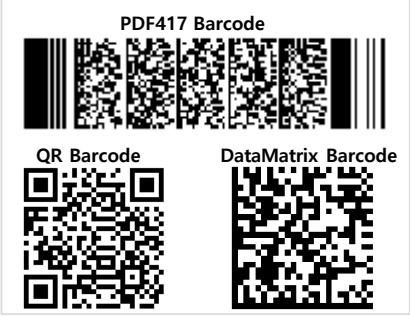


나. 바코드 종류

1차원 바코드는 '그리기 형태'를 '바코드'로 지정한 후 '바코드 타입' 속성에서 수 십 가지 바코드 종류 중 선택하여 그립니다. 2차원 바코드는 '그리기 형태'를 'PDF417바코드', 'QR 바코드', 'DataMatrix 바코드' 중에서 지정합니다.

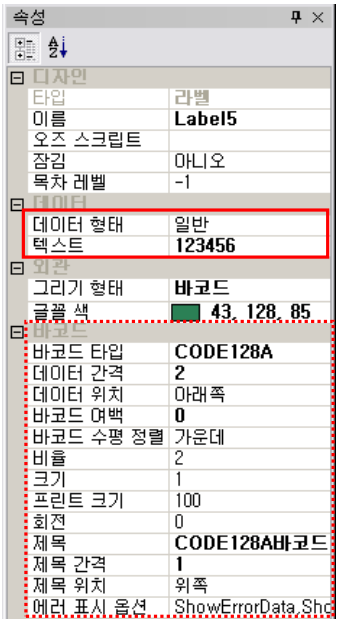


[1차원 바코드]



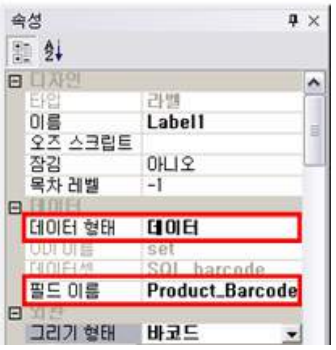
[2차원 바코드]

다. 바코드 데이터



- '데이터 형태' 속성이 '일반' 일 때, '텍스트' 속성 값으로 바코드 데이터 출력됨

- '데이터 형태' 속성이 '데이터' 일 때, '필드 이름'에 지정한 필드 데이터 값으로 바코드 데이터 출력됨

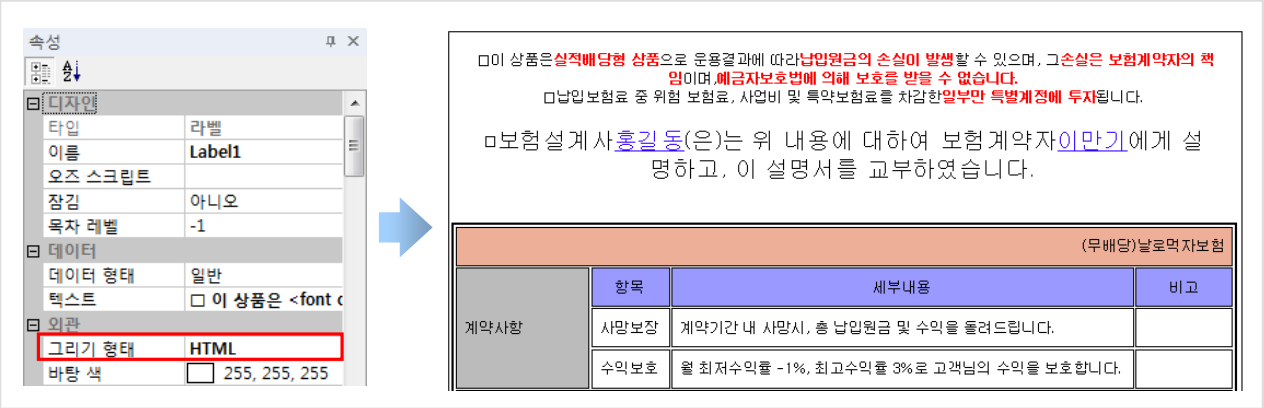


TIP

바코드 전문 업체의 2차원 바코드를 구입한 경우 '사용자 컴포넌트'로 연동하여 사용할 수 있습니다.
사용자 컴포넌트는 라벨의 '그리기 형태' 속성을 '사용자 컴포넌트'로 선택하고 '사용자 클래스' 속성에 뷰어에 추가되어 있는 연동 DLL이름을 선택하여 사용합니다. 관련 DLL은 사용자 PC에 배포되어야 합니다.

가. HTML 라벨 정의

HTML 라벨은 텍스트에 HTML 태그를 사용하여 다양한 글꼴 및 서식을 표현할 수 있는 컴포넌트 입니다.
라벨 컴포넌트를 추가한 후 '그리기 형태' 속성을 'HTML'으로 변경하여 사용합니다.



나. 지원되는 주요 HTML 태그

태그	설명	태그	설명
<FONT>	글꼴 설정 - face : 글꼴 이름, size : 글꼴 크기, color : 글꼴 색상 속성 사용 ※ 생략하면 라벨 컴포넌트의 기본 속성을 따릅니다.	<DIV>	문단 단위로 영역 지정 - align 속성 값에 left/center/right/justify 사용
<B>, <STRONG>	굵은 글꼴 설정	<SPAN>	문장 단위로 영역 지정
<I>	기울임 글꼴 설정	<UL>	순서 없는 목록 표시 (각 항목에 기호 표시)
<U>	텍스트에 밑줄 표시	<OL>	숫자 목록 표시 (각 항목에 숫자 표시)
<S>, <STRIKE>	텍스트에 가운데 줄 표시	<LI>	<UL>과 <OL> 사용 시 목록의 항목 표시
 	한 줄 띄우기	<DL>	정의 목록 표시
<TABLE>, <TR>, <TD>, <TH>	테이블 관련 설정	<DT>	<DL> 사용 시 정의 용어 항목을 표시
<SUB>, <SUP>	아래 첨자, 윗 첨자 표시	<DD>	<DT> 사용 시 정의 설명 항목을 표시
<IMG>	이미지 설정 - src, width, height, style 속성 사용 - style 속성에 width, height 사용	모든 태그의 style 속성	- style 속성으로 font-size, line-height, color 사용 - font-size와 line-height 단위는 pt, px 사용 <FONT style="font-size:10pt">가</FONT>

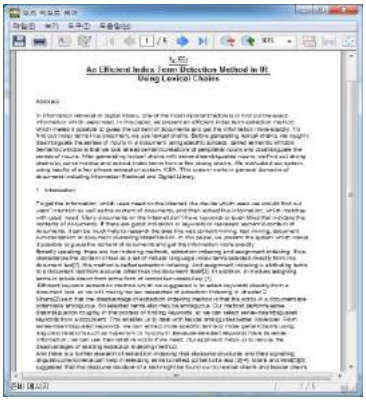
- **HTML 예약어**

코드	표현 문자	코드	표현 문자
	공백 문자	&qt;	<
&	&	<	>

가. 노트 라벨 정의

노트 라벨은 논문, 매뉴얼 등과 같은 대용량 텍스트 데이터를 보고서에 표시할 수 있는 컴포넌트 입니다.

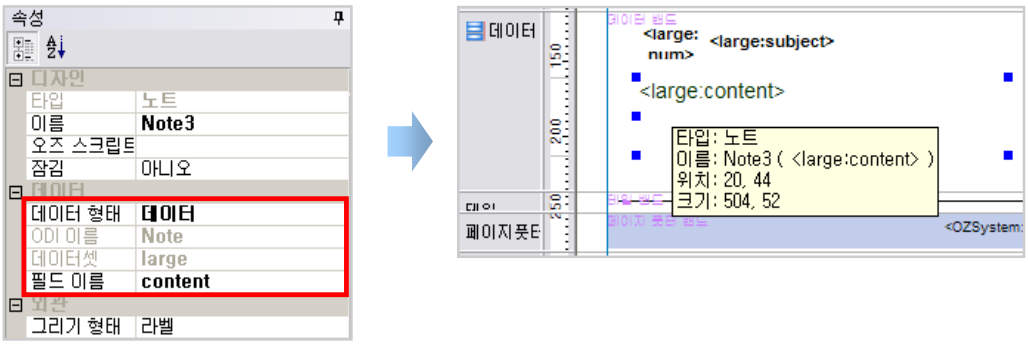
툴바의 '노트' 아이콘 을 클릭하여 밴드에 추가합니다.



나. 노트 데이터

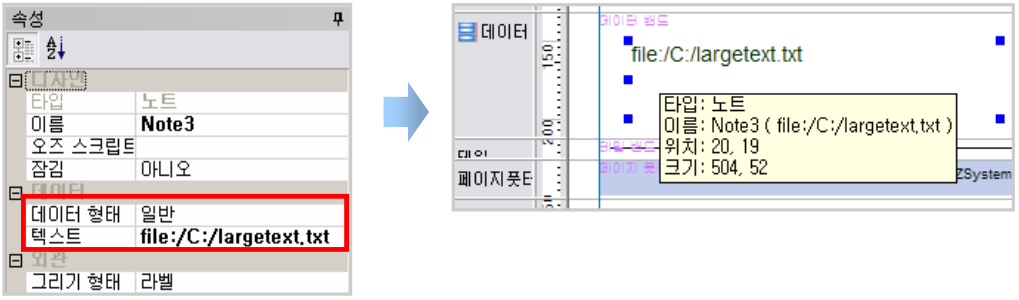
■ **데이터가 데이터베이스에 저장된 대용량 텍스트인 경우 (Binary, BLOB, Long VARCHAR, CLOB 등의 타입)**

① 노트 컴포넌트의 '데이터 형태' 속성을 '데이터'로 변경합니다.
② '필드 이름' 속성에 표현할 데이터 필드를 지정합니다.



■ **데이터가 웹 경로나 로컬 PC에 있는 텍스트 파일인 경우**

① 노트 컴포넌트의 '데이터 형태' 속성을 '일반'으로 변경합니다.
② '텍스트' 속성에 URL 이나 사용자 PC의 로컬 파일 경로를 입력합니다.
예) http://oz.forcs.com/demo_kr/largetext.txt, file:/C:/largetext.txt



가. 멀티플렉스 컴포넌트

라벨이 데이터를 표현하는 최소 단위의 컴포넌트라고 한다면 멀티플렉스 컴포넌트는 여러 데이터를 복합적으로 표현할 수 있는 컴포넌트입니다.

예를 들어, 라벨에는 하나의 데이터 필드를 매핑하여 보여줄 수 있는데 비해, 멀티플렉스 컴포넌트에는 여러 데이터 필드를 컴포넌트 타입에 따라 다양한 방식으로 보여줄 수 있습니다.

나. 멀티플렉스 종류

크로스탭, 차트, 테이블, 간트 차트, 리전, 고정 테이블 컴포넌트가 멀티플렉스 컴포넌트입니다.

- 멀티플렉스 컴포넌트는 '컴포넌트 추가' → '멀티플렉스' 메뉴에서 추가할 수 있습니다.

종류	설명
크로스탭	데이터 분석 및 통계 보고서 작성시 유용한 컴포넌트입니다. 기준 필드의 값에 따라 데이터들을 분류하고 합계, 평균 등을 표시합니다.
차트	2차원 또는 3차원 차트를 보고서에 표시합니다.
테이블	데이터 셋의 내용을 순차적으로 리스트 형태의 표로 출력하는 컴포넌트입니다. 테이블의 모든 라벨은 데이터 필드와 매핑됩니다.
간트 차트	작업 계획, 절차 계획과 일정 계획 등의 내용을 이해하기 쉽게 표시하는 도표입니다.
리전	하나의 보고서 안에 서브 보고서를 생성하고자 할 때 사용합니다. 데이터 밴드는 하나의 데이터 셋과 매핑되는데 데이터 밴드 내에 다른 데이터 셋을 표현하려면 리전을 이용해야 합니다.
고정 테이블	일반적인 표 형태의 보고서를 디자인하는데 사용하는 컴포넌트입니다. 테이블 컴포넌트는 데이터 필드로만 표현이 가능하나, 고정 테이블 컴포넌트는 데이터 필드와의 매핑없이 표를 그릴 수 있으며, 고정테이블 내의 임의의 라벨에 데이터 필드를 선택적으로 매핑할 수 있습니다. 셀 병합, 셀 삽입, 셀 분할, 셀 삭제 등의 기능으로 테이블 형태를 쉽게 변형할 수 있습니다.

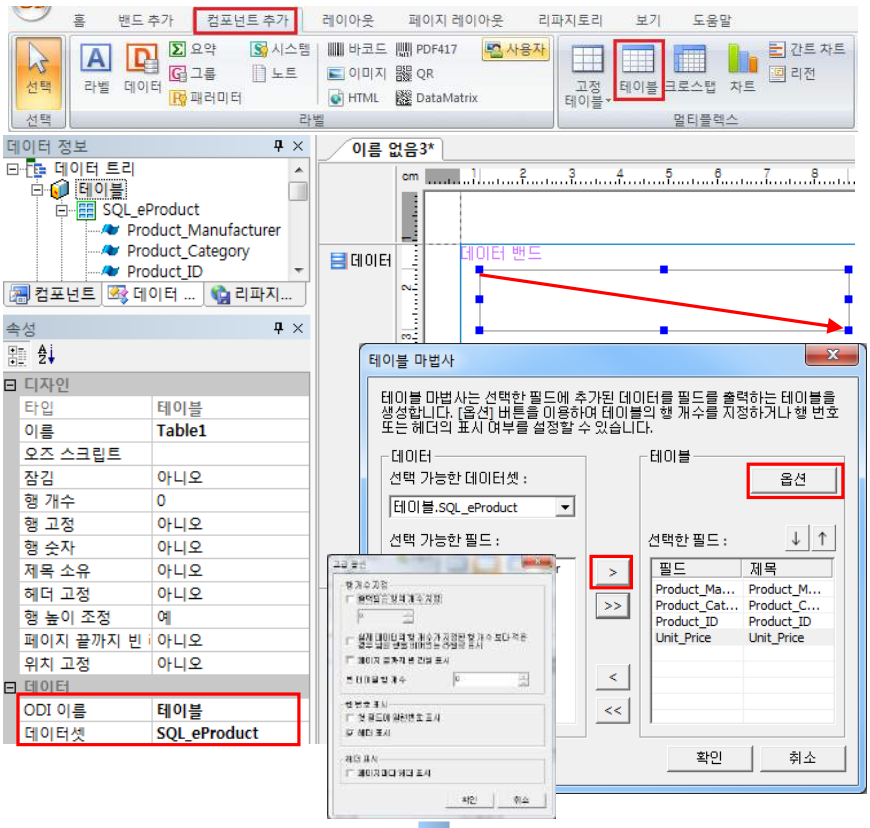
다. 컴포넌트별 사용 가능 밴드

- 고정 테이블 : 모든 밴드에 추가 가능
- 테이블, 크로스탭, 차트, 간트 차트 : 데이터 밴드(서브, 상위그룹 헤더·풋터), 써머리 밴드에 추가 가능
- 리전 : 데이터 밴드(서브, 상위그룹 헤더·풋터), 써머리 밴드, 데이터 익스팬더 밴드에 추가 가능

가. 테이블 정의

테이블은 데이터 셋을 리스트 형태의 표로 보여주기 위해 사용하는 컴포넌트입니다.
마법사를 이용하여 데이터 필드 선택 만으로 간편하게 테이블이 완성됩니다.
일련 번호 표시, 페이지 끝까지 빈칸 표시, 자동 크기 조정, 자동 분할 등의 유용한 옵션을 이용할 수 있습니다.

나. 테이블 구현 방법



① 테이블에 보여줄 데이터 셋을 쿼리 디자이너에서 생성

② 리포트에 ODI 추가

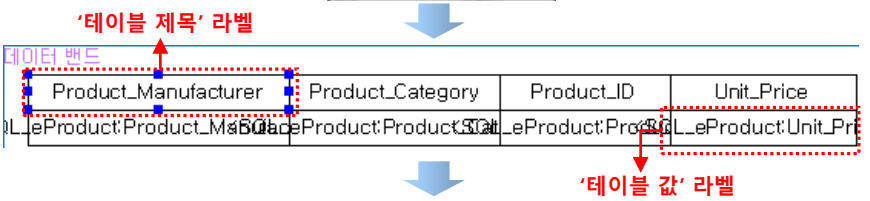
③ 데이터 밴드 추가 후 'ODI 이름', '데이터셋' 속성 설정

④ 툴바의 '테이블' 아이콘 클릭 후 밴드 위에 대각선으로 그리면 테이블 마법사 실행됨

⑤ 테이블 마법사 창의 '선택 가능한 필드'에서 보여줄 필드를 선택한 후 '선택한 필드'로 이동

⑥ '옵션' 버튼 클릭 후 고급 옵션 창에서 필요한 옵션 체크

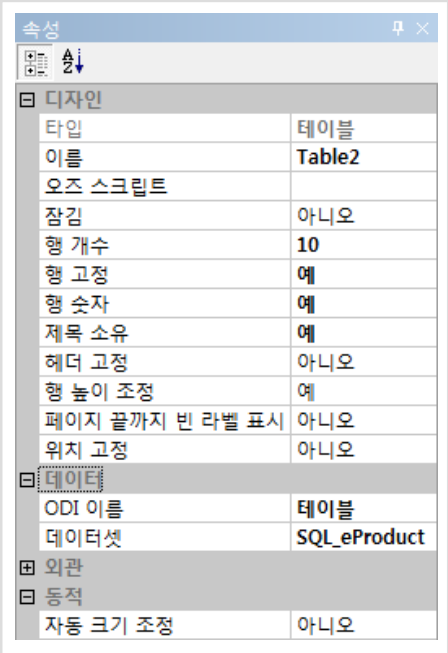
⑦ '확인' 버튼 클릭



⑧ '테이블 제목' 라벨을 더블 클릭 한 후 제목 텍스트 입력하고 배경색 지정

⑨ 툴바의 '미리보기' 아이콘 클릭 후 결과 확인

다. 테이블 속성



속성	설명
행 개수	테이블 출력되는 행의 최대 개수 지정
행 고정	실제 데이터의 행 개수가 지정된 행 개수 보다 적은 경우 남은 행을 비어 있는 라벨로 표시
행 숫자	첫 필드에 일련번호 표시
제목 소유	테이블의 헤더 표시
헤더 고정	페이지마다 테이블의 헤더 표시
행 높이 조정	테이블의 일부 라벨이 그룹핑 되어 있을 경우 다른 라벨의 높이가 조정될 때 균등 분할될지 여부 설정
페이지 끝까지 빈 라벨 표시	페이지의 끝까지 빈 라벨로 표시할지 여부
위치 고정	매 페이지마다 테이블의 시작 위치를 고정할 지 여부. '아니오'인 경우 다음 페이지 부터 용지 여백 만큼만 띄우고 출력됨
ODI 이름	테이블에 보여줄 데이터 셋이 있는 ODI 이름
데이터셋	테이블에 보여줄 데이터 셋 이름 (데이터 밴드의 데이터 셋과 다른 경우 데이터 밴드에 설정된 데이터 셋의 레코드 수만큼 테이블 전체가 반복 출력됨)
자동 크기 조정	테이블 내의 모든 라벨에 대한 자동 크기 조정

➢ **테이블 속성 사용 예**

- 행 개수(MaxRows) : 10, 실제 데이터 행 개수 : 9 인 경우

No.	제조사	제품분류	제품 ID	단가
1	SAMSUNG	TV	TO-142-LKM	412
2	SAMSUNG	VTR	TK-234-CV	253
3	SAMSUNG	AUDIO	JK-745-GG	1,634
4	LG	CAMERA	SD-644-FG	560
5	LG	TV	OP-263-343	1,244
6	LG	AUDIO	OP-643-KK	1,444
7	SONY	TV	SY-246-GS	699
8	SONY	AUDIO	SY-235-KQ	499
9	SONY	VTR	SY-233-DE	299

제목 소유 : 예

행 고정 : 예

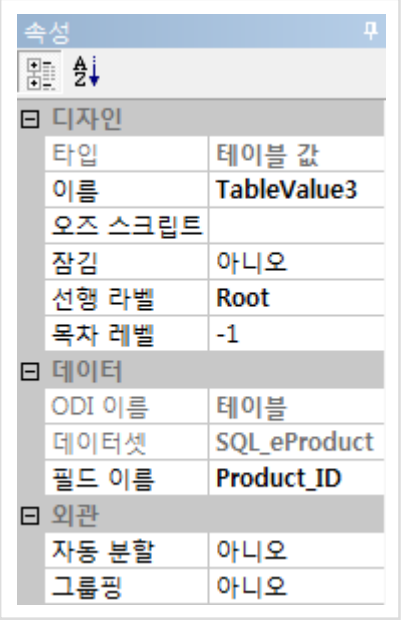
- 행 개수(MaxRows) : 5, 실제 데이터 행 개수 : 9 인 경우

No.	제조사	제품분류	제품 ID	단가
1	SAMSUNG	TV	TO-142-LKM	412
2	SAMSUNG	VTR	TK-234-CV	253
3	SAMSUNG	AUDIO	JK-745-GG	1,634
4	LG	CAMERA	SD-644-FG	560
5	LG	TV	OP-263-343	1,244

행 숫자 : 예

마. 테이블 값 속성

'테이블 값' 라벨은 테이블 마법사에서 선택한 필드에 대해 생성된 데이터 라벨입니다.



속성	설명
선행 라벨	같은 테이블의 다른 '그룹핑' 속성이 설정된 '테이블 값' 라벨을 선행 라벨로 지정하여 선행 라벨에 따라 그룹핑이 되도록 설정
ODI 이름	테이블의 ODI 이름으로 자동 설정됨
데이터셋	테이블의 데이터셋으로 자동 설정됨
필드 이름	테이블 마법사에서 선택한 필드로 변경 가능
자동 분할	테이블 값의 내용이 한 페이지를 넘어가는 경우 자동 분할을 설정하면 여러 페이지에 걸쳐 표현 가능
그룹핑	같은 필드의 동일한 데이터는 병합되어 한 라벨로 표현됨

※ '테이블 값' 라벨은 '데이터 타입' 속성이 지원되지 않으므로 '일반', '패러미터', '요약' 등의 라벨로 변환이 안되나, 스크립트로 보여지는 텍스트를 변경할 수는 있습니다.

➢ **테이블 값 속성 사용 예**

- '제조사', '제품분류' 라벨에 그룹핑 설정

제조사	제품분류	제품 ID	단가
LG	TV	OP-263-343	1,244
		OP-263-343	1,244
SAMSUNG		TO-142-LKM	412
		TO-142-LKM	412
SONY		SY-246-GS	699
		SY-246-GS	699

- '제품분류' 라벨에 '제조사' 라벨을 선행 라벨로 지정

제조사	제품분류	제품 ID	단가
LG	TV	OP-263-343	1,244
		OP-263-343	1,244
SAMSUNG	TV	TO-142-LKM	412
		TO-142-LKM	412
SONY	TV	SY-246-GS	699
		SY-246-GS	699

- '제조사 설명' 라벨에 자동분할 설정

제조사	제조사 설명	제품 ID	단가
SAMSUNG	2020년을 향한 삼성전자의 VISION은 Inspire the World, Create the Future	TO-142-LKM	412

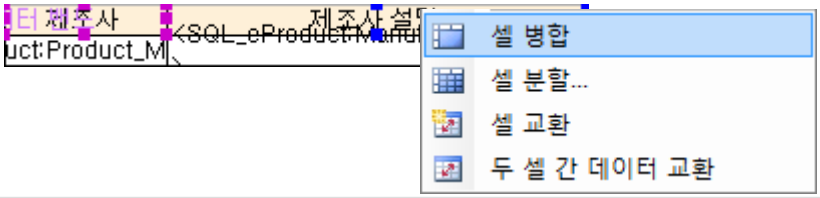
제조사	제조사 설명	제품 ID	단가
	입니다.	TK-234-CV	253
		JK-745-GG	1,634

- 자동분할 설정 안한 경우 다음 페이지로 넘어가서 그려짐

제조사	제조사 설명	제품 ID	단가
SAMSUNG	2020년을 향한 삼성전자의 VISION은 Inspire the World, Create the Future 입니다.	TO-142-LKM	412
		TK-234-CV	253
		JK-745-GG	1,634

라. 테이블 제목 기능

'테이블 제목' 라벨은 '셀 병합', '셀 분할', '셀 교환', '두 셀 간 데이터 교환' 팝업 메뉴를 제공합니다.



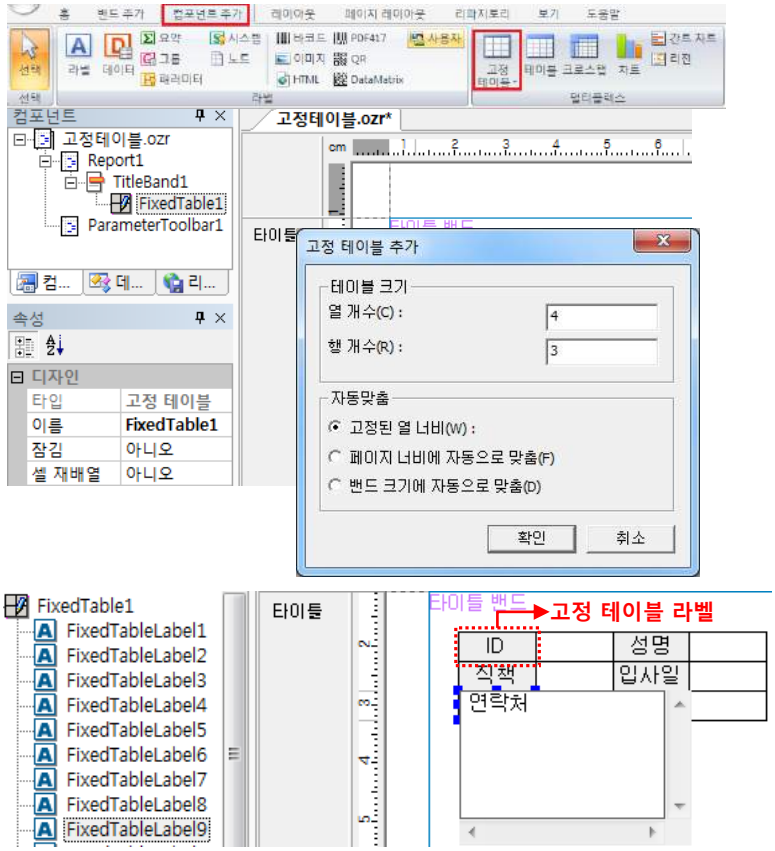
가. 고정 테이블 정의

고정 테이블은 임의의 행, 열 크기로 표를 작성할 수 있는 컴포넌트 입니다.
이력서, 세금계산서와 같이 라벨 크기가 일정하지 않은 형태의 보고서 작성에 유용합니다.
셀 추가, 셀 병합, 셀 분할, 셀 삭제 등의 부가 기능을 사용하여 쉽게 다양한 테이블 폼을 만들 수 있습니다.

고정 테이블은 '고정 테이블 라벨' 이라는 컴포넌트 단위로 이루어져 있습니다.
'고정 테이블 라벨'은 '데이터 형태', '그리기 형태' 등의 일반 라벨이 갖고 있는 속성을 그대로 사용할 수 있습니다.

나. 고정 테이블 구현 방법

1) 고정 테이블 추가



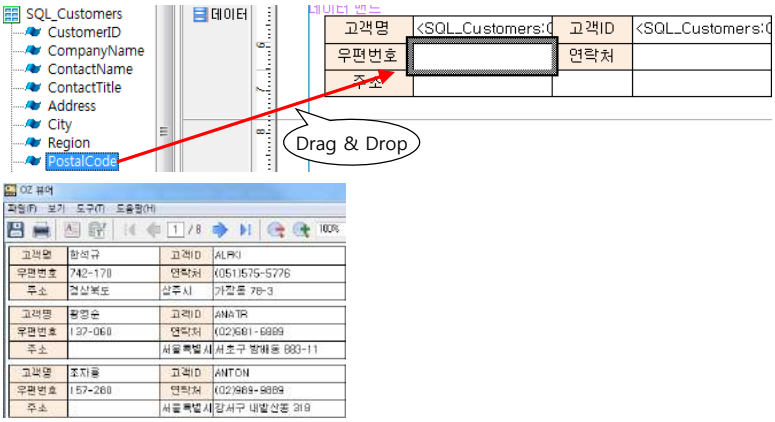
① 임의의 밴드 추가

② 툴바의 '고정 테이블' 아이콘 클릭 후 밴드 위에 대각선으로 그리면 '고정 테이블 추가' 창 나옴

③ 열 개수, 행 개수를 입력한 후 '확인' 버튼 클릭

④ '고정 테이블 라벨'을 더블 클릭한 후 임의의 텍스트 입력하고 배경색 지정

2) 데이터 필드 매핑

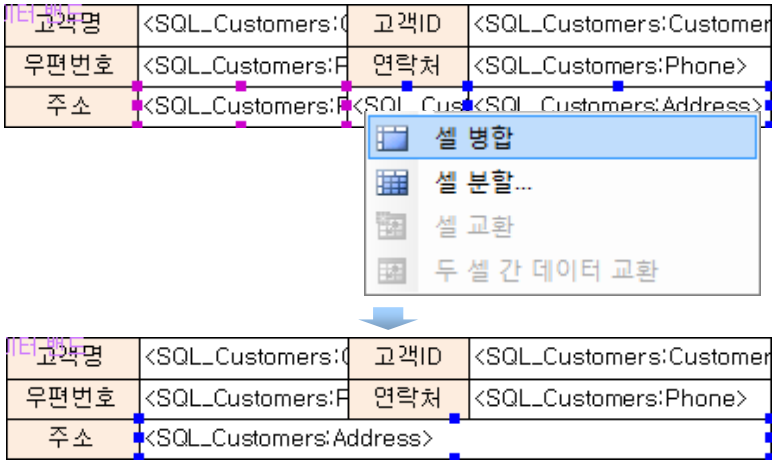


① 데이터 밴드 위에 고정 테이블 추가

② '데이터 정보' 창의 '데이터 필드'를 '고정 테이블 라벨'로 드래그&드롭

③ 미리보기하면 데이터 셋의 레코드 수만큼 고정 테이블이 반복 출력됨

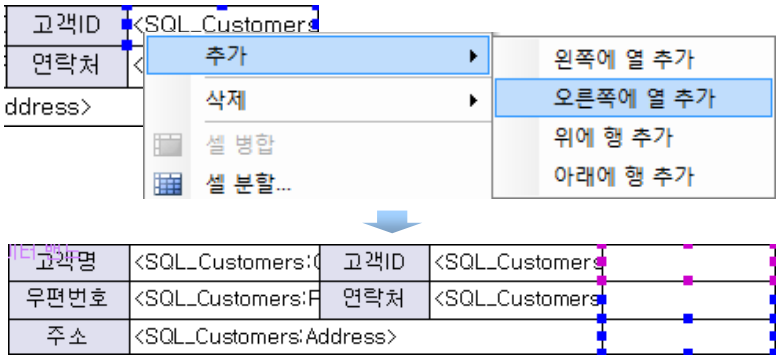
3) 셀 병합



① 병합하고자 하는 고정 테이블 라벨 여러 개를 Ctrl 키를 누른 상태에서 선택

② 팝업 메뉴에서 '셀 병합' 선택

4) 셀 추가



① 라벨 오른쪽에 새로운 열을 추가하기 위해서 고정 테이블 라벨의 팝업 메뉴에서 '추가' → '오른쪽에 열 추가' 메뉴 선택

다. 고정테이블 속성

속성	설명
셀 재배열	고정 테이블 라벨의 크기가 변경될 때 전체 고정 테이블 라벨을 재배열할지 여부 설정

➢ **'셀 재배열' 속성 사용 예**

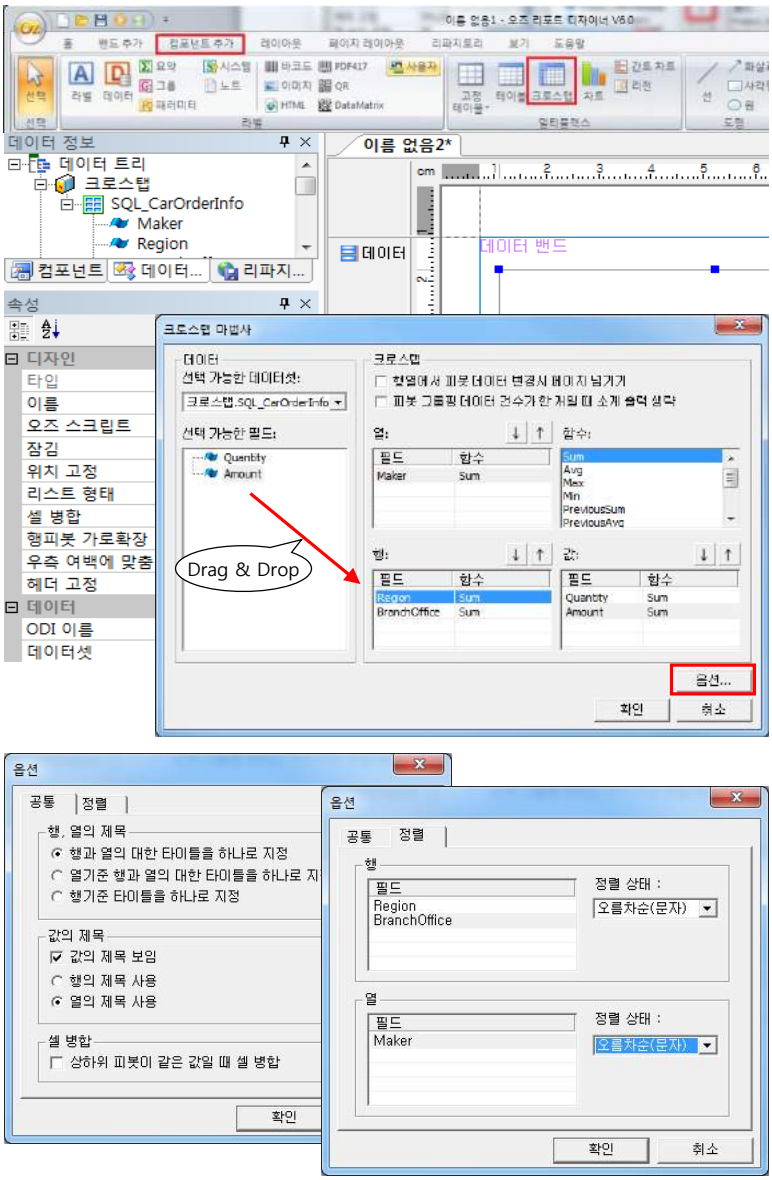
※ 고정 테이블 라벨의 '자동 크기 조정' 속성이 '예'인 경우 고정 테이블의 '셀 재배열' 속성 값과 상관없이 셀 재배열이 됩니다.

※ 셀 재배열 될 때 고정 테이블 높이가 추가되어 있는 데이터 밴드 높이보다 커지는 경우 데이터 밴드의 '자동 크기 조정' 속성을 '예'로 설정해야 고정 테이블 라벨이 잘려 보이지 않습니다.

가. 크로스탭 정의

크로스탭은 기준 필드에 따라 데이터들을 분류하고 합계, 평균 등을 포함하는 다차원 테이블 보고서를 작성하기 위한 컴포넌트입니다. 데이터 분석 및 통계 보고서 작성에 유용합니다.

나. 크로스탭 구현 방법



① 크로스탭에 보여줄 데이터 셋을 쿼리 디자이너에서 생성

② 리포트에 ODI 추가

③ 데이터 밴드 추가 후 'ODI 이름', '데이터셋' 속성 설정

④ 툴바의 '크로스탭' 아이콘 클릭 후 밴드 위에 대각선으로 그리면 크로스탭 마법사 실행됨

⑤ '크로스탭 마법사' 창의 '선택 가능한 필드'를 드래그&드롭으로 열, 행, 값 필드 창에 각각 추가

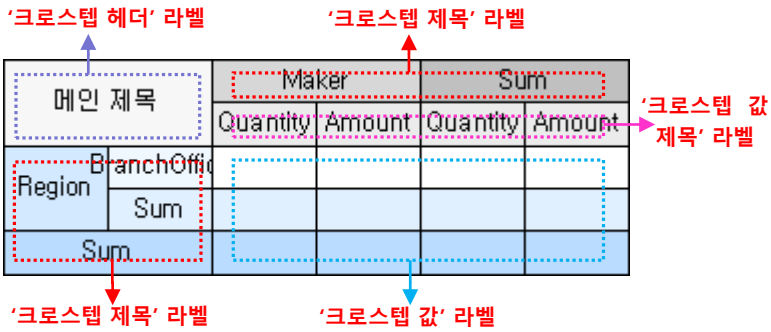
※ 추가된 필드를 제거할 때는 다시 '선택 가능한 필드'창으로 드래그&드롭

⑥ 합계를 보여줄 필드의 함수를 Sum으로 지정

⑦ '옵션' 버튼 클릭 후 '공통' 탭의 '값의 제목 보임' 체크

⑧ '정렬' 탭에서 행, 열 필드의 정렬 상태 지정

⑨ '확인 버튼 클릭



⑩ 생성된 크로스탭의 '크로스탭 헤더', '크로스탭 값 제목', 'Sum' 라벨의 텍스트 변경

⑪ '크로스탭 값' 라벨의 라벨 서식을 '숫자'로 지정

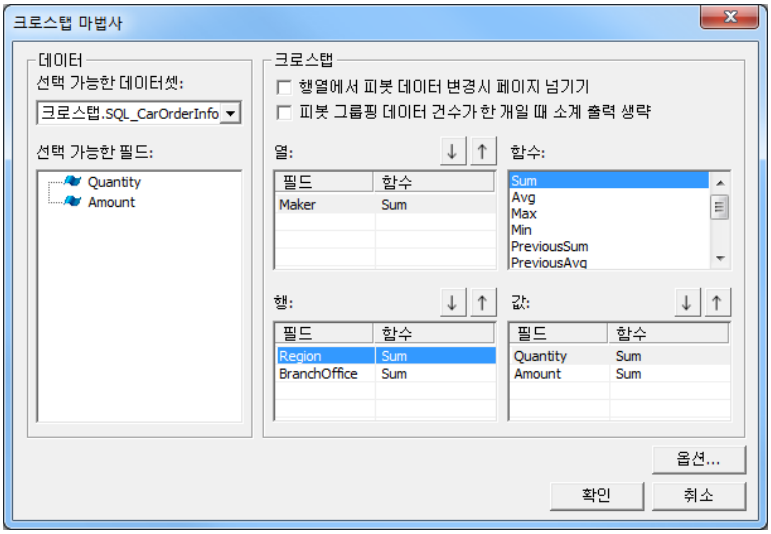
제조사 / 지역/지점		기아자동차		대우자동차		현대자동차		합계	
		수량	금액	수량	금액	수량	금액	수량	금액
경기	수원	3	67,020,000	2	46,610,000	2	32,000,000	7	145,630,000
	인천	2	39,000,000	1	32,520,000	4	85,810,000	7	157,330,000
	지역 소계	5	106,020,000	3	79,130,000	6	117,810,000	14	302,960,000
경상	대구	2	31,920,000			2	44,800,000	4	76,720,000
	부산	2	35,560,000	4	73,520,000	3	65,460,000	9	174,540,000
	지역 소계	4	67,480,000	4	73,520,000	5	110,260,000	13	251,260,000
서울	강남	4	82,560,000	3	58,470,000	5	93,060,000	12	234,090,000
	종로	4	84,520,000	3	48,060,000	3	88,600,000	10	221,180,000
	지역 소계	8	167,080,000	6	106,530,000	8	181,660,000	22	455,270,000
전라	광주	4	79,300,000			1	30,500,000	5	109,800,000
	전주	2	40,410,000	3	58,520,000	2	65,000,000	7	163,930,000
	지역 소계	6	119,710,000	3	58,520,000	3	95,500,000	12	273,730,000
충청	대전	4	66,270,000	2	34,560,000			6	100,830,000
	충주			1	27,410,000	4	77,000,000	5	104,410,000
	지역 소계	4	66,270,000	3	61,970,000	4	77,000,000	11	205,240,000
전국 합계		27	526,560,000	19	379,670,000	26	582,230,000	72	1,488,460,000

⑫ 미리보기 하면 행, 열 필드에 따라 값이 가변적으로 보여짐. 별도의 설정 없이 동일한 데이터는 그룹핑되어 보여짐

※ 크로스탭 스크립트 실행 순서

0		1		2
		3	4	5
6	7	8	9	10
	11	12	13	14
	15	16	17	18

다. 크로스탭 마법사



※ 함수는 Ctrl키나 Shift키를 누르고 여러 개 선택 가능

- **피봇** :

행, 열 필드를 가리킴

예) '인천'의 상위 피봇은 '경기'이고 '경상'의 하위 피봇은 '대구', '부산' 임

- **행, 열 필드의 함수** :

함수 결과를 보여주는 별도의 행, 열이 그룹핑된 데이터 별로 추가되어 보여짐

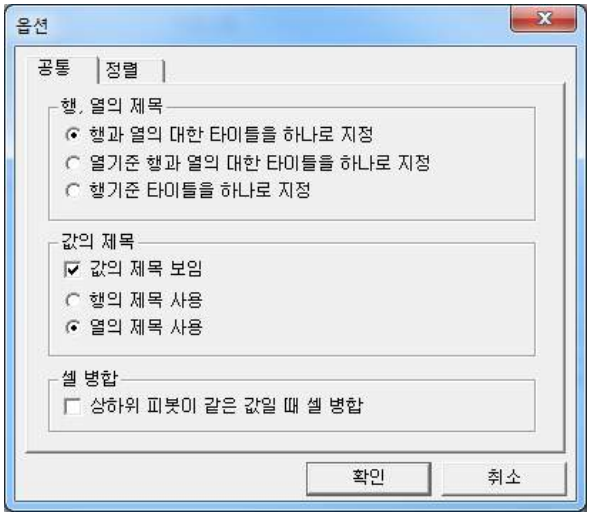
- **값 필드의 함수** :

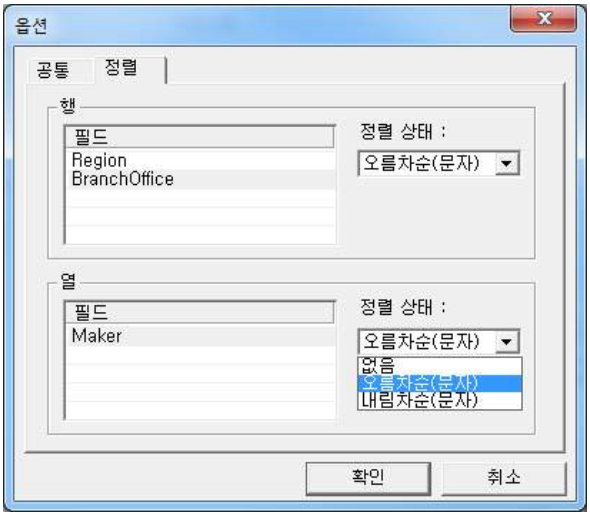
값 필드의 함수 결과가 '크로스탭 값' 라벨에 보여짐.

예) 동일 행, 열에 대한 값이 한 개 이상이고 값의 함수가 'Sum'이면 여러 값을 합산하여 '크로스탭 값' 라벨에 표시

▪ **함수 종류**

함수	설명
Sum, Avg, Max, Min	필드의 합계, 평균, 최대, 최소 값을 필드 아래 부분에 출력
Previous Sum, Previous Avg, Previous Max, Previous Min	필드의 합계, 평균, 최대, 최소 값을 필드 윗 부분에 출력
Invisible	필드 감추기. 출력은 안되나 필드에 따른 그룹핑 분류나 스크립트는 실행됨
Count	필드의 데이터 건수 출력
None	필드의 함수 사용 안함





■ '행, 열의 제목' 옵션

- 행과 열의 대한 타이틀을 하나로 지정

제조사	
지역/지점	수량
수원	3

- 열기준 행과 열에 대한 타이틀을 하나로 지정

지역	지점	
		수량
	수원	3

- 행 기준 타이틀을 하나로 지정

제조사	
값 구분	수량
수원	3

■ '값의 제목' 옵션

- '값의 제목 보임' 체크 해제

지역/지점/제조사		기아자동차	
경기	수원	3	67,020,000
	인천	2	39,000,000

- '값의 제목 보임' 체크 + 행의 제목 사용

지역/지점/제조사		기아자동차
수원	수량	3
	금액	67,020,000

- '값의 제목 보임' 체크 + 열의 제목 사용

제조사	기아자동차	
지역/지점	수량	금액
수원	3	67,020,000

■ '셀 병합' 옵션

- '상하위 피봇이 같은 값일 때 셀 병합' 체크 해제

메인 제목		기아자동차
경기	수원	2
서울	서울	3
충청	대전	1

- '상하위 피봇이 같은 값일 때 셀 병합' 체크

메인 제목		기아자동차
경기	수원	2
서울		3
충청	대전	1

■ '정렬' 옵션

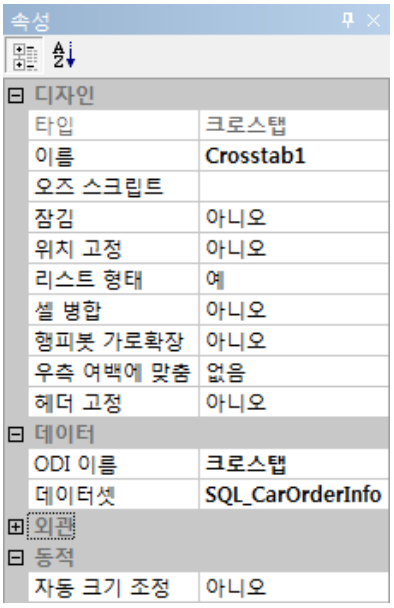
- 정렬 상태 : 없음

제조사		현대자동차		대우자동차	
지역/지점		수량	금액	수량	금액
서울	종로	3	88,600,000	3	48,060,000
	강남	5	93,060,000	3	58,470,000
	지역 소계	8	181,660,000	6	106,530,000
충청	충주	4	77,000,000	1	27,410,000
	대전			2	34,560,000

- 정렬 상태 : 오름차순(문자)

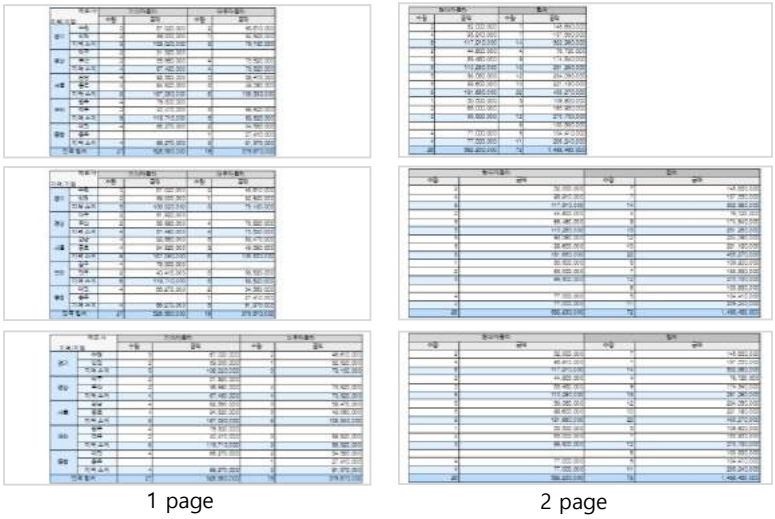
제조사		기아자동차		대우자동차	
지역/지점		수량	금액	수량	금액
경기	수원	3	67,020,000	2	46,610,000
	인천	2	39,000,000	1	32,520,000
	지역 소계	5	106,020,000	3	79,130,000
경상	대구	2	31,920,000		
	부산	2	35,560,000	4	73,520,000

라. 크로스탭 속성



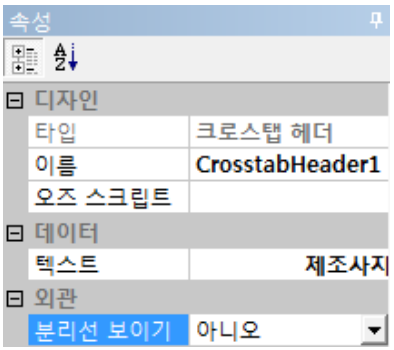
속성	설명
위치 고정	매 페이지 마다 크로스탭 시작 위치를 고정시킬지 여부. 고정하지 않으면 두 번째 페이지부터 용지 여백만 띄우고 출력됨
리스트 형태	동일한 행, 열 필드에 대한 값 필드가 다수일 때 값을 하나만 보여줄지 모두 보여줄지 여부. '예' 이면 모두 표시
행피봇 가로확장	'자동 크기 조정' 속성이 '예'일 때 행 피봇에 대해 높이가 아닌 너비로 자동 크기 조정되게 설정
우측 여백에 맞춤	크로스탭 내의 모든 라벨의 너비를 페이지 오른쪽 끝에 맞춰 조정할지 여부
헤더 고정	매 페이지마다 크로스탭의 행, 열을 반복 출력할지 여부
ODI 이름	크로스탭에 보여줄 데이터 셋이 있는 ODI 이름
데이터셋	크로스탭에 보여줄 데이터 셋 이름 (데이터 밴드의 데이터셋과 다른 경우 데이터 밴드에 설정된 데이터 셋의 레코드 수만큼 크로스탭 전체가 반복 출력됨)
자동 크기 조정	크로스탭 내의 모든 라벨의 텍스트가 잘리지 않도록 자동 크기 조정. 행 피봇의 경우 높이가 조정되고, 열 피봇의 경우 너비가 조정됨

➢ **'우측 여백에 맞춤' 속성 사용 예**



- '우측 여백에 맞춤'이 '없음' 일 때
- '우측 여백에 맞춤'이 '마지막 가로 페이지' 일 때
- '우측 여백에 맞춤'이 '모든 페이지' 일 때

마. 크로스탭 헤더 속성



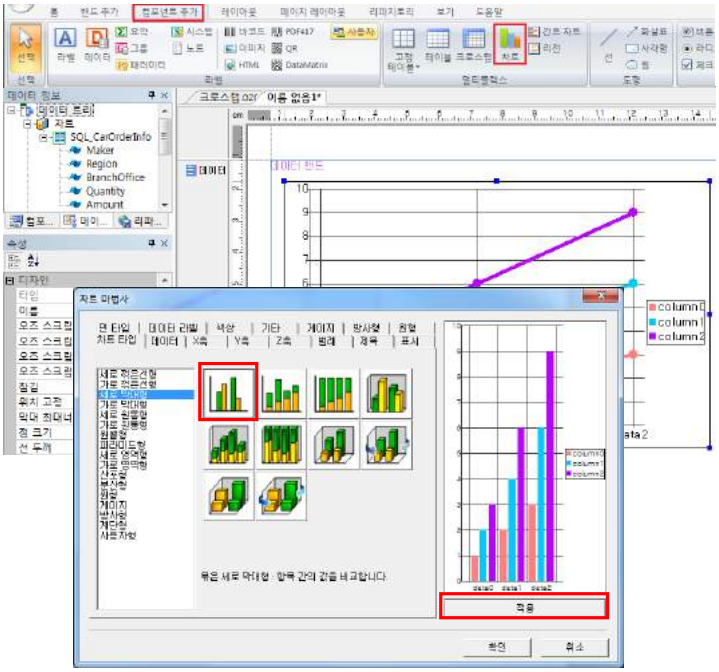
- '분리선 보이기' : 아니오

- '분리선 보이기' : 예

가. 차트 정의

차트는 데이터 셋의 데이터를 2차원, 3차원 그래프로 표현하는 컴포넌트입니다.
크로스탭과 함께 분석 및 통계 보고서 작성에 가장 많이 사용되는 컴포넌트입니다.

나. 차트 구현 방법



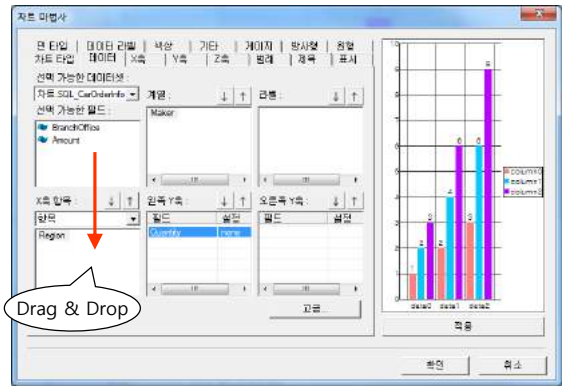
① 차트에 보여줄 데이터 셋을 쿼리 디자이너에서 생성

② 리포트에 ODI 추가

③ 데이터 밴드 추가 후 'ODI 이름', '데이터 셋' 속성 설정

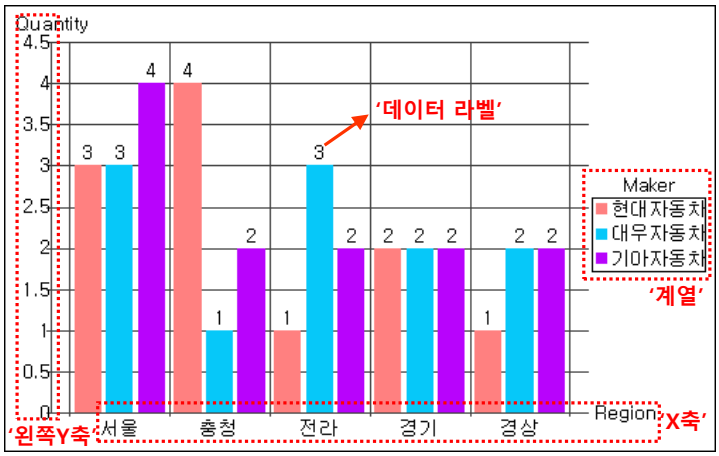
④ 툴바의 '차트' 아이콘 클릭 후 밴드 위에 대각선으로 그리면 차트 마법사 실행됨

⑤ '차트 마법사' 창의 '차트 타입' 탭에서 왼쪽 목록의 '세로 막대형' 선택. 오른쪽 차트 이미지 중에서 '묶은 세로 막대형' 선택. '적용' 버튼 클릭



⑥ '데이터' 탭에서 '선택 가능한 필드'를 드래그&드롭으로 'X축 항목', '왼쪽 Y축', '계열' 창에 각각 추가

⑦ '데이터 라벨' 탭에서 '보기' 체크



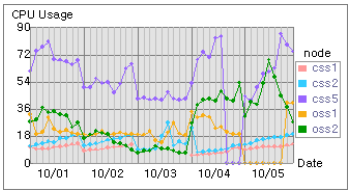
⑧ 툴바의 '미리보기' 아이콘 클릭 후 결과 확인

다. 차트 마법사

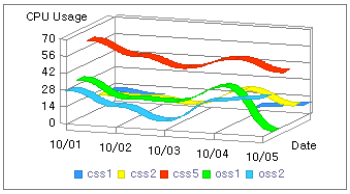
1) 차트 타입

차트 마법사의 '차트 타입' 탭에서 100여 가지 차트 종류를 지정할 수 있습니다.

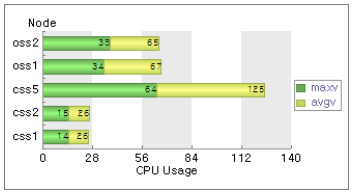
- 세로 꺾은선형



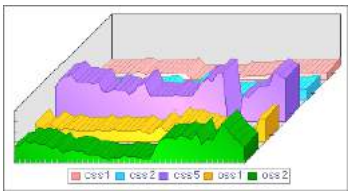
- 입체효과 꺾은선형



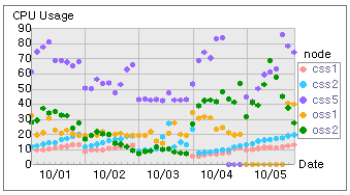
- 가로 막대형



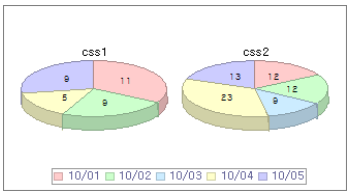
- 세로 영역형



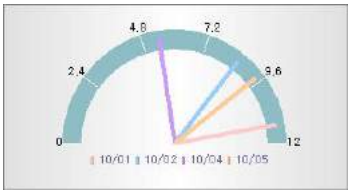
- 산포형



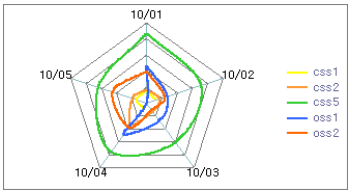
- 원형



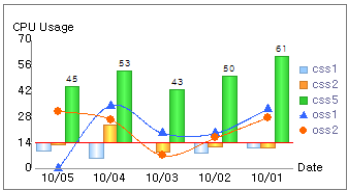
- 게이지



- 방사형

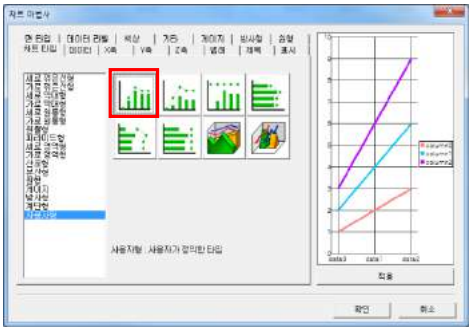


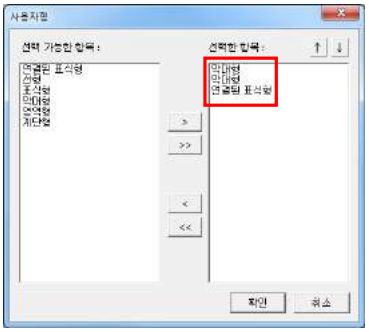
- 사용자형

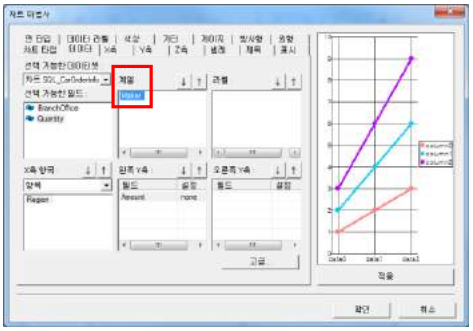


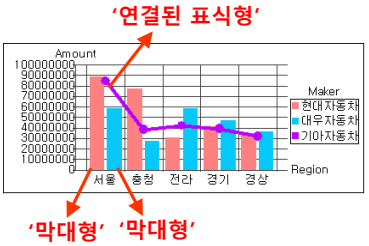
➢ **'사용자형' 차트 구현 예**

'사용자형' 차트는 계열 별로 원하는 차트 타입을 선택하여 그릴 수 있습니다.









① 차트 마법사의 '차트 타입' 탭에서 왼쪽 목록의 '사용자형' 클릭

② 차트 이미지 중에 하나 클릭하면 '사용자형' 창이 나옴

③ '사용자형' 창에서 '선택 가능한 항목'에서 '선택한 항목'으로 원하는 차트 타입 이동

④ '데이터' 탭에서 계열 지정

⑤ 미리보기 결과 선택한 차트가 계열 순서대로 그려짐

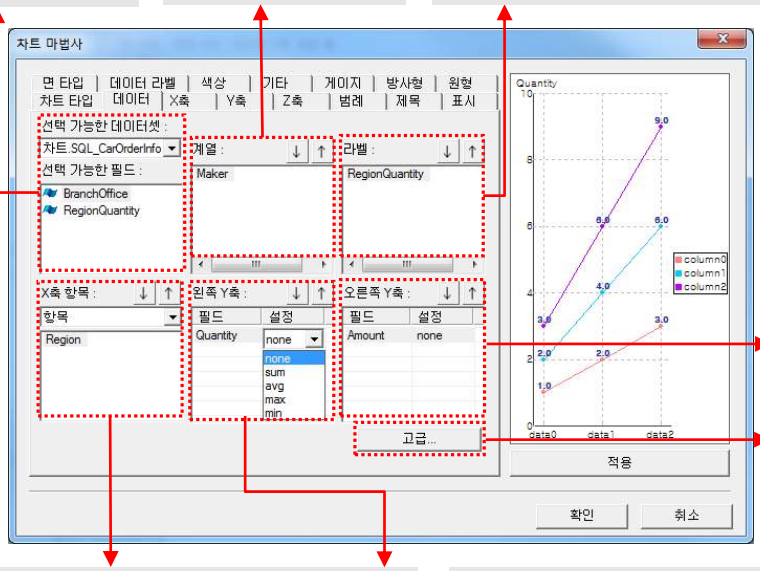
2) 데이터

차트 마법사의 '데이터' 탭에서 차트에 보여줄 데이터 셋의 데이터 필드를 설정합니다.

• **선택 가능한 필드** :
차트에 설정된 데이터 셋의 필드 목록

• **계열** :
차트의 범례에 사용될 필드 목록

• **라벨** :
차트의 데이터 라벨로 보여줄 필드 목록. Y축 필드 수와 맞춰야 하며, 설정 안하면 Y축 데이터가 데이터 라벨로 보여짐



• **오른쪽 Y축** :
차트 오른쪽 세로 축에 사용될 필드 목록

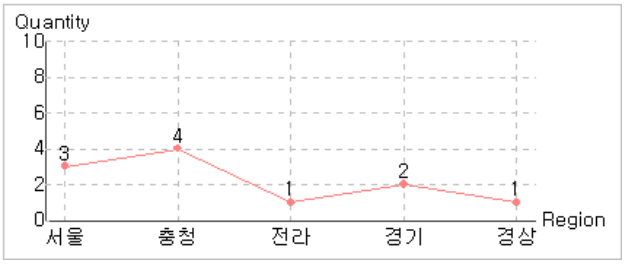
• **고급** :
에러 범위 설정

• **X축 항목** :
차트 가로 축에 사용될 필드 목록. X축 데이터 형태를 '항목', '숫자', '날짜' 타입 중에서 선택

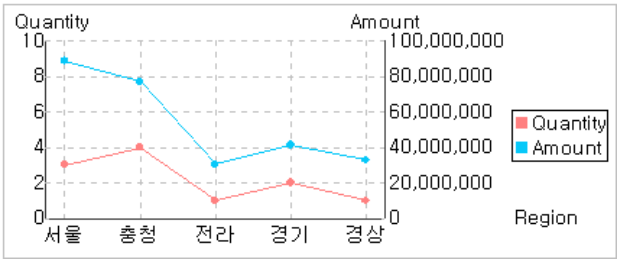
• **왼쪽 Y축** :
차트 왼쪽 세로 축에 사용될 필드 목록. 필드 별로 집계 함수 'sum', 'avg', 'max', 'min' 설정 가능

➢ **차트 데이터 필드 설정 예**

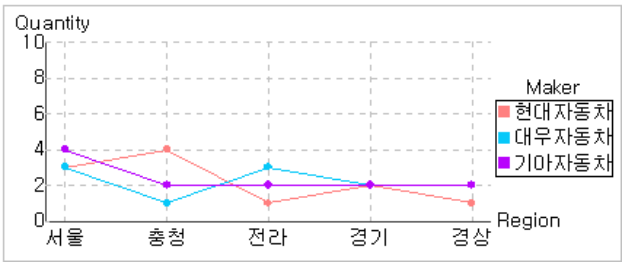
- X축 : Region
- 왼쪽 Y축 : Quantity



- X축 : Region
- 왼쪽 Y축 : Quantity
- 오른쪽 Y축 : Amount



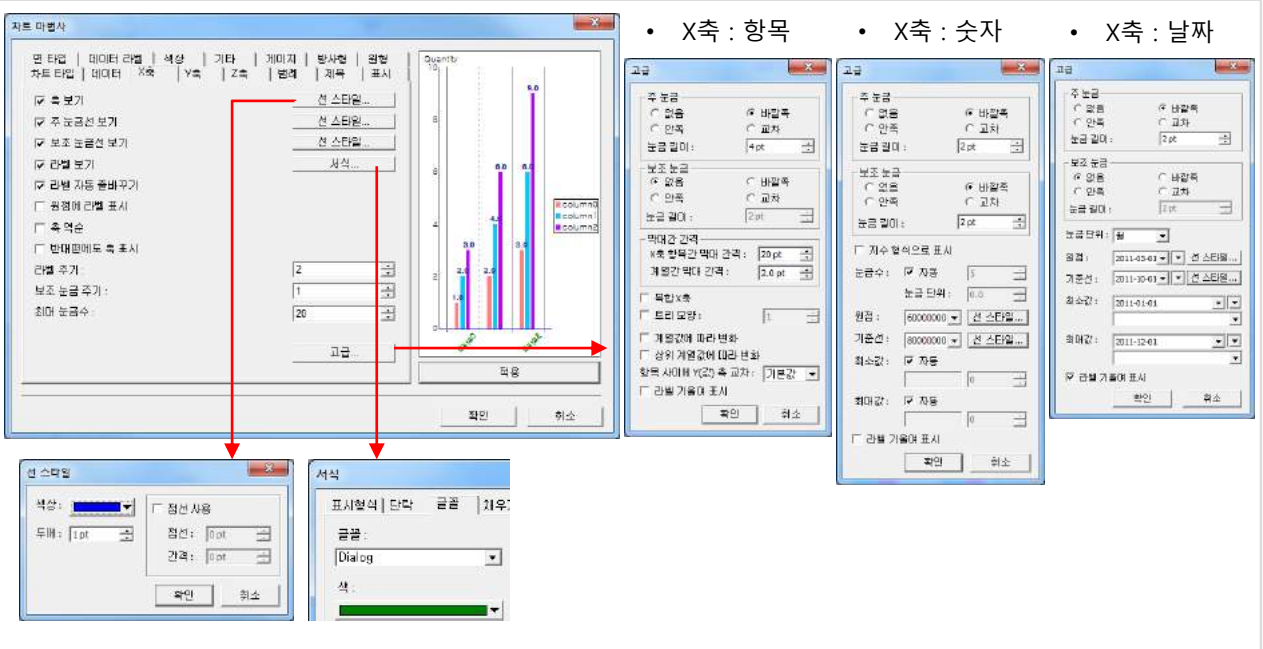
- X축 : Region
- 왼쪽 Y축 : Quantity
- 계열 : Maker



- X축 : Region
- 왼쪽 Y축 : Quantity
- 라벨 : RegionQuantity

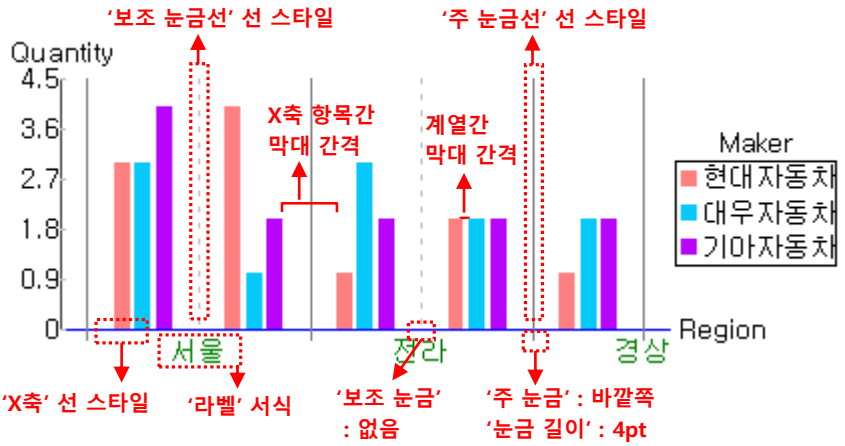
3) X축

차트 마법사의 'X축' 탭에서 차트의 X축 선 스타일, 서식, 눈금 등을 설정합니다.

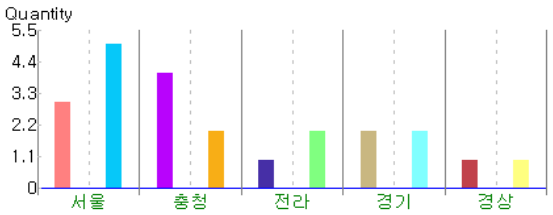


➢ **차트 X축 설정 예**

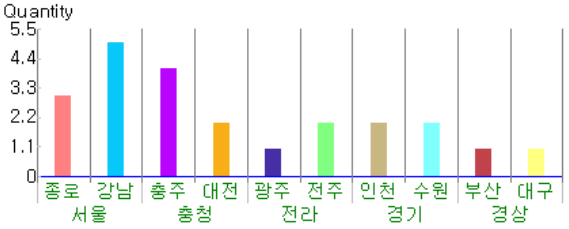
- '라벨 주기'를 '2'로 설정하면 X축 라벨이 눈금 2개 마다 한번씩 보여지고 주 눈금선도 눈금 2개 마다 보여짐



- '고급' 창의 '복합 X축' 체크한 경우



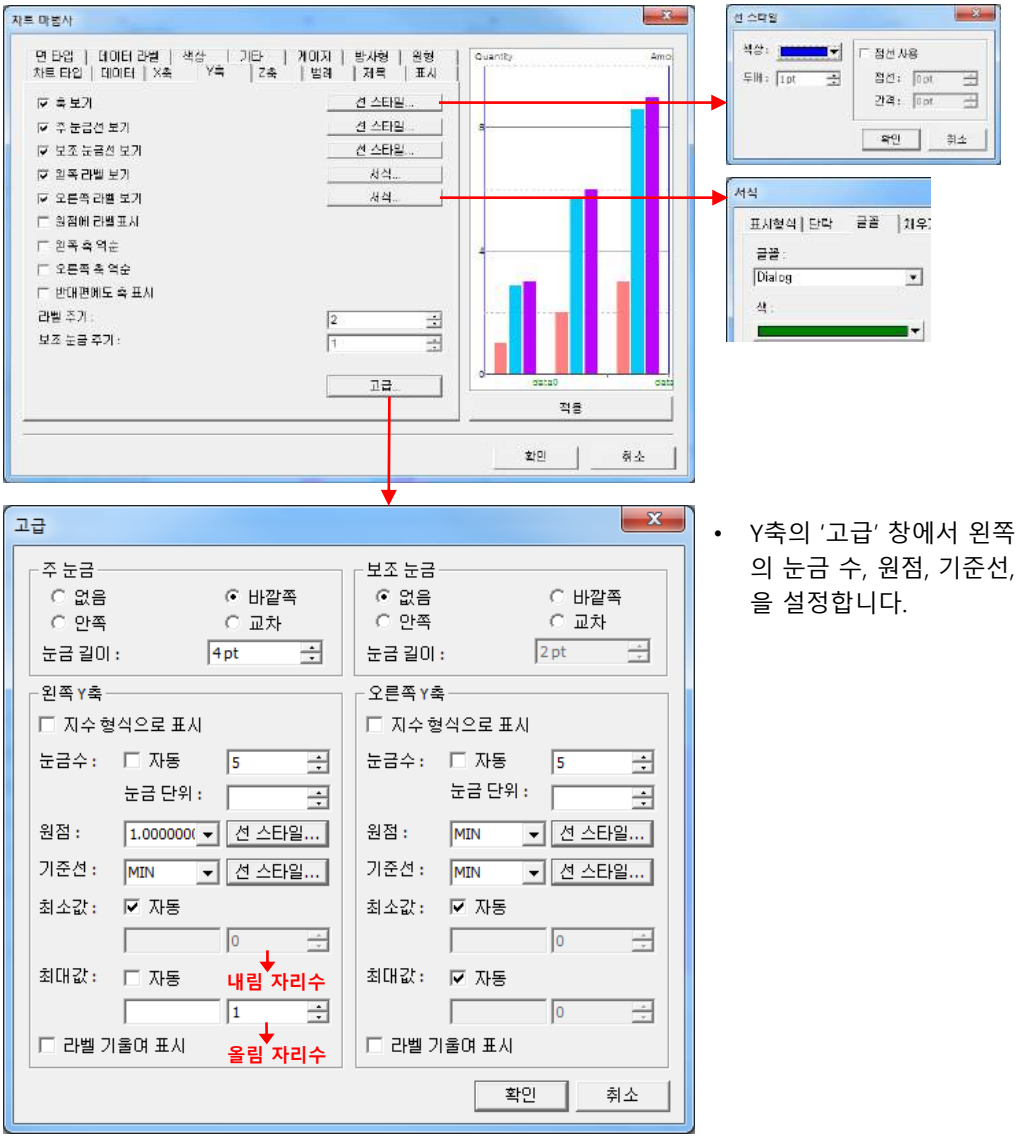
- '고급' 창의 '트리 모양'을 '2'로 설정한 경우



※ X축 데이터 형태가 '숫자', '날짜'인 경우의 고급 옵션은 '오즈 리포트 디자이너 도움말' → '마법사' → '차트 마법사' → 'X축' 참조

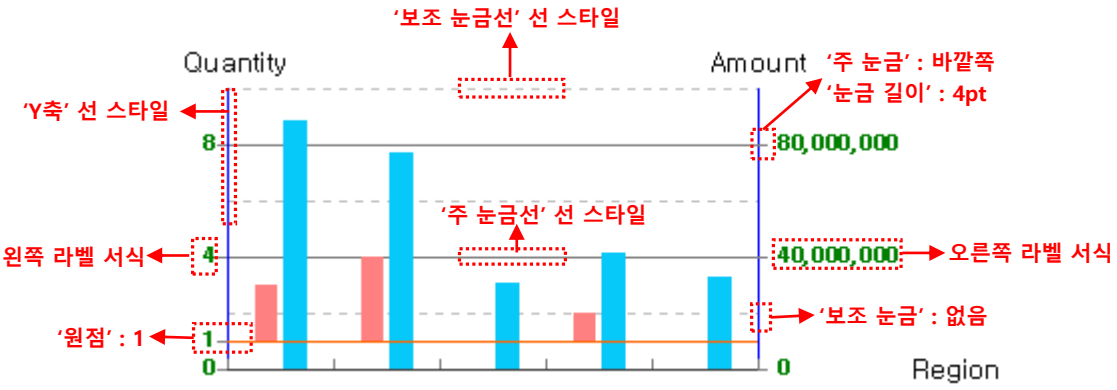
4) Y축

차트 마법사의 'Y축' 탭에서 차트의 Y축 선 스타일, 서식, 눈금, 최대값, 최소값, 기준선 등을 설정합니다.



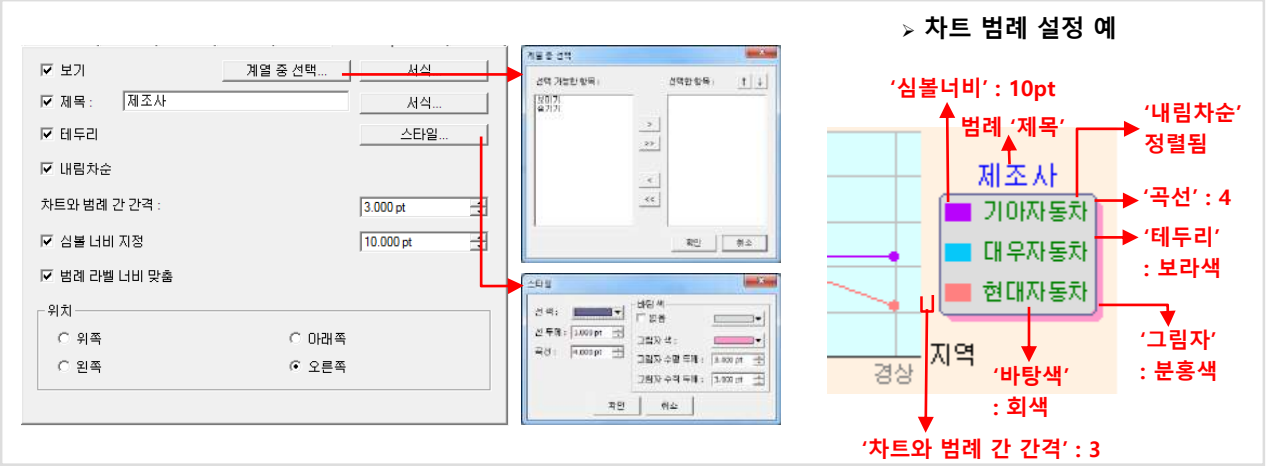
- Y축의 '고급' 창에서 왼쪽 Y축과 오른쪽 Y축의 눈금 수, 원점, 기준선, 최소값, 최대값 등을 설정합니다.

➢ **차트 Y축 설정 예**



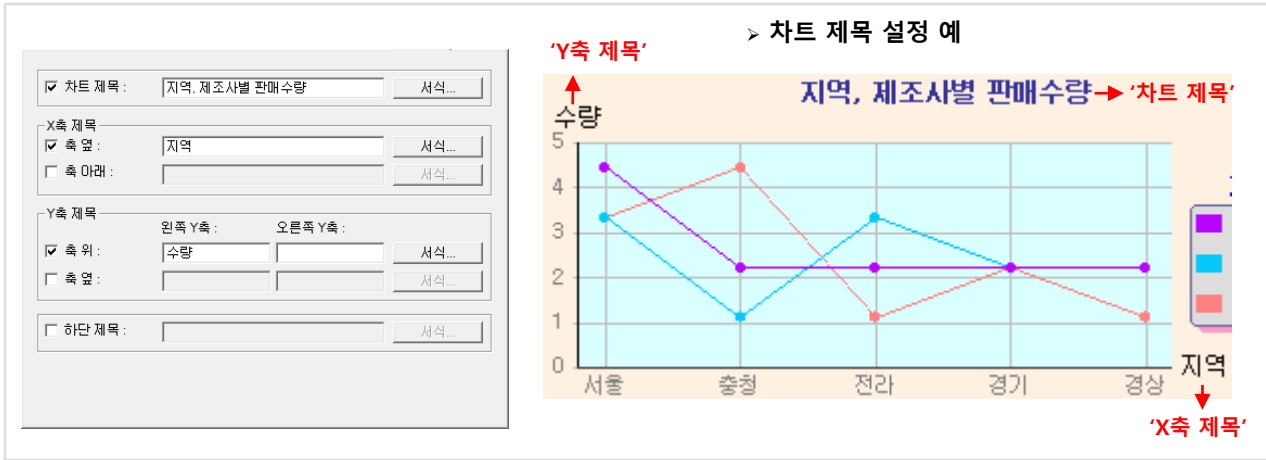
5) 범례

차트 마법사의 '범례' 탭에서 계열에 대한 속성을 설정합니다.



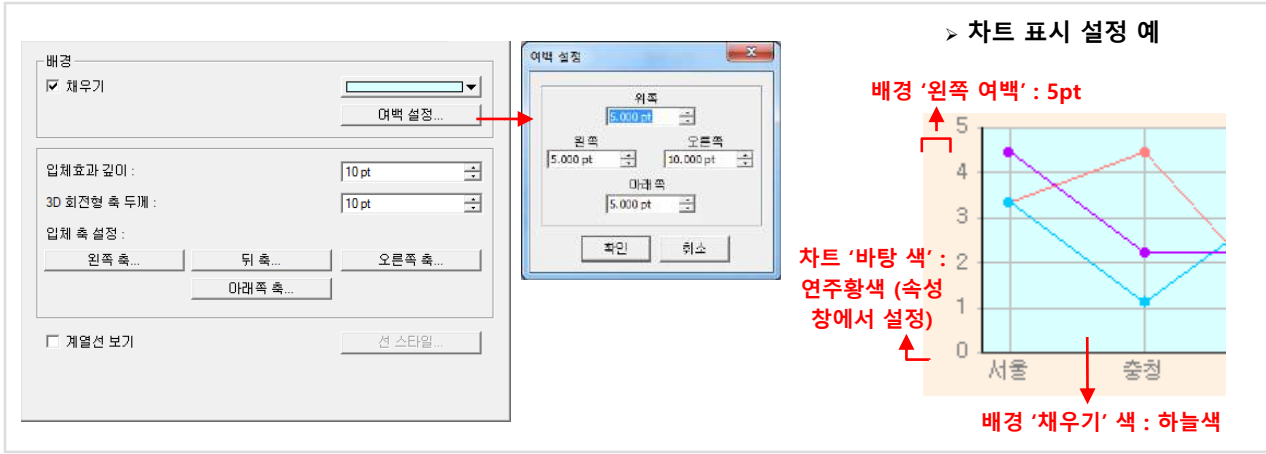
6) 제목

차트 마법사의 '제목' 탭에서 차트 제목과 X축, Y축 제목을 설정합니다.



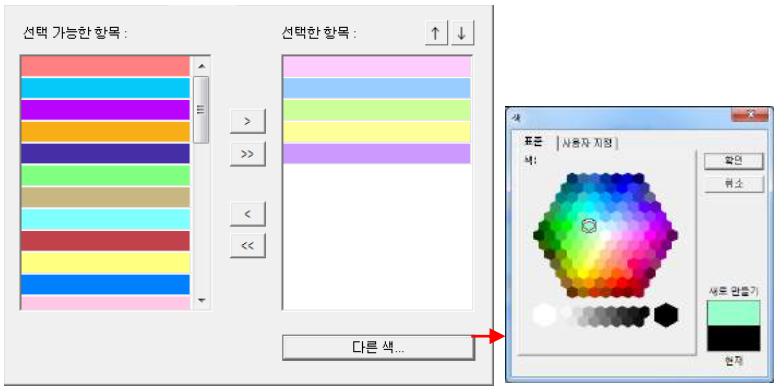
7) 표시

차트 마법사의 '표시' 탭에서 차트 채우기 색과 여백 등을 설정합니다.



8) 색상

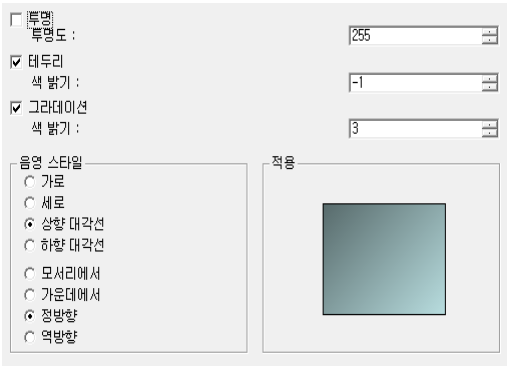
차트 마법사의 '색상' 탭에서 그래프의 색을 설정합니다.



➢ 차트 색상 설정 예

9) 면 타입

차트 마법사의 '면 타입' 탭에서 그래프의 투명, 테두리, 그라데이션을 설정합니다.

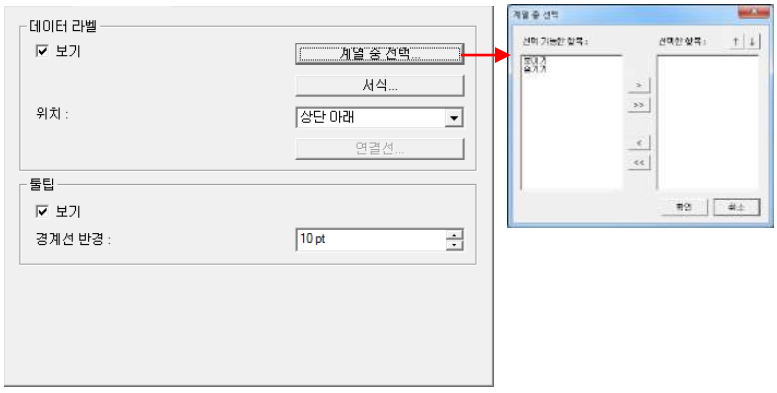


➢ 차트 투명 설정 예

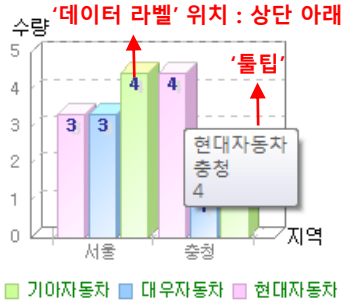
➢ 테두리, 그라데이션 설정 예

10) 데이터 라벨

차트 마법사의 '데이터 라벨' 탭에서 데이터 라벨과 툴팁을 설정합니다.

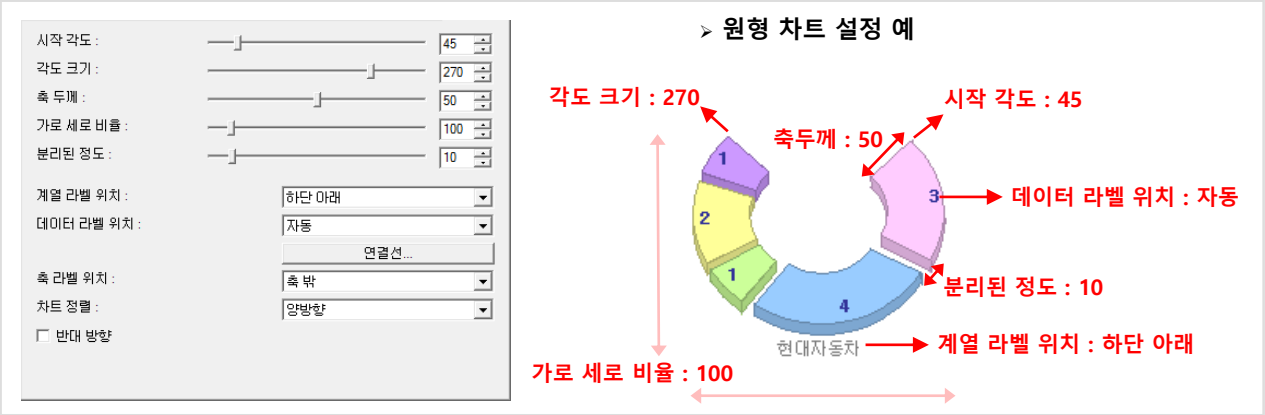


➢ 차트 데이터 라벨 설정 예



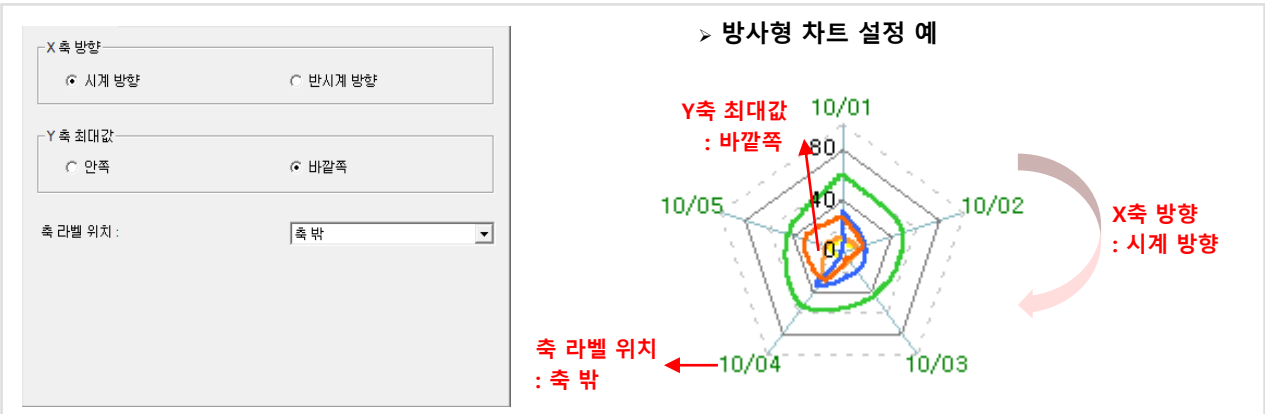
11) 원형

차트 마법사의 '원형' 탭에서 원형 차트를 설정합니다.



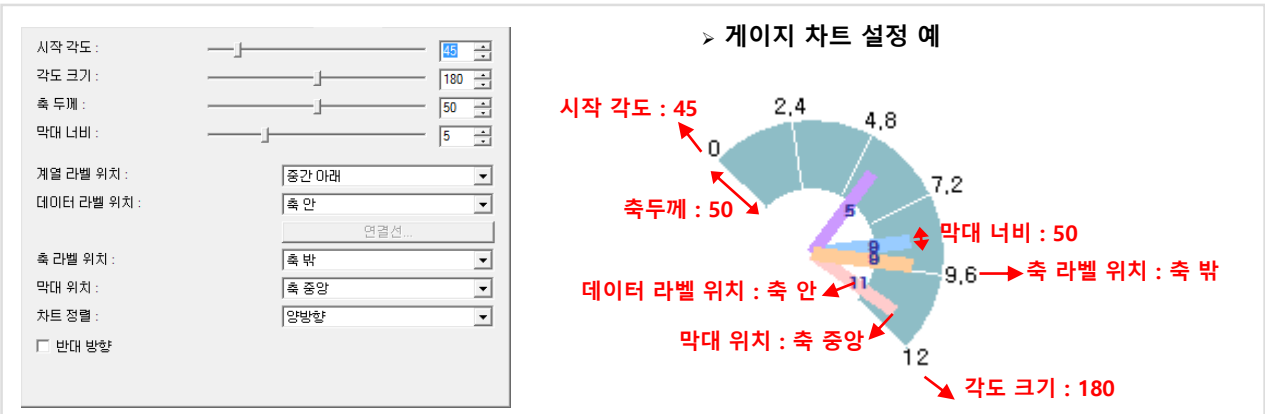
12) 방사형

차트 마법사의 '방사형' 탭에서 방사형 차트를 설정합니다.



13) 게이지

차트 마법사의 '게이지' 탭에서 게이지 차트를 설정합니다.

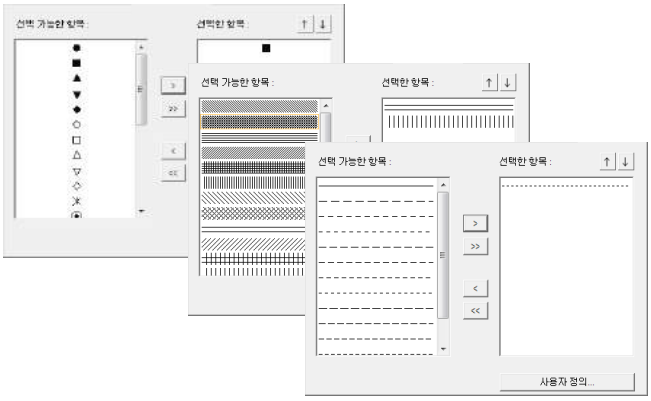


14) 기타

차트 마법사의 '기타' 탭에서 차트의 간격, 점/선/막대 크기, 패턴, 곡선 등을 설정합니다.
'기타' 탭의 '패턴 적용'을 체크하면 '마크 패턴', '면 패턴', '선 패턴' 탭이 생깁니다.

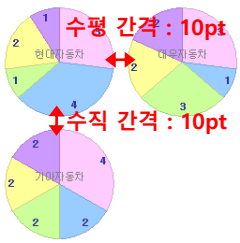
- '기타' 탭

- '마크 패턴', '면 패턴', '선 패턴' 탭

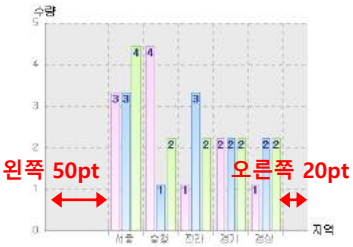


➢ **차트 기타 설정 예**

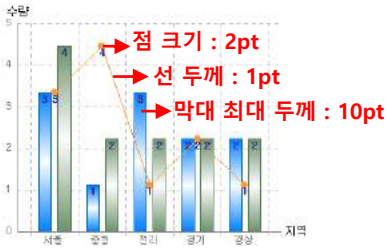
- 원형 타입



- 내부 간격



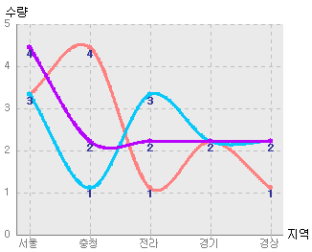
- 점 크기, 선 두께, 막대 최대 두께



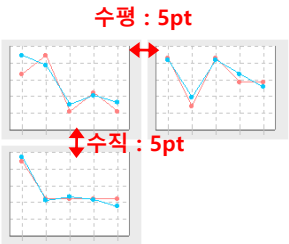
- 묶은 막대

- 누적 막대(영역, 계단)

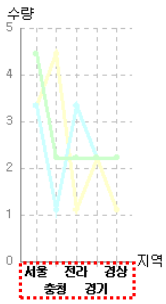
- 곡선



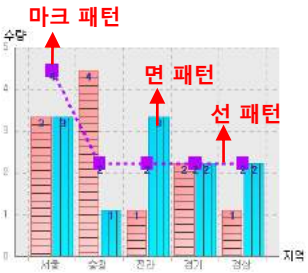
- 차트간 간격



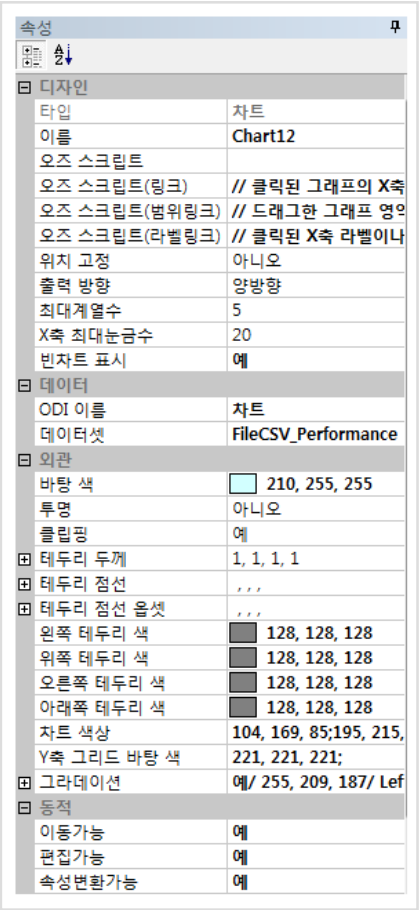
- 수평 축 라벨 2단 표시



- 패턴 적용



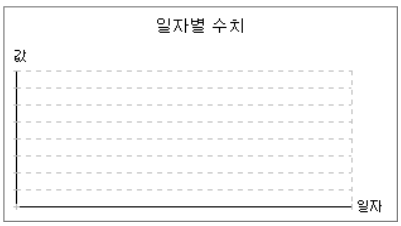
라. 차트 속성



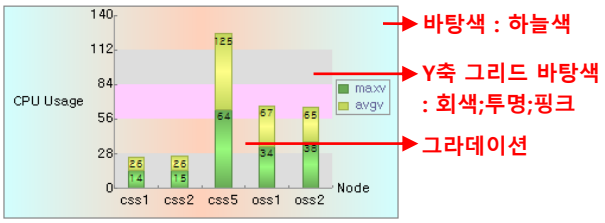
속성	설명
오즈 스크립트(링크)	그래프(바, 라인, 점 등)를 클릭할 때 실행되는 스크립트
오즈 스크립트(범위링크)	그래프를 드래그해서 범위를 선택할 때 실행되는 스크립트
오즈 스크립트(라벨링크)	X축 항목, 범례 라벨을 클릭할 때 실행되는 스크립트
출력 방향	차트가 여러 개 출력될 때 표시할 방향
최대계열수	차트에 표시할 계열의 최대 개수 (계열 수가 최대계열수를 초과하면 자동으로 차트 수가 증가됨)
X축 최대눈금수	X축의 최대 눈금수 (X축 눈금수가 최대 눈금수를 초과하면 자동으로 차트 수가 증가됨)
빈차트 표시	차트 데이터 셋의 데이터가 없을 때 차트의 윤곽 표시 (단, 데이터 밴드 '반복 횟수'를 0보다 큰 값으로 설정해야 함)
ODI 이름	차트에 보여줄 데이터 셋이 있는 ODI 이름
데이터셋	차트에 보여줄 데이터 셋 이름 (차트가 올라간 데이터 밴드의 데이터 셋과 다른 경우 데이터 밴드에 설정된 데이터 셋의 레코드 수만큼 차트가 반복 출력됨)
바탕 색	차트의 바탕색
투명	차트의 바탕색 투명 여부
Y축 그리드 바탕 색	차트 그래프 영역의 바탕색을 ';' 로 구분하여 여러 색상 설정
그라데이션	그라데이션 설정
이동가능	미리보기시에 차트의 이동 및 크기 조정 가능 여부
편집가능	미리보기시에 차트의 내용 수정 가능 여부
속성변환가능	미리보기시에 속성 변환 및 인쇄 가능 여부

➢ 차트 속성 사용 예

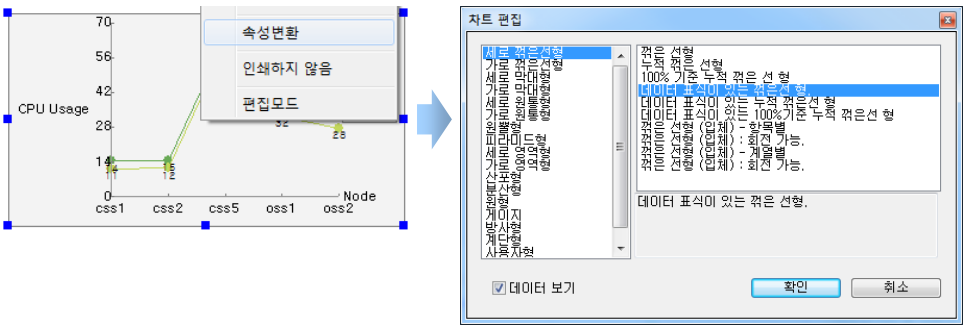
- 빈 차트 표시



- 바탕색, Y축 그리드 바탕색, 그라데이션



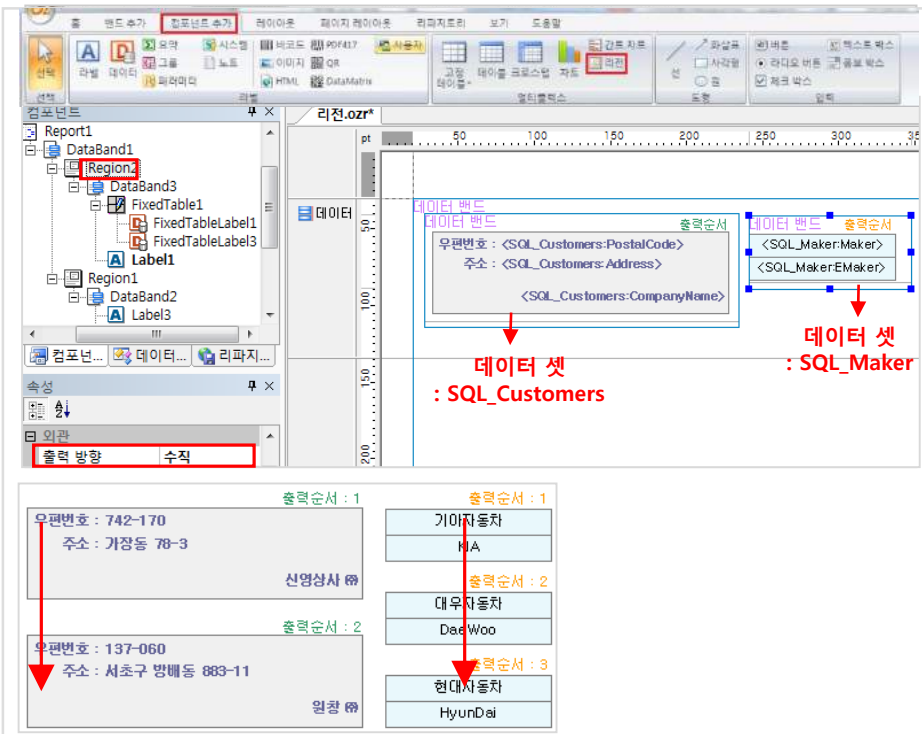
- 속성변환가능



가. 리전 정의

리전은 리포트와 같이 밴드를 위에 그릴 수 있는 컴포넌트 입니다. 하나의 데이터 밴드에서 두 개 이상의 데이터 셋을 사용해야 할 경우나 다단 보고서 작성이 필요한 경우 유용하게 활용할 수 있습니다.
리전 위에 서브 리포트 밴드를 제외한 모든 밴드를 그릴 수 있습니다.

나. 리전 구현 방법



① 데이터 밴드 추가 후 'ODI 이름', '데이터셋' 속성 설정

② 툴바의 '리전' 아이콘 클릭 후 데이터 밴드 위에 대각선으로 그림

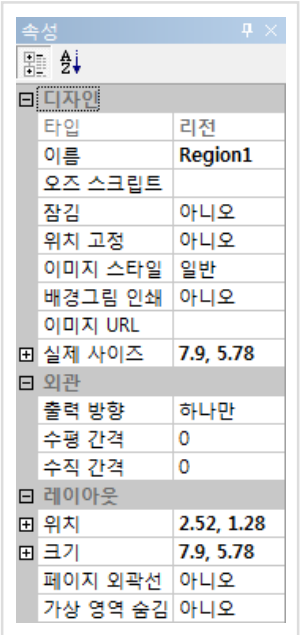
③ 리전 위에 '데이터 밴드' 추가 후 라벨 올려서 디자인

④ 리전의 '출력 방향' 속성을 '수직'으로 설정

⑤ 동일한 방법으로 데이터 밴드 위에 리전 하나 더 추가

⑥ 미리보기하여 리전 영역이 용지 수직 방향으로 반복되는 것 확인

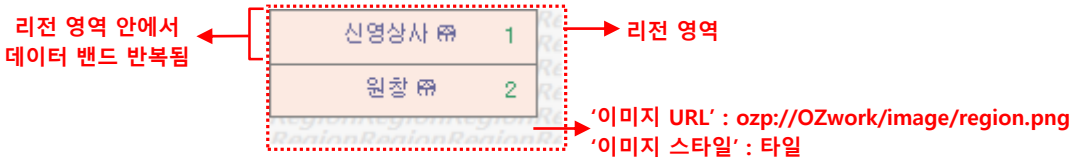
다. 리전 속성



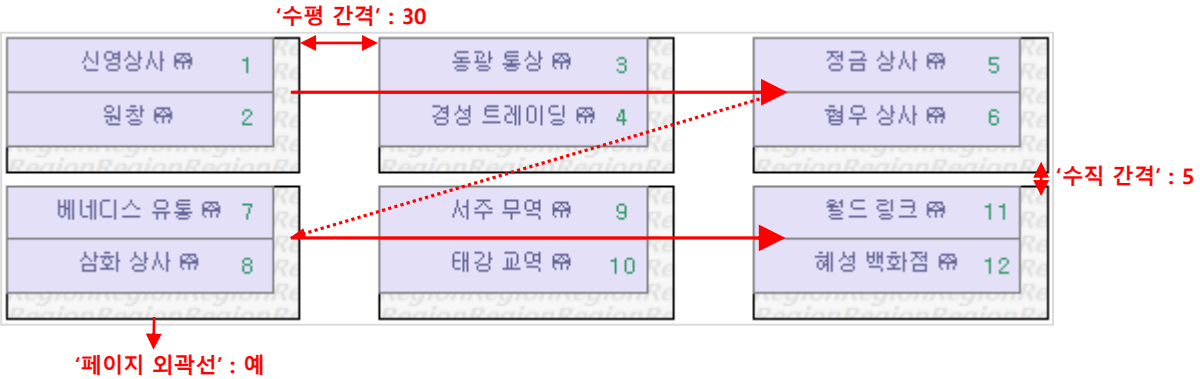
속성	설명
이미지 스타일	리전의 배경 이미지 스타일 설정 (이미지 라벨의 속성과 동일)
배경그림 인쇄	인쇄나 파일 저장할 때 리전의 배경 이미지 표시 여부
이미지 URL	리전의 배경 이미지 파일 경로 (http://, ozp:// 로 설정 가능)
실제 사이즈	보고서 출력될 때 리전이 반복되는 실제 크기
출력 방향	리전 내의 출력할 내용이 리전 실제 크기를 넘을 경우 리전의 반복 출력되는 방향 ('하나만', '수평', '수직', '양방향', '제한된 수평', '제한된 수직', '페이지 높이에 맞춤' 중에 선택)
수평 간격	리전이 반복 출력될 때 리전 간의 수평 간격
수직 간격	리전이 반복 출력될 때 리전 간의 수직 간격
크기	리전의 디자인된 크기
페이지 외곽선	리전 외곽선 표시 여부
가상 영역 숨김	리전의 '실제 사이즈' 보다 '크기'가 클 때 '실제 사이즈' 부분만 디자인 시에 보이게 함

➢ **리전 속성 사용 예**

- '출력 방향' 속성이 '하나만'인 경우 출력 내용이 리전 내에 모두 표현이 안되더라도 더 이상 출력하지 않습니다.

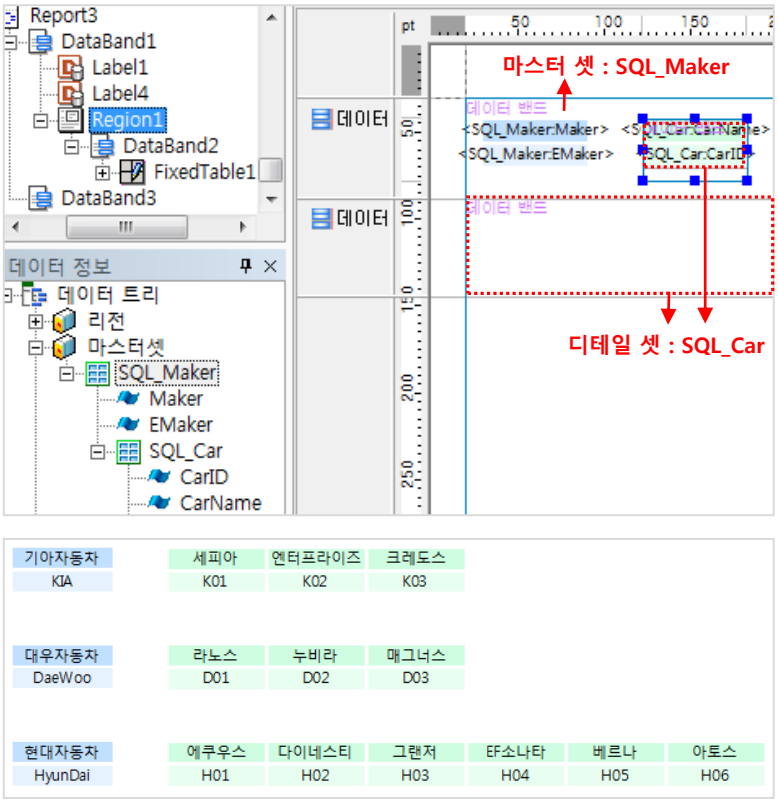


- '출력 방향' 속성이 '제한된 수평'인 경우 수평 방향으로 리전 영역이 반복 출력되고 용지 끝에서 아래로 내려가 다시 수평 방향으로 출력됩니다.



라. 리전 위에 디테일 밴드 구현 방법

동일한 데이터 밴드 위에 마스터 셋과 디테일 셋을 나란히 출력할 때 리전을 이용하여 디테일 셋을 표현합니다.



① 데이터 밴드 추가 후 마스터 셋 설정하고 데이터 필드 올려서 디자인(이하 DataBand1)

② DataBand1 위에 리전 추가 후 '출력 방향' 속성을 '수평'으로 변경

③ 리전 위에 데이터 밴드 추가 후 디테일 셋으로 설정 (이하DataBand2)

④ DataBand2는 리전 위에 있어서 마스터 밴드를 DataBand1로 설정할 수 없기 때문에 리포트 위에 데이터 밴드(DataBand3)를 추가한 후 DataBand2와 동일한 디테일 셋으로 설정하면 DataBand1과 Databand2의 마스터-디테일 관계가 성립됨

⑤ 임의로 추가한 DataBand3의 '감추기' 속성을 '예'로 설정하면 출력 안됨

⑥ 미리보기 하여 마스터-디테일 데이터가 나란히 출력되는 것 확인

가. 스크립트 용도

스크립트를 작성하여 보고서를 바인딩할 때 데이터 값에 따라 컴포넌트의 속성을 제어하고 데이터 연산 등을 통해 보다 유연하고 강력한 보고서를 만들 수 있습니다. 모든 컴포넌트에 스크립트를 작성할 수 있습니다.

나. 스크립트 종류

리포트 디자이너는 쿼리 디자이너와 마찬가지로 '오즈 스크립트', '자바 스크립트' 두 가지 스크립트 언어를 지원합니다. '오즈 스크립트'는 오즈에서 제공하는 자체 스크립트 언어입니다.

다. 오즈 스크립트

■ 문법

- 자료형 : int, double, string, Date
- 함수 : 수학 함수, 문자 함수, 날짜 함수, 데이터 셋 접근 함수 등 제공
- 변수 선언 및 초기화 예 : int i; i = 10; (주의 : 선언문에서 초기값을 지정하면 에러 발생)
- 연산자 : +, -, *, /, %, =, ==, !=, <, <=, >, >=, &, |, ^, ~, &&, ||, !
- 제어문 : if-else문, while 문
- 데이터 필드 정의 : #데이터셋명.필드명# 또는 #필드명#
- 사용자 지정 패러미터, 폼 패러미터 정의 : #OZParam.패러미터명#, #OZFormParam.패러미터명#

■ 전역 변수 사용

```
global double pagesum; // 전역 변수 선언
::pagesum = 0; // 전역 변수 초기화
```

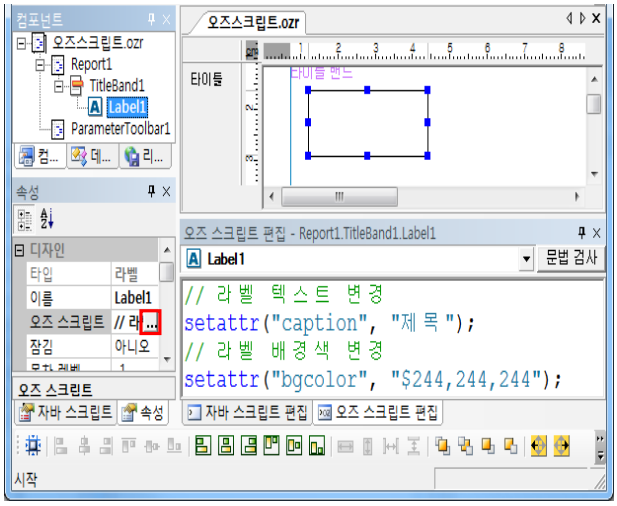
■ 지역 변수 사용

```
double pagesum; // 지역 변수 선언
pagesum = 0; // 지역 변수 초기화
```

■ 실행 순서

- 실행 순서 : 리포트템플릿 → 리포트 → 백그라운드밴드와 컴포넌트 → 디자인밴드와 컴포넌트 → 포그라운드밴드와 컴포넌트
- 밴드 위의 컴포넌트 실행 순서 : '컴포넌트' 창의 트리에 추가된 순서와 동일하게 실행됨
- 동일 컴포넌트에 대해 오즈 스크립트가 자바 스크립트보다 먼저 실행됨

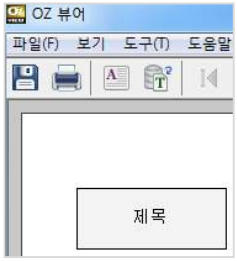
■ 작성 방법



① 컴포넌트 속성 창에서 '오즈 스크립트'의 ... 버튼을 클릭하면 '오즈 스크립트 편집' 창으로 포커스 이동됨

② '오즈 스크립트 편집' 창에 스크립트 작성

③ 미리보기 후 스크립트 반영된 것 확인



※ 자세한 내용은 리포트 디자이너의 '도움말' 탭 → '오즈 리포트 디자이너 도움말' 메뉴 → '스크립트 API' → '오즈 스크립트' 메뉴를 참고하시기 바랍니다.

라. 자바 스크립트

■ 문법

- 자료형 : var 로 선언하며 변수의 내용에 따라 데이터 형이 결정됨
- 함수 : 자바 스크립트에서 제공하는 함수, 리포트 디자이너에서 제공하는 함수, 사용자 정의 함수
- 연산자 : +, -, *, /, %, =, ==, !=, <, <=, >, >=, &, |, ^, ~, &&, ||, !
- 제어문 : if-else문, while 문, do while 문, for 문
- 예약어 This : 현재 컴포넌트
- 기타 : ReportTemplate, Global 객체의 함수 사용시 객체명을 생략할 수 있고 모든 컴포넌트에서 참조 가능

■ 전역 변수 사용

```
SetGlobal("pagesum", 0); // 전역 변수 초기화
This.SetText( GetGlobal("pagesum") ); // 전역 변수 사용
```

■ 지역 변수 사용

```
var pagesum = 0; // 지역 변수 선언 및 초기화
```

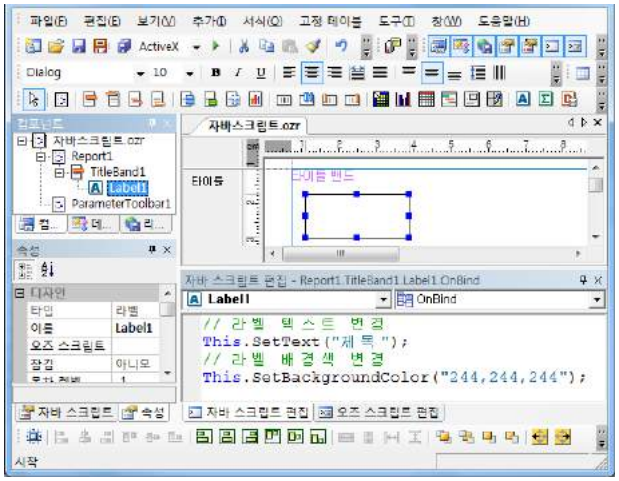
■ 주요 이벤트

- viewer.external_functions_path 패러미터로 공통 함수, 변수를 외부 텍스트 파일로 지정
- Functions : 리포트 템플릿 Functions에 정의한 함수는 해당 보고서에 추가된 모든 컴포넌트에서 호출 가능
- OnStartUp : 데이터 바인딩 직전에 발생하며 주로 뷰어 패러미터를 설정하는 스크립트 입력 용도로 사용됨
- OnInitialize : 컴포넌트 초기화 직후에 발생하는 이벤트
- OnBind : 컴포넌트가 바인딩 될 때 발생하는 이벤트로 컴포넌트 속성을 변경할 때 사용
- OnEndBind : 컴포넌트의 바인딩이 완료된 직후에 발생하며 바인딩된 정보를 가져오는 용도에 사용
- OnPostChartBuild : 차트의 그래프가 만들어진 후 발생하며 차트의 속성을 변경할 때 사용

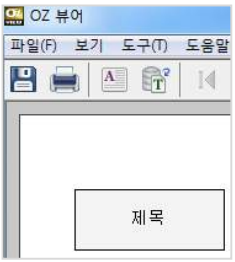
■ 실행 순서

① OnStartUp (리포트템플릿)
② OnInitialize (리포트템플릿 → 리포트 → 디자인밴드와 컴포넌트 → 백그라운드밴드 → 포그라운드밴드)
③ OnBind (리포트템플릿 → 리포트 → 백그라운드밴드 → 디자인밴드와 컴포넌트 → 포그라운드밴드)
④ OnEndBind (백그라운드밴드 → 디자인밴드와 컴포넌트 → 포그라운드밴드)

■ 작성 방법

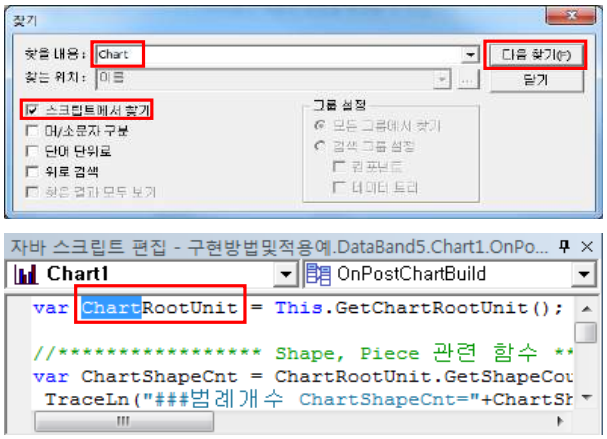


① 컴포넌트 선택 후 '자바 스크립트 편집' 창으로 이동
② 컴포넌트의 이벤트 선택 (예: OnBind)
③ 스크립트 작성
④ 미리보기 후 스크립트 반영된 것 확인



※ 자세한 내용은 리포트 디자이너의 '도움말' 탭 → '오즈 리포트 디자이너 도움말' 메뉴 → '스크립트 API' → '자바 스크립트' 메뉴를 참고하시기 바랍니다.

마. 전체 스크립트 검색



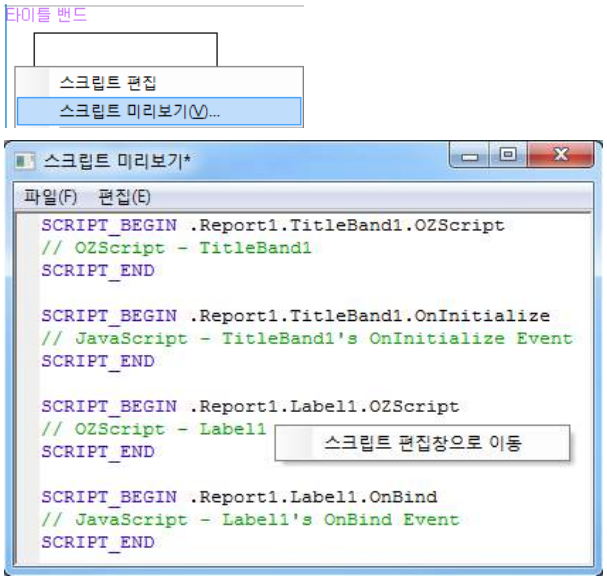
① 디자인 화면을 선택한 후 '편집' → '찾기' 메뉴 선택 (단축키 Ctrl + F)

② '찾기' 창에서 '스크립트에서 찾기' 선택

③ '찾을 내용' 입력 후 '다음 찾기' 버튼 클릭

④ '오즈 스크립트 편집' 창과 '자바 스크립트 편집' 창에서 일치하는 내용 검색됨

바. 스크립트 미리보기



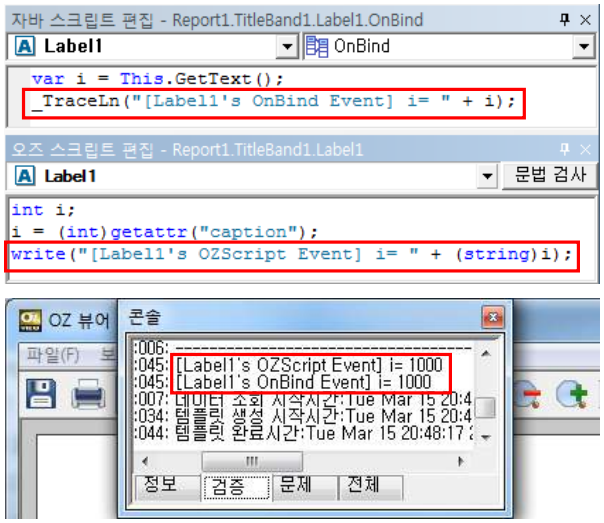
① 컴포넌트를 선택한 후 팝업 메뉴의 '스크립트 미리보기' 선택

② '스크립트 미리보기' 창에 현재 컴포넌트에 종속된 컴포넌트까지 오즈 스크립트와 자바 스크립트가 미리보기 됨

③ 'SCRIPT_BEGIN'~'SCRIPT_END' 절 사이를 클릭한 후 팝업 메뉴의 '스크립트 편집창으로 이동'을 선택하면 해당 스크립트 창으로 이동됨

④ '스크립트 미리보기' 창에서 수정된 내용은 실제 스크립트에 적용됨

사. 디버깅



① 디버깅할 위치에 오즈 스크립트의 write 함수나 자바 스크립트의 _TraceLn 함수 이용

② 미리보기 후 뷰어 콘솔을 'Ctrl + Z' 키로 열고 '검증' 탭으로 이동

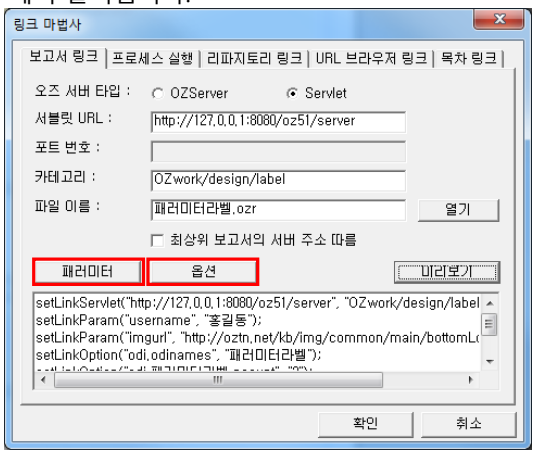
③ 함수의 패러미터로 넘긴 문자열 확인

④ 스크립트 에러는 '문제' 탭에서 확인 가능

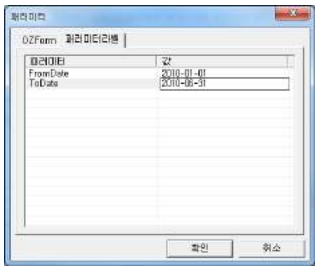
아. 링크

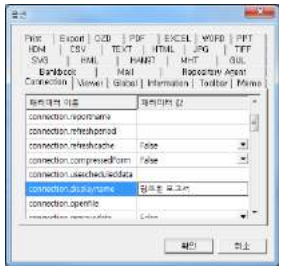
컴포넌트를 클릭했을 때 다른 보고서나 웹 페이지, 프로세스 등을 실행하는 기능을 스크립트로 작성할 수 있습니다. '오즈 스크립트 편집' 창의 팝업 메뉴에서 '링크 마법사'를 이용하거나 스크립트 편집 창에 직접 작성합니다.

- '**보고서 링크**'는 설정한 보고서가 동일 뷰어에서 출력됩니다.

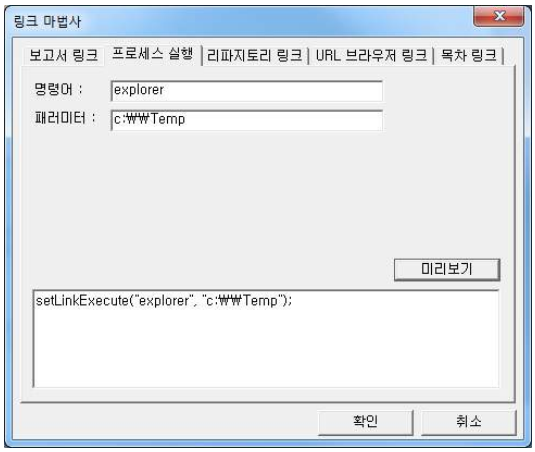


- '**패러미터**' 버튼과 '**옵션**' 버튼을 클릭하여 폼 패러미터, 사용자 지정 패러미터, 뷰어 옵션을 설정합니다.

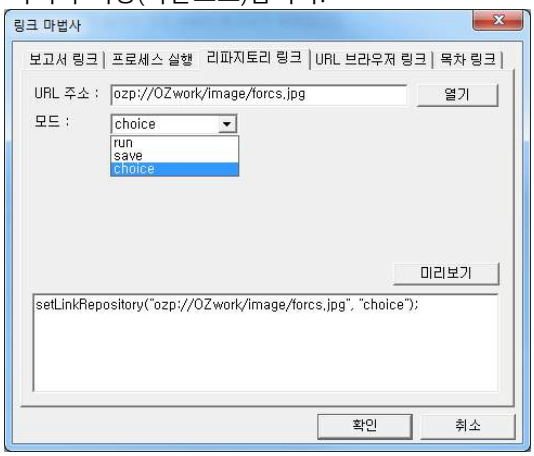




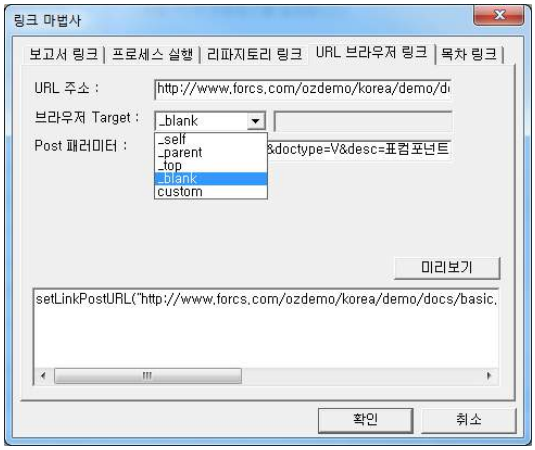
- '**프로세스 실행**'은 로컬 PC의 프로세스를 실행합니다.



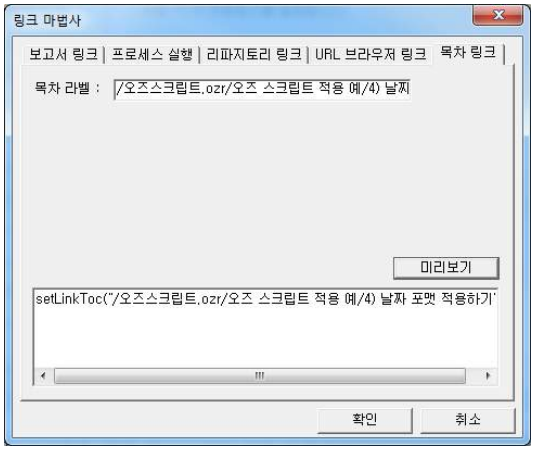
- '**리파지토리 링크**'는 오즈 서버의 리파지토리에 있는 파일을 로컬 PC의 연결 프로그램으로 실행하거나 저장(다운로드)합니다.



- '**URL 브라우저 링크**'는 웹 브라우저로 설정한 URL을 실행합니다. 웹 브라우저 Target과 Post 방식으로 넘기는 패러미터를 설정할 수 있습니다.



- '**목차 링크**'는 목차를 설정된 라벨로 이동됩니다. 패러미터로 링크할 목차 라벨의 경로를 설정합니다.



가. 오즈 스크립트 적용 예

➢ 데이터 리스트에 순번 보여주기

```
SCRIPT_BEGIN .Report1.FixedTableLabel1.OZScript
  int index; // 변수 선언
  index = index + 1; // 변수 값 증가
  setattr("caption", (string)index); // 라벨 텍스트 변경
SCRIPT_END
```

➢ 페이지 마다 5건씩 출력하기

```
SCRIPT_BEGIN .Report1.DataBand2.OZScript
  int idx;
  idx = currentRowIndex("CarSales"); // 현재 행 인덱스 변수에 설정
  write("row idx = " + (string)idx); // 디버깅을 위해 뷰어 콘솔에 행 인덱스 남김
  if (idx % 5 == 0) { // 행 인덱스를 5로 나눈 나머지 값이 0인 경우
    setattr("forcenewpage", "true"); // 강제로 페이지 넘기기
  }
SCRIPT_END
```

➢ 날짜 포맷 적용하기

```
SCRIPT_BEGIN .Report1.Label23.OZScript
  string currData, newText;
  currData = #CarSales.OrderDate#; // 데이터 필드 값을 변수에 설정 '2011-03-25'
  newText = formatDate( parseDate(currData, "yyyy-MM-dd"), "yyyy/MM/dd"); // Date타입 변환, 포맷 설정
  setattr("caption", newText); // 라벨 텍스트 변경 '2011/03/25'
SCRIPT_END
```

➢ 숫자 포맷 적용하기

```
SCRIPT_BEGIN .Report1.Label44.OZScript
  double oldvalue;
  string newvalue;
  oldvalue = (double)getattr("caption"); // 현재 라벨의 텍스트를 변수에 설정
  newvalue = formatNumber(oldvalue, "#,###;-#,###"); // 숫자 값을 포맷 설정한 텍스트로 변환
  setattr("caption", newvalue); // 라벨 텍스트 변경
SCRIPT_END
```

➢ 데이터에 따라 컴포넌트 속성 변경하기

```
SCRIPT_BEGIN .Report1.Table1.TableValue1.OZScript
  // Amount 필드 값이 70,000,000 이상인 경우 빨간색 글자로 표시
  if (#CarSales.Amount# > 70000000) {
    setattr("fontcolor", "$255,0,0");
  }
  // 수평, 수직 정렬 설정
  setattr("halign", "Right");
  setattr("valign", "Center");
SCRIPT_END
```

나. 자바 스크립트 적용 예

➢ 데이터 리스트에 순번 보여주기

```
SCRIPT_BEGIN .Report1.FixedTableLabel1.OnBind
  var index = 1;  // 임시 변수 선언
  if (GetGlobal("row_index")!= null) index = GetGlobal("row_index"); // 전역 변수가 존재하는 경우
  This.SetText(index); // 라벨 텍스트 변경
  SetGlobal("row_index", (index + 1)); // 전역 변수값 1 증가
SCRIPT_END
```

➢ 페이지 마다 5건씩 출력하기

```
SCRIPT_BEGIN .Report1.DataBand2.OnBind
  if (This.GetDataSetRowIndex() % 5 == 0) { // 행 인덱스를 5로 나눈 나머지 값이 0인 경우
    This.SetForceNewPage(true); // 강제로 페이지 넘기기
  }
SCRIPT_END
```

➢ 날짜 포맷 적용하기

```
SCRIPT_BEGIN .Report1.Label23.OnBind
  var currData = This.GetDataSetValue("CarSales.OrderDate"); // 데이터 필드 값을 변수에 설정 '2011-03-25'
  var newText = _FormatDate( _ParseDate(currData, "yyyy-MM-dd"), "yyyy/MM/dd"); // 문자열을 날짜값
(double)로 변환한 후 포맷 설정
  This.SetText(newText); // 라벨 텍스트 변경 '2011/03/25'
SCRIPT_END
```

➢ 숫자 포맷 적용하기

```
SCRIPT_BEGIN .Report1.Label44.OnBind
  This.SetFormat("#,###;-#,###"); // 현재 라벨의 텍스트에 숫자 포맷 적용
SCRIPT_END
```

➢ 데이터에 따라 컴포넌트 속성 변경하기

```
SCRIPT_BEGIN .Report1.Table1.TableValue1.OnBind
  // Amount 필드 값이 70,000,000 이상인 경우 빨간색 글자로 표시
  if (This.GetDataSetValue("CarSales.Amount") > 70000000) {
    This.SetTextColor("255,0,0");
  }
SCRIPT_END
```

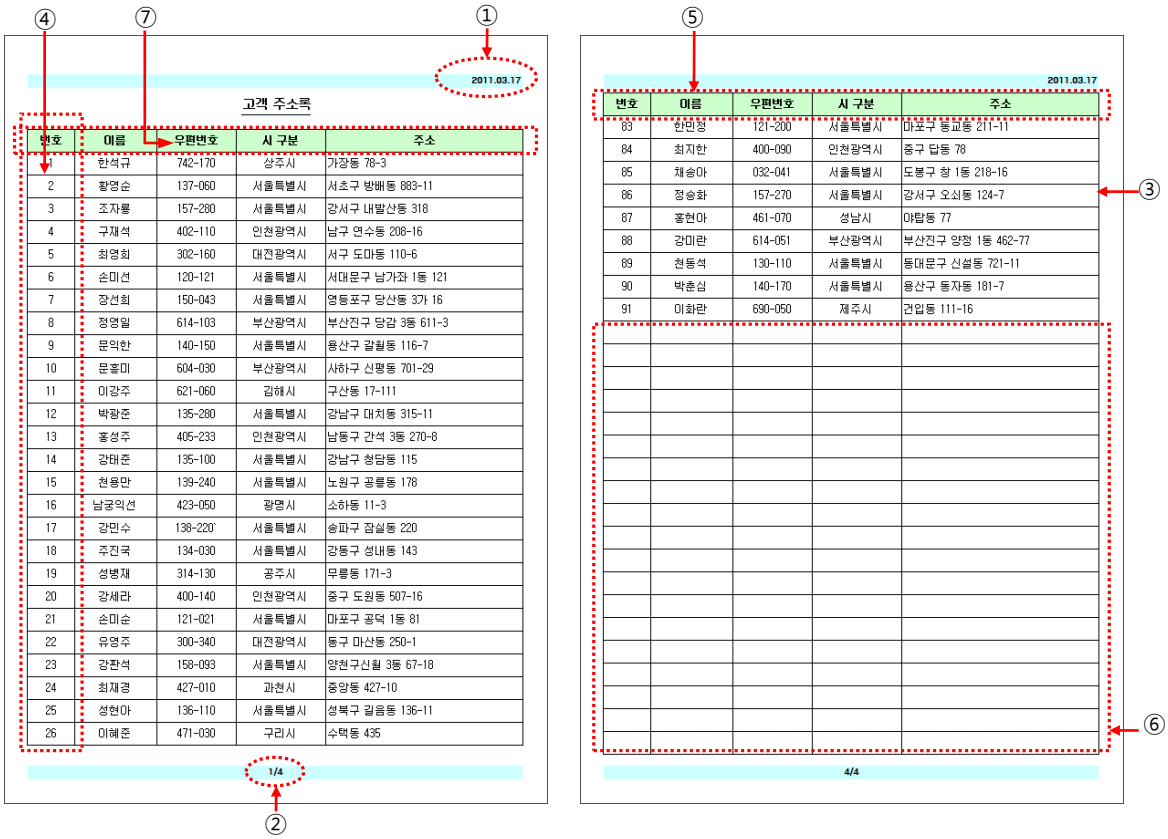
➢ 차트 속성 변경하기

```
SCRIPT_BEGIN .Report1.Chart1.OnPostChartBuild
  var ChartRootUnit = This.GetChartRootUnit();
  var XAxisLabel = ChartRootUnit.GetXAxisLabel(0); // X축 첫번째 라벨
  if (XAxisLabel.GetNodeName()=="서울") { // 라벨 텍스트가 '서울' 일 때 텍스트 색 변경, BOLD 표시
    XAxisLabel.SetTextColor("255,0,0");
    XAxisLabel.SetFontBold(true);
  }
SCRIPT_END
```

Chapter5. 보고서 유형별 실습

1. 리스트 보고서
2. 리스트 보고서(변형)
3. 고정 폼 보고서
4. 다단 보고서
5. 통계표 보고서(수직)
6. 통계표 보고서(복합)
7. 차트 보고서(막대)
8. 차트 보고서(사용자)
9. 차트 보고서(누적)
10. 드릴다운 보고서
11. 그룹 보고서
12. 마스터디테일 보고서

가. 목표 보고서 : 고객 주소록



나. 과제

① 페이지 마다 상단 우측에 '현재 날짜'를 'date_yyyy.MM.dd' 형식으로 표시 (시스템 라벨, 라벨 서식 이용)
② 페이지 마다 하단 중앙에 '현재 페이지 번호/전체 페이지 번호' 표시 (시스템 라벨 이용)
③ 고객 주소 정보를 테이블로 출력 (테이블 마법사 이용)
④ 테이블 첫 필드에 일련번호 표시 (테이블 마법사 옵션 이용)
⑤ 페이지 마다 테이블 헤더 표시 (테이블 마법사 옵션 이용)
⑥ 마지막 페이지 여백을 빈칸으로 채우기 (테이블 마법사 옵션 이용)
⑦ 테이블 헤더의 라벨 텍스트를 한글로 수정

다. 관련 정보

■ 쿼리문

```
SELECT ContactName, PostalCode, City, Address FROM Customers
```

■ 데이터

ContactName	PostalCode	City	Address
한석규	742-170	상주시	가장동 78-3
황영순	137-060	서울특별시	서초구 방배동 883-11
조자룡	157-280	서울특별시	강서구 내발산동 318
구재석	402-110	인천광역시	남구 연수동 208-16

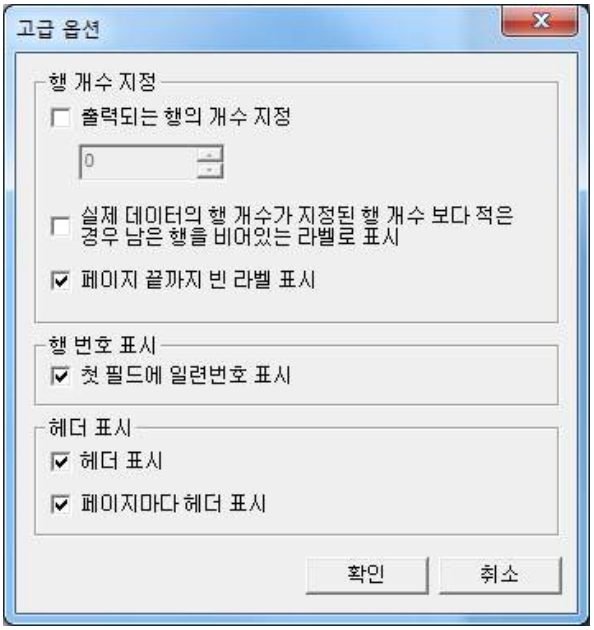
■ 보고서 파일 경로

· 내 문서\OZRepository\OZwork\exercise\ex1\고객주소록.odi, 고객주소록.ozr

(1) 시스템 필드 이름

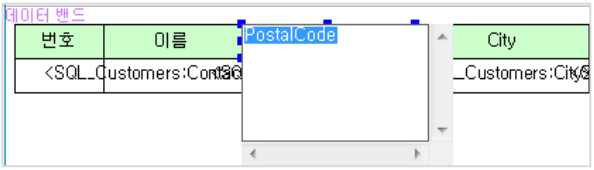
- Date : 현재 날짜
- Page_Of_Total : 현재 페이지 번호/전체 페이지 번호

(2) 테이블 마법사 고급 옵션



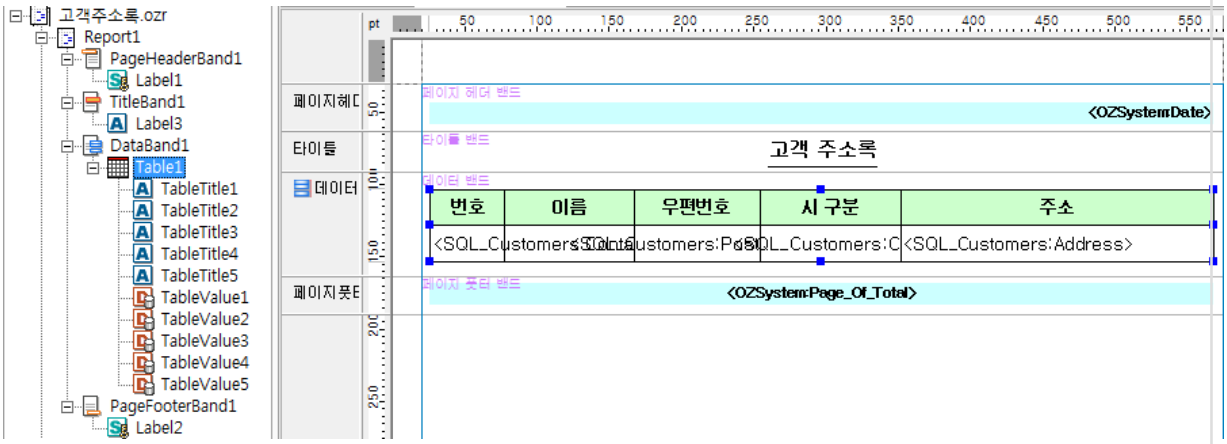
• 테이블 마법사의 '옵션' 버튼을 클릭하여 '고급 옵션' 창 띄운 후 필요한 옵션 선택

(3) 라벨 텍스트 편집



• 라벨의 텍스트를 편집하려면 라벨 속성 창에서 '텍스트' 값을 변경하거나 라벨을 선택한 후 '더블 클릭' 또는 'F2' 키를 눌러 텍스트 편집 창에서 수정할 수 있습니다.

(4) 디자인 화면



가. 목표 보고서 : 상품 재고 현황

분류	상품명	포장단위	단가	수량
가공 식품	대양 특선 건과(배)	12 - 1 lb pkgs.	30,000	15
	서울 구이 김	5 kg pkg.	10,000	4
	알파 특선 튀김 다시마	25 - 825 g cans	15,000	26
	유림 사과 통조림	50 - 300 g pkgs.	53,000	20
	유미 건조 다시마	40 - 100 g pkgs.	23,000	35
곡류	신성 시리얼	24 - 500 g pkgs.	21,000	104
	신성 쌀 튀김 과자	12 - 250 g pkgs.	9,000	61
	싱가폴 원산 옥수수	32 - 1 kg pkgs.	14,000	26
	알파 콘 플레이크	20 bags x 4 pieces	33,000	22
	필로 믹스	16 - 2 kg boxes	7,000	38
	한성 옥수수 가루	24 - 250 g pkgs.	38,000	21
	한성 통밀가루	24 - 250 g pkgs.	19,000	36
과자류	대양 마말레이드	30 gift boxes	81,000	40
	대양 핫 케이크 소스	24 pkgs. x 4 pieces	10,000	3
	미미 스카치 캔디	10 boxes x 8 pieces	12,000	6
	미황 계피 캔디	24 - 50 g pkgs.	20,000	10
	미황 코코아 샌드	12 - 100 g bars	16,000	65
	사계절 커스터드 파이	48 pies	49,000	17
	신한 초콜릿 소스	10 boxes x 12 pieces	19,000	25
	우미 코코넛 쿠키	100 - 250 g bags	31,000	15

분류	상품명	포장단위	단가	수량
조미료	루이지애나 특산 후추	32 - 8 oz bottles	21,000	76
	사계절 핫 소스	24 - 500 ml bottles	28,000	113
	신한 100% 복숭아 시럽	48 - 6 oz jars	22,000	53
	신한 100% 파인애플 시럽	36 boxes	21,000	0
	알파 샐러드 드레싱	12 boxes	13,000	32
	유미 간장	24 - 250 ml bottles	15,000	39
	유미 멸치 가루	24 - 8 oz jars	17,000	4
	태양 체리 시럽	12 - 550 ml bottles	10,000	13
해산물	노르웨이산 연어알 조림	12 - 200 ml jars	31,000	31
	대륙 냉동 참치	16 kg pkg.	62,000	42
	대성 어묵	24 pieces	13,000	62
	버뮤다 포장 문어	1k pkg.	9,000	5
	버뮤다 포장 참치	4 - 450 g glasses	12,000	95
	보스톤산 게살 통조림	24 - 4 oz tins	18,000	123
	원양 순 상어알	24 - 150 g jars	35,000	101
	유미 물김	2 kg box	16,000	24
	태평양 포장 파래	10 - 200 g glasses	25,000	10
	특제 꿀 통조림	24 - 250 g jars	19,000	112
	포장 건 오징어	12 - 500 g pkgs.	26,000	11
	훈제 대합조개 통조림	12 - 12 oz cans	9,000	85

나. 과제

① 상품 정보를 테이블로 출력 (테이블 마법사 이용)
② 테이블의 'CategoryName' 필드에 대해 연속적으로 같은 값을 병합하여 표시 (테이블 값의 '그룹핑' 속성 이용)
③ 테이블의 짝수 행 마다 배경색 표시 (자바 스크립트 이용)
④ 보고서 마지막에 '수량 합계' 표시 (요약 라벨 이용)
⑤ 테이블과 수량 합계의 숫자 데이터 값에 3 자리 마다 ',' 로 표시하고 오른쪽 정렬 (라벨 서식 이용)

다. 관련 정보

■ 쿼리문

```
SELECT CategoryName, ProductName, QuantityPerUnit, UnitPrice, UnitsInStock
FROM Categories, Products
WHERE Categories.CategoryID = Products.CategoryID
ORDER BY CategoryName ASC, ProductName ASC
```

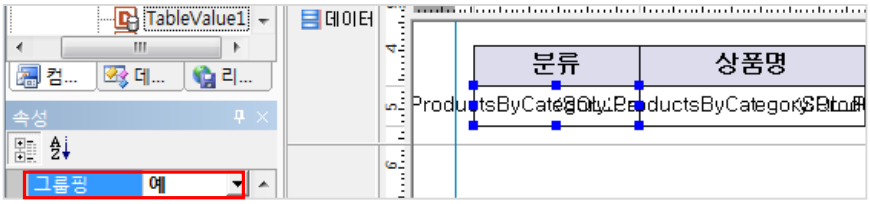
■ 데이터

CategoryName	ProductName	QuantityPerUnit	UnitPrice	UnitsInStock
가공 식품	대양 특선 건과(배)	12 - 1 lb pkgs.	30000.0000	15
가공 식품	서울 구이 김	5 kg pkg.	10000.0000	4
가공 식품	알파 특선 튀김 다시마	25 - 825 g cans	15000.0000	26
가공 식품	유림 사과 통조림	50 - 300 g pkgs.	53000.0000	20
가공 식품	유미 건조 다시마	40 - 100 g pkgs.	23000.0000	35
곡류	신성 시리얼	24 - 500 g pkgs.	21000.0000	104
곡류	신성 쌀 튀김 과자	12 - 250 g pkgs.	9000.0000	61

■ 보고서 파일 경로

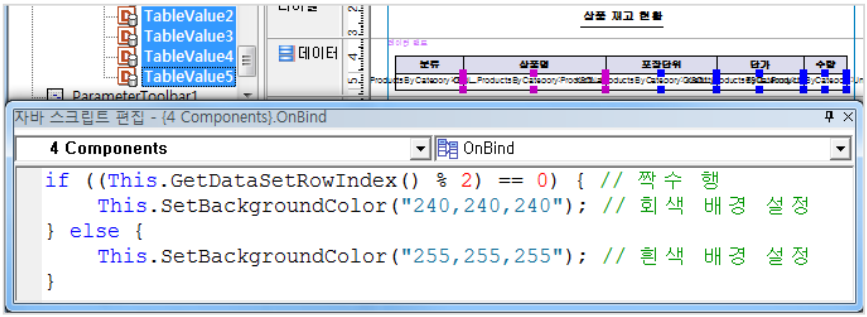
· 내 문서\OZRepository\OZwork\exercise\ex2\상품재고현황.odi, 상품재고현황.ozr

(1) 테이블 값 그룹핑



• 연속되는 같은 데이터를 가진 라벨이 병합되도록 '테이블 값'의 '그룹핑' 속성을 '예'로 설정

(2) 짝수 행마다 배경색 설정하는 스크립트

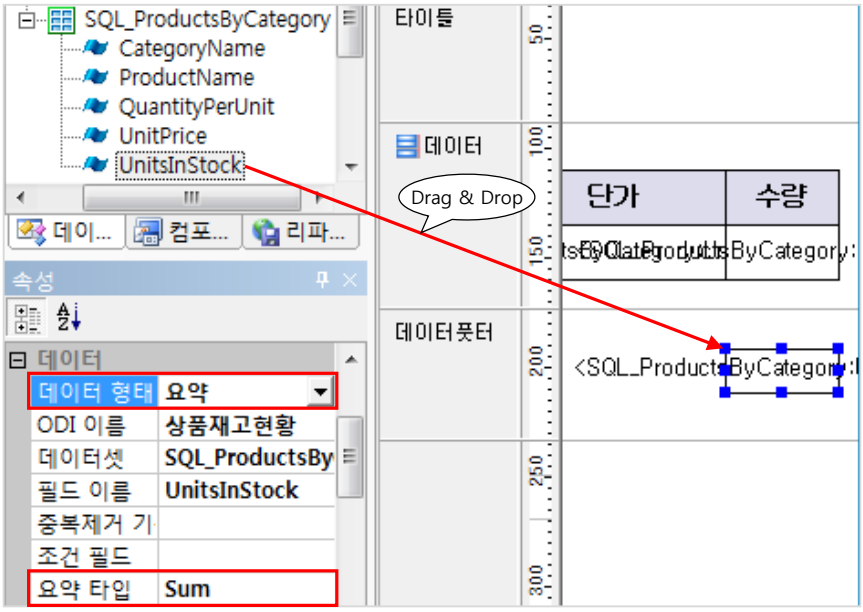


• '컴포넌트' 창에서 스크립트를 작성할 테이블 값 라벨들을 'Ctrl' 또는 'Shift' 키를 누르고 여러 개 선택

• '자바 스크립트 편집' 창의 OnBind 이벤트에 스크립트 작성

(3) 요약 라벨을 이용한 합계 표시

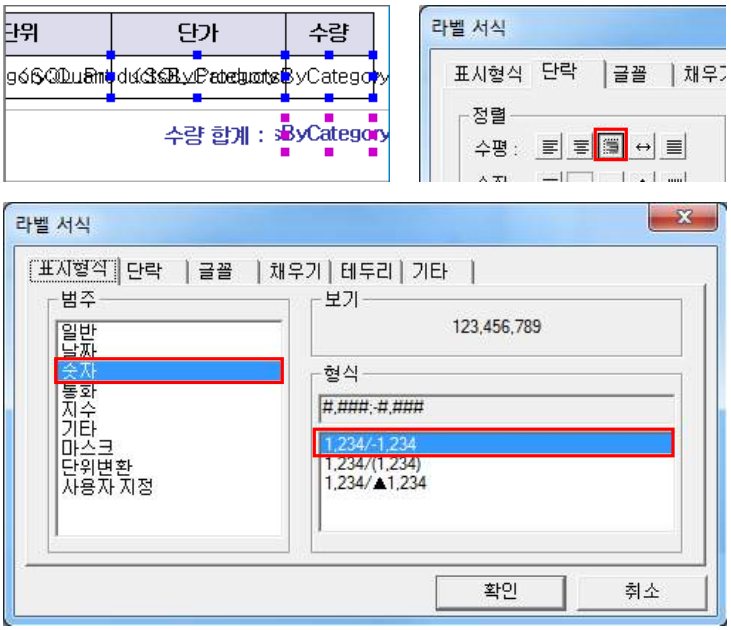
• 데이터 밴드 바로 아래 합계를 보여주기 위해 데이터 밴드의 '풋터밴드 소유'를 '예'로 설정하여 '데이터 풋터 밴드' 추가



• 데이터 풋터 밴드 위에 데이터 필드를 드래그&드롭으로 추가하면 데이터 라벨이 생성됨

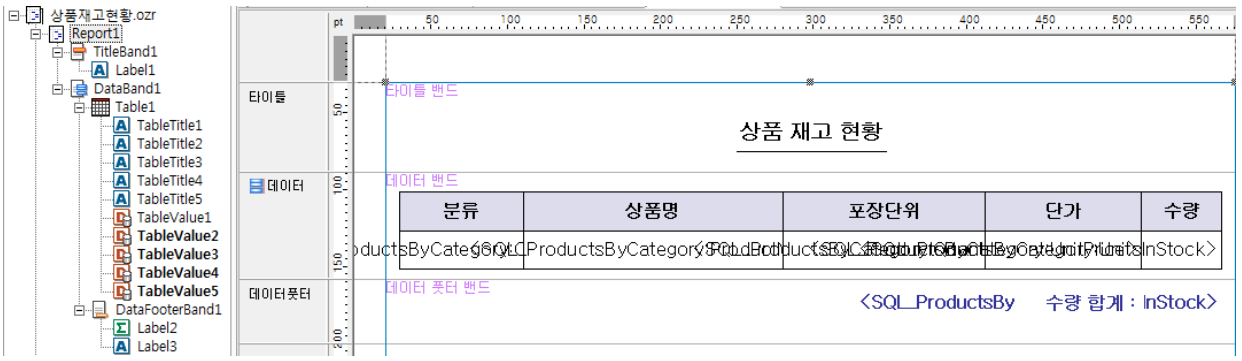
• 데이터 라벨의 '데이터 형태' 속성을 '요약'으로 설정하고, '요약 타입' 속성을 'Sum'으로 설정하면 데이터 라벨에 설정된 필드의 합계가 보여짐

(4) 숫자 표현을 위한 라벨 서식 설정

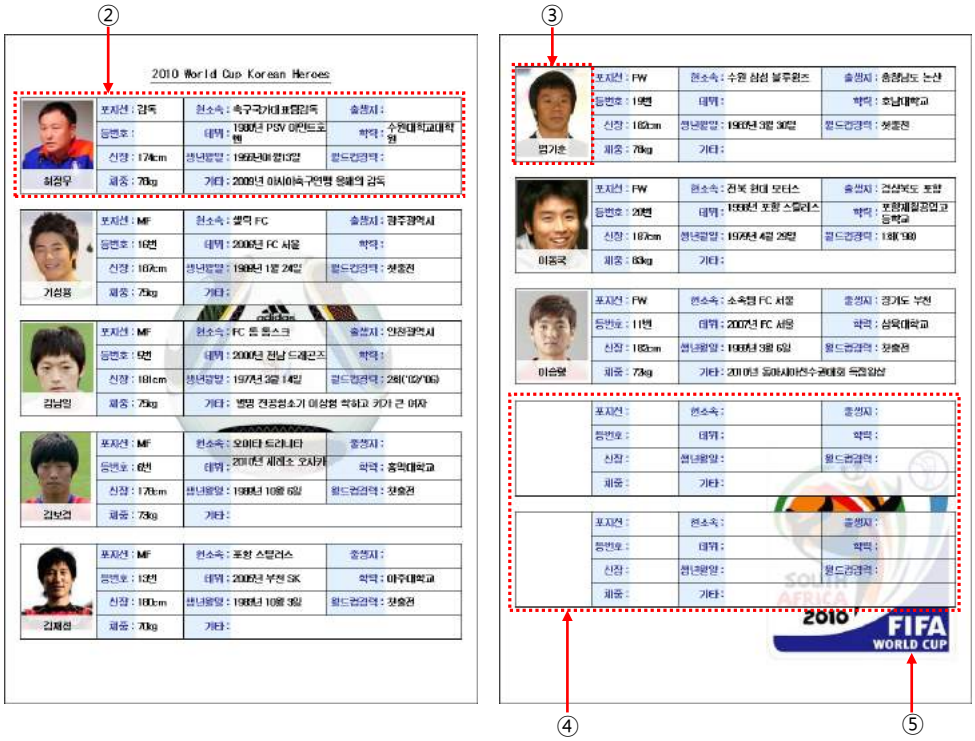


- 숫자를 보여주는 라벨들을 선택한 후 팝업 메뉴의 '라벨 서식' 선택
- '라벨 서식' 창의 '단락' 탭에서 수평 정렬 지정
- '표시형식' 탭에서 숫자 범주를 '#,###;-#,###'로 설정하면 3자리 마다 ',' 표시됨

(5) 디자인 화면



가. 목표 보고서 : 월드컵 국가대표 약력



나. 과제

① 텍스트 파일을 데이터 소스로 사용 (CSV 파일 스토어 이용)

② 국가대표 약력을 보여주는 폼 작성 (고정 테이블 이용)

③ 국가대표 사진 표시 (이미지 라벨 이용)

④ 마지막 페이지 여백을 비워있는 폼으로 채우기 (더미 풋터 밴드 이용)

⑤ 리포트 배경 이미지를 페이지마다 삽입 (백그라운드 모드 이용)

다. 관련 정보

■ 데이터, 이미지 파일 경로

- ozp://OZwork/data/월드컵국가대표약력.txt
- ozp://OZwork/image/*.JPG

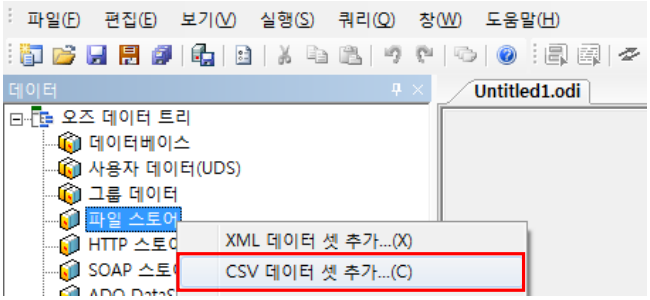
■ 데이터

이름	사진	포지션	등번호	월드컵...	생년월일	출생지	신장	체중	현소속	학력	데뷔	기타
허정무	ozp...	감독			1955년0...		174...	78kg	축구국...	수원...	1980...	200...
기성용	ozp...	MF	16번	첫출전	1989년 ...	광주...	187...	75kg	셀틱 FC		2006...	
김남일	ozp...	MF	5번	2회('02...	1977년 ...	인천...	181...	75kg	FC 톨 ...		2000...	별...
김보경	ozp...	MF	6번	첫출전	1989년 ...		178...	73kg	오이타...	홍익...	2010...	

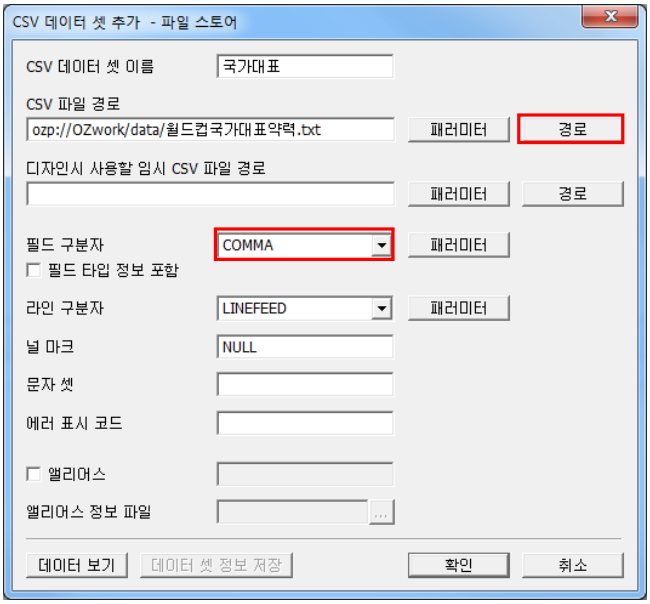
■ 보고서 파일 경로

- 내 문서\OZRepository\OZwork\exercise\ex3\월드컵국가대표약력.odi, 월드컵국가대표약력.ozr

(1) 텍스트 파일 데이터 사용



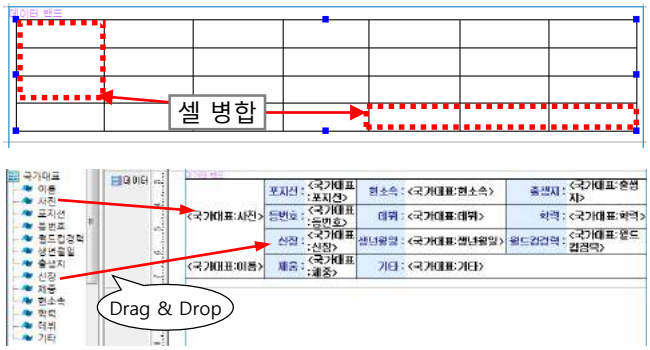
- 보고서에 파일 데이터를 사용하기 위해 쿼리 디자이너를 실행한 후 '파일 스토어'의 팝업 메뉴에서 'CSV 데이터 셋 추가' 선택



- 'CSV 데이터 셋 추가' 창의 '경로' 버튼을 클릭하여 데이터 파일 지정
- '필드 구분자'에 'COMMA' 선택

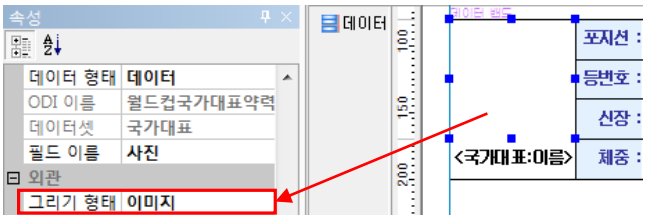
※ 'CSV 파일 경로'는 운영할 때 사용될 파일 경로입니다. 경로 앞부분을 'ozp://' 로 변경하면 오즈 서버 리파지토리를 기준으로 하는 경로가 설정됩니다.

(2) 고정 폼 작성



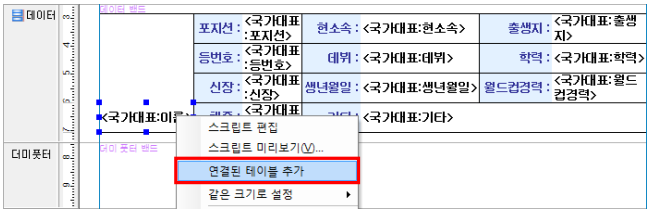
- 고정 테이블 추가 후 셀 병합
- 라벨의 테두리 선, 폰트, 배경색, 텍스트 디자인
- '데이터 정보' 창의 데이터 필드를 라벨로 드래그&드롭

(3) 사진 표시



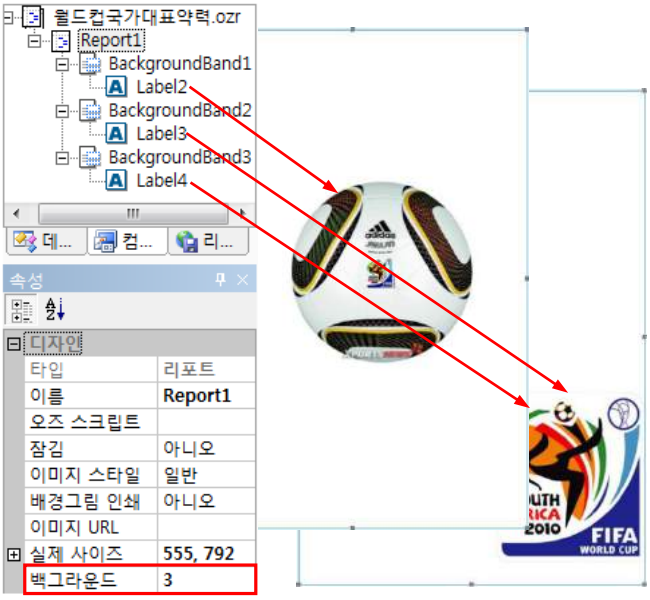
- 사진 라벨의 '그리기 형태' 속성을 '이미지'로 변경

(4) 페이지 여백을 빈 폼으로 채우기



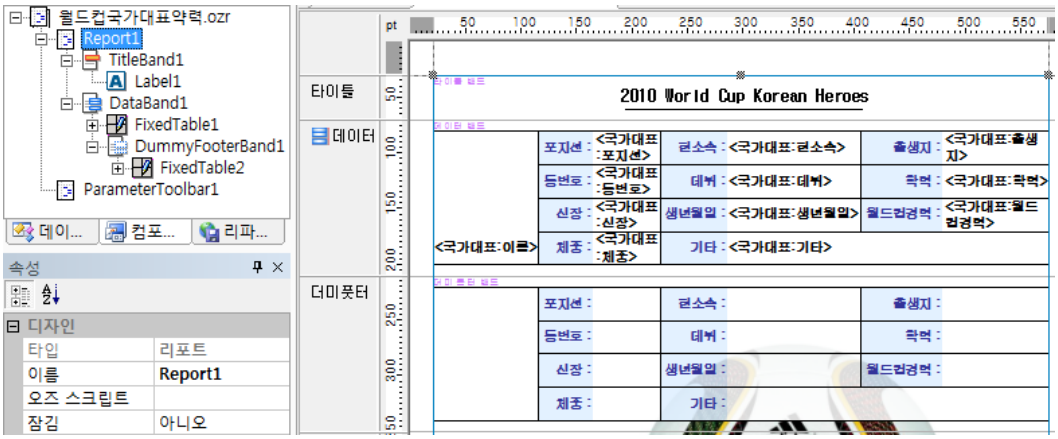
- 데이터 밴드의 '더미(풋터)' 속성을 '1'로 설정하면 더미 풋터 밴드가 하나 생성됨
- 데이터 밴드에 그려진 고정 테이블의 임의의 라벨을 선택한 후 팝업 메뉴의 '연결된 테이블 추가'를 선택하고 더미 풋터 밴드를 클릭하면 더미 풋터 밴드 위에 빈 폼이 그려짐
- 더미 풋터 밴드의 '반복 횟수' 속성을 한 페이지에 출력할 수 있는 개수 이상으로 설정
- 더미 풋터 밴드의 '페이지 끝에서 멈춤' 속성을 '예'로 설정하면 페이지 끝까지 더미 풋터 밴드가 반복됨

(5) 리포트 백그라운드 이미지 디자인



- Report1의 '백그라운드' 속성을 '3'으로 설정
- '보기' → '백그라운드 모드' 메뉴 선택
- 각각의 백그라운드 밴드에 이미지 라벨 추가
- 이미지 라벨의 팝업 메뉴에서 '이미지 URL 설정'을 선택한 후 ozp://OZwork/image/자블라니.JPG, ozp://OZwork/image/2010.JPG 경로 선택
- 배경 이미지가 보이도록 디자인 모드의 고정 테이블 라벨의 투명도를 255 이하 값으로 변경
예) 255 -> 220

(6) 디자인 화면



가. 목표 보고서 : 우편 발송용 고객 주소 라벨

나. 과제

① 고객 주소 정보를 2열로 표시 (리전 이용)

다. 관련 정보

■ 쿼리문

```
SELECT ContactName, PostalCode, City, Address FROM Customers
```

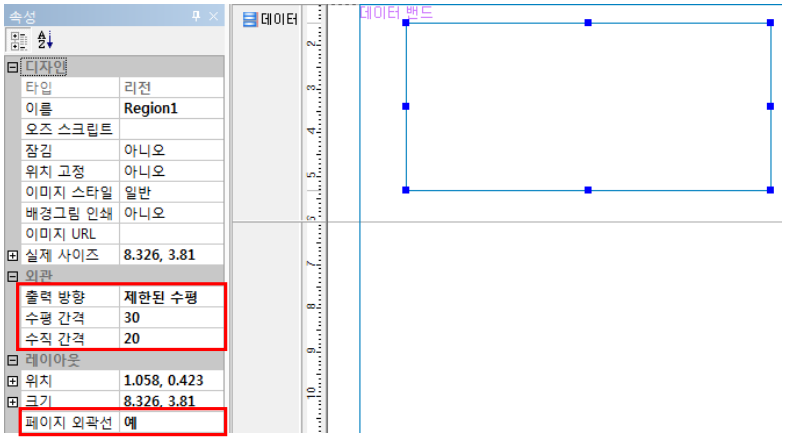
■ 데이터

ContactName	PostalCode	City	Address
한석규	742-170	상주시	가장동 78-3
황영순	137-060	서울특별시	서초구 방배동 883-11
조자룡	157-280	서울특별시	강서구 내발산동 318
구재석	402-110	인천광역시	남구 연수동 208-16

■ 보고서 파일 경로

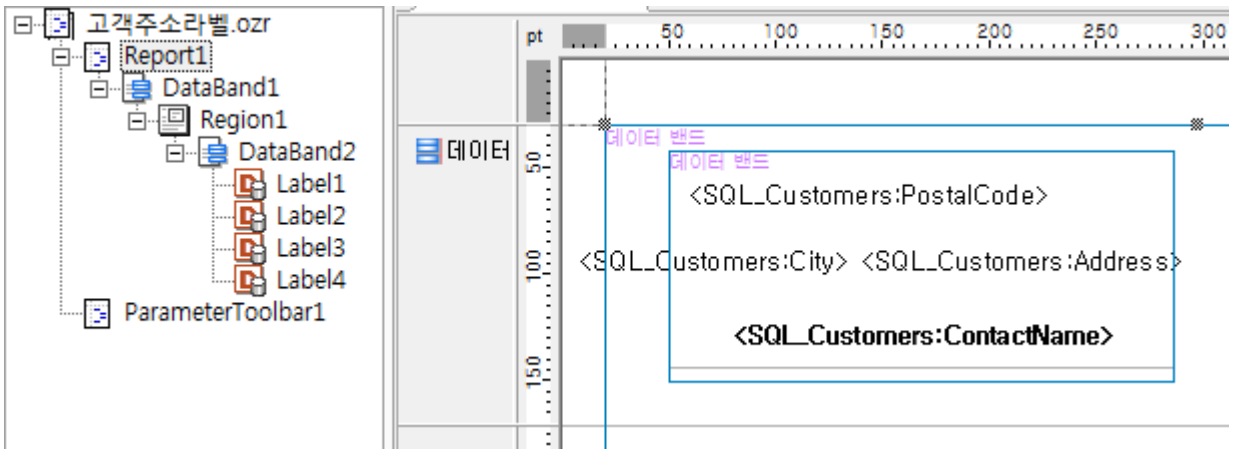
· 내 문서\OZRepository\OZwork\exercise\ex4\고객주소라벨.odi, 고객주소라벨.ozr

(1) 2열로 표시



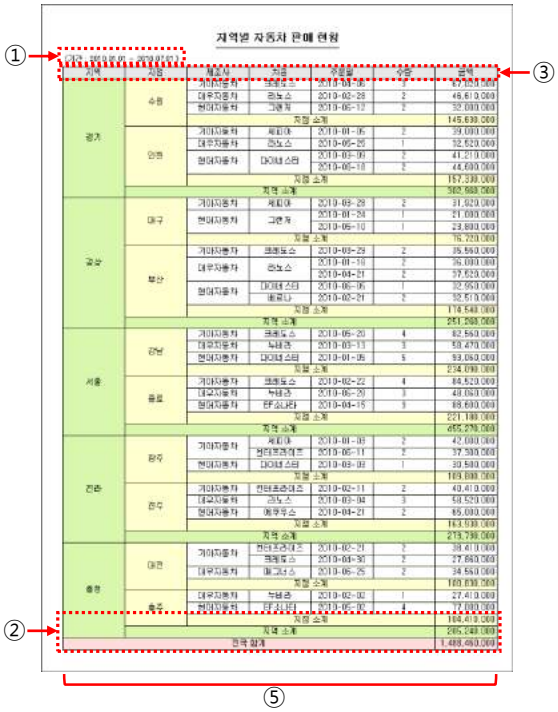
- 데이터를 2열로 표시하기 위해 데이터 밴드 위에 리전 추가
- 리전의 '출력 방향' 속성을 '제한된 수평'으로 설정하면 리전이 페이지 끝까지 수평 방향으로 반복되고 나서 수직방향으로 반복됨
- 리전의 '페이지 외곽선'을 '예'로 설정하면 테두리 보여짐
- 리전의 '수평 간격'을 '30', '수직 간격'을 '20' 과 같이 설정하면 리전 간의 간격이 설정됨

(2) 디자인 화면



5.1 목표 보고서 정의

가. 목표 보고서 : 지역별 자동차 판매 현황



나. 과제

① OrderDate 필드가 FromDate~ToDate 기간에 해당되는 데이터만 출력 (사용자 지정 패러미터 이용)

② Amount 필드의 지점 소계, 지역 소계, 전국 합계 표시 (크로스탭 마법사 Sum 함수 이용)

③ 표의 상단 제목 표시 (크로스탭 마법사의 옵션 이용)

④ 문자 데이터는 오름차순으로 정렬 (크로스탭 마법사의 옵션 이용)

⑤ 표의 열 너비를 보고서의 우측 여백에 맞추어 자동 조정 (크로스탭 속성 이용)

다. 관련 정보

■ **쿼리문**

```
SELECT CarOrders.OrderID, CarOrders.OrderDate, CarOrders.Region, CarOrders.BranchOffice,
CarOrders.Quantity, CarOrders.Amount, Car.Maker, Car.CarName
FROM Car, CarOrders
WHERE CarOrders.CarID = Car.CarID AND CarOrders.OrderDate >= '#OZParam.FromDate#' AND
CarOrders.OrderDate <= '#OZParam.ToDate#'
```

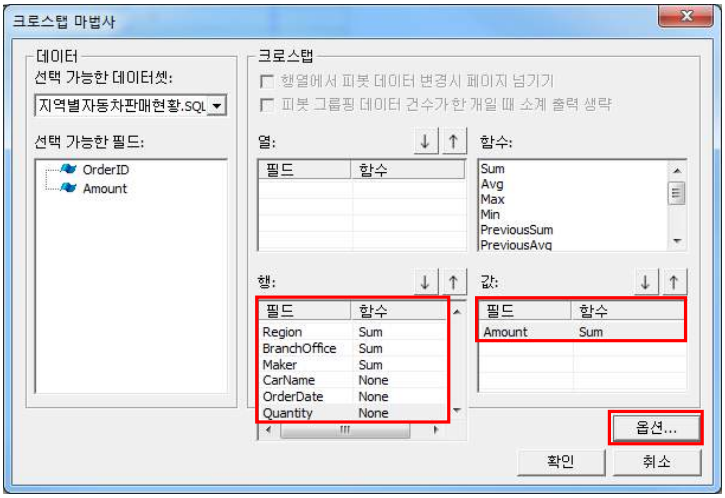
■ **데이터**

OrderID	OrderDate	Region	BranchOffice	Quantity	Amount	Maker	CarName
20100101	2010-01-03	전라	광주	2	42000000	기아자동차	세피아
20100102	2010-01-05	서울	강남	5	93060000	현대자동차	다이네스티
20100103	2010-01-05	경기	인천	2	39000000	기아자동차	세피아
20100104	2010-01-18	경상	부산	2	36000000	대우자동차	라노스

■ **보고서 파일 경로**

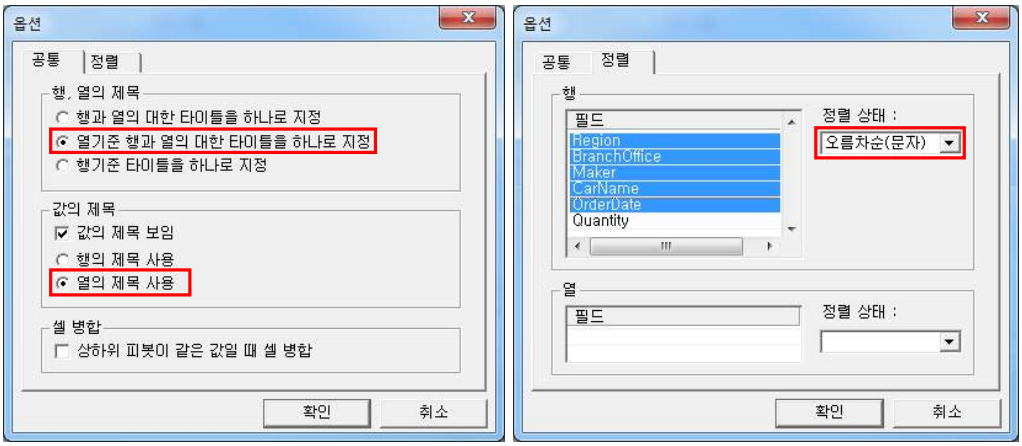
· 내 문서\OZRepository\OZwork\exercise\ex5\지역별자동차판매현황.odi, 지역별자동차판매현황.ozr

(1) 지점 소계, 지역 소계, 전국 합계 표시



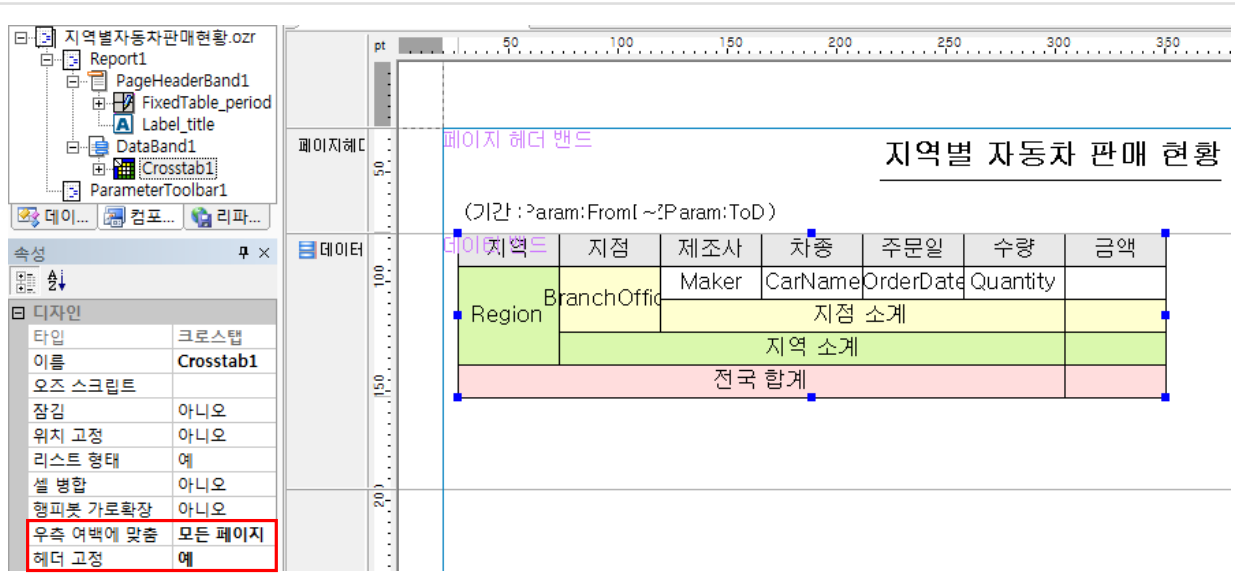
- 크로스탭 마법사의 행, 값 필드 지정
- 각각의 필드에 대해 함수 지정

(2) 표 제목 표시, 오름차순 정렬



- 필드 선택 후 정렬 상태 지정

(3) 디자인 화면



가. 목표 보고서 : 지역·제조사별 자동차 판매 현황

②

지역·제조사별 자동차 판매 현황

(기간 : 2010.01.01 ~ 2010.07.01)

지역/지점		전국 합계	경기			경상			서울			전라			충청		
제조사/차종			지역 소계	수원	인천	지역 소계	대구	부산	지역 소계	강남	종로	지역 소계	광주	전주	지역 소계	대전	충주
기아자동차	세피아	112,920,000	39,000,000		39,000,000	31,920,000	31,920,000					42,000,000	42,000,000				
	엔터프라이즈	116,120,000										77,710,000	37,300,000	40,410,000	38,410,000	38,410,000	
	크레도스	297,520,000	67,020,000	67,020,000		35,560,000		35,560,000	167,080,000	82,560,000	84,520,000				27,860,000	27,860,000	
	제조사 소계	526,560,000	106,020,000	67,020,000	39,000,000	67,480,000	31,920,000	35,560,000	167,080,000	82,560,000	84,520,000	119,710,000	79,300,000	40,410,000	66,270,000	66,270,000	
대우자동차	누비라	133,940,000							106,530,000	58,470,000	48,060,000				27,410,000		27,410,000
	라노스	211,170,000	79,130,000	46,610,000	32,520,000	73,520,000		73,520,000				58,520,000		58,520,000			
	매그너스	34,560,000													34,560,000	34,560,000	
	제조사 소계	379,670,000	79,130,000	46,610,000	32,520,000	73,520,000		73,520,000	106,530,000	58,470,000	48,060,000	58,520,000		58,520,000	61,970,000	34,560,000	27,410,000
현대자동차	EF소나타	165,600,000							88,600,000		88,600,000				77,000,000		77,000,000
	그랜저	76,800,000	32,000,000	32,000,000		44,800,000	44,800,000										
	다이네스티	242,320,000	85,810,000		85,810,000	32,950,000		32,950,000	93,060,000	93,060,000		30,500,000	30,500,000				
	베르나	32,510,000				32,510,000		32,510,000									
	에쿠우스	65,000,000										65,000,000		65,000,000			
	제조사 소계	582,230,000	117,810,000	32,000,000	85,810,000	110,260,000	44,800,000	65,460,000	181,660,000	93,060,000	88,600,000	95,500,000	30,500,000	65,000,000	77,000,000		77,000,000
업체 합계		1,488,460,000	302,960,000	145,630,000	157,330,000	251,260,000	76,720,000	174,540,000	455,270,000	234,090,000	221,180,000	273,730,000	109,800,000	163,930,000	205,240,000	100,830,000	104,410,000
업체 평균		51,326,207	50,493,333	48,543,333	52,443,333	41,876,667	38,360,000	43,635,000	75,878,333	78,030,000	73,726,667	45,621,667	36,600,000	54,643,333	41,048,000	33,610,000	52,205,000

①

나. 과제

① 제조사 소계, 업체 합계, 업체 평균, 지역 소계, 전국 합계 표시 (크로스탭 마법사 함수 이용)

② 메인 제목에 분리선 표시 (크로스탭 헤더 속성)

③ 미리보기할 때 모든 페이지를 한 장으로 표시 (미리보기 옵션에 viewer.largebundle=true 이용)

다. 관련 정보

■ 쿼리문

```
SELECT CarOrders.OrderID, CarOrders.OrderDate, CarOrders.Region, CarOrders.BranchOffice,
CarOrders.Quantity, CarOrders.Amount, Car.Maker, Car.CarName
FROM Car, CarOrders
WHERE CarOrders.CarID = Car.CarID AND CarOrders.OrderDate >= '#OZParam.FromDate#' AND
CarOrders.OrderDate <= '#OZParam.ToDate#'
```

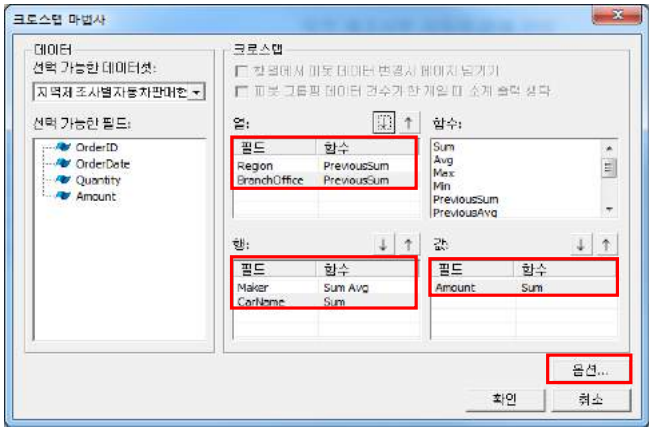
■ 데이터

OrderID	OrderDate	Region	BranchOffice	Quantity	Amount	Maker	CarName
20100101	2010-01-03	전라	광주	2	42000000	기아자동차	세피아
20100102	2010-01-05	서울	강남	5	93060000	현대자동차	다이네스티
20100103	2010-01-05	경기	인천	2	39000000	기아자동차	세피아
20100104	2010-01-18	경상	부산	2	36000000	대우자동차	라노스

■ 보고서 파일 경로

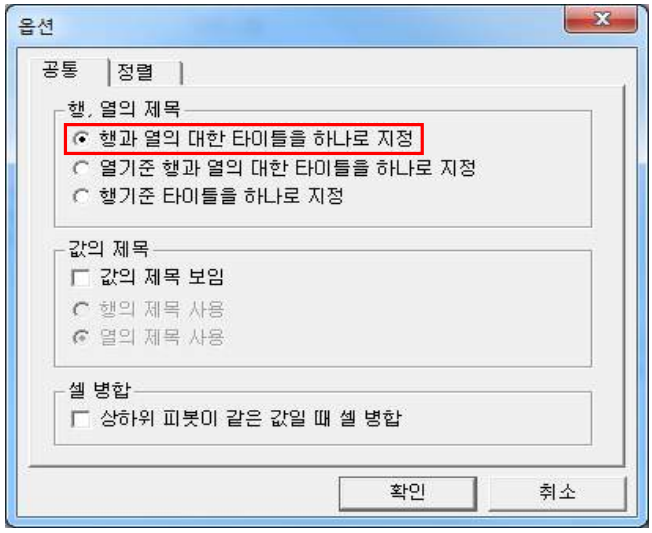
· 내 문서\OZRepository\OZwork\exercise\ex6\지역제조사별자동차판매현황.odi, 지역제조사별자동차판매현황.ozr

(1) 소계, 합계 표시



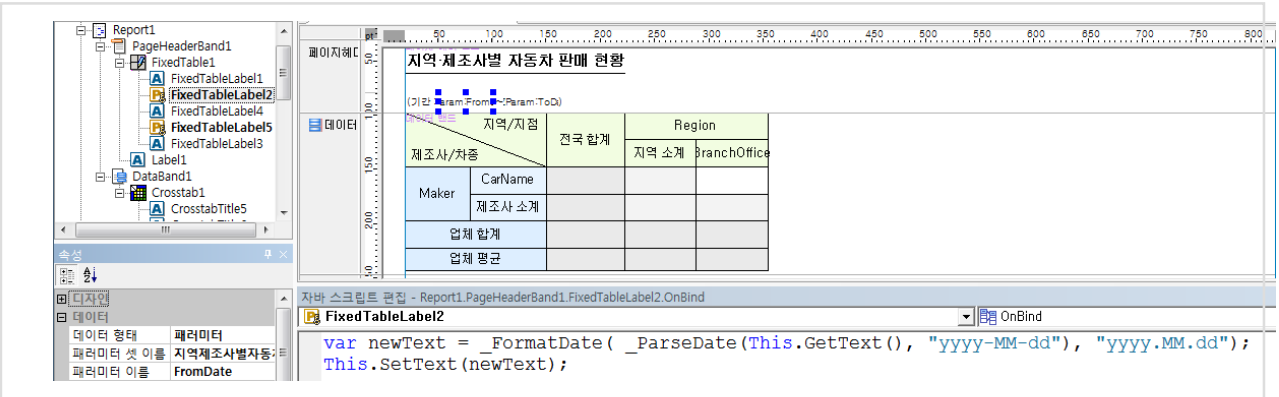
- 크로스탭 마법사의 열, 행, 값 필드 지정
- 각각의 필드에 대해 함수 지정

(2) 분리선 표시

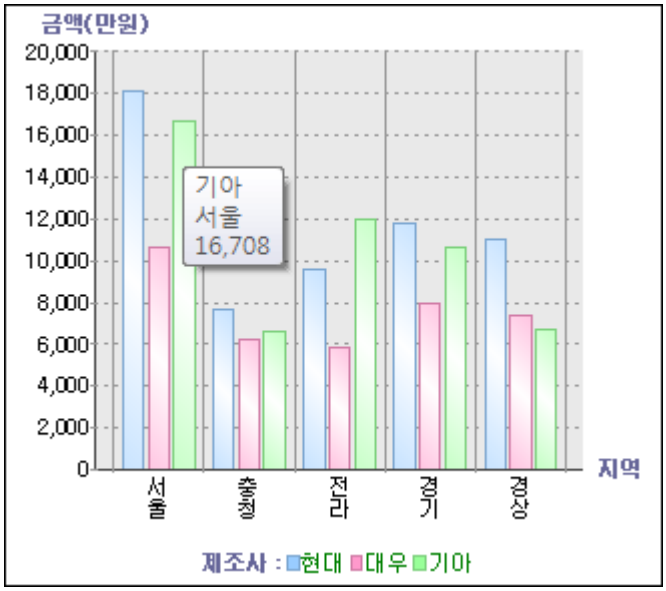


- 크로스탭 옵션 창에서 행, 열의 제목 선택
- 메인 제목의 '분리선 보이기' 속성 지정

(3) 디자인 화면



가. 목표 보고서 : 지역별 판매금액



나. 과제

① 묶은 세로 막대형 차트 구현 (X축:Region, 왼쪽 Y축:Amount1, 계열:Maker1)
② 동일 지역, 제조사에 해당되는 금액이 여러 개일 때 합쳐서 표시 (데이터 탭에서 Y축 필드를 sum으로 설정)
③ X축의 항목간 막대 간격을 기본값 보다 크게 조정 (차트 마법사의 X축 탭의 고급 창에서 설정)
④ Y축 라벨의 3자리마다 콤마 표시 (차트 마법사의 Y축 탭의 서식 창에서 설정)
⑤ 범례 하단으로 표시 (차트 마법사의 범례 탭에서 설정)
⑥ 툴팁 표시 (차트 마법사의 데이터 라벨 탭에서 설정)

다. 관련 정보

■ **쿼리문**

```
SELECT OrderID, OrderDate, Region, BranchOffice, Quantity, (Amount/10000) AS Amount1,
mid(Maker,1,2) AS Maker1, CarName
FROM Car, CarOrders
WHERE CarOrders.CarID = Car.CarID
```

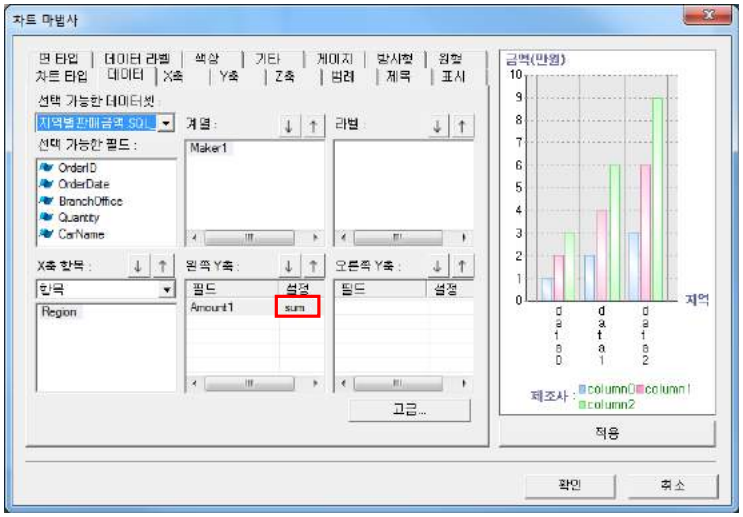
■ **데이터**

OrderID	OrderDate	Region	BranchOffice	Quantity	Amount1	Maker1	CarName
20100402	2010-04-15	서울	종로	3	8860.0	현대	EF소나타
20100501	2010-05-02	충청	충주	4	7700.0	현대	EF소나타
20100102	2010-01-05	서울	강남	5	9306.0	현대	다이네스티
20100301	2010-03-03	전라	광주	1	3050.0	현대	다이네스티

■ **보고서 파일 경로**

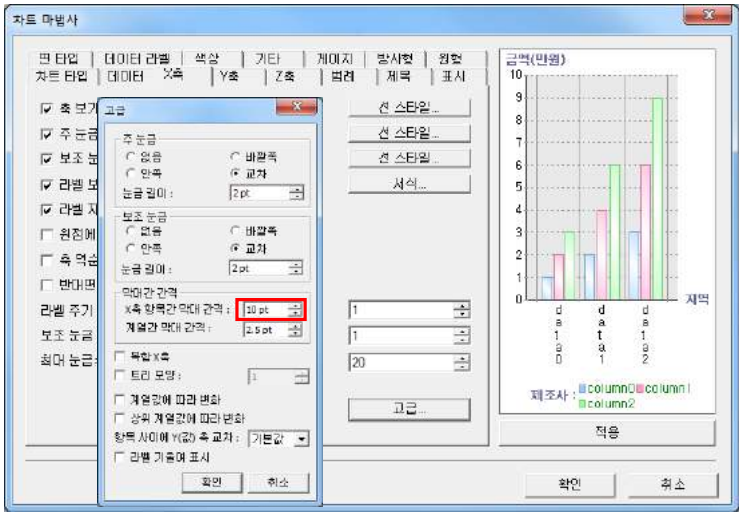
· 내 문서\OZRepository\OZwork\exercise\ex7\지역별판매금액.odi, 지역별판매금액.ozr

(1) 차트 데이터 설정



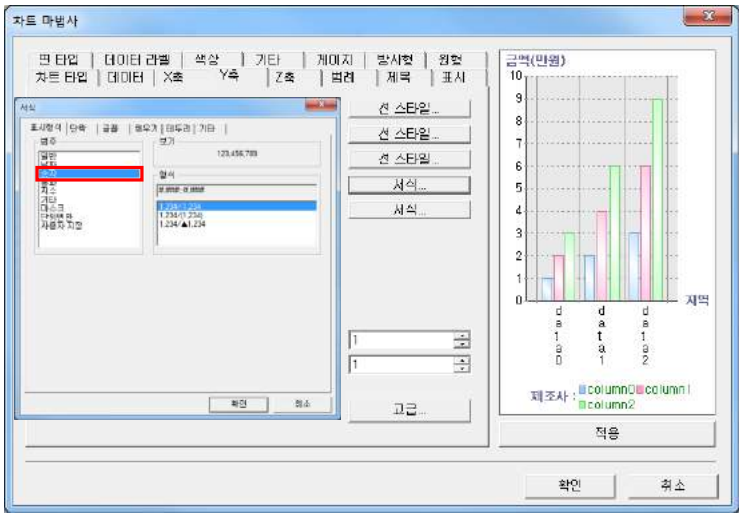
- 차트 마법사의 데이터 탭에서 X축, Y축, 계열에 필드를 드래그&드롭
- Y축 필드의 'none'을 'sum'으로 변경

(2) X축 설정



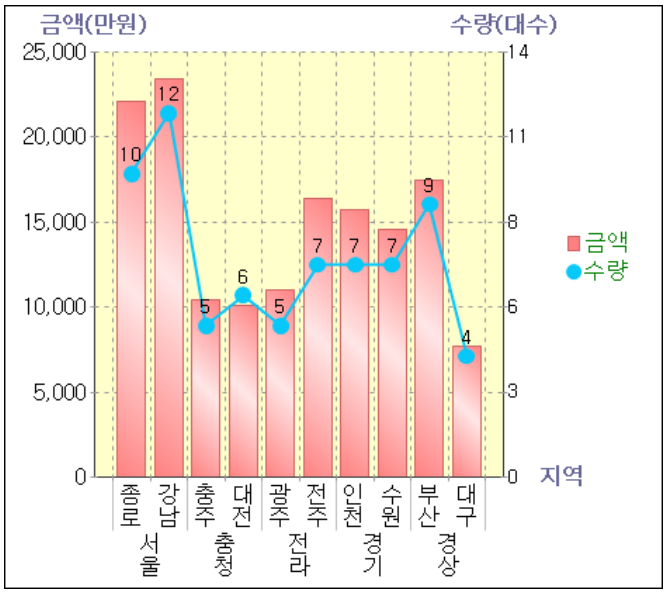
- X축 고급 창에서 'X축 항목간 막대 간격' 조절

(3) Y축 설정



- Y축 왼쪽 라벨의 서식 창에서 표시 형식을 '숫자'로 변경

가. 목표 보고서 : 지역별 판매금액수량



나. 과제

① '금액'은 막대, '수량'은 표식있는 선 그래프로 표시 (사용자형 차트 이용)

② X축 항목에 Region, BranchOffice 2개 필드를 지정하여 트리 모양으로 표현 (X축 탭의 고급 옵션 이용)

③ 왼쪽 Y축에는 Amount1(금액), 오른쪽 Y축에는 Quantity(수량) 필드를 표시하고 값이 여러 개일 때 합쳐서 표시

④ 양쪽 Y축 눈금수를 5개로 설정하고 왼쪽 Y축의 최소값 0, 최대값 25000으로 설정

⑤ '수량'에 대한 데이터만 그래프에 표시 (데이터 라벨 탭에서 계열 중 선택)

⑥ 범례의 영문 필드명을 한글 텍스트로 변경 (Chart의 OnPostChartBuild 이벤트에서 자바 스크립트 이용)

⑦ 차트의 선과 표식을 부드럽게 표시 (미리보기 옵션에 viewer.antialiasing=true 이용)

다. 관련 정보

■ **쿼리문**

```
SELECT OrderID, OrderDate, Region, BranchOffice, Quantity, (Amount/10000) AS Amount1,
mid(Maker,1,2) AS Maker1, CarName
FROM Car, CarOrders
WHERE CarOrders.CarID = Car.CarID
```

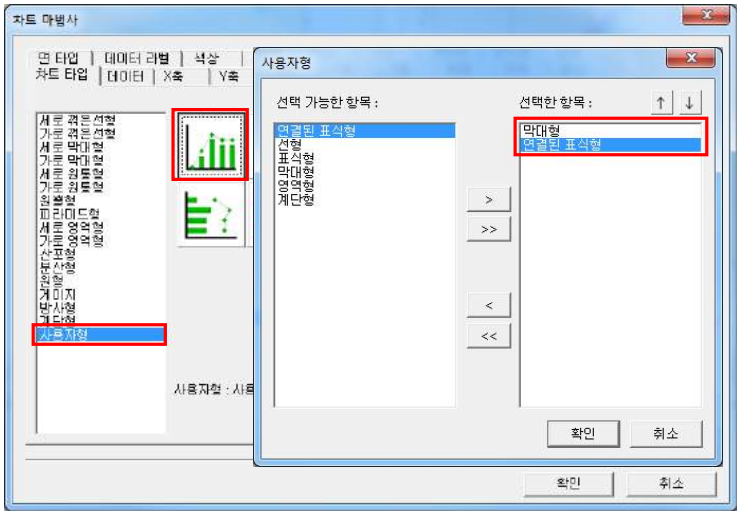
■ **데이터**

OrderID	OrderDate	Region	BranchOffice	Quantity	Amount1	Maker1	CarName
20100402	2010-04-15	서울	종로	3	8860.0	현대	EF소나타
20100501	2010-05-02	충청	충주	4	7700.0	현대	EF소나타
20100102	2010-01-05	서울	강남	5	9306.0	현대	다이네스티
20100301	2010-03-03	전라	광주	1	3050.0	현대	다이네스티

■ **보고서 파일 경로**

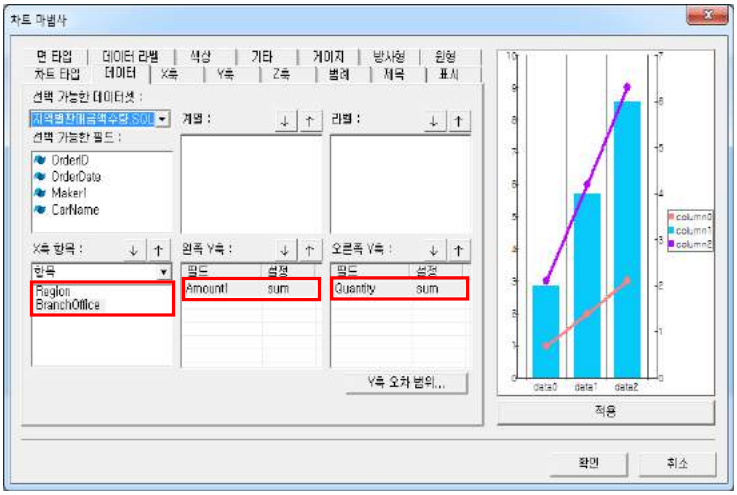
· 내 문서\OZRepository\OZwork\exercise\ex8\지역별판매금액수량.odi, 지역별판매금액수량.ozr

(1) 차트 타입 설정



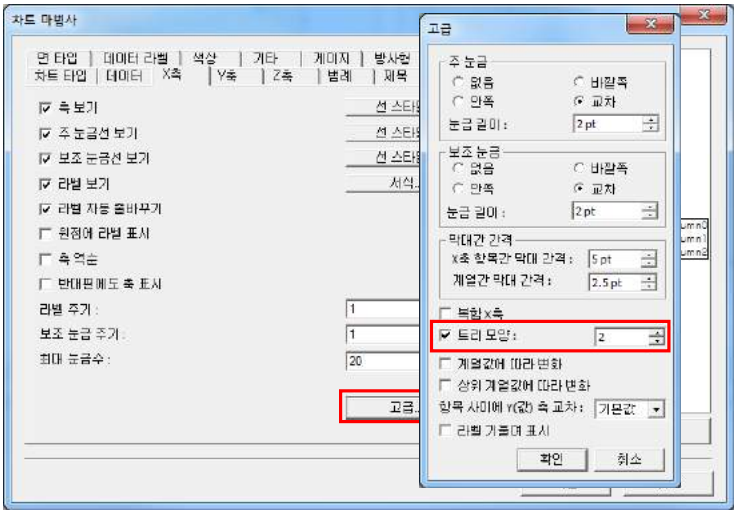
- 차트 마법사의 차트 타입 탭에서 '사용자형' 차트 선택
- 막대형, 연결된 표식형 (마크 있는 선 모양)을 선택하여 데이터 별로 다른 차트 표현 가능

(2) 차트 데이터 설정



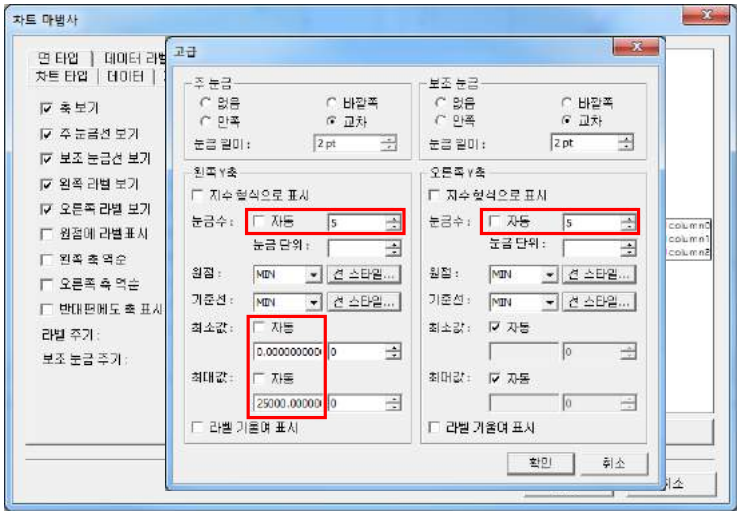
- X축 항목에 'Region', 'BranchOffice' 필드 설정
- 왼쪽 Y축에 'Amount1' 필드를 'sum'으로 설정
- 오른쪽 Y축에 'Quantity' 필드를 'sum'으로 설정

(3) X축 설정



- X축 탭의 '고급' 창에서 '트리 모양'을 체크하고 '2'로 설정하면 한 차트에 2개의 X축 항목이 통합되어 보여짐
- '트리 모양'을 체크하지 않으면 첫 번째 X축 필드의 값 별로 차트가 분리됨

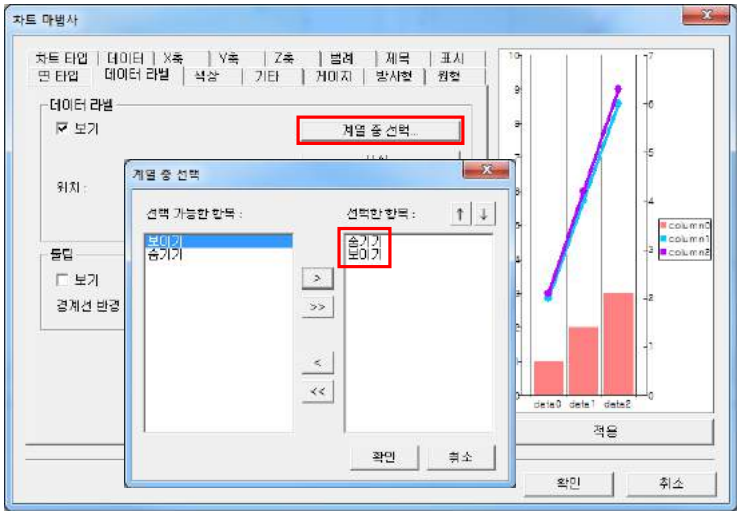
(4) Y축 설정



- Y축 탭의 '고급' 창에서 '눈금수'의 '자동'을 체크 해제하고 '5'를 입력하면 눈금수가 다섯 개로 고정됨

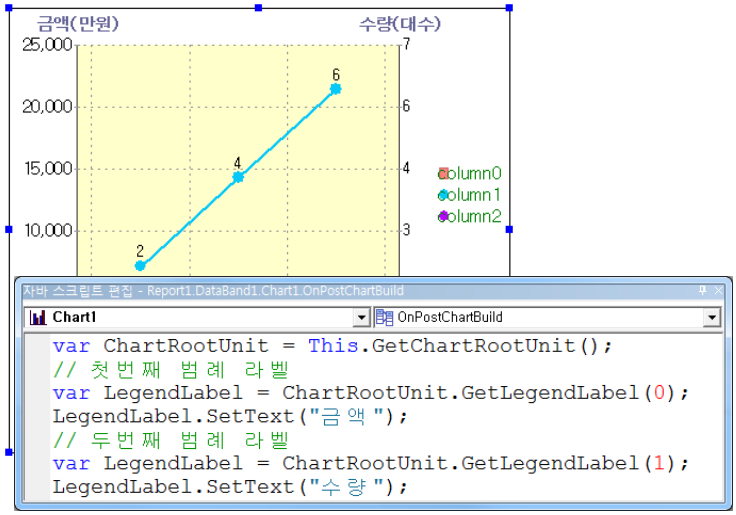
- '최소값', '최대값'의 '자동'을 체크 해제하고 '0', '25000'으로 각각 입력하면 최소/최대값이 고정됨

(5) 데이터 라벨 설정



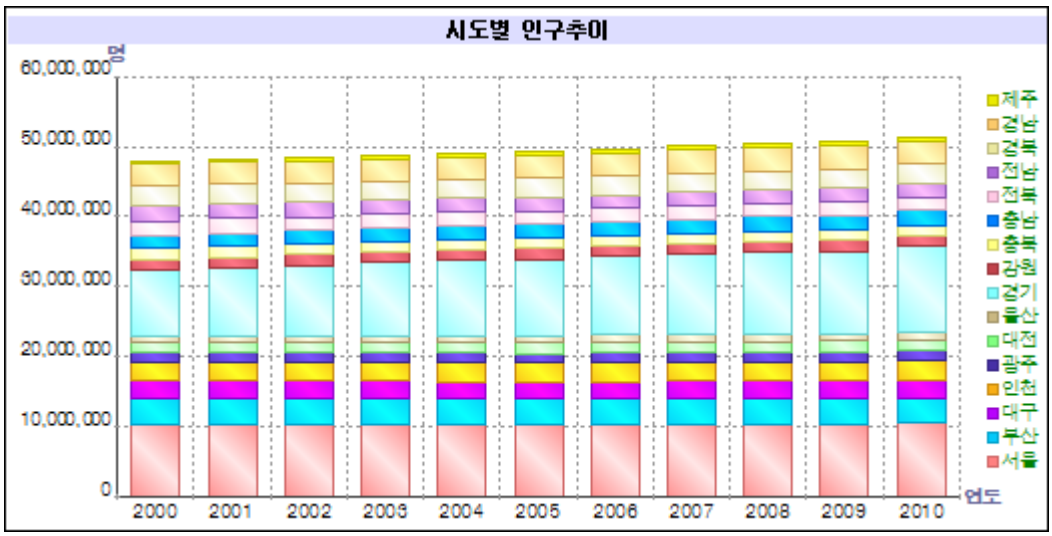
- 데이터 라벨 탭의 '계열 중 선택' 창에서 '숨기기', '보이기' 순으로 '선택한 항목'에 지정하면 왼쪽 Y축의 데이터 라벨은 안보이고 오른쪽 Y축의 데이터 라벨만 그래프에 표시됨

(6) 범례 텍스트 설정



- Y축 필드가 여러 개일 때 각 필드명이 범례 텍스트에 보여지는데, 필드명이 영문일 때 한글로 변경하려면 OnPostChartBuild 이벤트에 스크립트를 작성할 수 있음

가. 목표 보고서 : 시도별 인구추이



나. 과제

① 누적 세로 막대형 차트 구현 (X축:연도, 왼쪽 Y축:나머지 필드)

② 모든 시도 데이터를 한 차트로 표현 (Z축 탭의 최대계열수 설정)

③ 그래프와 범례의 심볼 색상을 순서대로 맞추기 (범례 탭의 내림차순 설정)

④ 차트 제목 표시 (제목 탭 설정)

다. 관련 정보

■ **데이터 파일 경로**

· ozp://OZwork/data/시도별인구추이.csv

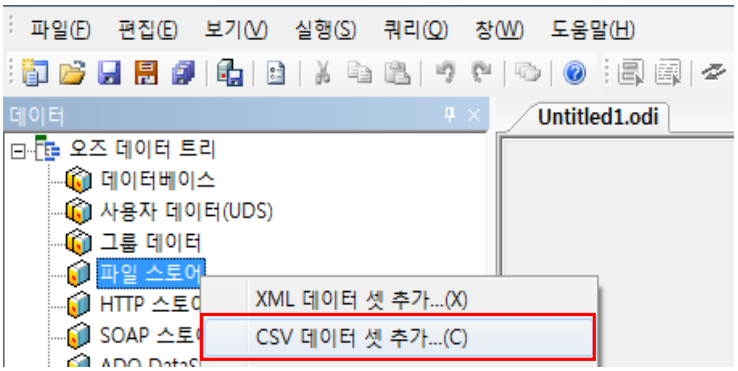
■ **데이터**

연도	서울	부산	대구	인천	광주	대전	울산	경기	강원	충북	충남	전북	전남	경북	경남	제주
2000	10...	38...	25...	25...	13...	13...	10...	92...	15...	15...	19...	20...	21...	28...	31...	54...
2001	10...	37...	25...	25...	13...	14...	10...	96...	15...	15...	19...	20...	21...	28...	31...	54...
2002	10...	37...	25...	25...	14...	14...	10...	10...	15...	15...	19...	19...	20...	27...	31...	55...
2003	10...	37...	25...	26...	14...	14...	10...	10...	15...	15...	19...	19...	20...	27...	31...	55...

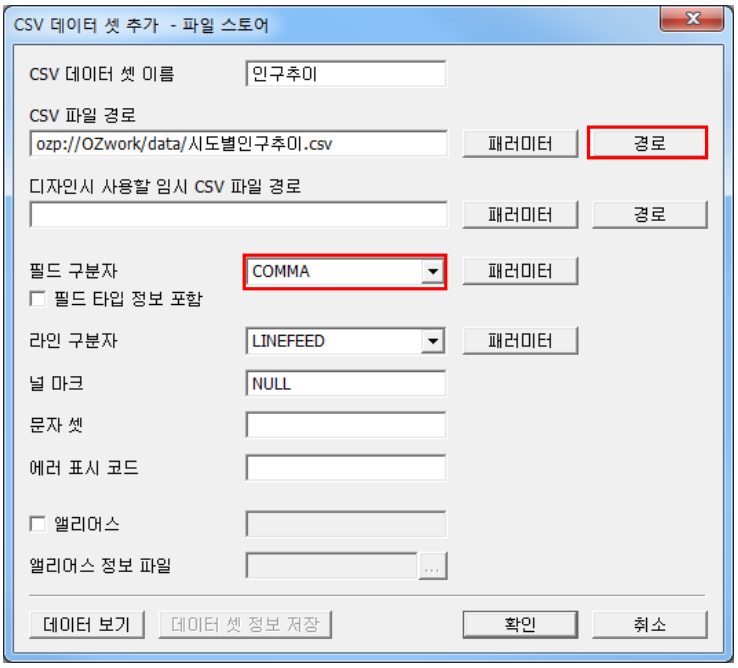
■ **보고서 파일 경로**

· 내 문서\OZRepository\OZwork\exercise\ex9\시도별인구추이.odi, 시도별인구추이.ozr

(1) CSV 파일 데이터 사용



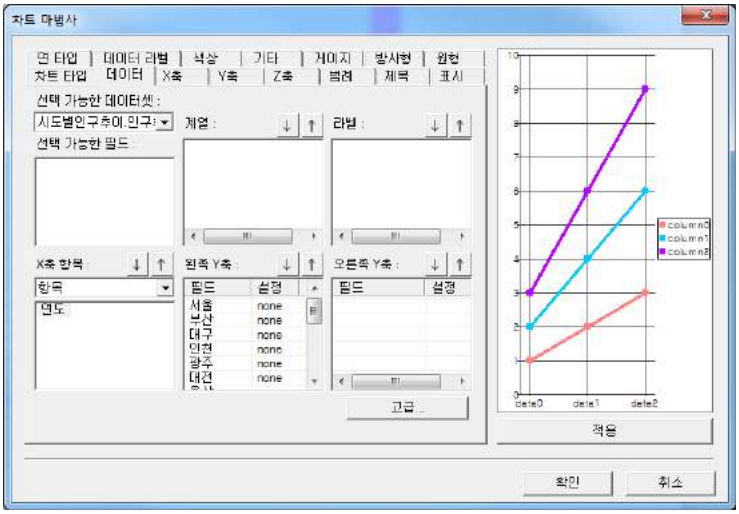
- 보고서에 csv 파일 데이터를 사용하기 위해 쿼리 디자이너에서 '파일 스토어'의 팝업 메뉴에서 'CSV 데이터 셋 추가' 선택



- 'CSV 데이터 셋 추가' 창의 '경로' 버튼을 클릭하여 데이터 파일 지정
- '필드 구분자'에 'COMMA' 선택

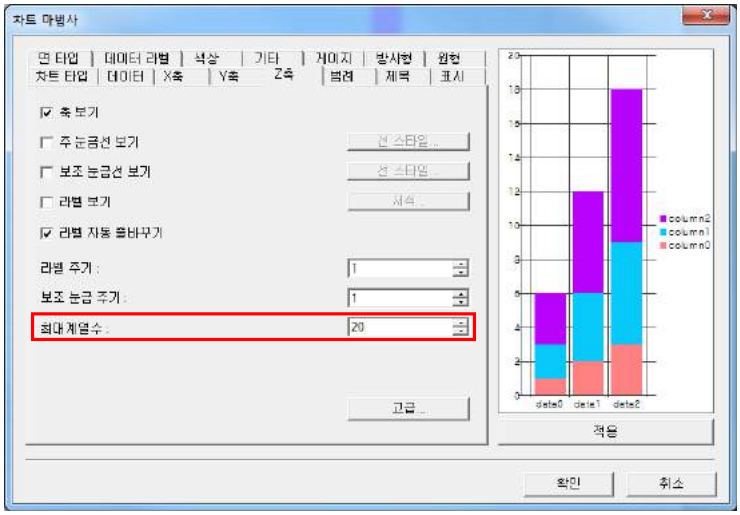
※ 'CSV 파일 경로'는 운영할 때 사용될 파일 경로입니다. 경로 앞부분을 'ozp://' 로 변경하면 오즈 서버 리파지토리를 기준으로 하는 경로가 설정됩니다.

(2) 차트 데이터 설정



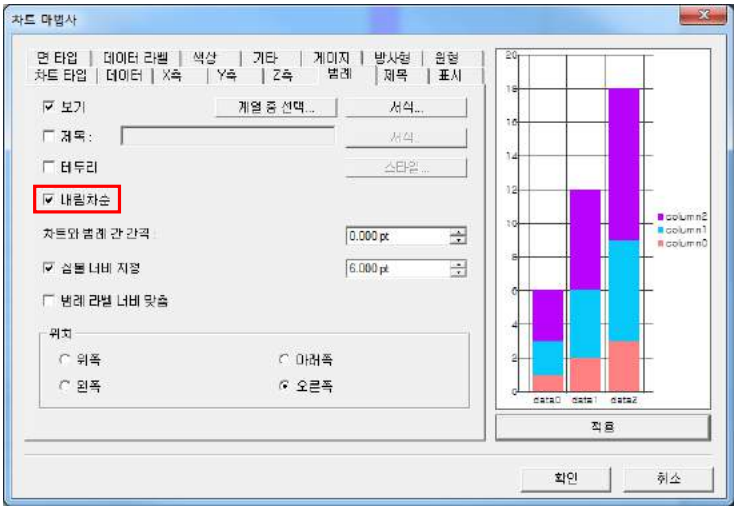
- X축 항목에 '연도' 필드 설정
- 왼쪽 Y축에 나머지 필드 모두 설정

(3) Z축 설정



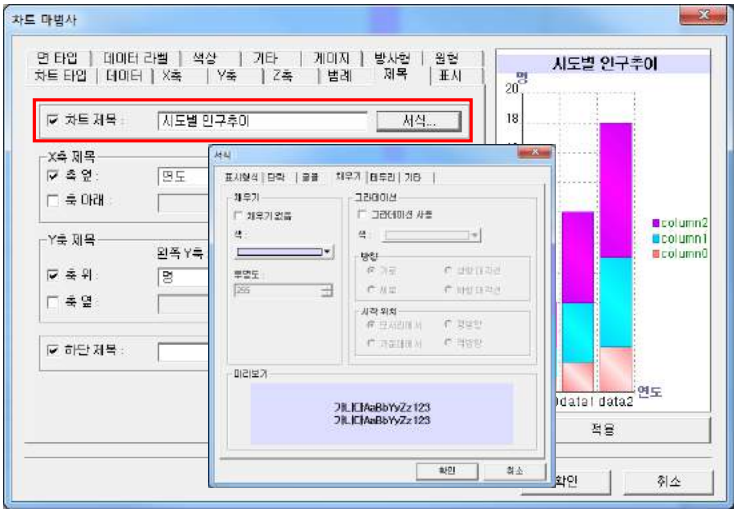
• Z축 탭의' 최대계열수'를 Y축의 필드 개수 이상으로 설정해야 한 차트에 모두 보여짐

(4) 범례 설정



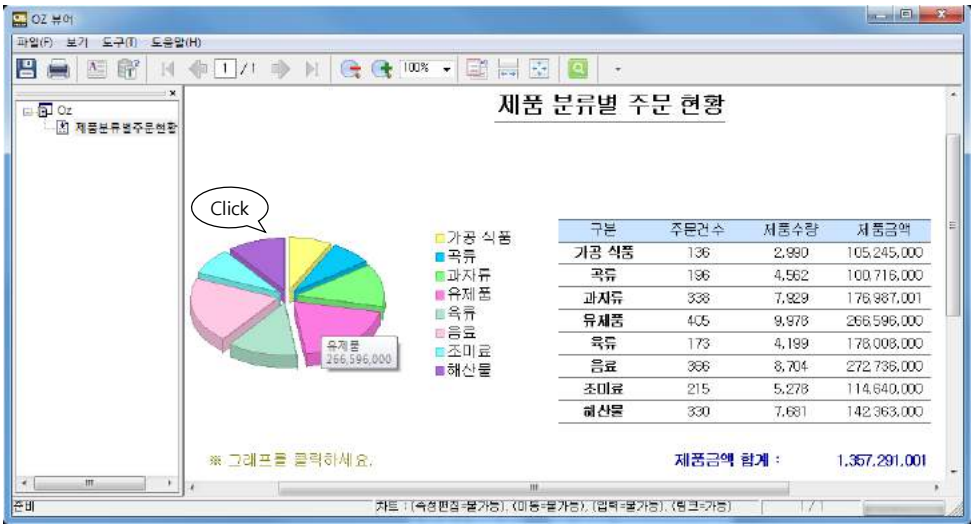
• 누적 막대가 아래에서 위로 그려지기 때문에 범례 심볼과 색상을 맞추기 위해 범례의 '내림차순' 체크

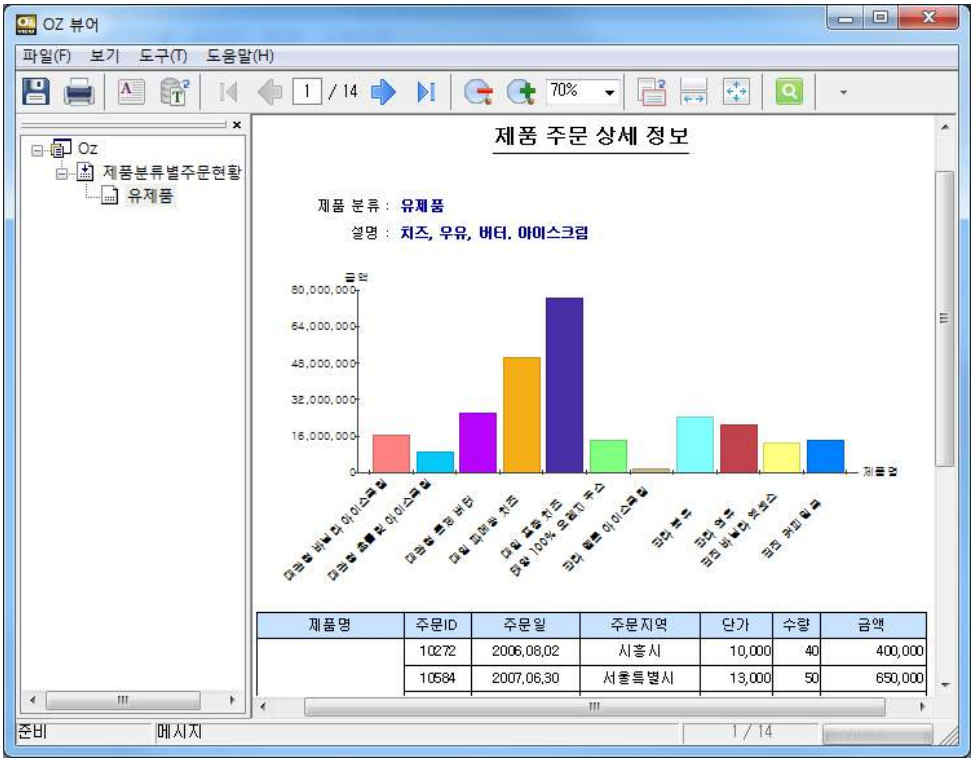
(5) 제목 설정



• 제목 탭에서 '차트 제목'을 입력하고, 서식에서 채우기 색 설정

가. 목표 보고서 : 제품 분류 별 주문 현황, 제품 주문 상세 정보





나. 과제

① 제품 분류 별 주문 현황 보고서 구현

② 특정 제품 분류에 대한 상세 주문 정보를 보여주는 보고서 구현

③ '제품 분류 별 주문 현황' 보고서의 그래프를 클릭했을 때 '제품 주문 상세 정보' 보고서 표시

④ 클릭된 제품 분류 이름을 패러미터로 넘겨 관련 제품 분류에 대해서 상세 정보 표시

다. '제품 분류 별 주문 현황' 보고서 관련 정보

■ 쿼리문

```
SELECT C.CategoryID, C.CategoryName, C.Description, COUNT(OD.OrderID) AS OrderCount,
SUM(OD.Quantity) AS Quantity, SUM(OD.UnitPrice*OD.Quantity) AS Amount
FROM Orders O, Categories C, Products P, [Order Details] OD, Customers Cust
WHERE C.CategoryID = P.CategoryID AND O.OrderID = OD.OrderID
AND Cust.CustomerID = O.CustomerID AND P.ProductID = OD.ProductID
GROUP BY C.CategoryID, C.CategoryName, C.Description
ORDER BY C.CategoryName ASC
```

■ 데이터

CategoryID	CategoryName	Description	OrderCount	Quantity	Amount
7	가공 식품	건과류 ,과일 통조림	136	2990.0	105245000.0000
5	곡류	빵, 파스타, 시리얼	196	4562.0	100716000.0000
3	과자류	사탕, 쿠키, 비스킷	338	7929.0	176987000.6000
4	유제품	치즈, 우유, 버터. 아이스크림	405	9978.0	266596000.0000
6	육류	가공육	173	4199.0	178008000.0000
1	음료	청량음료, 커피, 홍차, 맥주	366	8704.0	272736000.0000
2	조미료	감미료, 향신료, 양념, 스프레드	215	5278.0	114640000.0000
8	해산물	해초류, 생선, 건어물	330	7681.0	142363000.0000

■ 보고서 파일 경로

· 내 문서\OZRepository\OZwork\exercise\ex10\제품분류별주문현황.odi, 제품분류별주문현황.ozr

라. '제품 주문 상세 정보' 보고서 관련 정보

■ 쿼리문

```
SELECT C.CategoryName, C.Description, P.ProductName, Cust.City, O.OrderDate, OD.OrderID, OD.UnitPrice,
OD.Quantity, (OD.UnitPrice * OD.Quantity) AS Amount
FROM [Order Details] OD, Orders O, Customers Cust, Products P, Categories C
WHERE C.CategoryID = P.CategoryID AND P.ProductID = OD.ProductID
AND OD.OrderID = O.OrderID AND O.CustomerID = Cust.CustomerID
AND C.CategoryName = '#OZParam.CategoryName#'
ORDER BY P.ProductName ASC
```

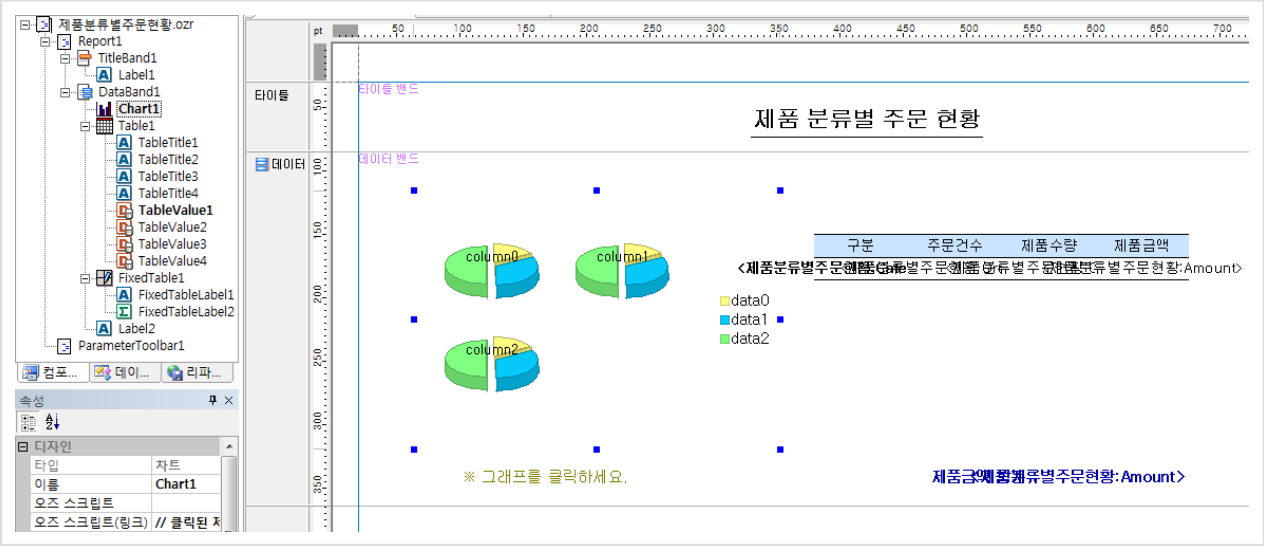
■ 데이터

CategoryName	Description	ProductName	City	OrderDate	OrderID	UnitPrice	Quantity	Amount
음료	청량음료, ...	미왕 초콜릿 드링크	서울특별시	2008-01-01 00:00:00.000	10808	18000.0000	50	900000.0000
음료	청량음료, ...	삼화 콜라	서울특별시	2007-08-07 00:00:00.000	10623	5000.0000	3	15000.0000
음료	청량음료, ...	삼화 콜라	평택시	2007-08-28 00:00:00.000	10648	5000.0000	15	75000.0000

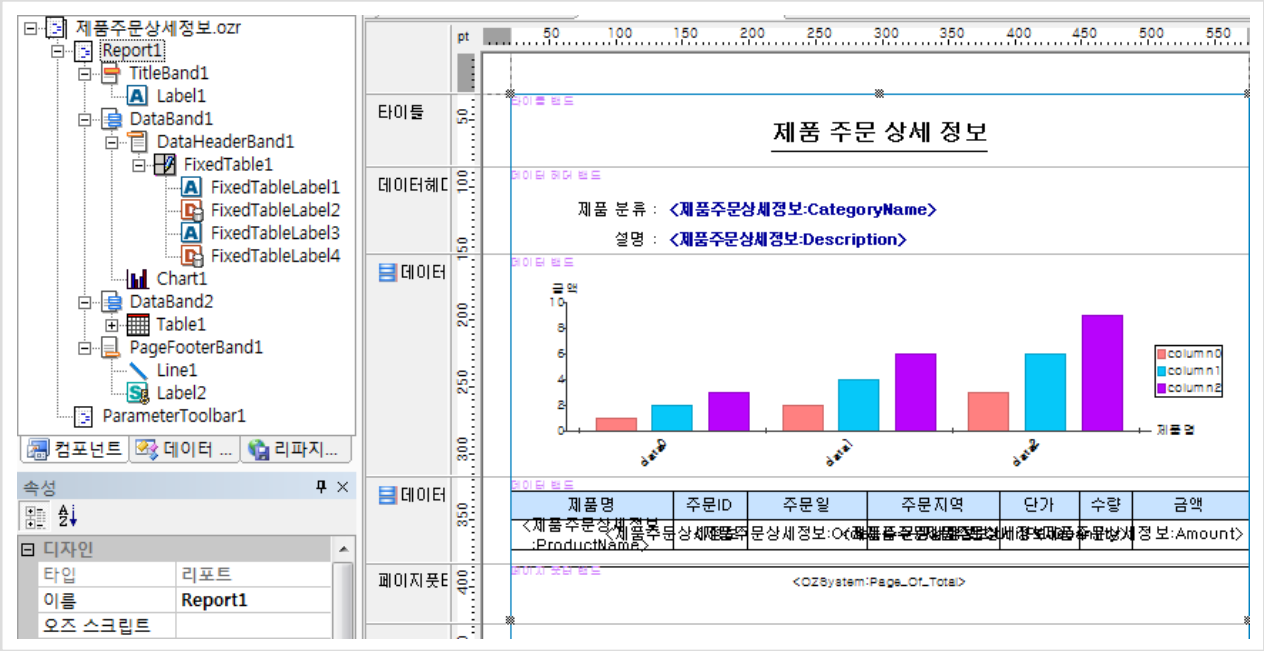
■ 보고서 파일 경로

· 내 문서\OZRepository\OZwork\exercise\ex10\제품주문상세정보.odi, 제품주문상세정보.ozr

(1) '제품 분류 별 주문 현황' 디자인 화면



(2) '제품 주문 상세 정보' 디자인 화면



(3) 차트 링크 스크립트

- '제품 분류 별 주문 현황' 보고서에서 차트의 '오즈 스크립트(링크)' 속성 선택
- '오즈 스크립트 편집' 창의 팝업 메뉴에서 '링크 마법사'의 '보고서 링크' 이용하여 아래 스크립트 작성

오즈 스크립트 편집 - Report1.DataBand1.Chart1

Chart1(링크) 문법 검사

```
// 클릭된 제품 분류에 대한 상세 보고서로 링크
setLinkServer("", 0, "OZwork/exercise/ex10/제품주문상세정보.ozr");
setLinkOption("odi.odinames", "제품주문상세정보");
setLinkOption("odi.제품주문상세정보.pcount", "1");
setLinkOption("odi.제품주문상세정보.args1", "CategoryName=" + (string)getattr("Chart_tooltip_rowpivot"));
setLinkOption("connection.displayname", (string)getattr("Chart_tooltip_rowpivot"));
```

가. 목표 보고서 : 제품 주문 정보

2011년03월24일 목 **제품 주문 정보** 30/60

유제품 : 치즈, 우유, 버터, 아이스크림

제품명	주문ID	주문일	주문지역	단가	수량	금액
현진 커피 밀크	10770	2007.12.09	제천시	21,000	15	315,000
	10797	2007.12.25	서울특별시	21,000	20	420,000
	10800	2007.12.26	서울특별시	21,000	50	1,050,000
	10823	2008.01.09	서울특별시	21,000	20	420,000
	10842	2008.01.20	서울특별시	21,000	15	315,000
	10862	2008.01.30	서울특별시	21,000	25	525,000
	10869	2008.02.04	서울특별시	21,000	10	210,000
	10889	2008.02.16	시흥시	21,000	40	840,000
	10912	2008.02.26	부산광역시	21,000	40	840,000
	10926	2008.03.04	서울특별시	21,000	2	42,000
	10944	2008.03.12	부산광역시	21,000	5	105,000
	10986	2008.03.30	서울특별시	21,000	30	630,000
	10989	2008.03.31	대전광역시	21,000	15	315,000
	11043	2008.04.22	서울특별시	21,000	10	210,000
	11073	2008.05.05	부산광역시	21,000	10	210,000
				주문 건수 소계 :		405
				주문 수량 소계 :		9,978
				주문 금액 소계 :		266,596,000

나. 과제

① 제품 분류(CategoryName) 별로 주문 목록 표시 (그룹 헤더 밴드 이용)

② 제품 분류 별로 하단에 소계 표시 (그룹 풋터 밴드, 요약 라벨 이용)

③ 매 페이지 마다 제품 분류 상단에 표시 (데이터 밴드의 '상위 그룹 헤더 고정' 속성 이용)

④ 제품 분류 별로 페이지 분리하기 (그룹 풋터 밴드의 '페이지 넘기기' 속성 이용)

다. 관련 정보

■ 쿼리문

```
SELECT C.CategoryName, C.Description, P.ProductName, OD.OrderID, O.OrderDate, Cust.City, OD.UnitPrice,
OD.Quantity, (OD.UnitPrice*OD.Quantity) AS Amount
FROM Orders O, Categories C, Products P, [Order Details] OD, Customers Cust
WHERE C.CategoryID = P.CategoryID AND O.OrderID = OD.OrderID
AND Cust.CustomerID = O.CustomerID AND P.ProductID = OD.ProductID
ORDER BY P.ProductName ASC, O.OrderDate ASC
```

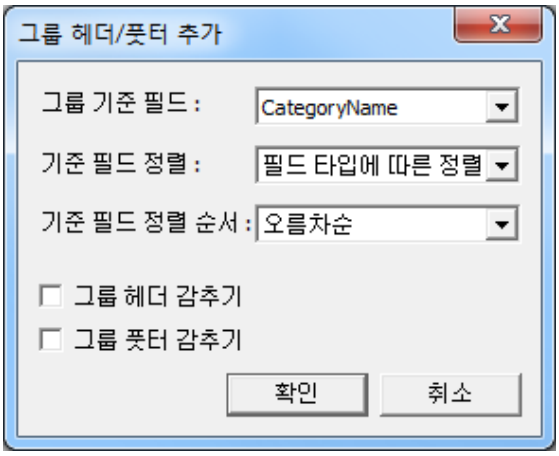
■ 데이터

CategoryName	Description	ProductName	OrderID	OrderDate	City	UnitPrice	Quantity	Amount
해산물	해초류, 생...	노르웨이산 ...	10273	2006-08...	인천...	25000.0...	24	600000.0...
해산물	해초류, 생...	노르웨이산 ...	10276	2006-08...	서울...	25000.0...	15	375000.0...
해산물	해초류, 생...	노르웨이산 ...	10357	2006-11...	서울...	25000.0...	30	750000.0...

■ 보고서 파일 경로

· 내 문서\OZRepository\OZwork\exercise\ex11\제품주문정보.odi, 제품주문정보.ozr

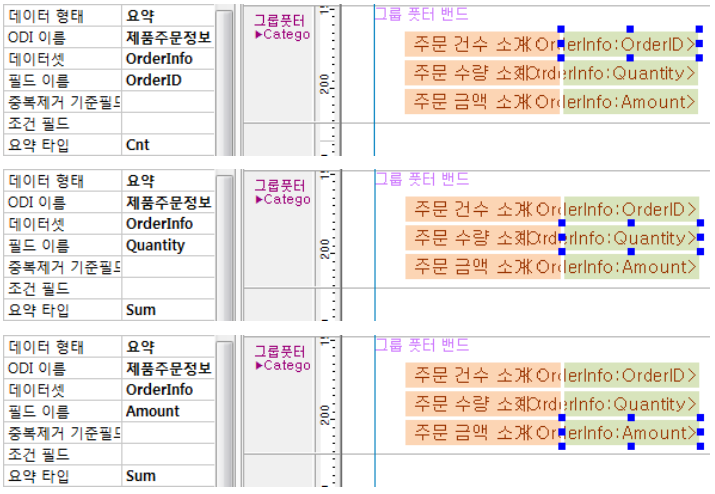
(1) 기준 필드로 그룹 설정



- 데이터 밴드의 팝업 메뉴에서 '밴드 추가' → '상위 그룹 헤더/풋터 밴드' 선택
- '그룹 기준 필드'에 'CategoryName' 선택
- '기준 필드 정렬'을 '필드 타입에 따른 정렬'로 선택

※ '그룹 보고서'는 하나의 데이터 셋 정보를 기준 필드 별로 분류해서 그리는 보고서 형태입니다.

(2) 소계 설정



- 그룹 풋터 밴드 위에 고정 테이블 추가 (2열 3행)
- 소계를 보여줄 라벨의 속성에서 '데이터 형태'를 '요약'으로 변경 후 '필드 이름', '요약 타입' 설정

(3) 디자인 화면

가. 목표 보고서 : 제품 분류 별 주문 정보

2011년03월24일 목 **제품 분류별 주문 정보** 10/61

제품 분류 : 곡류	설명 : 빵, 파스타, 시리얼
주문 건수 : 196	주문 수량 : 4,562 주문 금액 : 100,716,000

제품명	주문ID	주문일	주문지역	단가	수량	금액
한성 통밀가루	10597	2007,07,11	서울특별시	20,000	20	400,000
	10260	2006,07,19	안양시	16,000	50	800,000
	10282	2006,08,15	오산시	16,000	2	32,000
	10355	2006,11,15	인천광역시	16,000	25	400,000
	10368	2006,11,29	인천광역시	16,000	25	400,000
	10416	2007,01,16	성남시	16,000	20	320,000
	10438	2007,02,06	부천시	16,000	15	240,000
	10535	2007,05,13	서울특별시	20,000	5	100,000
	10578	2007,06,24	김해시	20,000	6	120,000
	10582	2007,06,27	서울특별시	20,000	4	80,000
	10251	2006,07,08	인천광역시	16,000	15	240,000

나. 과제

① 제품 분류(CategoryName) 별로 주문 목록 표시 ('마스터 밴드 설정' 이용)

② 매 페이지 마다 제품 분류 정보 상단에 표시 (디테일 밴드의 '마스터 고정' 속성 이용)

③ 제품 분류 별로 페이지 분리하기 (디테일 밴드의 '페이지 넘기기' 속성 이용)

다. 관련 정보

■ **'SalesByCategory' 데이터 셋의 쿼리문과 데이터**

```
SELECT C.CategoryID, C.CategoryName, C.Description, COUNT(OD.OrderID) AS OrderCount,
SUM(OD.Quantity) AS Quantity, SUM(OD.UnitPrice*OD.Quantity) AS Amount
FROM Orders O, Categories C, Products P, [Order Details] OD, Customers Cust
WHERE C.CategoryID = P.CategoryID AND O.OrderID = OD.OrderID
AND Cust.CustomerID = O.CustomerID AND P.ProductID = OD.ProductID
GROUP BY C.CategoryID, C.CategoryName, C.Description
ORDER BY C.CategoryName ASC
```

CategoryID	CategoryName	Description	OrderCount	Quantity	Amount
7	가공 식품	건과류 ,과일 통조림	136	2990.0	105245000.0000
5	곡류	빵, 파스타, 시리얼	196	4562.0	100716000.0000

■ **'SalesDetail' 데이터 셋의 쿼리문과 데이터**

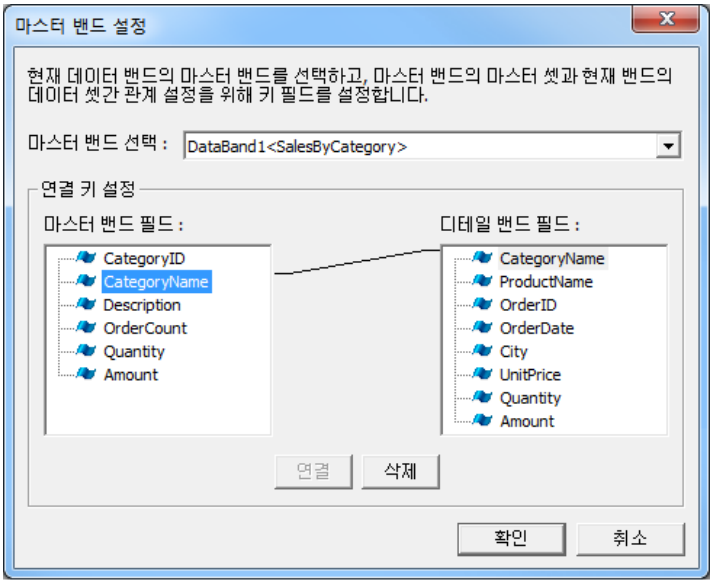
```
SELECT C.CategoryName, P.ProductName, OD.OrderID, O.OrderDate, Cust.City, OD.UnitPrice, OD.Quantity,
(OD.UnitPrice * OD.Quantity) AS Amount
FROM Categories C, Products P, [Order Details] OD, Orders O, Customers Cust
WHERE C.CategoryID = P.CategoryID AND P.ProductID = OD.ProductID
AND OD.OrderID = O.OrderID AND O.CustomerID = Cust.CustomerID
ORDER BY C.CategoryName ASC, P.ProductName ASC
```

CategoryName	ProductName	OrderID	OrderDate	City	UnitPrice	Quantity	Amount
가공 식품	대양 특선 건과(배)	10979	2008-03...	인천광역시	30000.0...	18	540000....
가공 식품	대양 특선 건과(배)	10262	2006-07...	시흥시	24000.0...	15	360000....

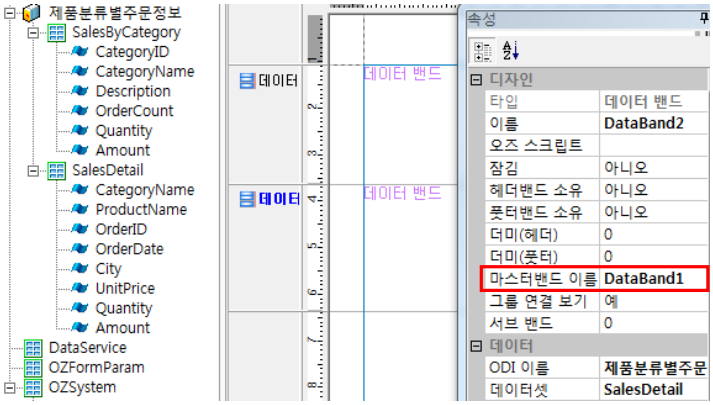
■ **보고서 파일 경로**

· 내 문서\OZRepository\OZwork\exercise\ex12\제품분류별주문정보.odi, 제품분류별주문정보.ozr

(1) 제품 분류 필드로 마스터 밴드 설정



- 데이터 밴드 두 개에 데이터 셋 각각 설정
- 'SalesDetail' 셋을 설정한 데이터 밴드의 팝업 메뉴에서 '마스터 밴드 설정' 선택
- '마스터 밴드 선택'을 'SalesByCategory' 셋을 설정한 데이터 밴드로 선택
- 마스터 밴드의 'CategoryName' 필드를 드래그해서 디테일 밴드의 'CategoryName' 필드에 연결



※ '마스터디테일 보고서'는 두 개의 데이터 셋 정보를 키 필드로 연결시켜서 두 개의 밴드를 교대로 보여주는 보고서 형태입니다.

(2) 디자인 화면

Chapter6. 보고서 배포 및 운영

1. 보고서 업로드
2. DB 커넥션 풀 생성
3. 보고서 출력

가. 리파지토리 정의

'리파지토리'는 보고서와 관련된 파일들(ozr, odi, 이미지 등)을 보관해 둔 저장소입니다.
리파지토리 내의 폴더는 '카테고리', 파일은 '아이템'이라 명명합니다.

- 오즈 서버 리파지토리 기본 경로 : %OZ_HOME%/repository_files/

나. 리파지토리 기능

쿼리 디자이너(odi 파일만 보임), 리포트 디자이너(ozr, odi 파일만 보임), 리파지토리 매니저(모든 파일 보임) 툴을 이용하여 리파지토리를 관리합니다.

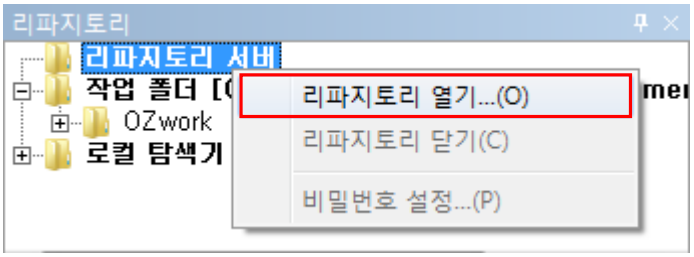
리파지토리는 'None', 'RDB', 'BuiltIn' 세 가지 유형이 있습니다. 유형별로 지원되는 기능에 차이가 있습니다.

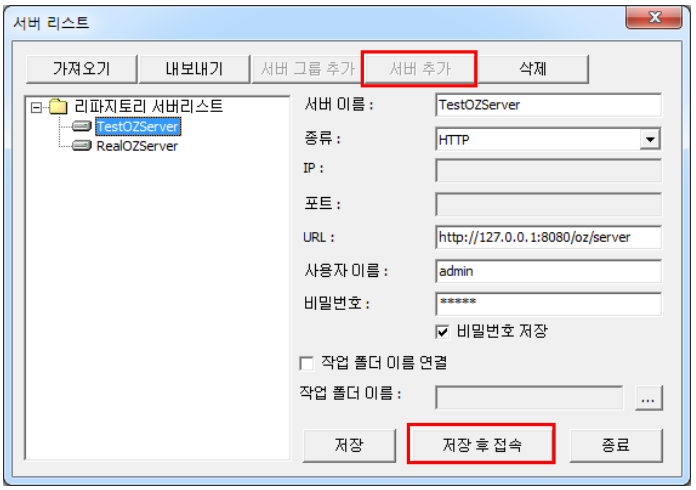
유형	기능
None	리파지토리 기본 유형으로 체크인, 체크아웃, 히스토리, 사용자 관리 등의 리파지토리 기능을 지원하지 않습니다. 그러나 파일 시스템에서 소스를 직접 접근할 수 있는 이점이 있습니다.
RDB	체크인, 체크아웃, 히스토리, 사용자 관리 등의 리파지토리 기능을 지원하며 관련된 정보를 RDB에 저장합니다. 파일 시스템에서 소스에 직접 접근하는 것은 허용되지 않습니다.
BuiltIn	체크인, 체크아웃, 히스토리, 사용자 관리 등의 리파지토리 기능을 지원하며 관련된 정보를 파일에 저장합니다. 파일 시스템에서 소스에 직접 접근하는 것은 허용되지 않습니다.

- 오즈 서버 리파지토리 유형 설정 파일 : %OZ_HOME%/conf/repository.properties

다. 리파지토리 서버 연결

오즈 서버는 사용하는 엔진에 따라 Java와 .Net 서버로 분류되며, 각각 연결 방식에 따라 HTTP와 TCP 로 분류됩니다. HTTP 연결 방식인 경우 오즈 서버의 URL 정보가 필요하며 TCP 연결 방식인 경우 서버의 IP, 포트 정보가 필요합니다.





① 디자이너의 리파지토리 창에서 '리파지토리 서버' 팝업 메뉴의 '리파지토리 열기' 선택

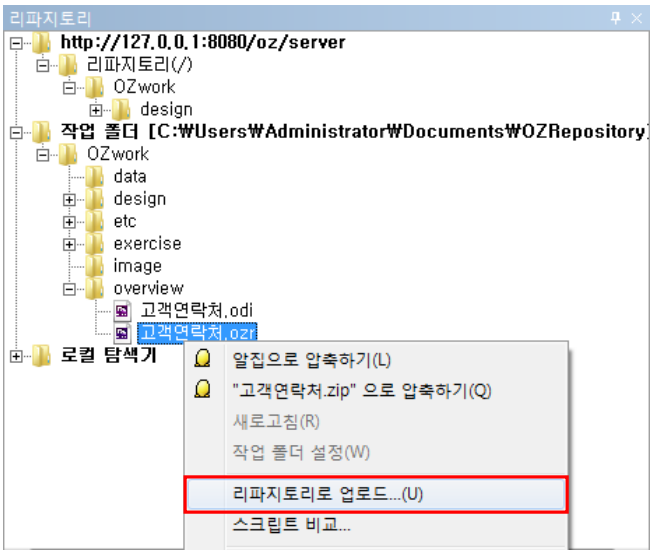
② '서버 추가' 버튼 클릭 후 '종류', 'IP', '포트', 'URL', 사용자 이름, 비밀번호 입력
- TCP일 때 기본 포트 : 8003
- 기본 사용자 : admin (비밀번호 동일)

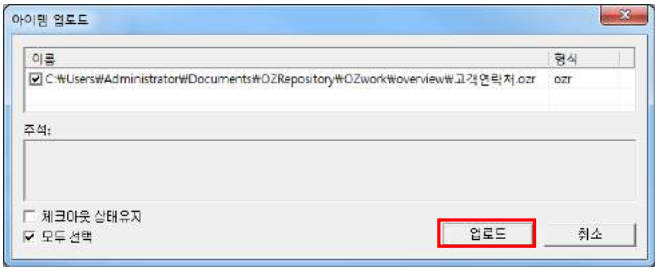
③ '저장 후 접속' 버튼 클릭하면 서버 연결됨

④ '리파지토리 서버' 팝업 메뉴의 '리파지토리 닫기'를 선택하면 서버와의 연결 종료됨

※ '리파지토리 서버' 팝업 메뉴의 '비밀번호 설정'에서 현재 접속한 사용자의 비밀번호를 변경할 수 있습니다.

가. 보고서 파일 업로드

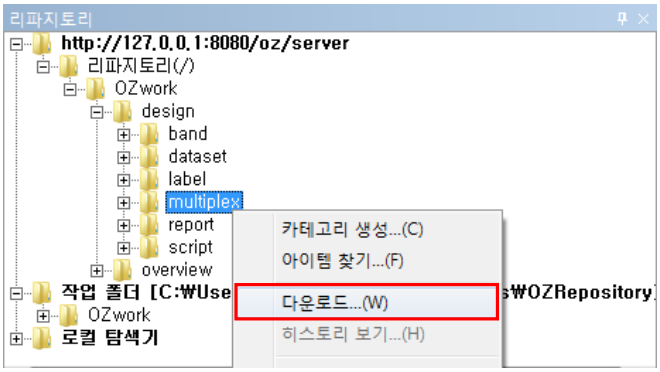


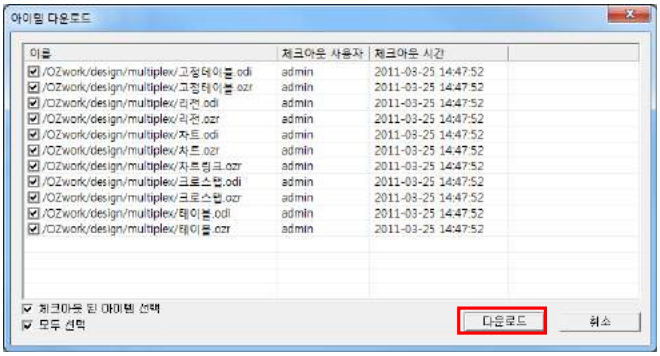


- 디자이너에서 개발된 보고서 파일은 작업 폴더에 있습니다. 보고서를 서비스하기 위해서는 관련 파일을 작업 폴더로부터 연결된 리파지토리 서버로 업로드해야 합니다.

- 파일이나 폴더의 팝업 메뉴에서 '리파지토리 업로드'를 선택하면 서버의 동일 경로에 업로드 됩니다.

나. 보고서 파일 다운로드



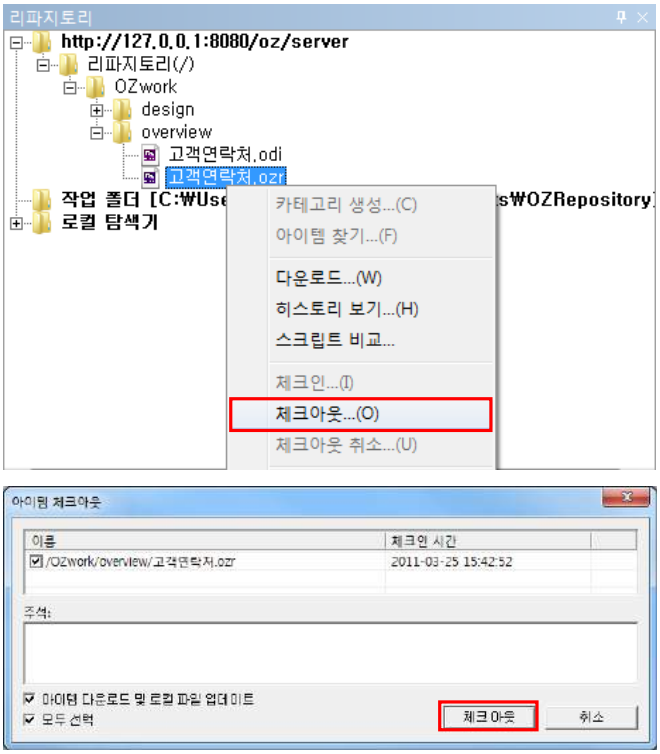


- 오즈 서버의 보고서 파일을 로컬 PC의 작업 폴더로 다운로드 받으려면 해당 파일이나 폴더의 팝업 메뉴에서 '다운로드'를 선택합니다.

- '아이템 다운로드' 창에서 선택된 파일만 다운로드 됩니다.

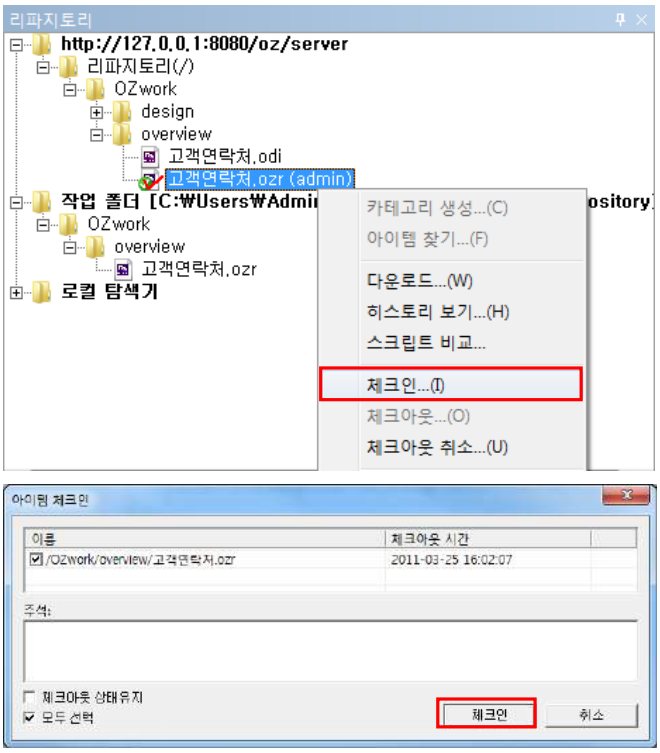
리파지토리 유형이 'RDB' 또는 'BuiltIn'인 경우 체크인, 체크아웃 기능을 이용하여 동일한 보고서 파일을 개발자들이 중복 수정하는 것을 방지할 수 있습니다.

가. 보고서 파일 체크아웃 (편집 권한 취득 및 파일 다운로드)



- 리파지토리의 파일을 체크아웃 하면 다른 사용자가 해당 파일을 변경하지 못합니다.
- 체크아웃한 사용자가 다시 체크인을 해야만 다른 사용자가 파일을 변경할 수 있는 권한이 생깁니다.
- 리파지토리 서버에 연결한 후 편집할 보고서 파일의 팝업 메뉴에서 '체크아웃'을 선택하면 해당 파일이 작업 폴더로 다운로드 됩니다.
- 다운로드 된 파일을 디자이너에서 열어 편집합니다.

나. 보고서 파일 체크인 (파일 업로드 및 편집 권한 반환)



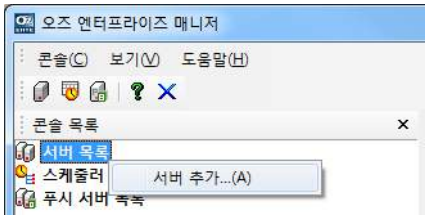
- 디자이너에서 수정한 보고서 파일을 서버에 반영하려면 리파지토리 서버에 연결한 후 해당 파일의 팝업 메뉴에서 '체크인'을 선택하면 작업 폴더의 동일 경로에 있는 파일이 서버로 업로드 됩니다.
- '체크인'은 이미 '체크아웃'된 파일에 대해서만 가능합니다.
- '체크인' 할 때마다 서버에 동일한 파일이 버전 별로 생성되기 때문에 '히스토리 보기' 메뉴에서 다른 버전의 파일을 다운로드 하거나 롤백 할 수 있습니다.

2.1 오즈 서버의 DB 연결 정보 설정

보고서에서 데이터베이스 연결로 데이터를 조회하려면 오즈 서버에 사용할 DB 연결 정보를 설정해야 합니다.
연결 정보 설정은 엔터프라이즈 매니저를 이용할 수 있습니다.

오즈 엔터프라이즈 매니저는 오즈 서버, 오즈 스케줄러, 오즈 푸시 서버를 관리하는 클라이언트 툴입니다. 오즈 서버의 메모리 상태 체크, 리파지토리 설정, 캐시 설정, 커넥션 풀 설정, 로그 설정 및 모니터링 등을 할 수 있습니다.

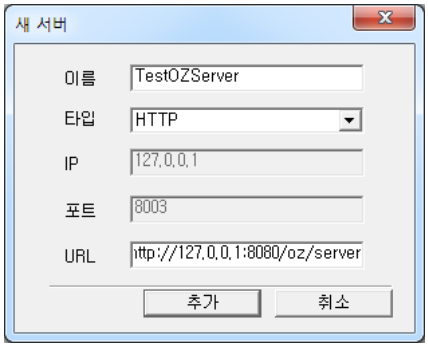
➢ **DB 커넥션 풀 설정 예**



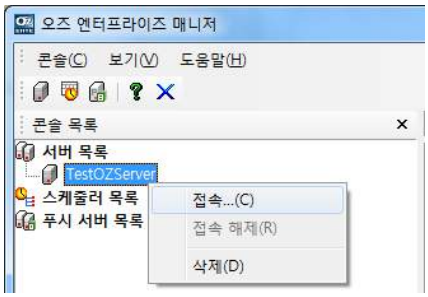
① 바탕화면의 오즈 엔터프라이즈 매니저 아이콘 실행

- 실행 파일 경로 : %ProgramFiles%\FORCS\OZ Report 7.0 \OZ Enterprise Manager 7.0\bin\OZEnterpriseManager.exe

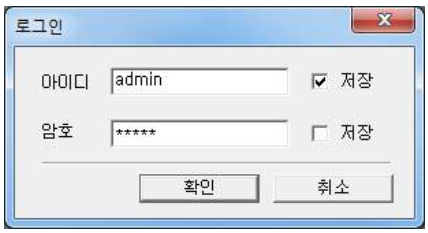
② '서버 목록'의 팝업 메뉴에서 '서버 추가' 클릭



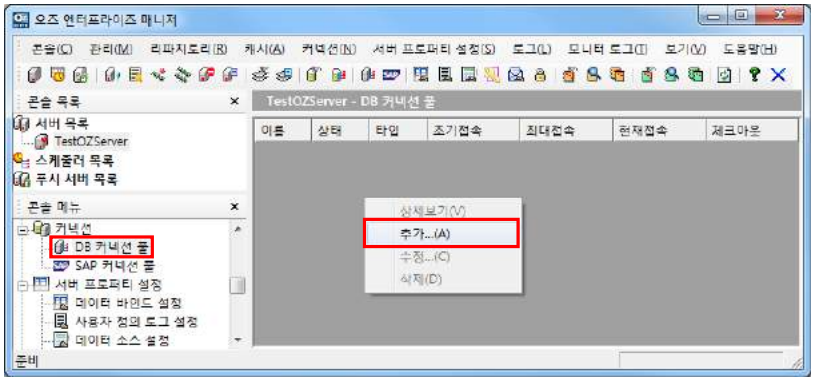
③ '새 서버' 창에서 오즈 서버의 타입을 TCP나 HTTP 중에서 선택하고 관련 IP, 포트 또는 URL 입력



④ 추가된 서버의 팝업 메뉴에서 '접속' 선택

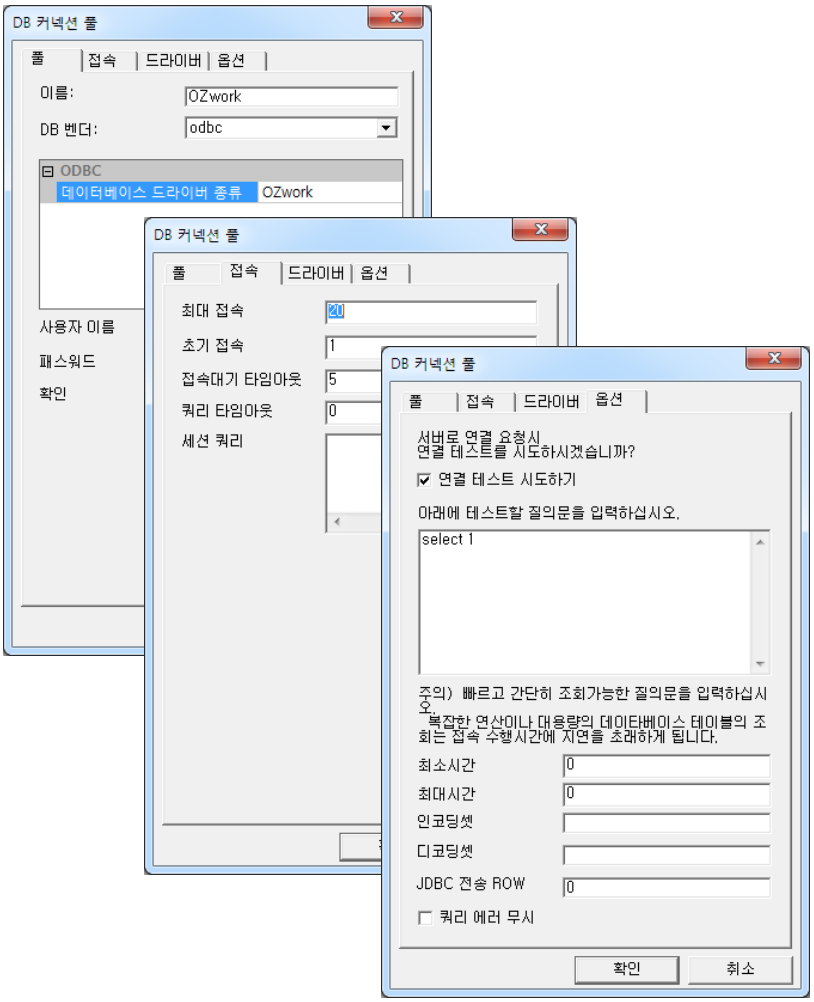


⑤ '로그인' 창에서 아이디, 암호 입력
(오즈 서버가 설치될 때 기본 아이디와 암호는 admin 임)



⑥ 연결된 오즈 서버의 왼쪽 트리 메뉴에서 'DB 커넥션 풀' 선택

⑦ 오른쪽 화면의 팝업 메뉴에서 '추가' 선택



⑧ 'DB 커넥션 풀' 창에 DB 연결 정보 입력

- '이름'에 앨리어스 명 입력
- DB 벤더 선택 후 관련 정보 입력
- DB의 사용자 이름, 패스워드 입력
- DB 재구동, 네크워크 재연결과 같은 이유로 DB와의 연결이 끊어졌을 때 자동으로 다시 연결되게 하려면 '옵션' 탭의 '연결 테스트 시도하기' 체크 후 연결 테스트할 간단한 질의문 입력

⑨ 생성된 DB 커넥션 풀의 '상태'가 'OK'인 경우 오즈 서버와 DB간의 연결이 정상적으로 수립된 것임

```
# db.properties의 OZwork 앨리어스 정보
OZwork.vendor=odbc
OZwork.dsn=OZwork
OZwork.password=
OZwork.user=
OZwork.maxconns=20
OZwork.initconns=1
OZwork.timeout=5
OZwork.doConnectionCheck=true
OZwork.testQueryString=select 1
```

- 수정된 오즈 서버 파일

: %OZ_HOME%/conf/db.properties

배포 마법사는 오즈 보고서를 웹 브라우저로 보여주기 위해 필요한 웹 페이지(HTML, JSP, ASP) 를 자동으로 생성해 줍니다.

가. 배포 마법사 실행

- 리포트 디자이너에서 배포할 보고서를 저장합니다.
- 디자이너 화면 좌측 상단의 OZ 버튼을 클릭한 후 '배포 마법사' 메뉴를 클릭합니다.

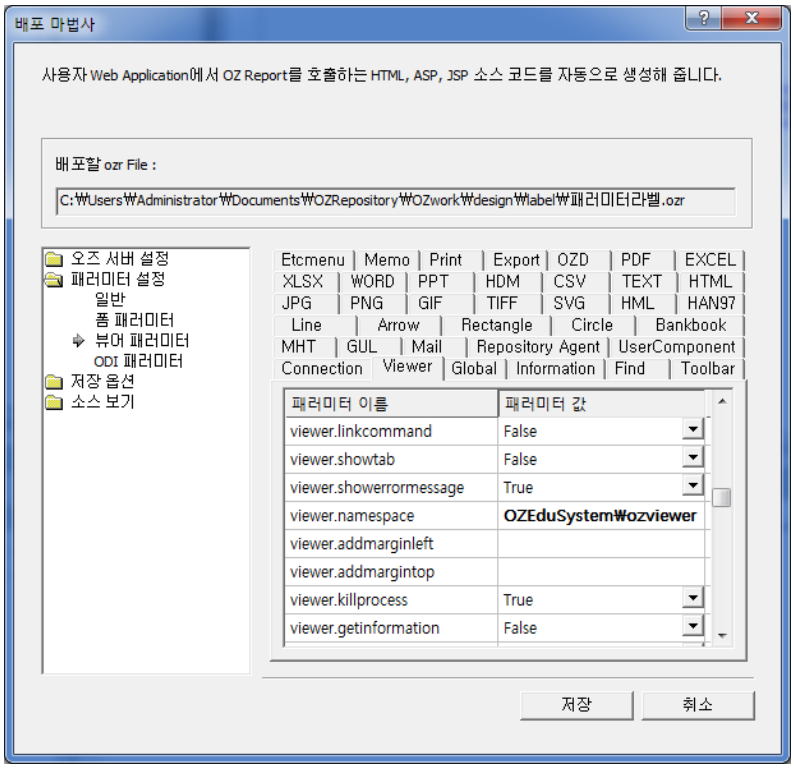
나. 배포 마법사 설정

① 배포할 ozr File : 작성 중인 보고서 파일 전체 경로가 표시됩니다.

② 오즈 서버 설정 : 오즈 서버 형태를 HTTP, TCP 중에서 선택한 후 관련 정보를 설정합니다.

③ 패러미터 설정 : 패러미터 암호화, 폼 패러미터, ODI 패러미터, 뷰어 패러미터를 설정합니다.

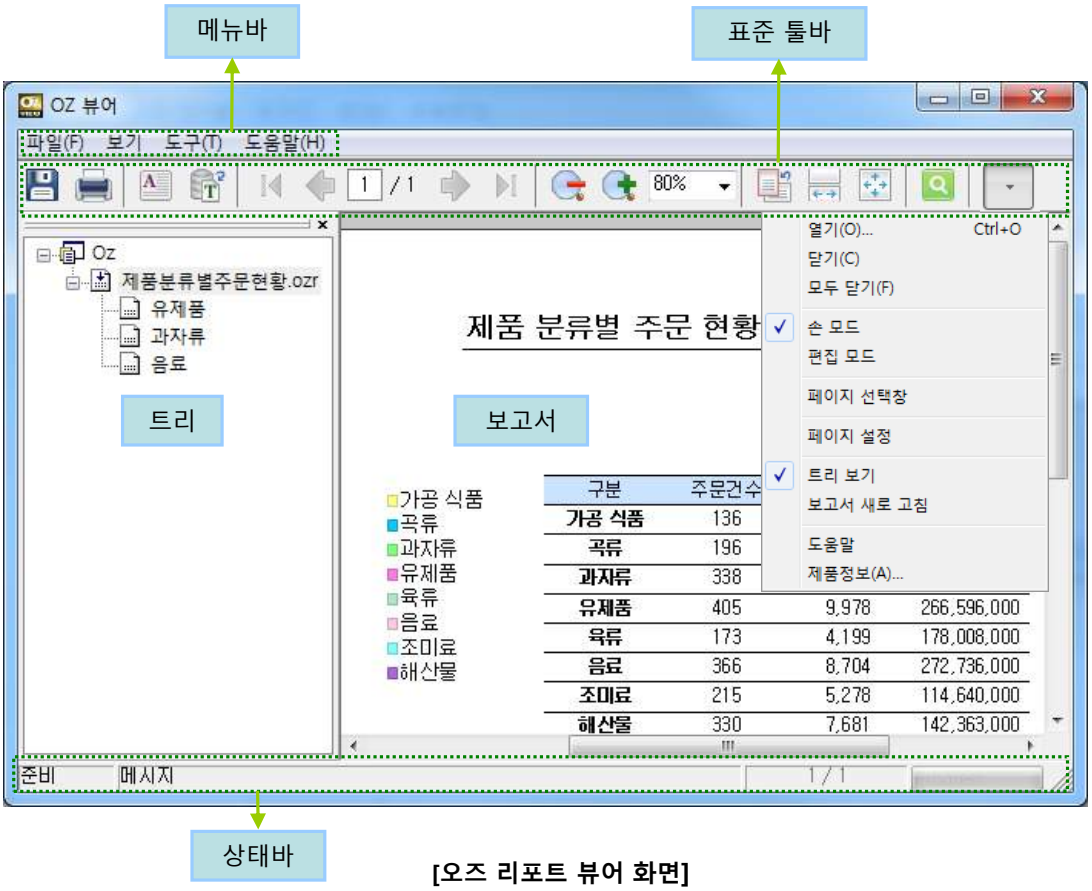
- 일반 : 패러미터 암호화 옵션을 선택할 수 있으며 암호화된 값은 오즈 뷰어와 서버가 복호화하여 자동으로 인식됩니다.
- 폼 패러미터 : 보고서에 추가되어 있는 폼 패러미터 값을 설정합니다.
- ODI 패러미터 : 보고서에 추가된 ODI의 사용자 지정 패러미터 값을 설정합니다.
- 뷰어 패러미터 : 각종 뷰어 패러미터 값을 설정합니다.



- 뷰어 패러미터 종류 (일부)
  - Connection : 서버 통신, 보고서 관련
  - Viewer : 뷰어 기본 옵션
  - Global : 뷰어 일반 옵션
  - Information : 트랜젝션 로그 옵션
  - Toolbar : 뷰어 툴바 관련 옵션
  - Print : 인쇄 관련 옵션
  - Export : 파일 변환 관련 옵션
  - Mail : 메일 관련 옵션
  - OZD : ozd 파일 변환 관련 옵션
  - PDF : pdf 파일 변환 관련 옵션
  - EXCEL : 엑셀 파일 변환 관련 옵션
  - XLSX : xlsx 엑셀 파일 변환 옵션
  - WORD : 워드 파일 변환 관련 옵션
  - PPT : 파워포인트 변환 관련 옵션
  - HTML : HTML 변환 관련 옵션
  - SVG : SVG 변환 관련 옵션
  - PNG : PNG 변환 관련 옵션
  - JPG : JPG 변환 관련 옵션

④ 저장 옵션 : 저장 형식('ASP', 'HTML', 'JSP')과 브라우저 종류, 뷰어 다운로드 위치를 설정합니다.

⑤ 소스 보기 : 작성된 웹 페이지 소스를 미리보기 합니다.



[오즈 리포트 뷰어 화면]

구분	기능		설명
메뉴바	파일 → 열기		로컬 PC의 보고서 파일(ozd)을 열어서 뷰어에 출력
	보기 → 트리 보기		뷰어에 트리 표시
	보기 → 페이지 설정		페이지 여백 설정
	도구 → 손 모드		페이지 이동시 마우스 커서를 손 모양으로 변경
	도구 → 편집 모드		보고서 편집시 마우스 커서를 화살표 모양으로 변경
	도움말 → 도움말		뷰어 소개 및 기능에 대한 도움말
	도움말 → 제품 정보		뷰어의 버전 정보 확인
툴바	저장		출력중인 보고서를 다른 형태의 파일로 변환하여 저장
	보고서 인쇄		보고서를 프린터로 인쇄
	메모 추가		보고서에 메모 컴포넌트 추가 후 텍스트 입력 가능
	데이터 저장		데이터 셋 별로 데이터를 텍스트 파일 형식으로 저장
	용지 방향 전환		보고서 용지 방향을 가로, 세로로 변경
	너비에 맞춰 한 페이지씩 연속 보기		뷰어 너비에 맞춰 보여짐. 스크롤로 페이지 이동 가능
	화면 크기에 맞춰 한 페이지씩 보기		뷰어 크기에 맞춰 보여짐. 한 페이지씩 페이지 이동
	찾기		보고서의 전체 페이지에서 문자열 검색
	다른 메뉴 → 보고서 새로 고침		일정 주기마다 보고서 새로 그림. 데이터도 갱신됨

가. ActiveX 뷰어 설치 태그

뷰어 설치에 필요한 <object> 태그입니다. 오즈 뷰어는 Ztransfer Control에 의해 자동 설치 및 업데이트 됩니다.

```
<object id="ZTransferX" width="0" height="0" CLASSID="CLSID:C7C7225A-9476-47AC-B0B0-FF3B79D55E67"
codebase="http://127.0.0.1:8080/oz/OZViewerActiveX/ZTransferX.cab#version=2,2,5,1">
    <param name="download.Server" value="http://127.0.0.1/oz/OZViewerActiveX">
    <param name="download.Port" value="8080">
    <param name="download.Instruction" value="ozrviewer.idf">
    <param name="install.Base" value="<PROGRAMS>/Forcs">
    <param name="install.NameSpace" value="OZEduSystem">
</object>
```

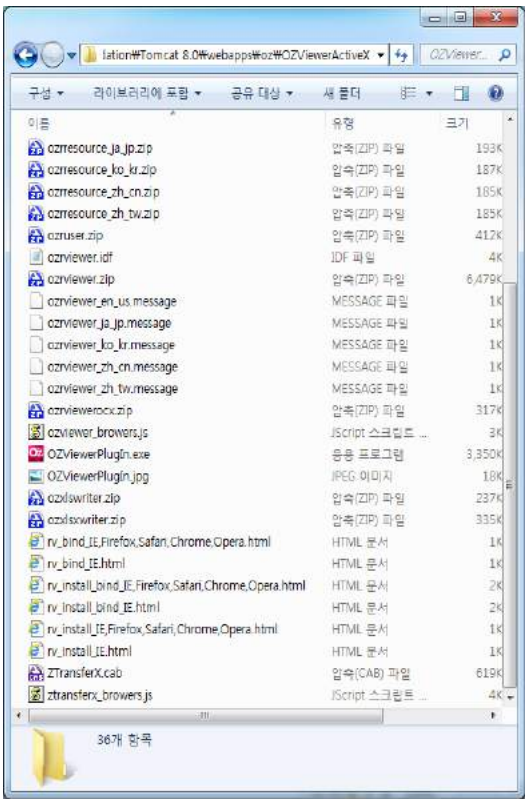
- **패러미터 설명**
  - object 태그의 codebase 속성에는 Ztransfer의 URL과 버전이 명시됩니다.
  - download.Server, download.Port, download.Instruction의 value를 합치면 오즈 버전 정보 파일의 URL이 됩니다. 이 URL이 인식되어야 뷰어 설치가 가능합니다.
    예) http://127.0.0.1:8080/oz/OZViewerActiveX/ozrviewer.idf
  - 뷰어 설치 경로는 *[install.Base]*/OZ Family/*[install.Namespace]*/ozviewer 입니다.

- **뷰어 설치 순서**

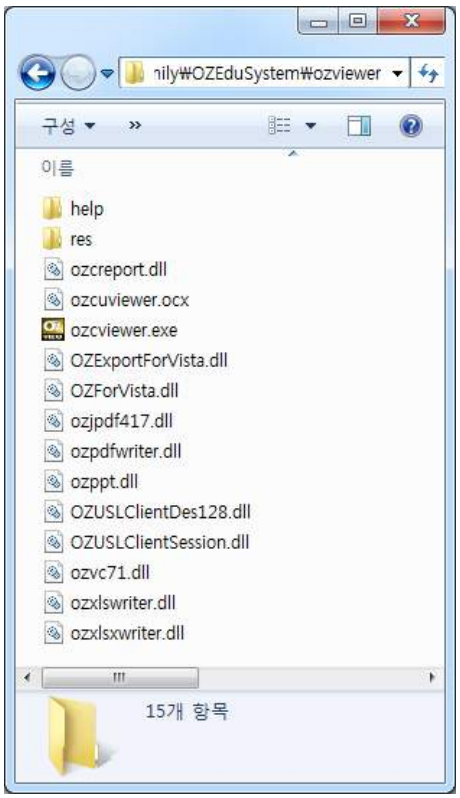
Ztransfer 설치 → Ztransfer 실행 → 뷰어 다운로드 → 뷰어 설치

- **뷰어 파일**

• 웹 서버의 오즈 뷰어 파일(설치될 파일)



• 사용자 PC에 설치된 오즈 뷰어 파일



나. ActiveX 뷰어 실행 태그

```
<object id="ozviewer" width="800" height="600" CLASSID="CLSID:0DEF32F8-170F-46f8-B1FF-4BF7443F5F25">
   <param name="connection.servlet" value="http://127.0.0.1:8080/oz/server">
   <param name="connection.reportname" value="/OZwork/design/label/패러미터라벨.ozr">
   <param name="viewer.configmode" value="html">
   <param name="viewer.namespace" value="OZEduSystem₩ozviewer">
   <param name="viewer.isframe" value="false">
   <param name="viewer.mode" value="preview">
   <param name="information.debug" value="true">
   <param name="connection.pcount" value="2">
   <param name="connection.args1" value="username=사용자1">
   <param name="connection.args2" value="imgurl=http://oztn.net/kb/img/common/main/bottomLogo_n.gif">
   <param name="odi.odinames" value="패러미터라벨">
   <param name="odi.패러미터라벨.pcount" value="2">
   <param name="odi.패러미터라벨.args1" value="FromDate=2010-02-01">
   <param name="odi.패러미터라벨.args2" value="ToDate=2010-05-31">
   <param name="odi.패러미터라벨.clientdmtype" value="Memory">
   <param name="odi.패러미터라벨.serverdmtype" value="Memory">
   <param name="odi.패러미터라벨.fetchtype" value="Concurrent">
</object>
```

- **패러미터 설명**

- object 태그의 뷰어 id는 스크립트에서 뷰어를 참조할 경우 사용
- object 태그의 width, height는 뷰어가 웹 페이지에 embed되는 경우의 뷰어 너비와 높이
- connection.servlet : 오즈 서버의 URL (오즈 서버가 TCP 타입인 경우 connection.server, connection.port 사용)
- connection.reportname : 뷰어에 출력할 리포트 파일 경로
- viewer.namespace : 로컬에 설치된 뷰어 중에서 실행할 뷰어 이름 (설치한 뷰어 이름 지정)
- viewer.isframe : 뷰어를 브라우저에 분리해서 실행할지 여부 설정
  - 'true' : 뷰어를 별도 창으로 표시. 브라우저와 별개의 프로세스로 실행됨 (기본값)
  - 'false' : 뷰어를 브라우저에 임베디드된 형태로 표시. 브라우저와 동일한 프로세스로 실행됨
  - 'inprocess' : 뷰어를 브라우저와 별도 창으로 표시하지만 브라우저와 동일한 프로세스로 실행됨
- viewer.mode : 뷰어 실행 모드
  - 'preview' : 뷰어에 보고서 미리보기 됨 (기본값)
  - 'print' : 보고서 미리보기 없이 바로 프린터로 인쇄
  - 'export' : 보고서 미리보기 없이 다른 형태의 파일로 저장
- information.debug : 뷰어 콘솔 로그를 남기는 수준 설정 ('Ctrl+Z' 단축키로 뷰어 콘솔 창 띄운 후 로그 확인)
- connection.pcount : 폼 패러미터 개수
- connection.args# : 폼 패러미터 이름과 값을 '=' 로 구분해서 입력하고 '#'은 1부터 순차적으로 증가
- odi.odinames : 리포트에 추가되어 있는 odi 이름 (여러 개일 경우 ',' 로 구분)
- odi.odi명.pcount : 'odi명'에 설정한 odi의 사용자 지정 패러미터 개수
- odi.odi명.args# : 사용자 지정 패러미터의 이름과 값을 '=' 로 구분해서 입력하고 '#'은 1부터 순차적으로 증가
- odi.odi명.fetchtype : 서버에서 실시간으로 데이터를 전송 받으려면 'Concurrent'로 설정 (첫 페이지 먼저 출력)

- **뷰어 실행 순서**

다. Flash 뷰어 다운로드 및 실행

Flash 뷰어는 사용자 PC에 설치된 Adobe Flash Player로 실행됩니다. Player가 없으면 설치 안내 메시지가 나옵니다.

```
<script src="http://127.0.0.1:8080/ozrviewer/AC_OETags.js" language="javascript"></script>
<script src="http://127.0.0.1:8080/ozrviewer/ozutil.js" language="javascript"></script>
<script src="http://127.0.0.1:8080/ozrviewer/ozjscript.js" language="javascript"></script>
<script language="JavaScript" type="text/javascript">
var requiredMajorVersion = 10; var requiredMinorVersion = 0; var requiredRevision = 0;
var hasProductInstall = DetectFlashVer(6, 0, 65);
var hasRequestedVersion = DetectFlashVer(requiredMajorVersion, requiredMinorVersion, requiredRevision);
if(hasProductInstall && !hasRequestedVersion) {
   var MMPlayerType = (isIE == true) ? "ActiveX" : "PlugIn";
   var MMredirectURL = window.location;
   document.title = document.title.slice(0, 47) + " - Flash Player Installation";
   var MMdoctitle = document.title;
   AC_FL_RunContent("src", "http://127.0.0.1:8080/ozrviewer/playerProductInstall",  ...(생략) );
} else if(hasRequestedVersion) {
   function SetOZParamters_OZViewer() {
      var oz;
      if(navigator.appName.indexOf("Microsoft") != -1) { oz = window["OZViewer"];
      } else { oz = document["OZViewer"]; }
      oz.sendToActionScript("connection.servlet", "http://127.0.0.1:8080/oz/server");
      oz.sendToActionScript("connection.reportname", "category_path/sample.ozr");
      return true;
   }
   AC_FL_RunContent(
      "src", "http://127.0.0.1:8080/ozrviewer/OZViewer10",
      "width", "100%",
      "height", "100%",
      "align", "middle",
      "id", "OZViewer",
      "quality", "high",
      "bgcolor", "#ffffff",
      "name", "OZViewer",
      "allowScriptAccess", "sameDomain",
      "type", "application/x-shockwave-flash",
      "pluginspage", "http://www.adobe.com/go/getflashplayer",
      "flashVars", "flash.objectid=OZViewer"
   );
} else { ...  <a href=http://www.adobe.com/go/getflash/>Get Flash</a>  ...(생략) }
</script>
```

- **호출 소스 설명**
  - 오즈 뷰어의 AC_OETags.js, ozutil.js, ozjscript.js 파일 경로를 설정합니다.
  - AC_FL_RunContent 함수를 이용하여 오즈 뷰어 swf 파일의 경로, 뷰어 크기, 뷰어 ID 등을 지정하고 실행합니다.
  - SetOZParamters_뷰어ID 함수를 이용하여 오즈 서버 경로, 보고서 경로와 같은 뷰어 패러미터를 설정합니다.

※ 전체 스크립트는 OZ Flash 뷰어 폴더의 sample.html 샘플 파일 참조

라. HTML5 뷰어 다운로드 및 실행

HTML5를 지원하는 웹 브라우저에서 실행됩니다.

```
<!DOCTYPE html>
<html style="height:100%">
<head>
<meta http-equiv="X-UA-Compatible" content="IE=edge"/>
<script src="http://code.jquery.com/jquery-1.8.3.min.js"></script>
<link rel="stylesheet" href="http://code.jquery.com/ui/1.9.2/themes/base/jquery-ui.css" type="text/css"/>
<script src="http://code.jquery.com/ui/1.9.2/jquery-ui.min.js"></script>
<link rel="stylesheet" href="http://127.0.0.1:8080/ozrviewer/ui.dynatree.css" type="text/css"/>
<script type="text/javascript" src="http://127.0.0.1:8080/ozrviewer/jquery.dynatree.js" charset="utf-
8"></script>
<script type="text/javascript" src="http://127.0.0.1:8080/ozrviewer/OZJSViewer.js" charset="utf-8"></script>
</head>

<body style="width:98%;height:98%">
<div id="OZViewer" style="width:98%;height:98%"></div>
<script type="text/javascript" >
  function SetOZParamters_OZViewer(){
      var oz;
      oz = document.getElementById("OZViewer");
      oz.sendToActionScript("connection.servlet","http://127.0.0.1:8080/oz/server");
      oz.sendToActionScript("connection.reportname","category_path/sample.ozr");
      return true;
  }
  start_ozjs("OZViewer","http://127.0.0.1:8080/ozrviewer/");
</script>
</body>
</html>
```

- **호출 소스 설명**
  - <!DOCTYPE html> : HTML5 문서라는 표시
  - html, body, div의 width, height 설정 : 오즈 뷰어 div의 크기가 %인 경우 body, html 크기가 지정되어야 함
  - meta 태그 : 렌더링하는 IE 버전 지정. Edge면 현재 설치된 ie로 표시할 수 있는 최신 버전으로 렌더링
  - 뷰어가 참조하는 jquery와 뷰어 모듈 js, css 지정
  - div id="OZViewer", getElementById : 뷰어 아이디
  - SetOZParamters_뷰어ID 함수 : 오즈 서버 경로, 보고서 경로 등 뷰어 패러미터 설정
  - start_ozjs 함수 : 뷰어 실행

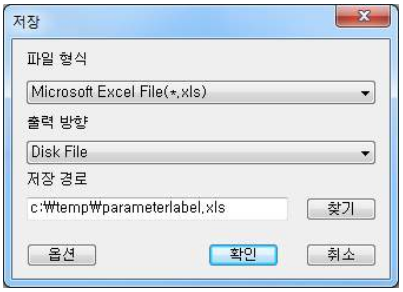
- **HTML5 뷰어 환경**
  - HTML5는 IE, Firefox, Chrome, Safari, Opera에서 지원되며 IE는 9이상 지원됩니다.
  - HTML5/Flash 기술은 크로스 도메인 환경 영향을 받습니다.
  - 클라이언트 환경의 폰트 영향을 최소화 하려면 viewer.fontdpi=auto로 하면 됩니다.

가. 미리보기 없이 보고서 인쇄

```
viewer.useprogressbar=false        <!-- 뷰어 실행시 프로그레스 바 감춤 -->
viewer.mode=print                  <!-- 미리보기 없이 바로 인쇄 -->
print.mode=true                    <!-- 인쇄창 표시 여부 : true(기본값)/showprogress/silent -->
print.copies=2                     <!-- 인쇄 매수 설정 -->
print.once=false                   <!-- 보고서 인쇄를 한번만 할지 여부 -->
print.printername=DEFAULT_PRINTER      <!-- 인쇄할 프린터명 -->
print.size=A4                      <!-- 인쇄 용지 크기 -->
```

나. 미리보기 없이 보고서를 다른 형태의 파일로 저장

```
viewer.mode=export                 <!-- 미리보기 없이 바로 저장 -->
export.mode=normal                 <!-- 저장 창 출력 여부 : normal/silent -->
export.confirmsave=true            <!-- silent일 때 저장 확인창 출력 여부 -->
export.applyformat=xls,xlsx,pdf,doc,csv,png,gif
<!-- 저장 창에 보여줄 파일 형식 목록 -->
export.format=xls                  <!-- 저장할 파일 형식 -->
export.path=c:₩temp₩               <!-- 파일 저장 위치 -->
export.filename=parameterlabel.xls     <!-- 저장할 파일명 -->
excel.removeblank=true             <!-- 페이지 사이 공백 제거 여부 -->
excel.numberformat=#,##0.00        <!-- 숫자 데이터에 대한 엑셀 셀 서식 설정 -->
```



다. 동시에 보고서 인쇄, 저장, 미리보기

보고서를 미리보기 하면서 동시에 인쇄를 하거나 다른 파일로 저장하려면 'viewer.mode' 패러미터에 'preview', 'export', 'print'를 ',' 나 '/'로 구분하여 여러 개를 설정해야 합니다. 'print' → 'export' → 'preview' 순서로 실행됩니다.

```
viewer.mode=print,export,preview
```

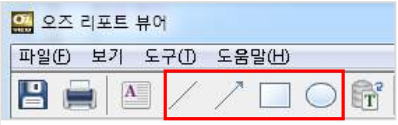
TIP

보고서 미리보기 없이 인쇄, 저장의 동작은 ActiveX, Applet, WPF 뷰어에서 가능합니다.

라. 콜아웃

보고서를 미리보기 한 후 선, 화살표, 사각형, 원을 추가하여 글 상자를 추가할 수 있습니다.

```
toolbar.addline=true               <!-- 툴바에 선 추가 아이콘 표시 -->
toolbar.addarrow=true              <!-- 툴바에 화살표 추가 아이콘 표시 -->
toolbar.addrectangle=true          <!-- 툴바에 사각형 추가 아이콘 표시 -->
toolbar.addcircle=true             <!-- 툴바에 원 추가 아이콘 표시 -->
```

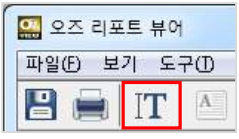


마. 텍스트 선택 모드

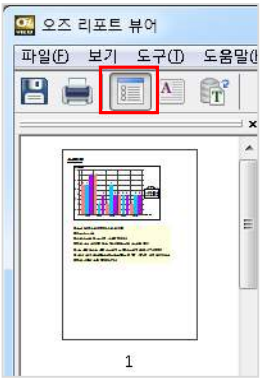
'텍스트 선택 모드'를 선택하고 뷰어에 표시된 텍스트를 복사하면 클립보드에 내용이 복사됩니다.

'텍스트 선택 모드'는 뷰어의 도구 메뉴, 팝업 메뉴, 다른 메뉴, 툴바에 활성화시켜 사용할 수 있습니다.

```
viewer.usetextselectmenu=true      <!-- 도구 메뉴와 팝업 메뉴에 표시 -->
toolbar.textselecttool=true        <!-- 툴바에 아이콘 추가 -->
```



바. 페이지 축소판 보기



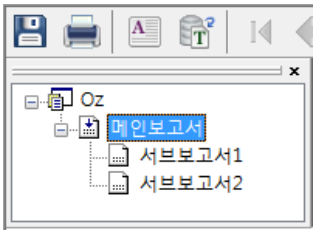
'페이지 축소판 보기' 를 선택하면 왼쪽 트리 창에 썸네일이 표시됩니다.

뷰어의 보기 메뉴, 다른 메뉴, 툴바에서 선택이 가능합니다.

```
viewer.showtree=true <!-- 보고서 바인딩 후 트리 창 바로 표시 -->
viewer.showthumbnail=true <!--보고서 바인딩 후 트리 창에 썸네일 표시-->
toolbar.showthumbnail=true <!-- 툴바에 페이지 축소판 보기 아이콘 표시 -->
```

사. 멀티 보고서

여러 보고서를 한 뷰어에 보여주는 것을 멀티 보고서라고 합니다.




```
viewer.childcount=2          <!-- 메인 보고서 외의 보고서 개수 -->
viewer.ismultidocassub=true  <!-- 차일드가 메인 보고서에 종속 여부 -->
viewer.focus_doc_index=0 <!-- 맨 위에 표시할 보고서번호(0:메인,-1:마지막)-->
viewer.showtree=true         <!-- 트리 보기 -->
global.concatpage=true       <!-- 한 보고서처럼 페이지 번호가 연결됨 -->
connection.displayname=메인보고서   <!-- 트리에 보여질 보고서 이름 -->
connection.servlet=http://127.0.0.1:8080/oz/server
connection.reportname=/OZwork/design/band/밴드개요.ozr
child1.connection.displayname=서브보고서1
child1.connection.servlet=http://127.0.0.1:8080/oz/server
child1.connection.reportname=/OZwork/design/band/더미밴드.ozr
child2.connection.displayname=서브보고서2
child2.connection.servlet=http://127.0.0.1:8080/oz/server
child2.connection.reportname=/OZwork/design/band/사이드밴드.ozr
```

아. 뷰어 이벤트

오즈 뷰어에서 발생하는 이벤트를 외부 프로그램에서 함수로 받아 처리할 수 있습니다.

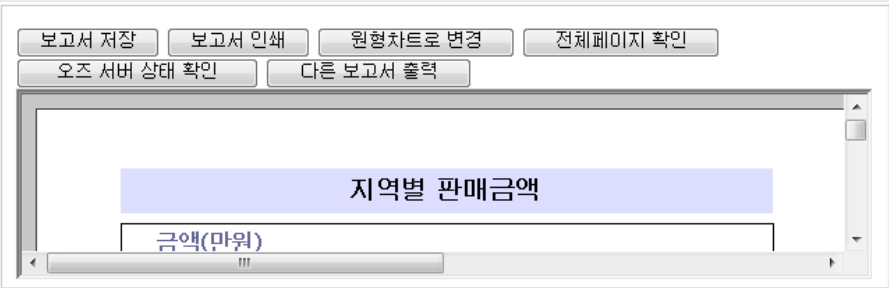
```
<!-- 프린트 이후 호출되는 이벤트  -->
<script language="JavaScript" for="ozviewer" event="OZPrintCommand(msg, code, reportname, printername,
printcopy, printpages, printrange, username)">
        if (code == "0") { // "1" : 출력실패, "0" : 출력성공
            alert("<OZPrintCommand> "+reportname+" 보고서를 " + printcopy + "매 인쇄하였습니다.");
        }
</script>
<object id="ozviewer" width="100%" height="100%" CLASSID="CLSID:0DEF32F8-170F-46f8-B1FF-
4BF7443F5F25">
        <param name="connection.servlet" value="http://127.0.0.1:8080/oz51/server">
        <param name="connection.reportname" value="/OZwork/exercise/ex5/지역별자동차판매현황.ozr">
        <param name="viewer.namespace" value="OZEduSystem₩ozviewer">
        <param name="viewer.isframe" value="false">
        <param name="viewer.printcommand" value="true"> <!-- 뷰어 인쇄 이벤트 사용 -->
        <param name="print.lockopt" value="true"> <!-- 인쇄 속성 사용자 변경 못하게 함 -->
        <param name="print.once" value="true"> <!-- 한번만 인쇄 가능 -->
</object>
```

이벤트 시점	뷰어 패러미터	함수
미리보기 완료 후	viewer.postcommand=cmd;msg	OZPostCommand(cmd, msg)
보고서 생성 중	viewer.progresscommand=true	OZProgressCommand(step, state, reportname)
보고서 인쇄 후	viewer.printcommand=true	OZPrintCommand(msg, code, reportname, printername, printcopy, printedpage, printrange, username, drivernaame)
보고서 저장 후	viewer.exportcommand=true	OZExportCommand(code, path, filename)
메일 전송 후	viewer.mailcommand=true	OZMailCommand(code)
링크 발생 시 (setLinkEvent 실행시)	viewer.linkcommand=true	OZLinkCommand(docindex, componentname, usertag, uservalue, mousebutton)
에러 발생 시	viewer.errorcommand=true	OZErrorCommand(code, msg, detailmsg)
특정 기능 실행 시	viewer.useractioncommand=true	OZUserActionCommand(type, attr)
별도 창으로 띄운 뷰어 종료 시	viewer.exitcommand=true	OZExitCommand()
화면에 표시되는 페이지가 변경된 후 발생	viewer.pagechangecommand=true	OZPageChangeCommand(docindex)
화면에 표시되는 보고서가 변경된 후 발생	viewer.reportchangecommand=true	OZReportChangeCommand(docindex)

자. 외부 프로그램에서 뷰어 기능 제어하기

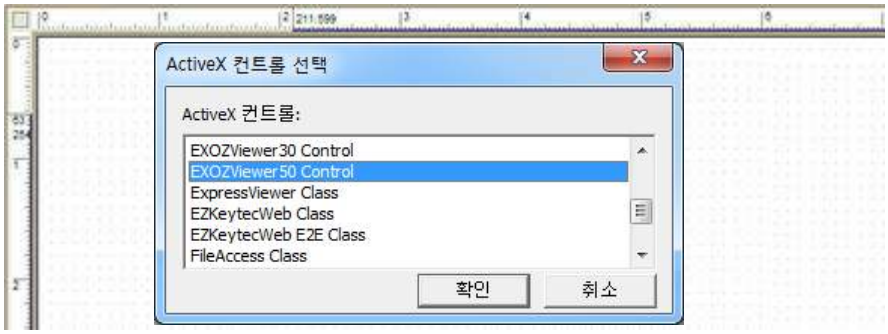
오즈 뷰어의 메뉴나 툴바의 기능을 외부 프로그램에서 제어할 수 있습니다.

```
<input type="button" value="보고서 저장" onclick="document.ozviewer.Script('save')">
<input type="button" value="보고서 인쇄"
  onclick="document.ozviewer.ScriptEx('print','print.mode=false;print.copies=2',';')"> <!-- 2매 바로 인쇄 -->
<input type="button" value="원형차트로 변경" onclick="document.ozviewer.Document.SetChartStyle('Pie')">
<input type="button" value="전체페이지 확인"
  onclick="alert(document.ozviewer.GetInformation('TOTAL_PAGE') + ' 페이지')">
<input type="button" value="오즈 서버 상태 확인"
  onclick="if(document.ozviewer.Document.PingOZServer('http://127.0.0.1:8080/oz/server')) {alert('정상') }
          else { alert('비정상') }">
<input type="button" value="다른 보고서 출력"
onclick="document.ozviewer.CreateReportEx('connection.servlet=http://127.0.0.1:8080/oz/server₩nconnection.reportname=/OZwork/overview/고객연락처.ozr₩nviewer.namespace=OZEduSystem₩₩ozviewer', '₩n')"> <!-- 설정한 뷰어 패러미터로 뷰어에서 보고서 다시 바인딩 -->
<object id="ozviewer" width="100%" height="100%" ... <!-- 보고서 호출 -->
<param name="connection.servlet" value="http://127.0.0.1:8080/oz/server">
<param name="connection.reportname" value="/OZwork/exercise/ex7/지역별판매금액.ozr">
<param name="viewer.usemenubar" value="false"> <!-- 뷰어 메뉴바 숨기기 -->
<param name="viewer.usetoolbar" value="false"> <!-- 뷰어 툴바 숨기기 -->
```



VB, 델파이, 닷넷, 마이플랫폼, 트러스트폼, 파워빌더와 같은 UI 툴에서 오즈 리포트 뷰어 ActiveX 컨트롤을 추가하거나 오즈 리포트 뷰어 프로세스를 실행하여 오즈 보고서를 출력할 수 있습니다.

- **ActiveX 컨트롤로 뷰어 추가 후 보고서 출력**



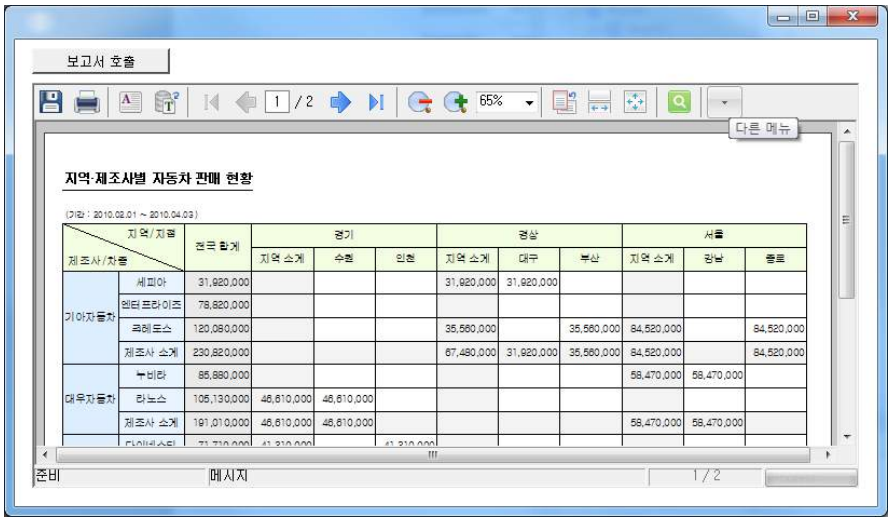
① ActiveX 컨트롤 추가하는 창 띄우기

② 디자인 화면에 'EXOZViewer50 Control' 추가

③ 보고서 호출하는 버튼 추가

```
var params = "connection.servlet=http://127.0.0.1:8080/oz/server" + "***";
params = params + "connection.reportname=/OZwork/exercise/ex6/지역제조사별자동차판매현황.ozr" + "***";
params = params + "viewer.namespace=OZEduSystem₩₩ozviewer" + "***";
params = params + "odi.odinames=지역제조사별자동차판매현황" + "***";
params = params + "odi.지역제조사별자동차판매현황.pcount=2" + "***";
params = params + "odi.지역제조사별자동차판매현황.args1=FromDate=2010-02-01" + "***";
params = params + "odi.지역제조사별자동차판매현황.args2=ToDate=2010-04-03" + "***";
ozviewer.CreateReportEx(params, "***"); // ozviewer는 추가된 ActiveX 컨트롤
```

④ 버튼의 클릭 이벤트에 보고서 보여주는 스크립트 작성



⑤ 미리보기 후 버튼 클릭하면 보고서 출력됨

- **프로세스 형태로 뷰어 실행**

```
// 프로세스 실행하는 함수와 문법은 UI제품마다 다릅니다.
var params = "connection.servlet=http://127.0.0.1:8080/oz/server" + "₩n";
params = params + "connection.reportname=/OZwork/보고서명.ozr" + "₩n";
_RunProcess("C:₩₩Program Files₩₩FORCS₩₩OZ Family₩₩OZEduSystem₩₩ozviewer₩₩ozcviewer.exe", "/string " + params)
```

① 보고서 호출하는 이벤트에 프로세스 실행하는 스크립트 작성

Chapter7. 기타 데이터 처리

1. 서버 바인딩 데이터
2. 뷰어 바인딩 데이터

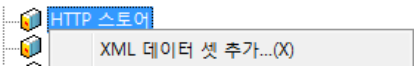
URL로 접근 가능한 XML이나 CSV 형식의 데이터를 보고서에 출력하고자 할 때 HTTP 스토어를 이용합니다.

- XML 데이터 : 지정된 형태의 XML 파일만 인식
- CSV 데이터 : 구분자로 분리된 텍스트 데이터

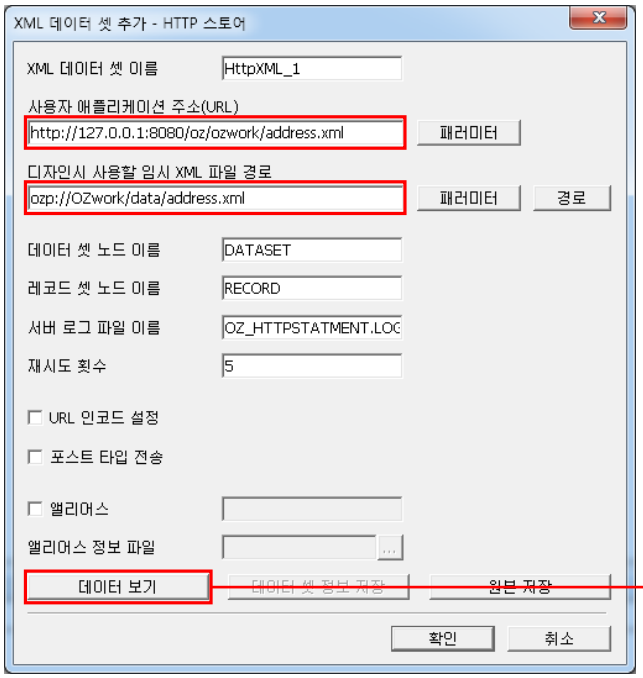
가. HTTP 스토어 - XML 데이터 셋

```xml
<?xml version="1.0" encoding="EUC-KR"?>
<XMLSET>
    <DATASET>
        <FIELDINFO>
            <FIELD NAME="ID" TYPE="INT"/>
            <FIELD NAME="NAME" TYPE="CHAR"/>
            <FIELD NAME="ADDRESS" TYPE="VARCHAR"/>
        </FIELDINFO>
        <RECORD>
            <ID>1</ID>
            <NAME>Park Chan Ho</NAME>
            <ADDRESS>Texas</ADDRESS>
        </RECORD>
        <RECORD>
            <ID>2</ID>
            <NAME>Shin Ki Sung</NAME>
            <ADDRESS>Seoul</ADDRESS>
        </RECORD>
        <RECORD>
            <ID>3</ID>
            <NAME>Kim Byung Hyun</NAME>
            <ADDRESS>Arizona</ADDRESS>
        </RECORD>
    </DATASET>
</XMLSET>
```

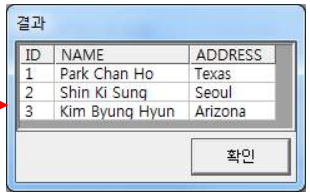
- XML 데이터 구조는 데이터 셋 노드 'DATASET' 과 레코드셋 노드 'RECORD', 필드 정보 노드인 'FIELDINFO' 로 이루어 집니다.
- FIELDINFO 노드를 생략하면 모든 필드가 Varchar 타입으로 인식됩니다.
- RECORD 노드와 FIELDINFO 노드 중에 하나는 있어야 인식됩니다.



- HTTP 스토어의 팝업 메뉴에서 'XML 데이터 셋 추가' 를 선택하면 나오는 'XML 데이터 셋 추가' 창에서 경로를 입력하여 데이터 셋을 추가합니다.



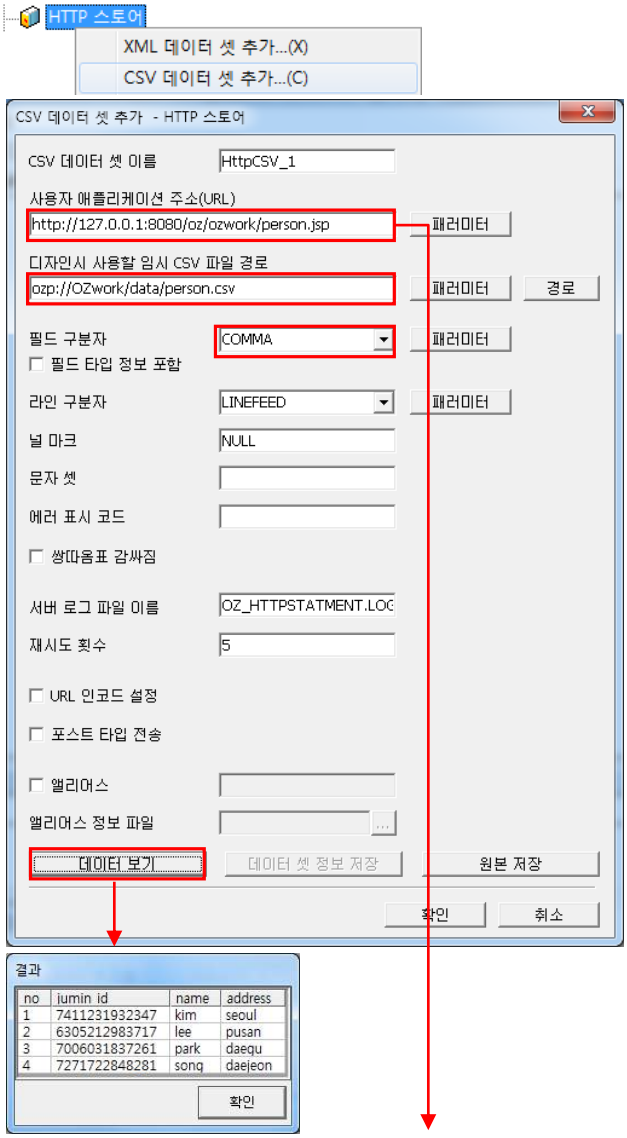
- **사용자 애플리케이션 주소** :
  - XML 파일을 다운로드 받을 수 있는 URL
  - XML 데이터를 반환하는 애플리케이션 URL
- **디자인시 사용할 임시 XML 파일 경로** :
  - 로컬 PC의 XML 파일 경로
  - 디자이너 작업 폴더의 XML 파일 경로(ozp://)



나. HTTP 스토어 - CSV 데이터 셋

```
no,jumin_id,name,address
INT,VARCHAR,VARCHAR,VARCHAR
1,7411231932347,kim,seoul
2,6305212983717,lee,pusan
3,7006031837261,park,daegu
4,7271722848281,song,daejeon
```

• CSV 데이터 구조는 첫번째 줄에는 필드 이름, 두번째 줄에는 필드 타입, 이후로는 레코드 데이터로 구성됩니다. 필드 타입을 생략하면 모든 필드가 Varchar 타입으로 인식됩니다.



• HTTP 스토어의 팝업 메뉴에서 'CSV 데이터 셋 추가'를 선택하면 나오는 'CSV 데이터 셋 추가' 창에서 경로를 입력하여 데이터 셋을 추가합니다.

• 사용자 애플리케이션 주소 :

운영시 사용되는 URL 경로를 입력합니다. 경로를 패러미터로 처리하여 동적으로 변경할 수 있습니다.

- CSV 파일을 다운로드 받을 수 있는 URL
- CSV 데이터를 반환하는 애플리케이션 URL

• 디자인시 사용할 임시 CSV 파일 경로 :

쿼리 디자이너와 리포트 디자이너에서만 사용하는 임시 경로입니다.

- 로컬 PC의 CSV 파일 경로
- 디자이너 작업 폴더의 CSV 파일 경로(ozp://)

• 필드 구분자 :

탭, 공백, ';', ':', ',' 중에서 선택하거나 직접 입력해서 변경 가능합니다.

```
<%
out.println("no,jumin_id,name,address");
out.println("INT,VARCHAR,VARCHAR,VARCHAR");
out.println("1,7411231932347,kim,seoul");
out.println("2,6305212983717,lee,pusan");
out.println("3,7006031837261,park,daegu");
out.println("4,7271722848281,song,daejeon");
%>
```

• CSV 데이터를 반환하는 애플리케이션을 웹 브라우저에서 실행한 후 소스 보기한 결과가 위의 CSV 데이터 구조와 같아야 인식됩니다.

1.2 XML 스토어

오즈 서버에 로컬 파일로 존재하거나 URL로 접근 가능한 XML 데이터를 보고서에 출력하고자 할 때 XML 스토어를 이용합니다. 파일 스토어나 HTTP 스토어와는 달리 XML 구조가 고정되어 있지 않고, 하나의 XML로부터 여러 데이터 셋을 만들 수 있습니다. XML의 구조를 정의한 DTD 또는 XML Schema(XSD) 파일이 디자인 시에 있어야 합니다.

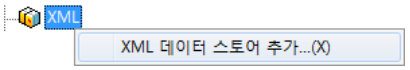
[messages.xml]

```
<?xml version="1.0"?>
<messages>
    <note id="501">
        <date>
            <day>10</day>
            <month>03</month>
            <year>2011</year>
        </date>
        <to>Tove</to>
        <from>Jani</from>
        <heading>Reminder</heading>
        <body>Don't forget me this weekend!</body>
    </note>
    <note id="502">
        <date>
            <day>11</day>
            <month>03</month>
            <year>2011</year>
        </date>
        <to>Jani</to>
        <from>Tove</from>
        <heading>Re: Reminder</heading>
        <body>I will not</body>
    </note>
</messages>
```

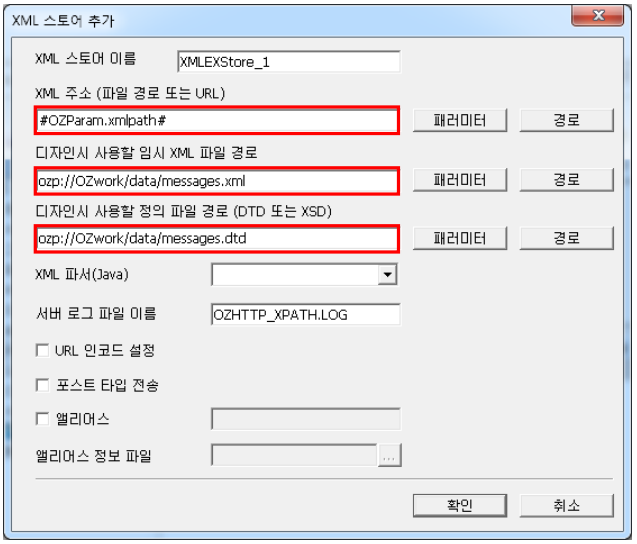
[messages.dtd]

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT messages (note+)>
<!ELEMENT note (date, to, from, heading, body)>
<!ELEMENT date (day, month, year)>
<!ELEMENT day (#PCDATA)>
<!ELEMENT month (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
<!ATTLIST note
    id CDATA #REQUIRED
>
```

• XML 데이터 구조는 DTD 또는 XSD에 정의된 구조입니다.



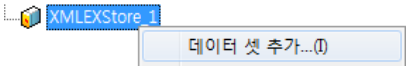
• 'XML'의 팝업 메뉴에서 'XML 데이터 스토어 추가'를 선택하면 나오는 'XML 스토어 추가' 창에서 경로를 입력하여 스토어를 추가합니다.



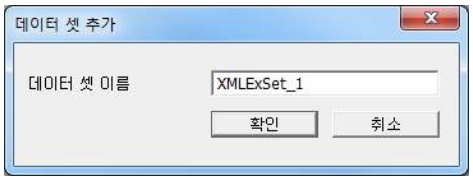
• **XML 주소** :
- XML 파일을 다운로드 받을 수 있는 URL
- XML 데이터를 반환하는 애플리케이션 URL
- 오즈 서버의 로컬 XML 파일 경로
- 오즈 서버 리파지토리의 XML 파일 경로(ozp://)

• **디자인시 사용할 임시 XML 파일 경로** :
- 로컬 PC의 XML 파일 경로
- 디자이너 작업 폴더의 XML 파일 경로(ozp://)

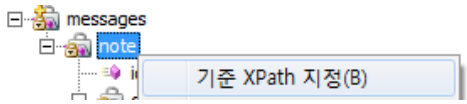
• **디자인시 사용할 정의 파일 경로 (DTD 또는 XSD)** :
- 로컬 PC의 DTD 또는 XSD 파일 경로
- 디자이너 작업 폴더의 DTD 또는 XSD 파일 경로



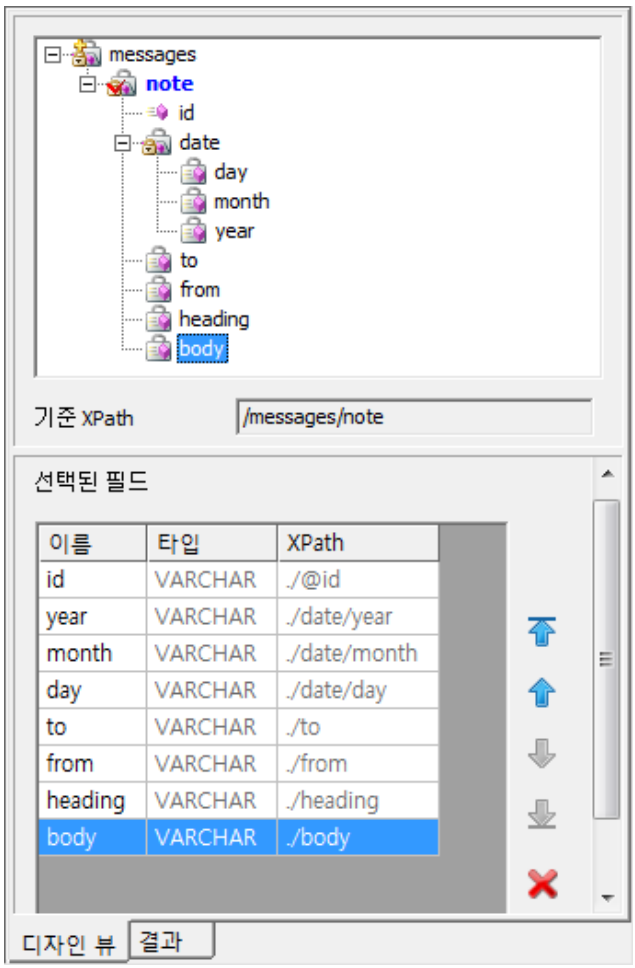
• 추가된 XML 스토어의 팝업 메뉴에서 '데이터 셋 추가'를 선택합니다.



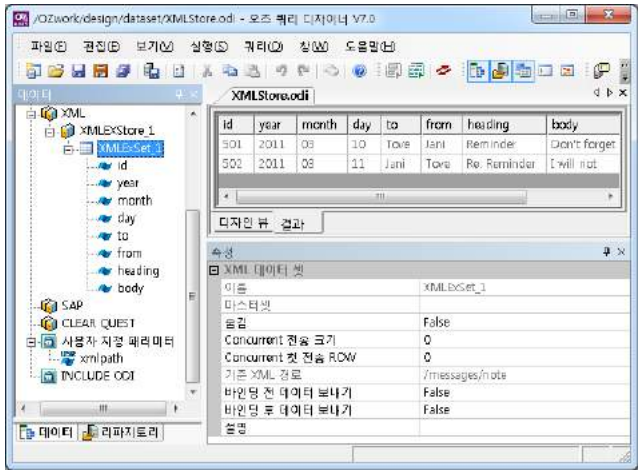
• 데이터 셋 이름을 입력한 후 '확인' 버튼을 클릭합니다.



• 기준 Xpath 를 지정합니다.



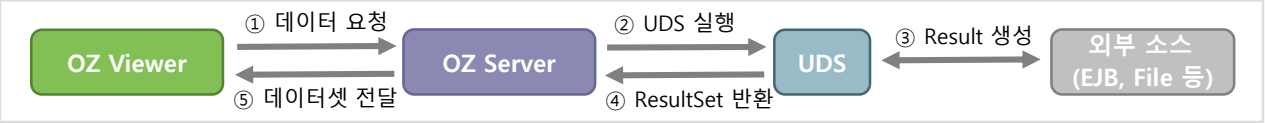
• 보고서에 출력할 노드를 상단 트리에서 더블 클릭하면 하단 '선택한 필드'에 추가됩니다.



• 쿼리문을 실행하면(F5) 데이터 필드가 생성되고, 결과 탭에서 데이터를 확인할 수 있습니다.

사용자 데이터 스토어(UDS)는 오즈에서 제공하는 Java 또는 .Net API를 이용하여 외부 데이터 소스로 부터 데이터 셋을 생성하는 스토어 입니다.

- **UDS 실행 순서**



- **UDS 클래스가 구현해야 하는 Interface**

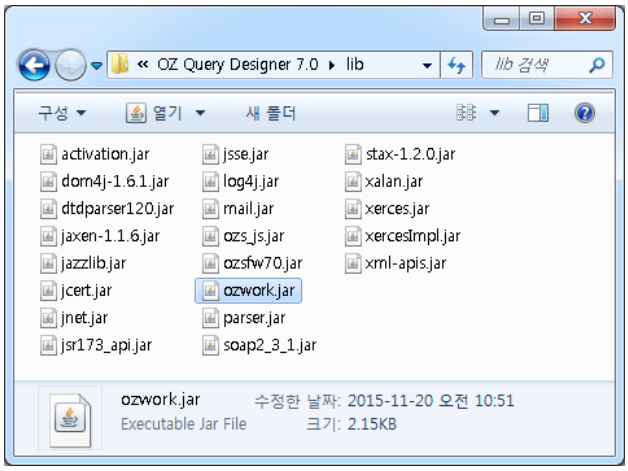
```
package oz.uds;
import java.sql.*;
public interface OZUserDataStore {
    public void init() throws OZUserDataStoreException; // init resource
    public ResultSet getResultSet(String argument) throws OZUserDataStoreException; // return ResultSet
    public void freeResultSet(ResultSet rst); // release resource for specific ResultSet
    public void close(); // release all resource
}
```

➢ **UDS 클래스 구현 예**

```
package ozworkuds;

import oz.uds.*;
import oz.uds.rs.*;
import java.sql.*;

public class ArrayUDS implements OZUserDataStore {

    public void init() throws OZUserDataStoreException { }

    public ResultSet getResultSet(String argument) throws OZUserDataStoreException {
        try {
            String[] fieldName = {"FIELD1","FILED2"};
            String[] fieldType = {"VARCHAR", "INTEGER"};
            String[][] data = { {"data11", "12"}, {"data21", "22"}, {"data31", "32"} };
            data[2][0] = data[2][0] + ", 실행문=" + argument;
            return new oz.uds.rs.ArrResultSet(fieldName, fieldType, data); // ResultSet 반환
        } catch(Exception ex){
            ex.printStackTrace();
            throw new OZUserDataStoreException(ex.getMessage());
        }
    }

    public void freeResultSet(ResultSet rst) { }

    public void close() { }
}
```

getResultSet 함수에서 반환하는 ResultSet 객체는 java.sql.ResultSet 인터페이스를 구현한 객체여야 합니다.
오즈 서버 모듈(ozsfw70.jar)의 oz.uds.rs 패키지에 있는 다양한 형태의 ResultSet을 이용할 수 있습니다.

- **UDS 설정**

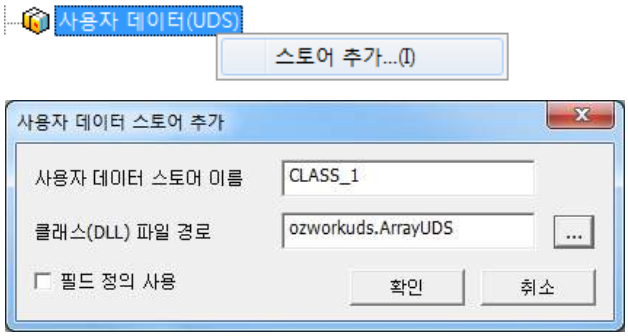


① '%OZD_HOME%\lib'폴더에 UDS 클래스(.jar)를 복사합니다.

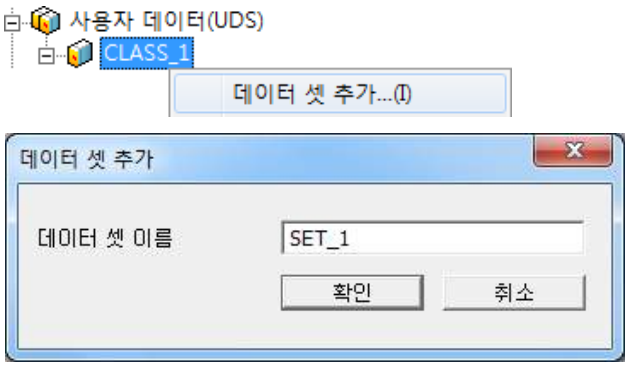
CLASSPATH = .₩lib₩ozsfw70.jar;...;**.₩lib₩ozwork.jar**

② '%OZD_HOME%\config\launch.cfg' 파일을 편집기로 열어 CLASSPATH 에 UDS 파일 경로를 추가합니다.

※ OZ Report Designer 도 동일하게 설정합니다.
※ 운영 시점에는 오즈 서버의 lib 폴더에 UDS 파일을 복사하고 클래스 패스로 인식되어야 합니다.

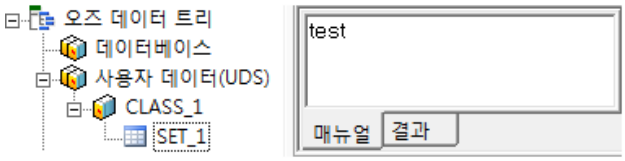


③ 쿼리 디자이너의 '사용자 데이터'의 팝업 메뉴에서 '스토어 추가'를 선택한 후 '클래스 파일 경로'를 입력합니다. '필드 정의 사용'을 체크하면 클래스 파일이 없어도 추가 가능합니다.

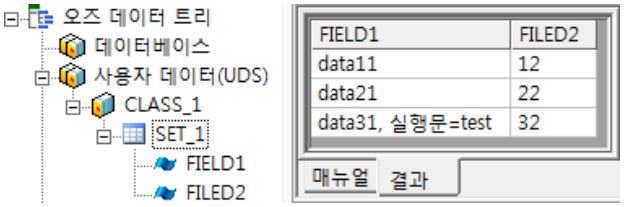


④ 추가된 스토어의 팝업 메뉴에서 '데이터 셋 추가'를 선택한 후 셋 이름을 입력합니다.

- 사용자 데이터 스토어에 추가된 데이터 셋 개수만큼 getResultSet 함수가 실행됩니다.



⑤ 쿼리 창에 실행문을 입력하고 '쿼리문 실행'(F5) 하면 데이터 필드가 생성됩니다.



- 실행문 : getResultSet 함수의 패러미터(argument)로 넘어가는 문자열

※ UDS에 대한 자세한 사항은 'oztn.net' → '자료실' → 'OZ 매뉴얼'의 'OZ UserDataStore Manual.pdf' 참조

앞에서 설명한 쿼리 디자이너에서 추가한 데이터 셋은 오즈 서버가 데이터를 조회하는 반면, 리포트 디자이너에서 추가한 데이터 셋은 오즈 리포트 뷰어가 직접 데이터를 조회합니다.

뷰어가 직접 데이터를 조회하는 방식에는 '데이터 서비스'와 'FX 데이터' 방식이 있습니다.

'데이터 서비스'는 CSV와 JSON 데이터를 사용할 수 있습니다.

지원되는 경로는 다음과 같습니다.

a. **뷰어에서 폼 패러미터로 넘긴 데이터**

b. **로컬 파일 경로(뷰어가 실행된 PC)** : 예) file://C:/temp/testdata.txt, C:/temp/testdata.csv, C:/temp/testdata.json

c. **오즈 서버 리파지토리 경로** : 예) ozp://OZwork/data/products.txt, ozp://OZwork/data/products.json

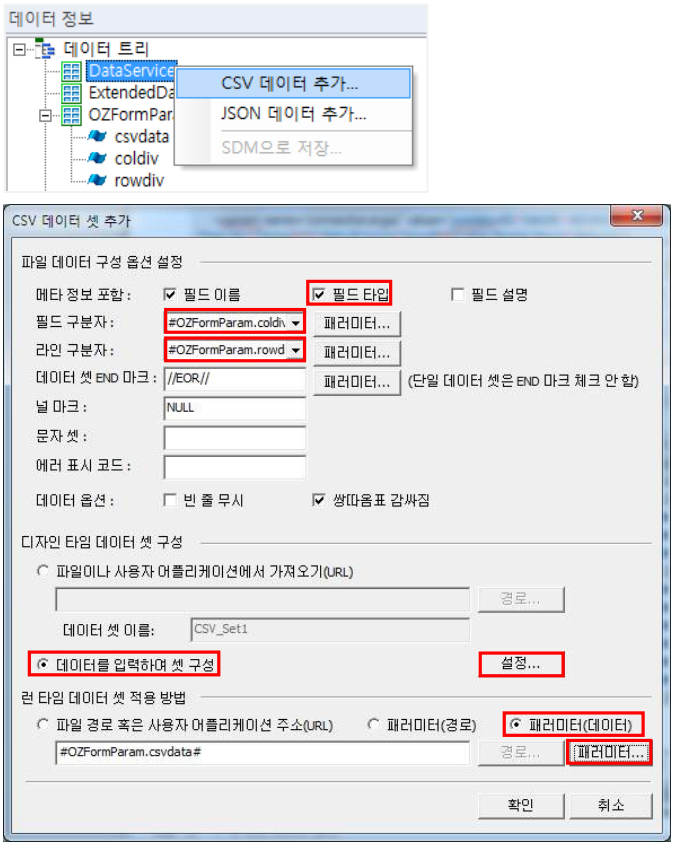
d. **URL** : 예) http://127.0.0.1/testdata.txt, http://127.0.0.1/getdata.jsp?id=123(csv, json 데이터를 반환하는 프로그램)

가. CSV 데이터

➢ 패러미터로 데이터를 넘기는 예

웹 페이지에서 뷰어를 실행할 때 아래와 같이 csvdata란 폼 패러미터로 데이터를 넘기는 방법입니다.

```
<param name="connection.servlet" value="http://127.0.0.1:8080/oz/server">
<param name="connection.reportname" value="/OZwork/design/dataset/DataService.ozr">
<param name="connection.pcount" value="3">
<param name="connection.args1" value="csvdata=ID^NAME^ADDRESS///INTEGER^VARCHAR^VARCHAR///1^Park Chan Ho^Texas///2^Shin Ki Sung^Seoul///3^Kim Byung Hyun^Arizona">
<param name="connection.args2" value="coldiv=^">
<param name="connection.args3" value="rowdiv=///">
```



① 리포트 디자이너의 '데이터 정보' 창에서 폼 패러미터 'csvdata', 'coldiv', 'rowdiv' 추가

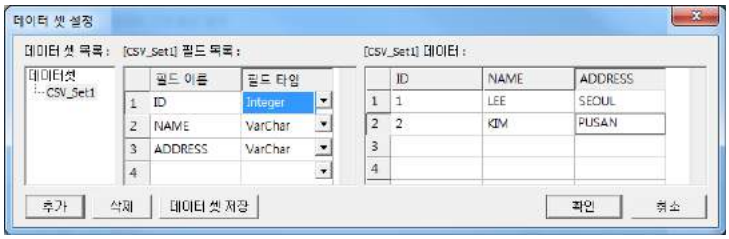
② 'DataService' 팝업 메뉴의 'CSV 데이터 추가' 선택

③ 데이터에 '필드 이름'과 '필드 타입'이 있으면 메타 정보 포함 항목에서 체크

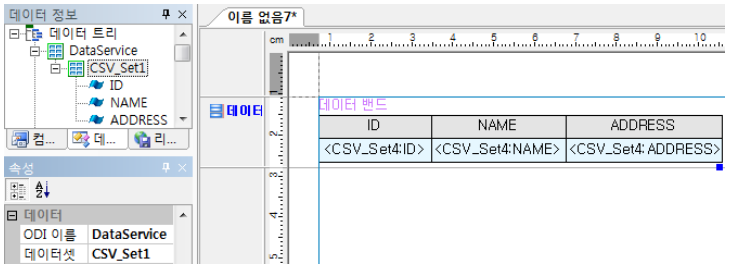
④ 필드 구분자, 라인 구분자에 'coldiv', 'rowdiv' 패러미터 설정

⑤ '런 타임 데이터 셋 적용 방법'의 '패러미터(데이터)' 체크하고 경로 옆의 '패러미터' 버튼 클릭 후 '#OZFormParam.csvdata#' 설정

⑥ '디자인 타임 데이터 셋 구성'의 '데이터를 입력하여 셋 구성'을 체크하고 옆의 '설정' 버튼 클릭



⑦ '추가' 버튼 클릭해서 데이터 셋 추가 후 '필드 이름', '필드 타입', '데이터' 입력

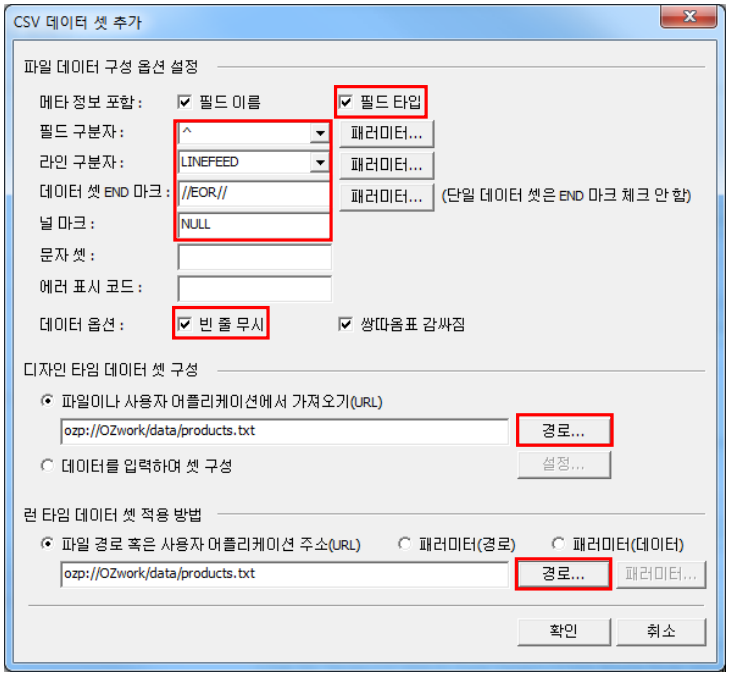


⑧ 데이터 셋이 만들어지면 데이터 밴드의 'ODI 이름' 속성에 'DataService' 를 선택하고 '데이터셋' 속성에 추가한 데이터 셋 설정

➢ 하나의 텍스트 데이터 파일로 두 개의 데이터 셋을 생성하는 예

아래와 같이 두 개의 데이터 셋 정보를 가진 데이터 파일(products.txt)로 데이터 셋을 추가하는 방법입니다.

```
CategoryID^CategoryName^Description
INTEGER^VARCHAR^VARCHAR
1^음료^청량음료, 커피, 홍차, 맥주
2^조미료^감미료, 향신료, 양념, 스프레드
//EOR//
ProductID^ProductName^CategoryID^QuantityPerUnit^UnitPrice^UnitsInStock
INTEGER^VARCHAR^INTEGER^VARCHAR^NUMERIC^SMALLINT
75^알파인 맥주^1^24 - 0.5 l bottles^24000.0000^125
39^OK 바닐라 셰이크^1^750 cc per bottle^28000.0000^69
65^루이지애나 특산 후추^2^32 - 8 oz bottles^21000.0000^76
61^사계절 핫 소스^2^24 - 500 ml bottles^28000.0000^NULL
(빈 줄)
```



① 데이터에 '필드 이름'과 '필드 타입'이 있으면 메타 정보 포함 항목에서 체크

② '필드 구분자', '라인 구분자', '데이터 셋 END 마크'가 실제 데이터 파일의 구분자와 같아야 하며 선택 목록에 없는 경우 직접 입력

※ 파일 스토어나 HTTP 스토어의 CSV 데이터 셋은 하나의 파일 경로에 하나의 데이터 셋만 생성할 수 있으나 데이터 서비스에서는 하나의 경로로 여러 데이터 셋을 생성할 수 있기 때문에 '데이터 셋 END 마크' 구분자가 필요합니다.

③ 널 데이터로 인식할 문자를 '널 마크'에 입력

④ 데이터 파일에 있는 빈 줄을 데이터로 인식하지 않으려면 '빈 줄 무시' 체크

⑤ '경로' 버튼 클릭 후 데이터 파일 지정

나. JSON 형식

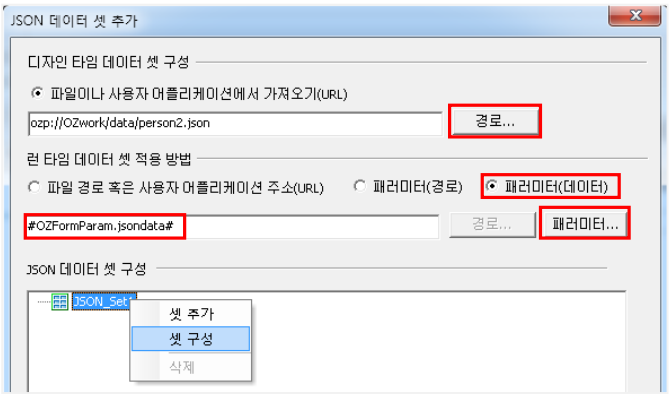
➢ 패러미터로 데이터를 넘기는 예

웹 페이지에서 뷰어를 실행할 때 아래와 같이 jsondata란 폼 패러미터로 데이터를 넘기는 방법입니다.

```
<param name="connection.servlet" value="http://127.0.0.1:8080/oz/server">
<param name="connection.reportname" value="/OZwork/design/dataset/DataServiceJSON.ozr">
<param name="connection.pcount" value="1">
<param name="connection.args1" value ="jsondata={'Person': [{'ID': 1, 'NAME': 'Park Chan Ho', 'ADDRESS': 'Texas'}, {'ID': 2, 'NAME': 'Shin Ki Sung', 'ADDRESS': 'Seoul'}, {'ID': 3, 'NAME': 'Kim Byung Hyun', 'ADDRESS': 'Arizona'}]}">
```

① 리포트 디자이너의 '데이터 정보' 창에서 폼 패러미터 'jsondata' 추가

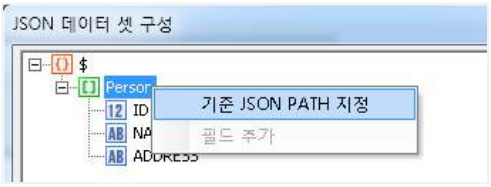
② 'DataService' 팝업 메뉴의 'JSON 데이터 추가' 선택



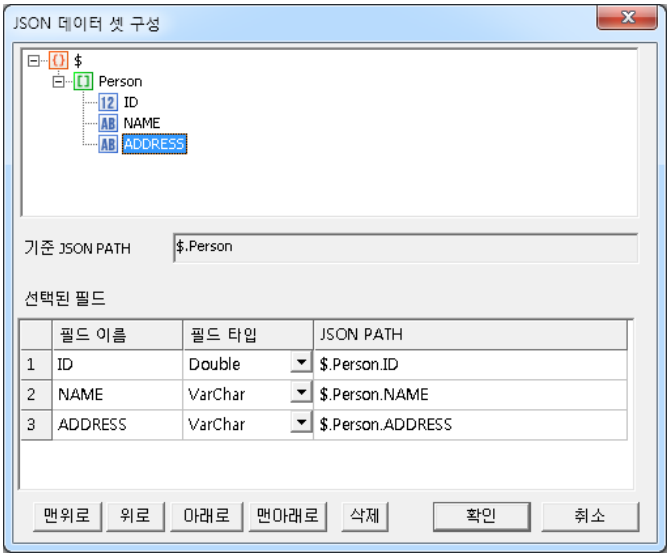
③ '디자인 타임 데이터 셋 구성'의 '경로' 버튼 클릭하고 디자인시 사용할 경로 설정

④ '런 타임 데이터 셋 적용 방법'의 '패러미터(데이터)' 체크하고 경로 옆의 '패러미터' 버튼 클릭 후 '#OZFormParam.jsondata#' 설정

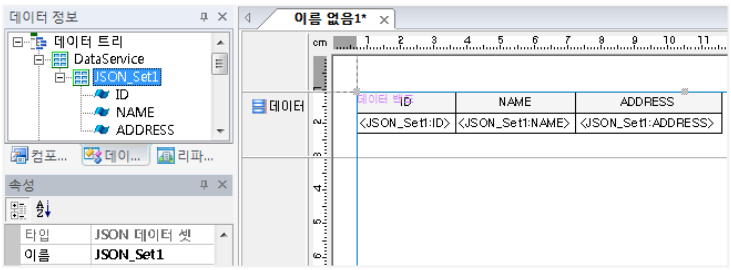
⑤ 'JSON 데이터 셋 구성' 아래에 있는 'JSON_Set1'의 팝업 메뉴 중 '셋 구성' 선택



⑥ 기준 JSON PATH로 설정할 노드를 마우스 오른쪽 버튼으로 클릭하여 팝업 메뉴 중 '기준 JSON PATH 지정' 선택



⑦ 기준 JSON PATH 아래에 있는 필드 중 추가할 필드를 더블 클릭하여 선택된 필드로 추가

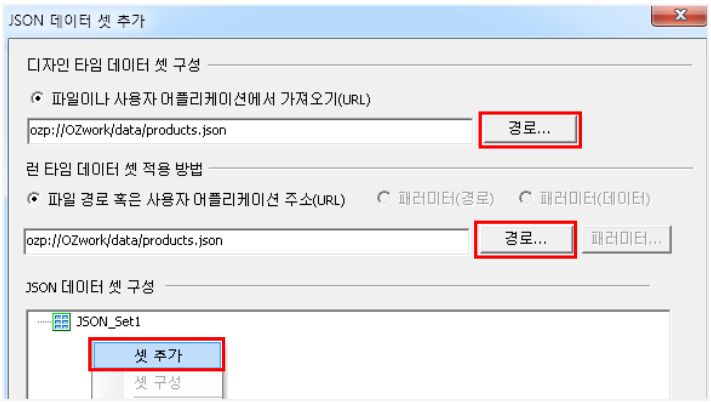


⑦ 데이터 셋이 만들어지면 데이터 밴드의 'ODI 이름' 속성에 'DataService' 를 선택하고 '데이터셋' 속성에 추가한 데이터 셋 설정

➢ 하나의 데이터 파일로 두 개의 데이터 셋을 생성하는 예

아래와 같이 두 개의 데이터 셋 정보를 가진 데이터 파일(products.json)로 데이터 셋을 추가하는 방법입니다.

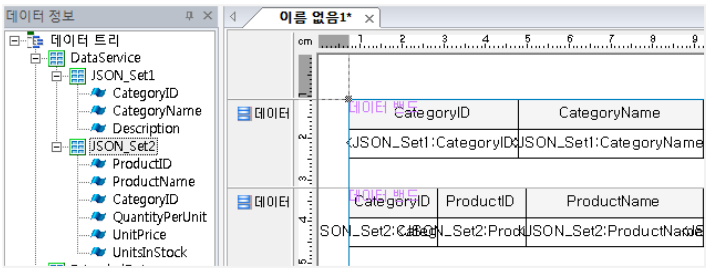
```
{ "Category": [
      {"CategoryID": 1, "CategoryName": "음료", "Description": "청량음료, 커피, 홍차, 맥주"},
      {"CategoryID": 2, "CategoryName": "조미료", "Description": "감미료, 향신료, 양념, 스프레드"},
    ],
   "Product": [
      {"ProductID": 75, "ProductName": "알파인 맥주", "CategoryID": 1, "QuantityPerUnit": "24 - 0.5 l bottles", "UnitPrice":
24000.0000, "UnitsInStock":125},
      {"ProductID": 39, "ProductName": "OK 바닐라 셰이크", "CategoryID": 1, "QuantityPerUnit": "750 cc per bottle", "UnitPrice":
28000.0000, "UnitsInStock":69},
      ...(생략)
   ]
}
```



① 디자인 시 사용할 경로와 런 타임 시 사용할 경로를 설정

② 'JSON 데이터 셋 구성' 의 빈 영역의 팝업 메뉴에서 '셋 추가' 클릭하여 두 번째 데이터 셋 생성

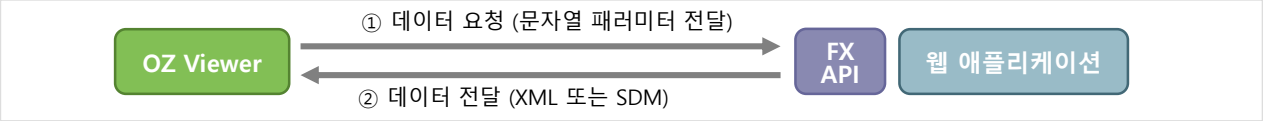
③ JSON_Set1, JSON_Set2 각각 '셋 구성' 팝업 메뉴에서 '기준 JSON PATH 지정' 및 필드 추가



④ JSON 데이터 셋을 데이터 밴드에 설정하여 컴포넌트와 연동

'데이터 서비스'와 같이 'FX 데이터'도 오즈 뷰어가 직접 데이터를 조회하는 방식입니다. 오즈 서버는 odi, ozr 등의 폼 파일을 서비스하고, 데이터는 오즈 뷰어의 요청으로 일반적인 웹 애플리케이션 프로그램에서 가져옵니다. 이 때, 웹 애플리케이션은 Java/.NET 으로 제공되는 FX API로 생성한 데이터 모듈을 뷰어에 전달해야 합니다.

- **FX 데이터 실행 순서**



오즈 뷰어가 FX API를 구현한 웹 애플리케이션 URL을 호출합니다. 이 때 데이터 조회에 필요한 조건 문자열을 POST 방식으로 전달합니다. 웹 애플리케이션은 데이터를 조회한 후에 FX API를 이용하여 데이터 모듈을 생성합니다. 데이터 모듈은 XML이나 SDM(오즈 뷰어가 인식하는 바이너리 데이터) 형태로 뷰어에 전달됩니다.

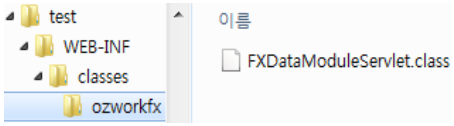
- **예약된 패러미터**

패러미터 명	설정 가능한 값	설명
type	xml, sdm	뷰어에 전달할 데이터 모듈 종류
compressed	true, false	뷰어에 전달할 데이터 압축 여부
concurrent	true, false	데이터 모듈을 뷰어에 전달하는 방식. true면 실시간 전송

➢ **FX API 구현 예 (FXDataModuleServlet.java)**

```
final String isSDM = "sdm".equalsIgnoreCase(request.getParameter("type"));
final boolean isCompressed = "true".equalsIgnoreCase(request.getParameter("compressed"));
final boolean concurrent = "true".equalsIgnoreCase(httpRequest.getParameter("concurrent"));

if (isSDM) {
    if (concurrent) {
        // 데이터 모듈 생성 작업과 전송 작업을 동시에 수행하는 형태의 SDM 데이터 모듈 생성
        dataModule = FXDataModuleFactory.createConcurrentDataModule();
    } else {
        // 전체 데이터에 대한 데이터 모듈 생성 작업을 완료한 후 전송되는 SDM데이터 모듈 생성
        dataModule = FXDataModuleFactory.createBatchDataModule();
    }
} else {
    // 전체 데이터에 대한 데이터 모듈 생성 작업을 완료한 후 전송되는 형태의 XML 데이터 모듈 생성
    dataModule = FXDataModuleFactory.createBatchXmlDataModule();
}
// 데이터 모듈을 기록할 정보를 초기화하며 데이터 압축 여부와 버퍼 크기를 설정할 수 있습니다.
dataModule.initialize(isCompressed, httpResponse.getOutputStream());
dataModule.addParameter("param1", Types.VARCHAR, httpRequest.getParameter("param1")); // 패러미터 추가

dataModule.addDataSet("SET_Maker"); // 'SET_Maker' 데이터 셋 추가
dataModule.addField("SET_Maker", "Maker", Types.VARCHAR); // 데이터 셋의 필드 추가

dataModule.startBinding(); // 메타 정보가 모두 등록되었을 때 데이터 모듈 바인딩을 시작하며 메타 정보의 무결성을 검증합니다.
dataModule.startSet("SET_Maker"); // 데이터 셋 바인딩을 시작합니다.
dataModule.setValue(1, "기아자동차"); // 현재 바인딩 중인 데이터 셋에 값을 설정합니다.
dataModule.completeRow(); // 하나의 Row에 대한 데이터 값 설정을 완료합니다.
dataModule.endSet("SET_Maker"); // 데이터 셋 바인딩을 종료합니다.
dataModule.endBinding(); // 데이터 모듈 바인딩을 종료합니다.
dataModule.finish(); // 모든 작업을 종료하며, 데이터 전송 타입이 batch 타입일 경우 데이터 전송 작업이 시작됩니다.
```

- **웹 애플리케이션 서버에서 FX 라이브러리 설정**



① 웹 애플리케이션에서 FX API를 사용하기 위해서는 웹 애플리케이션 서버의 클래스 패스에 ozsdmapi.jar, crimson.jar 파일을 등록해야 합니다.
[웹 모듈]\WEB-INF\lib\에 jar 복사

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee"...>
  <servlet>
    <servlet-name>ozworkfx.FXDataModuleServlet</servlet-name>
    <servlet-class>ozworkfx.FXDataModuleServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>ozworkfx.FXDataModuleServlet</servlet-name>
    <url-pattern>/OZworkFXServlet</url-pattern>
  </servlet-mapping>
</web-app>
```

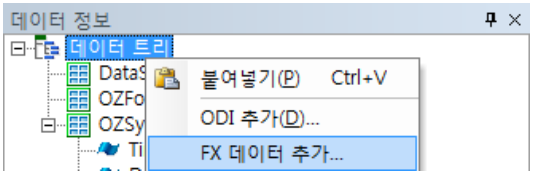
② 웹 애플리케이션이 서블릿인 경우 WEB-INF\web.xml 파일에 서블릿을 등록합니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<OZDATA>
    <VER>2</VER>
    <OZPARAMS>
        <PARAM NAME="param1" TYPE="VARCHAR">1</PARAM> ...
    </OZPARAMS>
    <OZDATASETINFOS>
        <DATASETINFO NAME="SET_Maker" MASTERSET="">
            <FIELDINFO>
                <FIELD NAME="Maker" TYPE="VARCHAR"/>
            </FIELDINFO>
        </DATASETINFO>
        <DATASETINFO NAME="SET_Car" MASTERSET="">
            <FIELDINFO>
                <FIELD NAME="Maker" TYPE="VARCHAR"/>
                <FIELD NAME="CarID" TYPE="VARCHAR"/>
                <FIELD NAME="CarName" TYPE="VARCHAR"/>
            </FIELDINFO>
        </DATASETINFO>
    </OZDATASETINFOS>
    <OZDATARESULT>
        <OZDATASET NAME="SET_Maker">
            <RECORD>
                <COL>기아자동차</COL>
            </RECORD>
        </OZDATASET>
        <OZDATASET NAME="SET_Car">
            <RECORD>
                <COL>기아자동차</COL>
                <COL>K01</COL>
                <COL>세피아</COL>
            </RECORD>
        </OZDATASET>
    </OZDATARESULT>
</OZDATA>
```

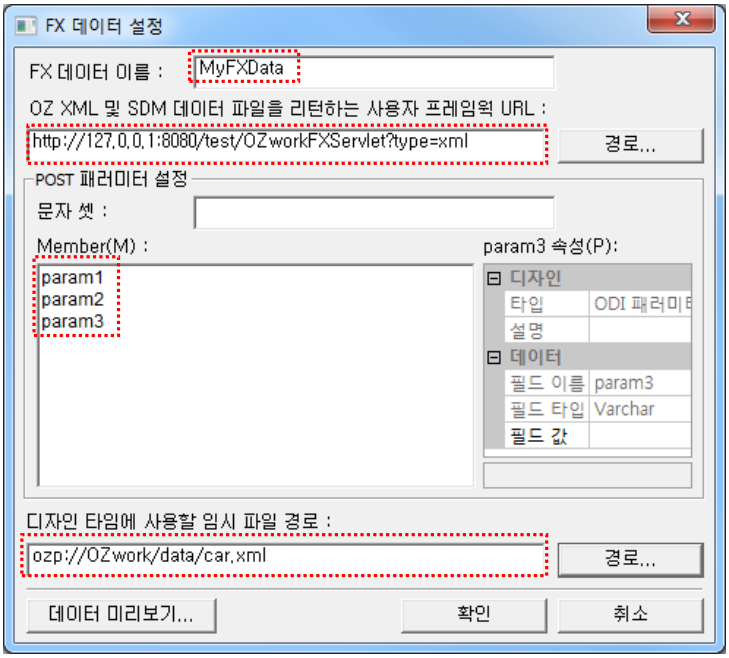
③ 서블릿 URL을 웹 브라우저에서 확인하면 생성된 데이터를 확인할 수 있습니다.
예)http://127.0.0.1:8080/test/OZworkFXServlet?type=xml

④ 웹 페이지에서 소스보기한 xml 데이터를 저장(car.xml)하여 디자인할 때 사용합니다.

- **리포트 디자이너에서 FX 데이터 사용**



① 리포트 디자이너의 '데이터 정보' 창에서 '데이터 트리' 팝업 메뉴의 'FX 데이터 추가' 선택



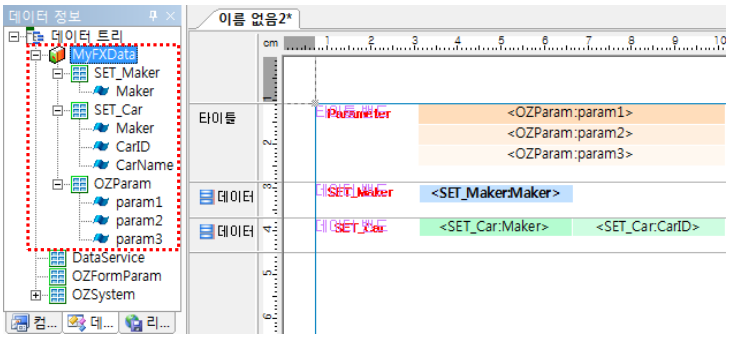
② 'FX 데이터 이름' 입력 (디자인할 때 'ODI 이름'에 사용됨)

③ 'OZ XML 및 SDM 데이터 파일을 리턴하는 사용자 프레임웍 URL'에 서블릿 URL 입력

④ 서블릿으로 넘길 패러미터를 Member에 입력

※ 프레임웍 URL에 입력한 패러미터는 GET방식으로 전달되며, Member에 입력한 패러미터는 POST방식으로 전달됩니다.

⑤ '디자인 타임에 사용할 임시 파일 경로'에 웹 페이지에서 저장한 xml 파일 경로 지정



⑥ '확인' 버튼 클릭하면 데이터 셋과 패러미터 구조가 '데이터 정보' 창에 만들어지며 디자인 방법은 동일

- **뷰어 실행 HTML 태그**

```
<object id="ozviewer" width="100%" height="100%" CLASSID="CLSID:0DEF32F8-170F-46f8-B1FF-4BF7443F5F25">
  <param name="viewer.namespace" value="OZEduSystem₩ozviewer">
  <param name="viewer.isframe" value="false">
  <param name="connection.servlet" value="http://127.0.0.1:8080/oz/server">
  <param name="connection.reportname" value="/OZwork/design/dataset/FXData.ozr">
  <param name="odi.odinames" value="MyFXData">
  <param name="odi.MyFXData.pcount" value="3">
  <param name="odi.MyFXData.args1" value="param1=100">
  <param name="odi.MyFXData.args2" value="param2=200">
  <param name="odi.MyFXData.args3" value="param3=300">
  <param name="odi.MyFXData.frameworkurl" value="http://127.0.0.1:8080/test/OZworkFXServlet?type=xml">
</object>
```

※ FX 데이터에 대한 자세한 사항은 'oztn.net' → '자료실' → 'OZ 매뉴얼'의 'OZ Framework Manual.pdf' 참조

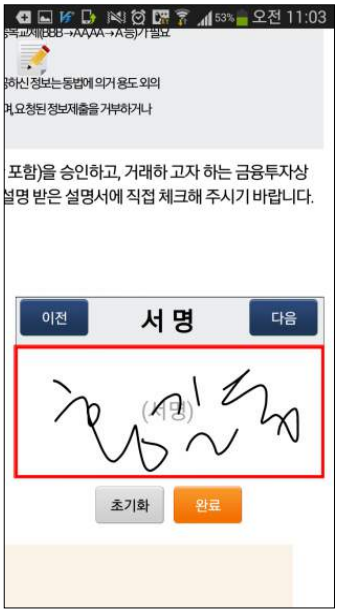
Chapter8. 오즈 e-Form

1. OZ e-Form 소개
2. OZ e-Form 기능
3. OZ e-Form Exercise

1. OZ e-Form 소개

■ **OZ e-Form 데모 (※ *LTE 사용*)**

- OZ Mobile 앱 설치 (핸드폰 > 애플 앱스토어 / 구글 스토어 > '**oz mobile**' 검색)
- OZ Mobile 앱 실행 (OZ 체험하기 > '계좌개설신청서' 클릭)



※ *언어 설정 방법 : 환경설정 > Current language : 한국어*

■ **http://192.168.***.*** > '이폼 기능' 메뉴 클릭 (※ *교육장 내부 무선 네트워크 사용*)

1) CheckBox : 여러 항목 중 하나 이상 선택할 수 있는 컴포넌트입니다.

2) RadioButton : 여러 항목 중 하나만 선택할 수 있으며 항목별로 그룹핑하여 사용할 수 있습니다.

3) TextBox : 숫자나 문자열을 입력할 때 사용하는 컴포넌트입니다.

4) SignPad : 문구 따라 쓰기, 서명을 입력할 수 있는 컴포넌트입니다.

5) Camera : 사진 촬영, 도장 촬영, 신분증 촬영을 제공하는 컴포넌트입니다.

6) ComboBox : 다수의 아이템을 리스트 형태로 보여주는 컴포넌트입니다.

7) NumericUpDown : 숫자 값을 직접 입력하거나 위/아래 버튼을 클릭하여 숫자 값을 증감시키고자 할 때 사용하는 컴포넌트입니다.

8) DateTimePicker : 날짜 또는 시간을 입력하거나 달력을 이용하여 날짜를 입력하는 컴포넌트입니다.

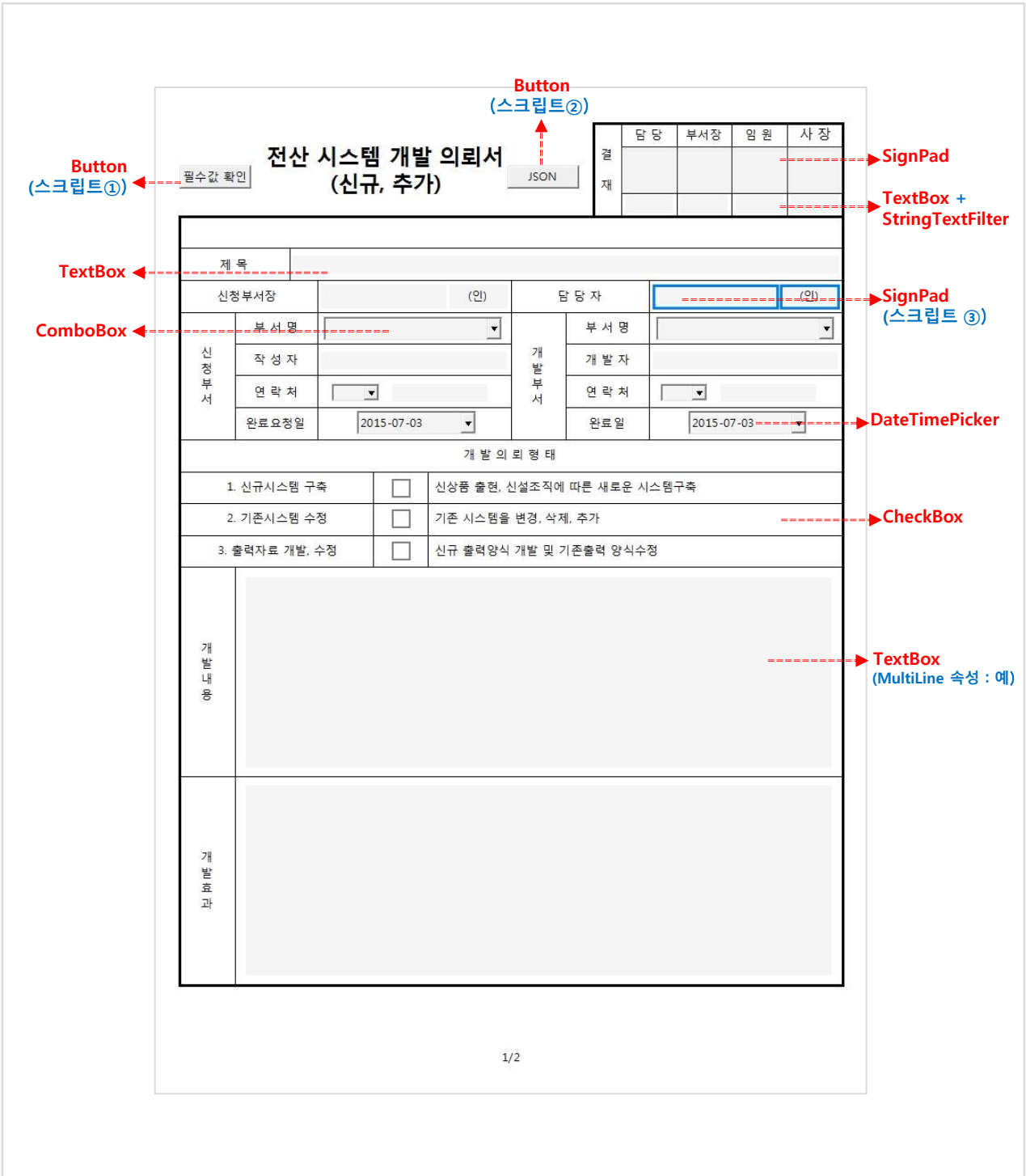
9) VoiceRecorder : 음성을 녹음하는 컴포넌트입니다.

가. 엑셀파일 다운로드 (※ *교육장 내부 무선 네트워크 사용*)

- http://192.168.***.*** > '이폼 실습' > '전산시스템개발의뢰서신규추가' 엑셀 다운로드

나. 실습 서식

- 바탕화면 > OZ e-Form 폴더 > 'OZ Report Designer 7.0' 실행



- 리포트 템플릿 선택 후 마우스 오른쪽 클릭 > '리포트 추가' 클릭하여 Report2 를 추가합니다.

- '내 문서₩OZRepository₩{오늘날짜(예. 20150707)}₩{교육생이름}.ozr' 로 완성한 서식을 저장합니다.

다. 구현 방법

■ **엑셀 복사**

엑셀 시트에서 전자 서식 영역을 복사한 후 이폼 디자이너의 더미밴드에 붙여넣기를 합니다.

시간은 30초 내외가 소요되며 엑셀 프로그램은 '편집모드'로 실행하셔야 합니다.

■ **결재란 날짜(MM/dd) 설정**

타입	속성	값
StringTextFilter	Pattern	##/##
TextBox	텍스트 필더 이름	생성한 'StringTextFilter'를 선택

■ **FormID 속성값 변경 (필수값 확인 시 필요)**

항목	FormID
제목	Title
담당자 – 이름 SignPad	OwnerName
담당자 – 서명 SignPad	OwnerSign

■ **스크립트**

(SCRIPT_BEGIN, SCRIPT_END 라인은 자바 스크립트 편집 창에 작성하지 마세요.)

① 필수값 확인 필수값 확인

```
SCRIPT_BEGIN .Report1.Button1.OnCheckValidity
if(GetInputValue("Title") == ""){
        GetInputComponent("Title").EnsureVisible(true);
        GetInputComponent("Title").SetFocus(true);
         _MessageBox("제목을 입력하세요.");
        return false;
}
if(GetInputValue("OwnerName") == "" || GetInputValue("OwnerSign") == ""){
        GetInputComponent("OwnerName").EnsureVisible(true);
        _MessageBox("담당자 이름/서명을 입력하세요.");
        return false;
}
SCRIPT_END
```

[참고]

- EnsureVisible : 컴포넌트가 화면에 보이도록 보고서를 스크롤 함
- SetFocus : 컴포넌트에 포커스를 설정함

② JSON 데이터 추출 JSON

```
SCRIPT_BEGIN .Report1.Button2.OnClick
_MessageBox(GetInputJSON(false));
SCRIPT_END
```

③ 담당자 이름/서명란

담 당 자		(인)

(담당자 이름/서명란에 아래 스크립트를 동일하게 작성해주세요.)

- 그룹서명

```
SCRIPT_BEGIN .Report1.SignPad7.OnBind
This.SetGroupName("SignGroup2");
SCRIPT_END
```

- 서명 작성 시 서명 테두리 삭제

```
SCRIPT_BEGIN .Report1.SignPad7.OnValueChanged
var id = This.GetFormID();
var value = This.GetInputValue(id);
if(value != ""){
        This.SetShowBorder(false);
}else{
        This.SetShowBorder(true);
}
SCRIPT_END
```

라. 리파지토리 업로드

작성한 보고서를 오즈 이폼 서버의 리파지토리에 업로드 합니다.

- http://192.168.***.*** > '이폼 실습' > 'OZ Repository 업로드 방법'

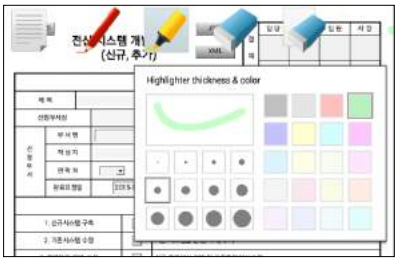
마. 실행

OZ Mobile 앱을 통해 작성한 보고서를 실행합니다.

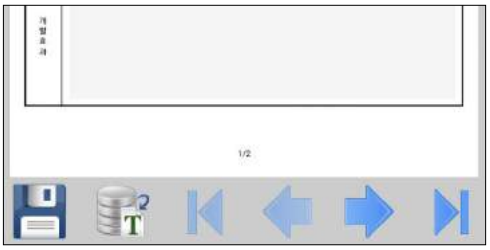
- OZ Mobile 앱은 LTE로 실행하신 후 아래 설정은 교육장 내부 무선 네트워크에서 진행
- http://192.168.***.*** > 'OZ Mobile 연결 방법'
- 'OZ Mobile' 앱 실행 > '이폼 실습' > '오즈 서버 리파지토리 연결' 화면에서 '확인' 클릭 > 오늘 날짜 (yyyyMMdd) 클릭 > '내이름.ozr' 보고서 클릭 > 'OK' > 보고서 실행됨

■ 뷰어 기능

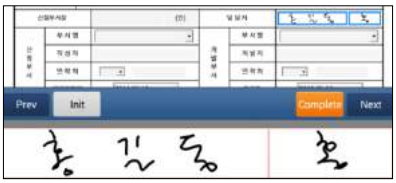
1) 주석 (뷰어 왼쪽 상단의 주석 아이콘 클릭)



2) 뷰어 툴바 (뷰어 하단 클릭)



3) 사인패드 그룹핑 (담당자 항목 클릭)



교재 샘플 소개

(1) 교재 샘플 폴더 구성

'OZwork' 폴더는 '내 문서\OZRepository\OZwork'에 복사되어야 합니다.

폴더	설명
/OZwork/data/	샘플 데이터 (MDB, 데이터 파일)
/OZwork/image/	이미지 파일
/OZwork/overview/	'Chapter2. 오즈 Overview' 관련 파일
/OZwork/design/	'Chapter3. 쿼리 디자이너', 'Chapter4. 리포트 디자이너', 'Chapter7. 기타 데이터 처리' 관련 파일
/OZwork/exercise/	'Chapter5. 보고서 유형별 실습' 관련 파일
/OZwork/viewer/	'Chapter6. 보고서 배포 및 운영' 관련 파일
/OZwork/etc/	기타 참조 파일 (db.properties, UDS 클래스, FX 라이브러리)
/OZwork/doc/	참고 문서 파일

※ 샘플 파일은 http://www.oztn.net/OZEdu/OZReportCourse 에서 다운로드 받을 수 있습니다.
주제 별 관련 파일명은 '/OZwork/doc/교재 샘플 파일 설명.txt'에서 확인할 수 있습니다.

(2) 교재 샘플 DB 설명

- 샘플 DB 위치 : 내 문서\OZRepository\OZwork\data\OZwork.mdb

구분	테이블 명	설명
자동차 관련	Car	자동차 정보
	CarOrders	자동차 판매 정보
식품 관련	Categories	제품 분류 정보
	Products	제품 정보
	Customers	고객 정보
	Orders	주문 배송 정보
	Order Details	주문 매출 정보
전자 제품 관련	eProduct	전자제품 정보

FORCS
FOR Client's Success

(주)포시에스 www.forcs.com
06106 서울시 강남구 논현로 646 포시에스 빌딩
TEL 02)6188-8200 FAX 02)6188-8280