# 목 차

Ⅰ. 시작하기 전에 ..... 3
본 매뉴얼의 목적 ..... 4
본 매뉴얼 구성 ..... 4
매뉴얼 로드맵 ..... 5

Ⅱ. 오즈 스크립트 개요 ..... 6
오즈 스크립트 개요 ..... 7
오즈 스크립트 활용 종류 ..... 7
오즈 스크립트 실행 순서 ..... 9

Ⅲ. 오즈 스크립트 문법과 함수 ..... 15
오즈 스크립트 문법 ..... 16
오즈 스크립트 함수 ..... 25
수학 함수 ..... 26
문자 함수 ..... 27
날짜 및 시간 함수 ..... 30
표준 입출력 함수 ..... 31
전역 플래그 설정 함수 ..... 32
데이터셋 접근 함수 ..... 32
수식 필드 값 설정 함수 ..... 34
데이터셋 SQL문 설정 함수 ..... 35
컴포넌트 속성 제어 함수 ..... 35
크로스탭 관련 함수 ..... 58
편의 함수 ..... 60
보고서 링크 관련 함수 ..... 61

뷰어 옵션 설정 함수 ..... 66
USL 서버 패러미터 관련 함수 ..... 67

Ⅳ. 제품별 활용 방법 ..... 68
오즈 리포트 디자이너 ..... 69
오즈 쿼리 디자이너 ..... 86

# Ⅰ. 시작하기 전에

- 본 매뉴얼의 목적
- 본 매뉴얼 구성
- 매뉴얼 로드맵

# 본 매뉴얼의 목적

본 매뉴얼은 오즈에서 제공하는 보고서 개발에 필요한 스크립트에 대한 지침서로 기본 문법과 제품별 활용법, 활용 예제 등 스크립트 전반에 대해 설명합니다.

# 본 매뉴얼 구성

본 매뉴얼 구성과 각 장에서 다루는 내용은 다음과 같습니다.

## 오즈 스크립트 개요

오즈에서 제공하는 스크립트의 개요와 활용 종류, 스크립트의 실행 순서에 대해 자세히 설명합니다.

## 오즈 스크립트 문법과 함수

오즈 스크립트의 문법과 기본 함수, 제어 가능한 속성 값들에 대해 자세히 설명합니다.

## 제품별 활용 방법

오즈 리포트 디자이너와 오즈 쿼리 디자이너에서 오즈 스크립트를 사용하는 방법과 활용 예를 살펴봅니다.

# 매뉴얼 로드맵

보고서 개발자와 웹 애플리케이션 개발자, 시스템 관리자가 알아야 할 내용을 전달하기 위해 다음과 같은 매뉴얼을 제공합니다.

| Manual | Description |
|---|---|
| **오즈 리포트 Getting Started** | 오즈 소개와 간단한 보고서 작성을 통해 오즈의 개념을 설명합니다. |
| **오즈 리포트 디자이너 사용자 가이드** | 오즈 리포트 디자이너 사용 방법과 보고서 작성하고 작성된 보고서를 웹에 배포하는 방법에 대해 설명합니다. |
| **오즈 리포트 디자이너 사용자 레퍼런스** | 오즈 리포트에서 제공하는 컴포넌트와 마법사에 대해 자세히 설명합니다. |
| **오즈 리포트 뷰어 사용자 가이드** | 보고서를 웹을 통해 조회하고 사용자 애플리케이션과 연동할 수 있도록 뷰어 패러미터와 활용 방법 등에 대해 설명합니다. |
| **오즈 엔터프라이즈 서버 관리자 가이드** | 서버 관리자가 오즈 서버를 효율적으로 관리할 수 있도록 오즈 서버의 설치 및 관리 및 운영 방법을 설명합니다. |
| **에러 메시지 레퍼런스** | 오즈 제품에서 보여지는 에러 메시지에 대해 설명합니다. |

# Ⅱ. 오즈 스크립트 개요

- 오즈 스크립트 개요
- 오즈 스크립트 활용 종류
- 오즈 스크립트 실행 순서

# 오즈 스크립트 개요

오즈는 실행 중 데이터셋, 밴드, 컴포넌트의 속성을 제어하고, 데이터의 연산 작업 등을 할 수 있도록 자체 스크립트 언어를 제공합니다.

# 오즈 스크립트 활용 종류

오즈 스크립트는 다양한 목적으로 활용될 수 있으며 어디에서 어떤 목적으로 사용되었는지에 따라 실행되는 장소와 순서가 달라질 수 있습니다.

## 데이터셋 스크립트 (Scripted Query)

쿼리 데이터셋을 생성하기 위한 쿼리문을 작성할 때 오즈 스크립트를 사용할 수 있으며, 조건에 따라 동적으로 변하는 쿼리를 만들기 위해 사용합니다. 이때 적용된 스크립트는 오즈 서버에서 데이터셋을 생성할 때 실행됩니다.

## 수식 필드 스크립트

수식 필드는 오즈 스크립트를 이용하여 데이터셋에 사용자가 정의한 임의의 필드를 추가할 때 사용합니다. 수식 필드 내 스크립트는 일반적으로 해당 필드가 사용될 때 실행되므로 오즈 뷰어에서 보고서에 해당 수식 필드를 사용하는 라벨을 생성할 때 실행하게 됩니다. 그러나 수식 필드가 타 데이터셋의 쿼리문이나 실행문의 조건 등에 사용될 경우에는 오즈 서버에서 해당 데이터셋을 생성할 때 실행됩니다.

## 컴포넌트 스크립트

컴포넌트의 속성을 동적으로 제어하기 위해 사용하며, 오즈 뷰어에서 해당 컴포넌트 객체를 보고서에 그릴 때 실행됩니다.

### ■ 컴포넌트 스크립트 종류

- 리포트 스크립트

  보고서 레벨 스크립트로 해당 보고서 생성시 한번 실행됩니다.

- 리포트 배경 스크립트

  보고서 배경을 그릴 때 실행되며, 매 페이지마다 실행됩니다.

- 밴드 스크립트

  컴포넌트를 담고 있는 밴드 스크립트로 해당 밴드가 담고 있는 컴포넌트를 생성하기 전에 실행됩니다.

- 컴포넌트 스크립트

  컴포넌트 자체 스크립트로 해당 컴포넌트를 생성할 때 실행됩니다.

## 초기화 스크립트

주로 보고서 관련 초기 설정 값(오즈 뷰어 옵션 설정), 전역 변수 선언 등을 위해 사용하며, 오즈 뷰어에서 보고서 생성 로직 수행 이전에 실행됩니다.

| 스크립트 활용 종류 | 실행 장소 | 작성지원 모듈 |
|---|---|---|
| **데이터셋 스크립트** | 오즈 서버 | 오즈 쿼리 디자이너 |
| **수식필드 스크립트** | 오즈 서버<br>오즈 뷰어 | 오즈 쿼리 디자이너<br>오즈 리포트 디자이너 |
| **컴포넌트 스크립트** | 오즈 뷰어 | 오즈 리포트 디자이너 |
| **초기화 스크립트** | 오즈 뷰어 | 오즈 리포트 디자이너 |

# 오즈 스크립트 실행 순서

기본적으로 오즈 서버에서 실행되는 데이터셋 스크립트나 수식 필드 스크립트가 먼저 실행되고, 오즈 뷰어에서 초기화 스크립트, 컴포넌트 스크립트 순으로 실행됩니다. 오즈 뷰어에서 실행되는 수식 필드 스크립트는 해당 수식 필드를 사용하는 컴포넌트 실행 순서가 적용됩니다.

## 일반적인 실행 순서

리포트 스크립트 → 리포트 배경 스크립트 → 리포트 배경 내 컴포넌트 스크립트 → 밴드 스크립트 → 밴드 내 컴포넌트 스크립트

## 복합 보고서에서 리포트 스크립트 실행 순서

보고서 생성 순서대로 실행되며, 하나의 보고서 내 스크립트 실행이 완료되고 다음 보고서의 스크립트가 실행됩니다.
보고서 생성 순서는 오즈 보고서 디자이너의 컴포넌트 트리에서 확인할 수 있습니다. 복합 보고서에서는 보고서간에 전역 변수 값을 공유하며 먼저 실행되는 보고서에서 선언된 전역 변수는 나중에 실행되는 보고서에서 공유하여 사용할 수 있지만 그 반대의 경우에는 사용할 수 없습니다.

## 밴드 스크립트 실행 순서

밴드의 스크립트 실행 순서는 디자인 시 리포트에 올려지는 밴드 순서를 따르며 윗쪽 밴드에서 아래쪽 밴드 순으로 스크립트가 실행됩니다. 사이드 밴드 내에서의 실행 순서도 사이드 밴드 위에 올려지는 밴드 순서를 따르며 좌측에서 우측 밴드 순으로 스크립트가 실행됩니다.

① 페이지헤더밴드 → ② 타이틀밴드 → ③ 데이터헤더밴드 → ④ 데이터밴드(그려져 있는 상단

의 밴드부터 순서대로) → ⑤ 데이터풋터밴드 → ⑥ 써머리밴드 → ⑦ 테일밴드 → ⑧ 페이지풋터밴드

### ■ 밴드 실행 규칙

보고서의 페이지 생성 과정에서 밴드는 실제 보고서 페이지에 그려지지는 않지만 밴드의 속성 값이나 스크립트 실행 결과로 밴드 크기만큼의 공간을 확보하여 그 공간에 밴드 위의 컴포넌트들이 그려지게 됩니다.

이때 밴드와 해당 밴드위의 컴포넌트의 스크립트가 수행된 후 밴드의 최종 크기가 결정되면 해당 밴드가 현재 페이지에 들어갈 공간이 남아 있는지 확인합니다. 공간이 남아 있으면 현재 페이지에 그리고, 그렇지 않으면 다음 페이지로 넘어가서 그려지게 됩니다.

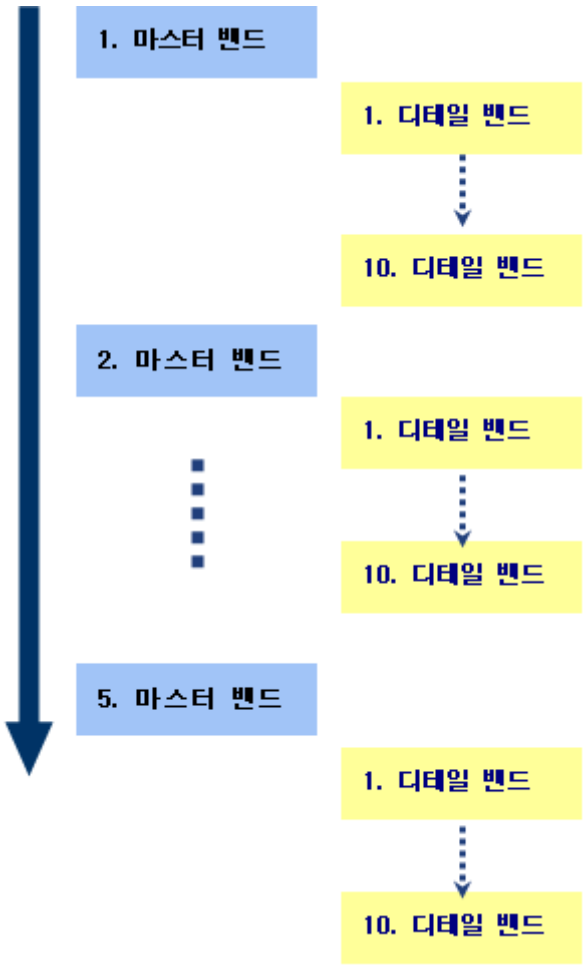



이러한 이유로 페이지 헤더 밴드/풋터 밴드, 테일 밴드와 같이 페이지마다 그려지는 밴드와 이전 밴드에서 페이지 넘김이 명확하게 설정되어 처리된 경우를 제외하고는 두번째 이상의 페이지 제일 윗 부분에 출력되는 밴드의 경우 이전 페이지에서 실행되는 것이 일반적입니다. 이 때문에 밴드 스크립트나 밴드에 속한 컴포넌트에서 현재 페이지를 출력하는 경우 이전 페이지의 번호를 출력할 수도 있고, 또 전역 변수를 사용하여 라벨에 값을 출력하고자 할 때 이전 페이지에서 설정된 값을 가져올 수 있으므로 주의하여야 합니다.

### ■ 마스터-디테일 관계의 데이터 밴드 실행 규칙

데이터 밴드 스크립트는 데이터 건수만큼 반복 수행되며, 마스터-디테일 관계에 있을 때 다음과 같은 순서로 실행됩니다.

예를 들어, 마스터 데이터 건수가 5이고, 각 마스터 데이터당 디테일 데이터셋 건수가 10인 경우에는 아래 그림과 같은 순서로 실행됩니다.

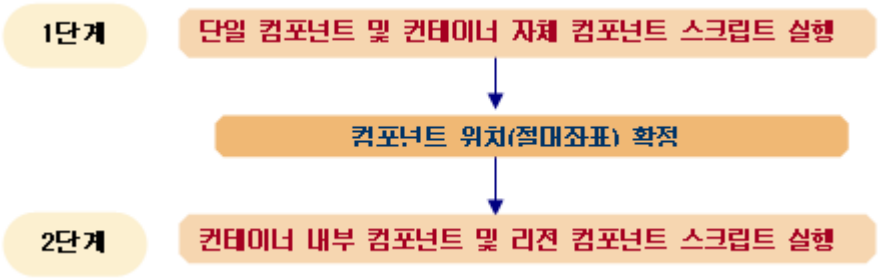



## 같은 밴드 내에서 컴포넌트 실행 순서

보고서 디자인 시 밴드 위에 컴포넌트를 생성한 순서대로 스크립트가 실행되며, 컴포넌트 생성 순서는 오즈 리포트 다자이너의 컴포넌트 트리에서 확인할 수 있습니다. 이미 생성된 컴포넌트의 실행 순서를 바꾸기 위해서는 해당 컴포넌트를 선택하고 마우스 오른쪽 버튼을 클릭하여 [앞으로] / [뒤로] 메뉴를 이용합니다. [뒤로] 기능은 해당 컴포넌트 스크립트 실행 순서를 앞으로 당기고, [앞으로] 기능은 해당 컴포넌트 스크립트 실행 순서를 뒤로 미루게 됩니다.

같은 밴드에서 겹쳐져 있는 컴포넌트의 경우 기본적으로 뒤에 있는 컴포넌트를 먼저 그려야 하기 때문에 뒤에 위치한 컴포넌트의 스크립트를 먼저 실행하게 됩니다. 그러므로 컴포넌트를 [뒤로] 할 경우 해당 컴포넌트를 먼저 실행되게 하기 위해 해당 컴포넌트 먼저 생성된 것처럼 컴포넌트 트리에서 생성 순서를 위로 이동시켜줍니다.

컴포넌트의 스크립트는 다음과 같이 2단계로 나누어 실행됩니다.




다음은 각 컴포넌트별 스크립트 실행 규칙입니다.

■ **단일 컴포넌트**

라벨, 이미지, 바코드 등의 단일 컴포넌트는 해당 컴포넌트 생성 1단계에서 스크립트가 실행됩니다.

예) 컴포넌트 생성 순서 : 라벨1 → 라벨2 → 이미지1

스크립트 실행 순서 : [1단계] 라벨1 → 라벨2 → 이미지1

■ **컨테이너 컴포넌트**

크로스탭, 테이블, 고정 테이블, 간트 차트 등 컴포넌트 내부에 또다시 단일 컴포넌트가 존재할 경우 2단계에 걸쳐 스크립트가 실행되며, 테이블이나 간트 차트 내부 컴포넌트 스크립트는 왼쪽에서 오른쪽, 위에서 아래로 보여지는 순서 그대로 실행됩니다.

[1단계] 컨테이너 컴포넌트 자체 스크립트 수행

[2단계] 컨테이너 내부 단일 컴포넌트들의 스크립트 수행

예) 컴포넌트 생성 순서 : 라벨1 → 테이블1 → 크로스탭1 → 라벨2

스크립트 실행 순서 : [1단계] 라벨1 → 테이블1자체 → 크로스탭1자체 → 라벨2

[2단계] 테이블1의 내부 컴포넌트 → 크로스탭1의 내부 컴포넌트

■ **리전(서브 리포트 개념) 컴포넌트**

리전 컴포넌트는 리전 자체 스크립트 수행과 리전 내부 컴포넌트 스크립트 수행 과정이 별도로 있지만 리포트 스크립트 실행과 유사한 룰을 가지기 때문에 하나의 리전 스크립트가 모두 수행된 후 다음 리전 스크립트가 수행되며, 내부 컴포넌트가 수행되는 2단계에서 리전 자체 스크립트부터 내부 컴포넌트 스크립트까지 한꺼번에 실행됩니다.

예1) 컴포넌트 생성 순서 : 리전1 → 리전2

스크립트 실행 순서 : 리전1자체 → 리전1 내부 컴포넌트 → 리전2 → 리전2 내부 컴포넌트

예2) 컴포넌트 생성 순서 : 리전1 → 테이블1 → 라벨1 → 리전2

스크립트 실행 순서 : [1단계] 테이블1자체 → 라벨 1

[2단계] 리전1자체 → 리전 내부 컴포넌트 → 테이블1 내부 컴포넌트 → 리전2자체 → 리전2 내부 컴포넌트

## ■ 크로스탭 내부 컴포넌트 스크립트 실행 순서

크로스탭은 데이터셋의 전체 데이터에 대해 행과 열에 설정된 필드 단위로 그룹핑 로직이 수행되기 때문에 크로스탭내 각 라벨 스크립트에서 해당 데이터셋의 필드 값을 참조할 수 없습니다.

크로스탭 내 라벨들의 스크립트 실행 순서는 다음과 같습니다.

<table>
<tr><td colspan="2" rowspan="2">18</td><td colspan="2">16</td><td rowspan="2">17<br>(계)</td></tr>
<tr><td>14</td><td>15<br>(소계)</td></tr>
<tr><td rowspan="2">12</td><td>10</td><td>1</td><td>2</td><td>3</td></tr>
<tr><td>11<br>(소계)</td><td>4</td><td>5</td><td>6</td></tr>
<tr><td colspan="2">13<br>(계)</td><td>7</td><td>8</td><td>9</td></tr>
</table>

## ■ 스크립트 실행 순서 관련 주의 사항

- 전역 변수를 사용할 경우 반드시 먼저 실행되는 곳에 선언을 하고 사용하여야 합니다. 초기화 스크립트에 전역 변수를 선언하면 선언되지 않은 전역 변수를 참조하는 등의 실수를 줄일 수 있습니다.
- 자동크기조정이 설정되어 있는 라벨이라도 해당 라벨이 이미 생성된 후에 나중에 타 라벨에서 스크립트로 해당 라벨의 텍스트나 자동 줄바꾸기 등의 속성을 변경하더라도 자동크기조정이 반영되지는 않습니다. 이것은 자동크기조정은 라벨 생성 시점에 한번만 적용되며, 타 속성은 라벨이 생성된 후에도 변경할 수 있기 때문입니다. 이런 경우 해당 라벨의 텍스트 등의 속성을 스크립트로 설정하고 있는 라벨의 실행 순서를 해당 라벨보다 앞으로 당겨주면 변경된 텍스트에 맞게 자동크기조정이 되도록 할 수 있습니다.
- 컴포넌트 위치(Toppose, Leftpose)는 기본적으로 해당 컴포넌트를 담고 있는 밴드내에서의 좌표(상대좌표)로 설정되는데 1단계 스크립트가 실행된 후 각 컴포넌트의 페이지내

에서의 좌표(절대 좌표)가 확정됩니다. 그러므로 1단계에서 실행되는 컴포넌트의 스크립트에서 컴포넌트의 위치를 설정하는 것은 정상적으로 반영되나, 2단계에서 실행되는 컴포넌트(리전 및 컨테이너 내부 컴포넌트)의 스크립트에서 컴포넌트 위치를 설정하면 해당 컴포넌트의 절대좌표를 변경하게 되므로 밴드 내의 좌표 위치가 아닌 페이지 내에서의 좌표위치로 이동하게 됩니다.

# Ⅲ. 오즈 스크립트 문법과 함수

- 오즈 스크립트 문법
- 오즈 스크립트 함수

# 오즈 스크립트 문법

오즈 스크립트는 C 언어와 유사한 문법 구조를 가지고 있습니다.
오즈 스크립트는 대소문자를 구분합니다.

## 주석

// Comment
/* Comment */
"//"에서부터 줄의 끝까지 존재하는 모든 문자는 무시됩니다.
"/*"와 "*/" 사이에 있는 모든 문자는 무시됩니다.

## 자료형

- **int : 32 bit 정수(decimal 표현)**

  초기 값 : 0
  표현 범위 : -2,147,483,648 ~ 2,147,483,647
  예) i = 126; // Ok : decimal '126'
  i = 0; // Ok : i = zero
  i = 00; // Error : oct 표현
  i = 0x666; // Error : hex 표현

- **double : 배정도 실수(소수점 및 지수표현)**

  초기 값 : 0
  최소 값 : -(+)4.94065645841246544E-324
  최대 값 : - (+)1.79769313486231570E+324
  유효 숫자 : 15 ~ 16 자리
  예) f = 10.00;
  f = 1E1;

■ **boolean : 진위 값 ('true' or 'false')**

초기 값 : false

■ **string : 문자열**

초기 값 : ""(Null String)

문자열 안에 특수 문자를 표현할 경우에는 '\'을 이용합니다.

예) 수평 탭 : "문자열\t"

라인 필드 : "문자열\n"

캐리지 리턴 : "문자열\r"

따옴표 : "\"문자열\""

기타 문자 : \+8진수 : "\177"(0~255 이하 문자만 가능)

■ **Date : 날짜 및 시간**

초기 값 : 현재 날짜

| 서식 | 의미 | 형태 | 예제 |
|---|---|---|---|
| G | Year designator | Text | AD |
| y | Year | Number | 1996 |
| M | Month in year | Text or Number | July or 07 (지역 설정에 따라 다름. 한국은 무조건 숫자만 됨) |
| d | day in month | Number | 10 |
| h | hour in am/pm (1~12) | Number | 12 |
| H | Hour in day (0~23) | Number | 0 |
| m | Minute in hour | Number | 30 |
| s | Second in minute | Number | 55 |
| S | Millisecond | Number | 978 |
| E | day in week | Text | Thu (지역 설정에 따라 다름. 한국은 '목') |
| D | day in year | Number | 189 |
| F | day of week in month | Number | 2 (2nd Wed in July) |
| w | Week in year | Number | 27 |
| W | Week in month | Number | 2 |
| a | am/pm marker | Text | PM |
| k | hour in day (1~24) | Number | 24 |
| K | hour in am/pm(0~11) | Number | 0 |

| | | | |
|---|---|---|---|
| z | Time zone | Text | Pacific Standard Time |
| ' | Escape for text | Delimiter | ' |
| '' | Single quote | Literal | '' |

예) 표시형태 적용 예 (지역 설정이 한국인 경우)

| 표시 형태 | 출력 형태 |
|---|---|
| date_EEE, MMM d, ''yy | 토, 7월 25, '09 |
| date_h:mm a | 12:00 오전 |
| date_hh 'o''''clock' a, zzzz | 12 o'clock 오전, GMT+09:00 |
| date_K:mm a, z | 0:00 오전, GMT+09:00 |
| date_yyyy'년' MM'월' dd'일' E'요일' | 2009년 07월 25일 토요일 |
| date_yy년 | 09년 |
| date_yyyy년 | 2009년 |
| date_E요일 | 토요일 |
| date_EEE | 수 |

## 변수와 상수

### ■ 변수

오즈 스크립트는 다른 프로그래밍 언어들과 마찬가지로 지역 변수와 전역 변수의 사용을 지원합니다.

지역 변수 선언 : 자료형 변수명;

전역 변수 선언 : global 자료형 변수명;

- 변수 이름 규칙
  - 영문자, 숫자, 언더라인(Underline)으로 구성되어야 합니다.
  - 공백을 포함할 수 없습니다.
  - 숫자만으로 구성될 수 없으며, 이름에 숫자를 먼저 사용할 수 없습니다.
  - 대소문자를 구분하여 사용하여야 합니다.
- 변수의 유효 범위
  - 지역 변수는 선언된 컴포넌트의 인스턴스(운영 시 생성된 컴포넌터의 객체들) 안에서만 참조할 수 있습니다.

- 전역 변수는 보고서의 모든 컴포넌트의 인스턴스(운영 시 생성된 컴포넌터의 객체들)에서 참조할 수 있습니다. 그러나 전역 변수를 선언한 컴포넌트가 해당 전역 변수를 참조하는 컴포넌트보다 늦게 실행되면 런타임 시 에러가 발생합니다.

- 변수 초기화

변수 선언 시 숫자형은 0, 문자형은 공백으로 초기화됩니다. 단, 변수 선언문 내에서는 변수의 초기 값을 지정할 수 없습니다.

```
int i = 126; // Error : 선언문 내에서 변수를 초기화하면 에러 발생
```

- 전역 변수의 사용

전역 변수임을 나타내기 위하여 변수명 앞에 "::"을 붙여 사용합니다.

```
global int a;
::a = 1;
```

※ 참고사항 : 전역 변수 값을 외부 애플리케이션에서 얻는 방법

예를 들어 오즈 리포트에서 다음과 같은 형태로 전역 변수를 선언하고

```
global string strMsg;
::strMsg = "Hello";
```

자바 스크립트에서 전역 변수 값을 가져오려면 다음과 같이 코딩합니다.

```
...
function GetGlobal(){
var  str;
str = document.OZReportViewerID.Document.GetGlobal('strMsg');
alert(str);
}
...
<input type="button" value="GetValue" onClick="GetGlobal()">
...
```

자바 스크립트에 입력되어있는 "OZReportViewerID"는 오즈 뷰어 호출 태그에서 설정한 Object ID를 입력하여야 합니다.

예를 들어 호출 태그를 다음과 같이 입력한 경우에는

```
...
<OBJECT ID = "ozrviewer" CLASSID="CLSID:
...
```

자바 스크립트에서는

```
...
document.ozrviewer.Document.GetGlobal('strMsg');
...
```

형태로 입력하여야 합니다.

■ **상수**

- null

데이터 값이 없음을 나타내는 상수로 데이터 필드의 값이 null인지를 체크하기 위해 사용하며, null 값을 대입문에서는 사용할 수 없습니다.

```
if (#ODI.Dataset.fieldName# ==null)
{  /* Null 처리 */
}
```

## 연산자

오즈 스크립트에서 사용 가능한 연산자는 다음과 같습니다.

| 연산자 | | 설명 |
|---|---|---|
| 산술 연산자 | + | 더하기/문자열 접합(Concatenation) |
| | - | 빼기/단항 연산자(Unary operator) |
| | * | 곱하기 |
| | / | 나누기 |
| | % | 나머지 연산자(Modules) |
| 대입 연산자 | = | 단순대입 |
| 비교 연산자 | == | 동등 |
| | != | 동등하지 않음 |
| | < | 작다 |
| | <= | 작거나 같다 |
| | > | 크다 |
| | >= | 크거나 같다 |
| 비트 연산자 | & | 비트 AND(int형만 가능) |
| | \| | 비트 OR(int형만 가능) |
| | ^ | 비트 XOR(int형만 가능) |
| | ~ | 부정(int형만 가능) |

| | | |
|---|---|---|
| | && | 논리 AND(boolean형만 가능) |
| | \|\| | 논리 OR(boolean형만 가능) |
| | ! | 부정(boolean형만 가능) |

※ 참고사항

- 연산자별 우선 순위는 "산술연산자 > 비교연산자 > 비트연산자 > 대입연산자" 순입니다.
- 산술 연산자의 우선 순위는 *, /, %가 우선 순위가 높고 +, - 가 우선 순위가 낮습니다.
- 비교 연산자의 리턴값은 boolean형입니다.

자료형별로 사용 가능한 연산자는 다음과 같습니다.

| 연산자 | 설명 |
|---|---|
| **int** | =, ~, ^ , &, \| |
| **int,double** | =, >, <, <=, >=, ==, !=, *, /, +, -, % |
| **boolean** | =, ==, !=, !, &&, \|\| |
| **string,date** | =, ==, != |

## 자료형 변환

자료형에 따른 연산 시 우선 순위는 double, int, boolean 순이며, 대입 시 자동 자료형 변환은 다음과 같습니다.

| 대입문 | | 설명 |
|---|---|---|
| **int=** | double | 소수점 이하 버림 |
| | boolean | true면 1, false면 0 |
| | string | 값 유지(decimal 표현인 경우만 유지) 소수점 이하 버림 |
| **double=** | int | 값 유지 |
| | boolean | true면 1, false면 0 |
| | string | 값 유지(decimal 표현인 경우만 유지) 소수점 이하 버림 |

| | | |
|---|---|---|
| **boolean=** | int | 0이면 false, 아니면 true |
| | double | 0.0이면 false, 아니면 true |
| | string | true면 true, 나머지는 false |
| **string=** | int | 값 유지 |
| | double | 값 유지 |
| | boolean | true면 true, false면 false |
| **Date=** | double | 대입 불가 |
| | boolean | 대입 불가 |
| | string | 대입 불가 |

※ 참고사항

- 다른 자료형과의 연산은 숫자형끼리만 가능하며, 묵시적 형변환은 double형이 우선순위가 높습니다.
- 명시적으로 자료형을 변환하고자 할 때는 변수 혹은 상수 앞에 변환하고자 하는 자료형을 표시합니다. 코드의 명확성을 위하여 명시적으로 자료형을 변환하는 방법을 권장합니다.

```
a = (자료형)b;
string s;
s = "123";
int i = (int)s;
i = (int)"12" + (int)"45";       // i = 57
s = (string)12 + (string)45;     // s = "1245"
```

- Date Type 변수는 자료형 변환이 지원되지 않습니다. 필요한 경우에는 날짜 및 시간 함수(parseDate, formatDate)를 이용할 수 있습니다.

## 제어문

### ■ if-else 문

주어진 조건에 따라 문장을 선택적으로 실행합니다. 제어문에서 Statement가 한 라인일 경우 { }로 묶어 주지 않아도 됩니다.

```
if(expression) {
```

```
    statement ;
} else if(expression) {
    statement ;
} else {
    statement ;
}
```

■ **while 문**

주어진 조건이 true인 동안 문장을 반복 실행합니다. 제어문에서 Statement가 한 라인일 경우 { }로 묶어 주지 않아도 됩니다.

```
while(expression) {
    statement ;
}
```

## 필드 정의

오즈 스크립트에서는 데이터 필드, ODI 필드, 패러미터 필드, 폼 필드, 시스템 필드 등의 값을 가져올 때 "#"을 사용할 수 있습니다.

■ **데이터 필드**

#ODI 이름.데이터셋 이름.필드 이름# 형태로 정의합니다.

예) ODI 이름이 Car, 데이터셋 이름이 Order, 데이터 필드 이름이 Region일 때

```
string FieldValue;
FieldValue = #Car.Order.Region#;
```

※ 참고사항 : ODI 이름 또는 데이터셋 이름 생략 시 필드 값을 가져오는 규칙

- 데이터 밴드 또는 데이터 라벨에서 생략한 경우에는 데이터 밴드 또는 데이터 라벨에 설정된 ODI와 데이터셋으로 설정됩니다. 만일 디테일셋이 설정된 데이터 밴드일 경우 디테일셋에서 필드 이름을 찾지 못하면 마스터 셋으로 설정됩니다.
- 데이터 밴드 외의 다른 밴드 또는 일반 라벨에서 생략한 경우에는 지정한 데이터셋이 있는 첫 번째 ODI로 설정됩니다. 만일 설정된 ODI의 데이터셋에 지정한 필드

이름이 없을 경우 OZ 데이터 트리에 추가된 ODI 중 지정한 필드 이름을 가지고 있는 첫 번째 ODI와 데이터셋으로 설정됩니다

.

■ **ODI 패러미터 필드**

#ODI 이름.OZParam.필드 이름# 형태로 정의합니다.

ex) ODI 이름이 Car, ODI 패러미터 필드 이름이 Region일 때

```
string FieldValue;
FieldValue = #Car.OZParam.Region#;
```

※ 참고사항 : ODI 이름 생략 시 필드 값을 가져오는 규칙

- 데이터 밴드 또는 데이터 라벨에서 생략한 경우에는 데이터 밴드 또는 데이터 라벨에 설정된 ODI로 설정됩니다.
- 데이터 밴드 외의 밴드 또는 일반 라벨에서 생략한 경우에는 지정한 필드 이름을 가진 OZ 데이터 트리에 추가된 ODI 중 지정한 필드 이름을 가지고 있는 첫 번째 ODI로 설정됩니다.

■ **폼 패러미터 필드**

#OZFormParam.필드 이름# 형태로 정의합니다.

ex) 폼 패러미터 필드 이름이 Title일 때

```
string FieldValue;
FieldValue = #OZFormParam.Title#;
```

■ **시스템 필드**

#OZSystem.필드 이름# 형태로 정의합니다.

ex) 시스템 필드가 PageNumber일 때

```
string FieldValue;
FieldValue = #OZSystem.PageNumber#;
```

※ 주의사항 : 시스템 필드 중 Page_of_Total, Total_Page 필드는 지원하지 않습니다.

# 오즈 스크립트 함수

오즈 스크립트는 다양한 종류의 기본 함수를 제공함으로써 보고서 출력을 위한 데이터 조작과 데이터 값에 따른 속성 변환 등을 손쉽게 구현할 수 있도록 합니다.
오즈 스크립트에는 아래와 같은 함수를 지원합니다.

| 함수 | 지원되는 스크립트 |
|---|---|
| **수학 함수** | 모든 스크립트 (데이터셋, 수식 필드, 컴포넌트, 초기화 스크립트) |
| **문자 함수** | 모든 스크립트 (데이터셋, 수식 필드, 컴포넌트, 초기화 스크립트) |
| **날짜 및 시간 함수** | 모든 스크립트 (데이터셋, 수식 필드, 컴포넌트, 초기화 스크립트) |
| **표준 입출력 함수** | 모든 스크립트 (데이터셋, 수식 필드, 컴포넌트, 초기화 스크립트) |
| **전역 플래그 설정 함수** | 모든 스크립트 수식 필드에서 전역 변수 선언이 안되므로 일반적으로 수식 필드에서 전역 변수 대체용으로만 사용할 것을 권장 |
| **데이터셋 접근 함수** | 수식 필드 스크립트 / 컴포넌트 스크립트 |
| **수식 필드 값 설정 함수** | 수식 필드 스크립트 |
| **데이터셋 SQL문 설정 함수** | 데이터셋 스크립트 |
| **크로스탭 관련 함수** | 컴포넌트 스크립트 |
| **컴포넌트 속성 제어 함수** | 컴포넌트 스크립트 |
| **편의 함수** | 컴포넌트 스크립트 |
| **보고서 링크 관련 함수** | 초기화 스크립트 |
| **뷰어 옵션 설정 함수** | 초기화 스크립트 / 컴포넌트 스크립트 |
| **USL 서버 패러미터 관련 함수** | 모든 스크립트 |

## 수학 함수

- **double abs(double n)**

  n의 절대값

- **double acos(double n)**

  n= -1~1, n의 arc cosine 값(0 ~ pi)

- **double asin(double n)**

  n= -1~1, n의 arc sine 값(-pi/2 ~ pi/2)

- **double atan(double n)**

  n의 arc tangent 값(-pi/2 ~ pi/2)

- **double atan2(double n, double m)**

  n/m의 arc tangent 값(-pi ~ pi)

- **double ceil(double n)**

  n보다 같거나 큰 가장 작은 소수점, 소수점 이하의 수를 무조건 올림한 수

  ex) ceil(1.4235) → 2.0

- **double cos(double n)**

  n의 cosine 값

- **double exp(double n)**

  e의 n승, e = 2.71828183…

- **double floor(double n)**

  n보다 작거나 같은 가장 큰 소수점, 소수점 이하의 수를 무조건 버린 수

  ex) floor(1.4235) → 1.0

- **double log(double n)**

  n의 log 값

- **max(arg1, arg2)**

  argument 값 중에서 가장 큰 값, 결과 값의 타입은 첫 번째 인수의 타입을 따름

- **min(arg1, arg2)**

  argument 값 중에서 가장 작은 값, 결과 값의 타입은 첫 번째 인수의 타입을 따름

- **double pow(double m, double n)**

  m의 n승 값

- **double random(double n)**

  0 ~ n 사이의 난수 값

- **double sin(double n)**

  n의 sin 값

- **double sqrt(double n)**

  n의 square root 값 (단, n>0)

- **double tan(double n)**

  n의 tangent 값

## 문자 함수

문자열의 인덱스는 1이 아닌 0부터 시작합니다.

2 byte 문자의 경우에는 한 문자로 인식합니다. 단, strlenb(), substrb() 함수는 예외입니다.

- **string getchar(string src, int n)**

  src의 n+1번째 문자를 리턴

  ex) getchar("abcdef", 2) → "c"

  ex) getchar("가나다라", 2) → "다"

- **string setchar(string src, int n, string dst)**

  src 문자열의 n+1번째 문자를 dst의 첫 번째 문자로 치환한 문자열을 리턴

  dst는 하나의 문자만 가능하며 2 byte 문자의 경우도 한 문자로 인식함

  ex) setchar("abcdef",2, "1") → "ab1def"

  ex) setchar("가나다라",2, "아") → "가나아라"

- **boolean strcmp(string src, string dst)**

  대소문자를 구분하여 src와 dst가 동일한 문자열이면 true, 아니면 false를 리턴

- **boolean stricmp(string src, string dst)**

  src와 dst가 동일한 문자열이면 true, 아니면 false를 리턴

  src와 dst 문자열을 비교할 때 대소문자는 구분하지 않고 비교함

- **int strstr(string str1, string str2)**

  문자열 str1에서 문자열 str2를 찾아서 시작 위치를 리턴

  ex) strstr("abcdef", "cd") → 2

  ex) strstr("가나다라", "다") → 2

- **int strlen(string src)**

  문자열 src의 문자 개수를 리턴

  ex) strlen("abc") → 3

  ex) strlen("가나다") → 3

- **int strlenb(string src)**

  문자열 src의 byte 수를 리턴

  ex) strlenb("abc") → 3

  ex) strlenb("가나다") → 6

- **string strtrim(string src)**

  문자열 src의 앞뒤 공백을 제거한 문자열을 리턴

  strtrim("  a b c d  ") → "a b c d"

- **string substr(string src, int start, int len)**

  문자열 src의 start+1번째부터 len 길이만큼의 문자열을 리턴

  substr("abc가나다", 2,3) → "c가나"

- **string substrb(string src, int start, int len)**

  문자열 src의 start+1번째부터 len 길이만큼의 문자열을 리턴

  길이와 위치정보는 byte로 계산함

  ex) substrb("abc가나다", 2,3) → "c가"

- **string strreplace(string src, string mat, string dst)**

  문자열 src에서 mat를 dst로 치환한 문자열을 리턴

  ex) strreplace("abcdef", "cd", "가나") → "ab가나ef"

- **string strlower(string src)**

  문자열 src 중 [A-Z]의 영문 대문자를 모두 소문자로 바꾼 문자열을 리턴

  ex) strlower("ABC") → "abc"

- **string strupper(string src)**

  문자열 src 중 [a-z]의 영문 소문자를 모두 대문자로 바꾼 문자열을 리턴

  ex) strupper("abc") → "ABC"

- **string strleft(string src, int len)**

  문자열 src의 왼쪽에서 len 길이만큼 잘라낸 문자열을 리턴

  ex) strleft("가나abc",3) → "가나a"

- **string strright(string src, int len)**

  문자열 src의 오른쪽에서 len 길이만큼 잘라낸 문자열을 리턴

  ex) strright("가나abc",4) → "나abc"

- **string strshift(string src, int len)**

  len이 양수면 src가 len만큼 오른쪽으로 옮겨가고, 음수면 왼쪽으로 옮겨감

  ex) strshift("abcdefg",3) → "abcd"

  ex) strshift("abcdefg",-3) → "defg"

- **string strdouble(double value, int option)**

  double 타입의 변수를 문자열로 변환할 때 option 값을 통해 scientific notation을 허용할 것인지 허용하지 않을 것인지를 설정, option을 0으로 설정하면 허용하고, 1로 설정하면 허용하지 않음

  double형인 값을 문자열로 변환하면 0.0001은 "1E-4"와 같이 표현되어 문자열로 변환됨

ex) strdouble(0.0001,0) → "1.0E-4"

ex) strdouble(0.0001,1) → "0.0001"

- **string formatNumber(double db | int i, string strFormat)**

  double, int 타입의 숫자에 strFormat 패턴을 적용시킨 문자열을 리턴

  ex) formatNumber(1234, "$#,###") → $1,234

- **string convertCharCode(string src)**

  입력된 문자열 코드를 문자열로 변환하여 리턴

  ex) convertCharCode("Forcs") → Forcs

## 날짜 및 시간 함수

- **Date setDate(Date date, string flag, int value)**

  flag에 지정된 부분을 date에서 value 값으로 치환하여 리턴

  ex) date가 "2004-06-04"일 때 setDate(date, "day", 12) → "2004-06-12"

- **int getDate(Date date, string flag)**

  flag에 지정된 부분 값을 리턴

  ex) date가 "2004-06-04"일 때 getDate(date, "day") → 4

| Flag | 리턴값 |
|---|---|
| dayofweek | 일요일부터 토요일까지의 요일을 value 값으로 설정/리턴 단, 일요일이 1, 월요일이 2, …, 토요일은 7 |
| day | 일을 value 값으로 설정/리턴 |
| month | 월을 value 값으로 설정/리턴 |
| year | 년을 value 값으로 설정/리턴 |
| hour | 시간대를 value 값으로 설정/리턴 |
| minute | 분을 value 값을 설정/리턴 |
| second | 초를 value 값으로 설정/리턴 |

- **Date parseDate(string DATE_STRING, string format)**

  날짜 또는 시간을 표현하는 문자열 DATE_STRING을 Date형으로 변환

  이때 format은 DATE_STRING이 어떤 형식으로 표현되어 있는지를 나타냄

  ex) parseDate("2004-06-04", "yyyy-MM-dd")

- **string formatDate(Date date, string format)**

  date를 문자열로 변환

  이때 format을 사용하면 특정 형식으로 변환할 수 있음

  ex) date가 "2004-06-04"일 때 formatDate(date, "yyyy/MM/dd") → "2004/06/04"

- **int dateInterval(Date from, Date to)**

  from과 to 사이의 일수를 리턴

  ex) from이 "2004-06-04"이고, to가 "2004-06-06"이고 dateInterval(from, to) → 2

- **Date dateAdd(Date date, int day_amount)**

  date에 day_amount 일만큼을 더한 날짜를 리턴

  ex) date가 "2004-06-04"일 때 dateAdd(date, 2) → 2004-06-06

- **int dateRollAndGet(string flag, int year, int month, int day, int day_amount)**

  year년 month월 day일에 day_amount 만큼을 더한 날짜를 리턴

  이때 지정한 flag에 따라 다른 값을 리턴, flag에 대한 설명은 위의 getDate 함수 설명 부분을 참조바람

  ex) dateRollAndGet("dayofweek", 2001,12,5,30) → 6(금요일)

## 표준 입출력 함수

보고서 디자인 시 디버깅 용도로 사용하는 함수입니다.

- **write(argument, ...)**

  시스템의 표준 출력(system.out)으로 argument의 값을 출력

  argument의 타입에는 제한이 없으며, 오즈 뷰어에서 <Ctrl+Z> 키를 눌러 출력 내용 확인 가능함

ex) int i; write(i);
float f; write(f);

## 전역 플래그 설정 함수

해당 함수는 변수명과 값이 모두 string형 이여야만 하므로 전역 변수 사용을 권장합니다. 전역 변수 사용에 대한 자세한 내용은 앞의 "기본 문법에서 변수" 부분을 참조하시기 바랍니다.

- **void setglobal("flag_name", "value")**
  보고서 전체의 스크립트에서 접근 가능한 전역 플래그 값을 설정
  기본적으로 global 선언문에 의해 선언된 전역 변수와 동일한 효과를 가지나 value 값은 문자열만 허용함

- **string getglobal("flag_name")**
  setglobal 함수에 의해 전역으로 설정된 플래그 값을 리턴
  setglobal로 플래그 값을 설정하지 않은 상태에서 호출하면 런타임 에러가 발생하므로 주의 바람
  ex) setglobal("변수명", "값") → getglobal("변수명")="값"
  setglobal("abc", "3") → getglobal("abc")="3"

## 데이터셋 접근 함수

데이터셋 접근 함수는 밴드나 라벨 등의 컴포넌트에서 사용할 수 있습니다.

- **getdata(string odi_name.dataset_name.field_name, int index)**
  index는 1부터 시작하며, 명시한 데이터셋의 해당 필드에 대한 Index번째 데이터 값을 리턴
  리턴 타입은 가져온 데이터의 타입에 따라 변환됨

- **getdata("field_name")**
- **getdata("odi_name.dataset_name.field_name")**
- **getdata("OZFormParam.param_name")**
- **getdata("odi_name.OZParam.param_name")**
- **getdata("OZSystem.sys_field_name")**

  명시한 데이터 필드, ODI 패러미터 필드, 폼 패러미터 필드, 시스템 필드의 값을 리턴

  index를 지정하지 않고 필드 경로만 설정할 경우 레코드 커서 인덱스에 있는 데이터 값을 리턴

  시스템 필드 중 Page_of_Total, Total_Page 필드는 지원하지 않음

  결과 타입은 필드의 타입에 따라 적절히 double, int, string으로 변환됨

  ※ 참고사항

  (1) 필드 값을 가져올 때 getdata 함수뿐만 아니라 "#"을 이용하여 가져올 수 있습니다. 예를 들어, Car.odi의 Order 데이터셋에 있는 Region 필드 값을 가져올 경우 아래 두 가지 방식으로 가져올 수 있습니다.

  - 함수 사용

```
string FieldValue;
FieldValue = getdata("Car.Order.Region");
```

  - # 사용

```
string FieldValue;
FieldValue = #Car.Order.Region#;
```

  (2) 'odi_name'과 'dataset_name'은 상황에 따라 생략할 수 있으며 생략 시 적용되는 규칙은 본 장의 "오즈 스크립트 문법" - "필드 정의" 부분을 참고하시기 바랍니다.

- **double dbsum("odi_name.dataset_name.field_name")**

  명시한 데이터셋 필드 값들에 대한 합계 값을 리턴

- **double dbavg("odi_name.dataset_name.field_name")**

  명시한 데이터셋 필드 값들에 대한 평균 값을 리턴

- **double dbmax("odi_name.dataset_name.field_name")**

  명시한 데이터셋 필드 값들 중 최대 값을 리턴

- **double dbmin("odi_name.dataset_name.field_name")**
  명시한 데이터셋 필드 값들 중 최소 값을 리턴

- **int dbfreq("odi_name.dataset_name.field_name", "value")**
  명시한 데이터셋 필드 값 중 "value" 값이 나타나는 빈도(횟수)를 리턴
  Null 값은 빈도(횟수)를 리턴하지 않음

- **int totalRowCount("odi_name.dataset_name")**
  명시한 데이터셋의 전체 레코드 수를 리턴
  데이터 밴드나 데이터 라벨에 한해서 데이터셋의 이름 생략 가능. 즉 totalRowCount()와 같이 사용할 수 있으며, 이 경우 현재 컴포넌트나 밴드가 사용하는 데이터셋을 기본 값으로 사용

- **int currentRowIndex("odi_name.dataset_name")**
  명시한 데이터셋의 현재 레코드 인덱스를 리턴
  데이터 밴드나 데이터 라벨에 한해서 데이터셋의 이름 생략 가능. 즉 currentRowIndex()와 같이 사용할 수 있으며, 이 경우 해당 컴포넌트 혹은 데이터 밴드가 사용하는 데이터셋을 기본 값으로 사용

## 수식 필드 값 설정 함수

수식 필드에서는 전역 변수를 선언하거나 사용할 수 없고 수학 함수, 문자 함수, 전역 플래그 설정 함수, 데이터셋 접근 함수의 사용만이 허용됩니다.

- **setFieldData(<data>)**
  수식 필드에서만 사용 가능한 함수로 수식 필드의 값을 설정
  <data>에는 string, int, double, boolean, date 모든 형식의 데이터가 가능하며, 결과 데이터 형식에 맞게 필드 타입을 바꿔 주여야 함
  ex) setFieldData(3.25); → [필드 타입]은 'DOUBLE'
  위와 같이 <data>가 double형인 경우 "필드 타입" 속성을 'DOUBLE'로 바꿔야 함

## 데이터셋 SQL문 설정 함수

SQL 데이터셋에서 스크립트 사용은 오즈 쿼리 디자이너의 쿼리 데이터셋 속성 중 "스크립트 사용" 속성 값을 "true"로 설정한 경우에만 가능하며, 스크립트는 서버에서 실행됩니다. 데이터셋 내 스크립트에는 global 변수를 선언하거나 사용할 수 없으며 수학 함수, 문자 함수, 데이터셋 접근 함수의 사용만이 허용됩니다.

- **setQueryString( "SQL String" )**

  SQL 데이터셋을 생성할 때만 사용되는 함수로 SQL 문장을 설정합니다. SQL 문장내에는 #table.field# 또는 getdata(...)와 같은 문장이 들어가서는 안되며 순수하게 DB에 넘겨 처리될 수 있는 SQL문만 포함되어야 합니다.

  ex) string s;

  ```
  s = "select * from car where Maker = '기아자동차'";
  setQueryString(s)
  ```

  ex) string s;

  ```
  s = "select * from car where Maker = " + "'" + #OZParam.Maker + "'";
  setQueryString(s)
  ```

## 컴포넌트 속성 제어 함수

- **setattr("attr", "value")**

  속성 "attr"을 "value" 값으로 설정

- **getattr("attr")**

  속성 "attr"의 값을 리턴

  결과 타입은 속성 타입을 따름

- **setcompattr("name", "attr", "value")**

  "name"으로 구별되는 외부 컴포넌트(같은 밴드내에 있는 다른 컴포넌트들)의 속성(attr)을 "value" 값으로 설정

- **getcompattr("name", "attr")**

  "name"으로 구별되는 외부 컴포넌트(같은 밴드내에 있는 다른 컴포넌트들)의 속성(attr)

값을 리턴

결과 타입은 속성 타입을 따름

- **setuserattr("attr", "value")**

  사용자 컴포넌트의 속성(attr)을 "value" 값으로 설정

  사용자 컴포넌트에서만 사용할 수 있음

- **getuserattr("attr")**

  사용자 컴포넌트의 속성(attr)의 값을 리턴

  결과 타입은 속성 타입을 따르며, 사용자 컴포넌트에서만 사용할 수 있음

- **setusercompattr("name", "attr", "value")**

  "name"으로 구별되는 외부 사용자 컴포넌트의 속성(attr)을 "value" 값으로 설정

- **getusercompattr("name", "attr")**

  "name"으로 구별되는 외부 사용자 컴포넌트의 속성(attr) 값을 리턴

  결과 타입은 속성 타입을 따름

## 컴포넌트 속성 제어 함수 - 컴포넌트별 제어할 수 있는 속성

- **라벨 / 바코드 / 이미지 / 도형 / 입력 컴포넌트 속성**

| 속성 | 값 유형 | 설명 |
|---|---|---|
| ***Color*** | | |
| bgcolor / fillcolor | Hexa 또는 Decimal 형태의 RGB 값<br>ex) #hhhhhh<br>ex) $ddd,ddd,ddd | 배경색, 글꼴색, 테두리색<br>RGB에 대한 hex 표현과 Decimal표현(ddd) 모두 지원<br>setattr, setcompattr에서만 사용 가능 |
| fgcolor / fontcolor | | |
| framecolor / bordercolor | | |
| ***Geometry*** | | |
| topline | 숫자 값 (타입:DOUBLE) 또는 문자열 | 컴포넌트의 상, 하, 좌, 우측 라인의 두께<br>0일 경우 두께 없음<br>※ 참고사항 : 테두리 선이 한 줄인 경우 숫자 |
| bottomline | | |
| leftline | | |

| | | |
|---|---|---|
| rightline | | 값으로 리턴되고 여러 중인 경우 문자열로 리턴됨. |
| topborderdash<br>bottomborderdash<br>leftborderdash<br>rightborderdash | 문자열 | 왼쪽, 위쪽, 오른쪽, 아래쪽 테두리를 점선으로 설정합니다.<br>※ 주의사항 : 테두리에 점선을 그리기 위해서는 "[점선 길이] [점선 사이의 간격];[점선 길이] [점선 사이의 간격] ..." 형식으로 설정합니다. 설정한 점선은 라벨 안쪽에서 바깥쪽으로 그려지며 맨 마지막 값은 반드시 "점선 사이의 간격"으로 설정하여야 합니다.<br>ex) 6 10;20 5 |
| topborderdashoffset<br>bottomborderdashoffset<br>leftborderdashoffset<br>rightborderdashoffset | 숫자 값 (타입:INT) | 왼쪽, 위쪽, 오른쪽, 아래쪽 테두리를 점선으로 설정 시 점선이 처음 시작되는 시작 지점을 설정합니다. |
| leftpose<br>toppose | 숫자 값 (타입:FLOAT) | 좌측, 상단 위치 |
| height<br>width | 숫자 값 (타입:FLOAT) | 컴포넌트의 높이와 너비 |
| TOPMARGIN<br>BOTTOMMARGIN<br>LEFTMARGIN<br>RIGHTMARGIN | 숫자 값 (타입:FLOAT) | 컴포넌트의 상, 하, 좌, 우측 여백 |
| preferedheight<br>preferedwidth | 숫자 값 (타입:FLOAT) | 텍스트, 바코드, 이미지를 표시하기에 적절한 라벨의 최적 높이와 너비<br>※ 주의사항 : 바코드 컴포넌트의 경우 높이가 자동으로 조정되지 않으므로 "preferedheight" 값을 가져올 경우 디자인된 height 값이 리턴됨. |
| halign | Center / Left / Right / Distributed / Justify | 텍스트 수평 정렬 |
| valign | Center / Top / Bottom / Distributed / Justify | 텍스트 수직 정렬 |
| ***Component Type*** | | |

| | | |
|---|---|---|
| drawtype | Label / Image / Barcode / PDF417 / HTML / User | 그리기 형태 |
| paintallow | Label / Image / Barcode / PDF417 / HTML / User | 그리기 형태 리스트<br>ex) Label, Image |
| ***Font & Display*** | | |
| strech / textdirection | Vertical / Horizontal / Vertical_2BYTE / Clockwise CounterClockwise / Horizontal_2BYTE | 텍스트 방향 |
| writedirection | LeftToRight_TopToBottom / LeftToRight_BottomToTop / RightToLeft_TopToBottom / RightToLeft_BottomToTop | 쓰기 방향 |
| autofontsize | true / false / largeronly / smalleronly | 폰트크기 자동조절 |
| autosize | true / false | 자동크기조정 |
| wordwrap | true / false | 자동 줄 바꾸기 |
| wordwraptype | default / ignorespace / nonasciiword | 줄바꿈 시에 맨 첫글자 공백 표시 여부와 띄어 쓰기 단위로 줄바꿈을 하여 단어가 잘리지 않도록 표시할지 여부 |
| caption | 문자열 | 캡션 |
| value | 문자열 | 텍스트 또는 데이터 값<br>※ 참고사항 : 서식이 적용되어 있을 경우 바인딩 시 표시되는 값이 아닌 서식 적용 전의 값을 가져옴. |
| clip | true / false | 클립핑 (데이터가 라벨 크기를 벗어날 경우의 처리) |
| effect | Basic / Shadow / Engrave / Outline / Hollow / Segments / Dimension | 효과 |

| | | |
|---|---|---|
| spacing | 숫자 값 (타입:DOUBLE) | 글자 간격, 기본 값은 0, 음수 허용 |
| transparent / nofill | true / false | 투명 |
| Transparency | 숫자 값 (타입:INT) | 투명도, 기본 값은 255, 0~255 사이의 값으로 설정 |
| wrapspace | 숫자 값 (타입:DOUBLE) | 줄 간격, 기본 값은 0, 음수 허용 |
| crlftolf | true / false | ASCII 13\|\|10을 ASCII 10으로 변환 |
| fontname | 문자열 | 폰트명 |
| fontsize | 문자열 | 폰트 크기 |
| fontstyle | Plain / Bold / Italic / Bold&Italic | 폰트 스타일 |
| fontstyleex | None / UnderLine / StrikeOut / DoubleStrikeOut | 글꼴 옵션 |
| shrink | true / false | 자동 크기 줄어듬 |
| noborder | true / false | 테두리 없음 |
| FrameDrawingMode | Inside / Center/ Outside | 테두리 라인 위치 |
| visible | true / false | 뷰어 화면에 출력 |
| printable | true / false | 프린트 출력 또는 파일 익스포트 시 포함 여부 |
| format | 문자열 | 수식 혹은 날짜 표현 형식 |
| ***Export*** | | |
| Export_All | true / false | 파일 익스포트 시 모든 포맷에 포함시킴<br>※ 주의사항 : 특정 포맷에서만 익스포트하지 않음으로 설정하려면 먼저 Export_All을 true로 설정한 후 특정 포맷을 false로 설정해야 함.<br>ex)<br>setattr("Export_All","true");<br>setattr("Export_txt","false");|

| | | |
|---|---|---|
| Export_Nothing | true / false | 파일 익스포트 시 모든 포맷에 포함시키지 않음<br>※ 주의사항 : 특정 포맷에서만 익스포트하도록 설정하려면 먼저 Export_Nothing을 true로 설정한 후 특정 포맷을 true로 설정해야 함.<br>ex)<br>setattr("Export_Nothing","true");<br>setattr("Export_txt","true");|
| Export_pdf | true / false | pdf 포맷으로 익스포트 시 포함 여부 |
| Export_xls | true / false | xls 포맷으로 익스포트 시 포함 여부 |
| Export_doc | true / false | doc 포맷으로 익스포트 시 포함 여부 |
| Export_ppt | true / false | ppt 포맷으로 익스포트 시 포함 여부 |
| Export_html | true / false | html 포맷으로 익스포트 시 포함 여부 |
| Export_csv | true / false | csv 포맷으로 익스포트 시 포함 여부 |
| Export_txt | true / false | txt 포맷으로 익스포트 시 포함 여부 |
| Export_jpg | true / false | jpg 포맷으로 익스포트 시 포함 여부 |
| Export_tif | true / false | tif 포맷으로 익스포트 시 포함 여부 |
| Export_svg | true / false | svg 포맷으로 익스포트 시 포함 여부 |
| Export_hml | true / false | hml 포맷으로 익스포트 시 포함 여부 |
| Export_Bankbook | true / false | 통장 프린터로 인쇄 시 포함 여부<br>※ 주의사항 : 라벨, 선 컴포넌트에서만 통장 프린터로 저장 가능합니다. |
| ***Customizable On Viewer*** | | |
| movable | true / false | 뷰어 화면에서 이동 가능 |
| editable | true / false | 뷰어 화면에서 입력/수정 가능 |
| changeable | true / false | 뷰어 화면에서 속성 변환 가능 |
| ***Auto Shift*** | | |
| downrelated | 문자열<br>ex) 컴포넌트1,컴포넌트2,... | 컴포넌트의 자동크기 조절 기능과 관련하여 아래쪽에 영향을 주는 컴포넌트 리스트 |
| rightrelated | 문자열<br>ex) 컴포넌트1,컴포넌트2,... | 컴포넌트의 자동크기 조절 기능과 관련하여 오른쪽에 영향을 주는 컴포넌트 리스트 |

■ 데이터 라벨 컴포넌트 속성

| 속성 | 값 유형 | 설명 |
|---|---|---|
| fieldname | 문자열 | 데이터셋의 필드명 |

■ 수식 라벨 컴포넌트 속성

| 속성 | 값 유형 | 설명 |
|---|---|---|
| expression | 문자열 | 표현식 |

※ 참고사항 : caption 속성으로 표현식 설정 가능합니다.

■ 이미지 라벨 컴포넌트 속성

| 속성 | 값 유형 | 설명 |
|---|---|---|
| imagestyle | Normal / Center / Fit / Tile / FitByWidth / FitByHeight / Isotropic / TopCenter / TopRight / MiddleLeft / MiddleRight / BottomLeft / BottomCenter / BottomRight | 이미지 표현 스타일 |

※ 참고사항 : caption 속성으로 이미지 URL 설정 가능합니다.

■ 바코드 컴포넌트 속성

| 속성 | 값 유형 | 설명 |
|---|---|---|
| BarCode_dataposition | None / Top / Bottom | 바코드 데이터 위치 |
| BarCode_titleposition | None / Top / Bottom | 바코드 타이틀 위치 |
| BarCode_datagap | 숫자 값 (타입:DOUBLE) | 바코드 데이터 문자 간격 |
| BarCode_titlegap | 숫자 값 (타입:DOUBLE) | 바코드 타이틀 문자 간격 |

| | | |
|---|---|---|
| BarCode_type | CODABAR / CODABAR_MOD16 / CODE11 / CODE11_MOD11 / CODE11_MOD11_MOD11 / CODE39 / CODE39_MOD43" / CODE39EXTENDED / CODE39EXTENDED_MOD43 / CODE93 / CODE93_MOD47 / CODE93FULL / CODE93FULL_MOD47 / EAN8 / EAN8PLUS2 / EAN8PLUS5 / EAN13 / EAN13PLUS2 / EAN13PLUS5 / STANDARD2OF5 / STANDARD2OF5_MOD10 / INDUSTRIAL2OF5 / INDUSTRIAL2OF5_MOD10 / INTERLEAVED2OF5 / INTERLEAVED2OF5_MOD10 / MSI / MSI_MOD10 / MSI_MOD11 / MSI_MOD10_MOD10 / MSI_MOD10_MOD11 / OPC / OPC_MOD10 / POSTNET / UPCA / UPCAPLUS2 / UPCAPLUS5 / UPCE / UPCEPLUS2 / UPCEPLUS5 / CODE128A / CODE128A_MOD103 / CODE128B / CODE128B_MOD103 / CODE128C / CODE128C_MOD103 / CODE128AUTO / CODE128AUTO_MOD103 | 바코드 타입 |
| BarCode_halignment | 문자열 | 바코드 수평 정렬 |
| BarCode_scale | 1 / 2 / ... / 9 / 10 | 크기 |
| BarCode_aprintscale | 숫자 값 (타입:INT) | 프린터 출력 크기 |

| | | |
|---|---|---|
| BarCode_ratio | 2 / 3 | 비율 |
| BarCode_margins | 숫자 값 (타입:DOUBLE) | 여백 |
| BarCode_rotation | 0 / 90 / 270 | 회전 각도 |

■ PDF417 바코드 컴포넌트 속성

| 속성 | 값 유형 | 설명 |
|---|---|---|
| PDF417_ROWNUM | 숫자 값 (타입:INT) | 행 수 |
| PDF417_COLNUM | 숫자 값 (타입:INT) | 열 수 |
| PDF417_ECC | 숫자 값 (타입:INT) | 에러 교정 |
| PDF417_XSCALE | 숫자 값 (타입:INT) | X 크기 |
| PDF417_YSCALE | 숫자 값 (타입:INT) | Y 크기 |

■ 테이블 컴포넌트 속성

| 속성 | 값 유형 | 설명 |
|---|---|---|
| EvenHeightAtGrouping | true / false | 행 높이 조정 |

■ 크로스탭 컴포넌트 속성

| 속성 | 값 유형 | 설명 |
|---|---|---|
| CrossTab_fittorightmargin | None / LastHorzPage / AllPage | 우측 여백에 맞춤 |

■ 차트 컴포넌트 속성

| 속성 | 값 유형 | 설명 |
|---|---|---|
| ***Common*** | | |
| Chart_horizontal_axislabel_2stair | true / false | 수평 축 라벨 2단 표시 |
| Chart_style | 문자열 | 차트 스타일<br>차트 스타일 예약어는 아래 표 참조 |

| | | |
|---|---|---|
| Chart_color | 문자열<br>ex) #hhhhhh;;#hhhhhh<br>ex)<br>$ddd,ddd,ddd;;$ddd,ddd,ddd | 차트 색상<br>RGB에 대한 hexa 표현과 Decimal 표현을 모두 지원하며 여러 개의 색을 설정할 경우에는 ";"을 구분자로 하여 설정<br>※ 참고사항 : 차트 색상을 투명으로 설정하고자 할 경우에는 색상 값을 입력하지 않고 세미 콜론(;)만 입력하여 차트 색상을 설정합니다. |
| topborderdash<br>bottomborderdash<br>leftborderdash<br>rightborderdash | 문자열 | 왼쪽, 위쪽, 오른쪽, 아래쪽 테두리를 점선으로 설정<br>※ 주의사항 : 테두리에 점선을 그리기 위해서는 "[점선 길이] [점선 사이의 간격];[점선 길이] [점선 사이의 간격] ..." 형식으로 설정합니다. 설정한 점선은 라벨 안쪽에서 바깥쪽으로 그려지며 맨 마지막 값은 반드시 "점선 사이의 간격"으로 설정하여야 합니다.<br>ex) 6 10;20 5 |
| topborderdashoffset<br>bottomborderdashoffset<br>leftborderdashoffset<br>rightborderdashoffset | 숫자 값 (타입:INT) | 왼쪽, 위쪽, 오른쪽, 아래쪽 테두리를 점선으로 설정 시 점선이 처음 시작되는 시작 지점을 설정합니다. |
| ShowFigure | true / false | 패턴 적용 여부 |
| ChartLineFigures | 문자열 | 선 패턴<br>※ 주의사항 : 선 패턴은 "[점선 길이] [점선 사이의 간격],[점선 길이] [점선 사이의 간격] ..." 형식으로 설정합니다. 설정 시 맨 마지막 값은 반드시 "점선 사이의 간격"으로 설정하여야 합니다.<br>ex) 3 3,5 5,33 33 |

| | | |
|---|---|---|
| ChartAreaFigures | LeftInclined / Scissor / Horizontal / RightInclined / Plus / Vertical / LeftInclined2 / Scissor2 / Horizontal2 / RightInclined2 / Plus2 / Vertical2 / Empty | 면 패턴<br>※ 참고사항 : 여러 개의 면 패턴을 설정할 경우에는 ","를 구분자로 하여 설정.<br>ex) Plus,Scissor,Vertical |
| AreaFigureColor | 문자열<br>ex) #hhhhhh<br>ex) $ddd,ddd,ddd | 면 패턴 색상<br>RGB에 대한 hexa 표현과 Decimal 표현을 모두 지원 |
| ChartMarkFigures | Circle / Rectangle / Triangle / InvertedTriangle / Diamond / Circle_Transparent / Rectangle_Transparent / Triangle_Transparent / InvertedTriangle_Transparent / Diamond_Transparent / Snow / LongBar / ShortBar / Scissor / Plus / HorizontalLine1 / HorizontalLine2 / HorizontalLine3 / HorizontalLine4 / HorizontalLine5 / Empty / VerticalLine | 마크 패턴<br>※ 참고사항 : 여러 개의 마크 패턴을 설정할 경우에는 ","를 구분자로 하여 설정.<br>ex) Circle,Triangle,Snow |
| ***X Axis - Common*** | | |
| Chart_max_#ofitem | 문자열 | X축 최대 눈금 수 |
| Chart_period_x | 문자열 | X축의 라벨 주기 |
| Chart_sub_period_x | 숫자 값(타입:INT) | X 축의 보조 눈금선 주기, 기본 값은 1<br>1보다 작은 값을 설정하였을 경우에는 기본 값인 1로 재설정됨 |
| Chart_tooltip_caption | 문자열 | 해당 차트의 툴팁에 표시될 캡션 값<br>오즈 스크립트(링크)에서만 사용 가능 |

| | | |
|---|---|---|
| Chart_tooltip_colpivot | 문자열 | 해당 차트에서 현재 마우스가 위치한 곳의 계열 값<br>계열에 다중 항목이 설정되어 있을 경우 콤마(,)를 구문자로 하여 여러 개 값 리턴(계열 value1, 계열 value2,.....)<br>오즈 스크립트(링크)에서만 사용 가능<br>getattr 함수에서만 사용 가능 |
| Chart_tooltip_contents | 숫자 값(타입:DOUBLE) | 해당 차트에서 현재 마우스가 위치한 곳의 Y 축 값<br>오즈 스크립트(링크)에서만 사용 가능<br>getattr 함수에서만 사용 가능 |
| Chart_tooltip_rowpivot | 문자열 | 해당 차트에서 현재 마우스가 위치한 곳의 X 축 값<br>X 축에 다중 항목이 설정되어 있는 경우 콤마(,)를 구문자로 하여 여러 개 값 리턴(XValue, XValue2,.....)<br>오즈 스크립트(링크)에서만 사용 가능<br>getattr 함수에서만 사용 가능 |
| Chart_tooltip_text | 문자열 | 해당 차트에서 현재 마우스가 위치한 곳의 데이터 라벨 값<br>차트 마법사의 라벨 쪽에 필드를 지정하지 않은 경우 Y축 값 리턴 |
| Chart_tooltip_value | 숫자 값(타입:DOUBLE) | 해당 차트에서 현재 마우스가 위치한 곳의 실제 Y 축 값<br>만약 누적 차트인 경우 Chart_tooltip_contents는 누적 값을 리턴하지만 Chart_tooltip_value는 실제 해당하는 Y 값을 리턴함 |
| XAxisBottomTitle | 문자열 | X축 아래쪽 타이틀 |
| XAxisRightTitle | 문자열 | X축 오른쪽 타이틀 |
| XAxisReverse | true / false | X축 역순 |
| ***X Axis - Item Type*** | | |
| YCrossBetweenX | Default / Yes / No | 항목 사이에 Y(값) 축 교차 |
| ***X Axis - Date Type*** | | |
| Chart_baseline_x | Min / Max / 문자열 | X 축의 기준선<br>- MIN : X 축 최소 값으로 설정<br>- MAX : X 축 최대 값으로 설정<br>- 날짜를 직접 설정 (ex.2008-05-28) |
| Chart_max_x | Auto / 숫자 값 (타입:DOUBLE) | X 축 최대 값<br>Auto : 데이터 값에 따라 자동으로 설정 |

| | | |
|---|---|---|
| Chart_min_x | Auto / 숫자 값 (타입:DOUBLE) | X 축 최소 값<br>Auto : 데이터 값에 따라 자동으로 설정 |
| Chart_origin_x | Min / Max / 문자열 | X 축 원점<br>- MIN : X 축 최소 값으로 설정<br>- MAX : X 축 최대 값으로 설정<br>- 날짜를 직접 설정<br>ex) 2008-05-27 |
| Chart_roundoff_x<br>Chart_roundup_x | year_january / year_april / halfyear_january / halfyear_april / quarter / month / week / day / hour / minute / second | X 축 최소 값에 대한 내림 자릿수<br>X 축 최대 값에 대한 올림 자릿수 |
| Chart_unitofticks_x | year / 2year / halfyear / quarter / 2month / month / 2week / week / 2day / day/ hour/ 10minute / minute / 10second / second | X 축에 대한 눈금 간격 |
| ***X Axis - Number Type*** | | |
| Chart_#ofticks_auto_x | true / false | X 축에 대한 눈금 수 자동 설정 여부 |
| Chart_#ofticks_x | 숫자 값 (타입:INT) | X 축에 대한 눈금 수<br>눈금 간격과 함께 설정된 경우 눈금 수가 우선 처리됨 |
| Chart_baseline_x | Min / Max / 숫자 값 (타입:INT) | X 축의 기준선<br>- MIN : X 축 최소 값으로 설정<br>- MAX : X 축 최대 값으로 설정<br>- 값을 직접 설정 |
| Chart_max_auto_x | true / false | X 축 최대 값 자동 지정 여부 |
| Chart_max_x | Auto / 숫자 값 (타입:DOUBLE) | X 축 최대 값<br>Auto : 데이터 값에 따라 자동으로 설정 |
| Chart_min_auto_x | true / false | X 축 최소 값 자동 지정 여부 |
| Chart_min_x | Auto / 숫자 값 (타입:DOUBLE) | X 축 최소 값<br>Auto : 데이터 값에 따라 자동으로 설정 |
| Chart_origin_x | Min / Max / 숫자 값 (타입:DOUBLE) | X 축 원점<br>- MIN : X 축 최소 값으로 설정<br>- MAX : X 축 최대 값으로 설정<br>- 값을 직접 설정 |
| Chart_roundoff_x<br>Chart_roundup_x | 숫자 값(타입:INT) | X 축 최소 값에 대한 내림 자릿수<br>X축 최대 값에 대한 올림 자릿수 |
| Chart_unitofticks_x | 숫자 값(타입:INT) | X축에 대한 눈금 간격 |
| ***Y Axis*** | | |

| | | |
|---|---|---|
| Chart_#ofticks_auto_y1<br>Chart_#ofticks_auto_y2 | true / false | Y축에 대한 눈금 수 자동 설정 여부 |
| Chart_#ofticks_y1<br>Chart_#ofticks_y2 | 숫자 값 (타입:INT) | Y 축에 대한 눈금 수<br>(눈금 간격과 함께 설정된 경우 눈금 수가 우선 처리됨)<br>※ 참고사항 : 눈금 수를 설정하려면 먼저 "Y축에 대한 눈금 수 자동 설정 여부"를 "false"로 설정하여야 합니다. |
| Chart_baseline_y1<br>Chart_baseline_y2 | Min / Max / 숫자 값 (타입:INT) | Y 축의 기준선<br>- MIN : Y 축 최소 값으로 설정<br>- MAX : Y 축 최대 값으로 설정<br>- 값을 직접 설정 |
| Chart_max_auto_y1<br>Chart_max_auto_y2 | true / false | Y축 최대 값 자동 지정 여부 |
| Chart_max_y1<br>Chart_max_y2 | Auto / 숫자 값 (타입:DOUBLE) | Y 축 최대 값<br>Auto : 데이터 값에 따라 자동으로 설정<br>※ 참고사항 : 최대 값을 특정한 값으로 설정하려면 먼저 "Y축 최대 값 자동 지정 여부" 여부를 "false"로 설정하여야 합니다. |
| Chart_min_auto_y1<br>Chart_min_auto_y2 | true / false | Y축 최소 값 자동 지정 여부 |
| Chart_min_y1<br>Chart_min_y2 | Auto / 숫자 값 (타입:DOUBLE) | Y 축 최소 값<br>Auto : 데이터 값에 따라 자동으로 설정<br>※ 참고사항 : 최소 값을 특정한 값으로 설정하려면 먼저 "Y축 최소 값 자동 지정 여부" 여부를 "false"로 설정하여야 합니다. |
| Chart_origin_y1<br>Chart_origin_y2 | Min / Max / 숫자 값 (타입:DOUBLE) | Y 축 원점<br>- MIN : Y 축 최소 값으로 설정<br>- MAX : Y 축 최대 값으로 설정<br>- 값을 직접 설정 |
| Chart_period_y | 문자열 | Y축의 라벨 주기 |
| Chart_roundoff_y1<br>Chart_roundoff_y2<br>Chart_roundup_y1<br>Chart_roundup_y2 | 숫자 값(타입:INT) | Y 축 최소 값에 대한 내림 자릿수<br>Y축 최대 값에 대한 올림 자릿수 |

| | | |
|---|---|---|
| Chart_sub_period_y | 숫자 값(타입:INT) | Y 축의 보조 눈금선 주기, 기본 값은 1<br>1보다 작은 값을 설정하였을 경우에는 기본 값인 1로 재설정됨 |
| Chart_unitofticks_y1<br>Chart_unitofticks_y2 | 숫자 값(타입:INT) | Y축에 대한 눈금 간격 |
| YAxisLeftReverse | true / false | 왼쪽 Y축 역순 여부 |
| YaxisLeftTitle | 문자열 | 왼쪽 Y축 타이틀 |
| YaxisLeftTopTitle | 문자열 | 왼쪽 Y축 상단 타이틀 |
| YAxisRightReverse | true / false | 오른쪽 Y축 역순 여부 |
| YaxisRightTitle | 문자열 | 오른쪽 Y축 타이틀 |
| YaxisRightTopTitle | 문자열 | 오른쪽 Y축 상단 타이틀 |
| ***Z Axis*** | | |
| Chart_period_z | 문자열 | Z축의 라벨 주기 |
| Chart_sub_period_z | 숫자 값(타입:INT) | Z 축의 보조 눈금선 주기, 기본 값은 1<br>1보다 작은 값을 설정하였을 경우에는 기본 값인 1로 재설정됨 |

- 차트 스타일 예약어

| 차트 모양 | 차트 이름 | 예약어 |
|---|---|---|
| **세로 꺾은 선형** | | |
|  | 꺾은 선형 | VLine_NoPoint |
|  | 누적 꺾은 선형 | VLine_Accumulative_NoPoint |
|  | 100% 기준 누적 꺾은 선형 | VLine_Percent_NoPoint |
|  | 데이터 표식이 있는 꺾은 선형 | VLine |
|  | 데이터 표식이 있는 누적 꺾은 선형 | VLine_Accumulative |
|  | 데이터 표식이 있는 100% 기준 누적 꺾은 선형 | VLine_Percent |
|  | 입체 효과 꺾은 선형 | VLine_3D |

| | | |
|---|---|---|
|  | 입체 효과 꺾은 선형 | VLine_Rotational_3D |
|  | 입체 효과 꺾은 선형 | VLine_3D_Zwise |
|  | 입체 효과 꺾은 선형 | VLine_Rotational_3D_Zwise |
| **가로 꺾은 선형** | | |
|  | 꺾은 선형 | HLine_NoPoint |
|  | 누적 꺾은 선형 | HLine_Accumulative_NoPoint |
|  | 100% 기준 누적 꺾은 선형 | HLine_Percent_NoPoint |
|  | 데이터 표식이 있는 꺾은 선형 | HLine |
|  | 데이터 표식이 있는 누적 꺾은 선형 | HLine_Accumulative |
|  | 데이터 표식이 있는 100% 기준 누적 꺾은 선형 | HLine_Percent |
| **세로 막대형** | | |
|  | 묶은 세로 막대형 | VBar |
|  | 누적 세로 막대형 | VBar_Accumulative |
|  | 100% 기준 누적 세로 막대형 | VBar_Percent |
|  | 입체 효과의 묶은 세로 막대형 | VBar_3D_Bundle |
|  | 입체 효과의 누적 세로 막대형 | VBar_3D_Accumulative |
|  | 입체 효과의 100% 기준 누적 세로 막대형 | VBar_3D_Percent |
|  | 입체 효과 세로 막대형 | VBar_3D |

| | | |
|---|---|---|
|  | 입체 효과 세로 막대형 | VBar_Rotational_3D |
|  | 입체 효과 세로 막대형 | VBar_3D_Zwise |
|  | 입체 효과 세로 막대형 | VBar_Rotational_3D_Zwise |
| **가로 막대형** | | |
|  | 묶은 가로 막대형 | HBar_Bundle |
|  | 누적 가로 막대형 | HBar_Accumulative |
|  | 100% 기준 누적 가로 막대형 | HBar_Percent |
|  | 입체 효과의 묶은 가로 막대형 | HBar_3D_Bundle |
|  | 입체 효과의 누적 가로 막대형 | HBar_3D_Accumulative |
|  | 입체 효과의 100% 기준 누적 가로 막대형 | HBar_3D_Percent |
| **세로 원통형** | | |
|  | 묶은 세로 원통형 | VCylinder |
|  | 누적 세로 원통형 | VCylinder_Accumulative |
|  | 100% 기준 누적 세로 원통형 | VCylinder_Percent |
|  | 입체 효과의 묶은 세로 원통형 | VCylinder_3D_Bundle |
|  | 입체 효과의 누적 세로 원통형 | VCylinder_3D_Accumulative |
|  | 입체 효과의 100% 기준 누적 세로 원통형 | VCylinder_3D_Percent |
|  | 입체 효과 세로 원통형 | VCylinder_3D |

| | | |
|---|---|---|
|  | 입체 효과 세로 원통형 | VCylinder_3D_Zwise |
| **가로 원통형** | | |
|  | 묶은 가로 원통형 | HCylinder |
|  | 누적 가로 원통형 | HCylinder_Accumulative |
|  | 100% 기준 누적 가로 원통형 | HCylinder_Percent |
|  | 입체 효과의 묶은 가로 원통형 | HCylinder_3D_Bundle |
|  | 입체 효과의 누적 가로 원통형 | HCylinder_3D_Accumulative |
|  | 입체 효과의 100% 기준 누적 가로 원통형 | HCylinder_3D_Percent |
| **원뿔형** | | |
|  | 입체 효과의 묶은 세로 원뿔형 | VCone_3D_Bundle |
|  | 입체 효과의 누적 세로 원뿔형 | VCone_3D_Accumulative |
|  | 입체 효과의 100% 기준 누적 세로 원뿔형 | VCone_3D_Percent |
|  | 입체 효과의 묶은 가로 원뿔형 | HCone_3D_Bundle |
|  | 입체 효과의 누적 가로 원뿔형 | HCone_3D_Accumulative |
|  | 입체 효과의 100% 기준 누적 가로 원뿔형 | HCone_3D_Percent |
|  | 입체 효과 세로 원뿔형 | VCone_3D |
|  | 입체 효과 세로 원뿔형 | VCone_3D_Zwise |
| **피라미드형** | | |

| | | |
|---|---|---|
|  | 입체 효과의 묶은 세로 피라미드형 | VPyramid_3D_Bundle |
|  | 입체 효과의 누적 세로 피라미드형 | VPyramid_3D_Accumulative |
|  | 입체 효과의 100% 기준 누적 세로 피라미드형 | VPyramid_3D_Percent |
|  | 입체 효과의 묶은 가로 피라미드형 | HPyramid_3D_Bundle |
|  | 입체 효과의 누적 가로 피라미드형 | HPyramid_3D_Accumulative |
|  | 입체 효과의 100% 기준 누적 가로 피라미드형 | HPyramid_3D_Percent |
|  | 입체 효과 세로 피라미드형 | VPyramid_3D |
|  | 입체 효과 세로 피라미드형 | VPyramid_3D_Zwise |
| **세로 영역형** | | |
|  | 영역형 | VArea |
|  | 누적 영역형 | VArea_Accumulative |
|  | 100% 기준 누적 영역형 | VArea_Percent |
|  | 입체 효과의 영역형 | VArea_3D |
|  | 입체 효과의 누적형 | VArea_3D_Accumulative |
|  | 입체 효과의 100% 누적형 | VArea_3D_Percent |
|  | 입체 효과의 영역형 | VArea_Rotational_3D |
|  | 입체 효과의 영역형 | VArea_3D_Zwise |

| | | |
|---|---|---|
|  | 입체 효과의 영역형 | VArea_Rotational_3D_Zwise |
| **가로 영역형** | | |
|  | 영역형 | HArea |
|  | 누적 영역형 | HArea_Accumulative |
|  | 100% 기준 누적 영역형 | HArea_Percent |
|  | 입체 효과의 누적형 | HArea_3D_Accumulative |
|  | 입체 효과의 100% 누적형 | HArea_3D_Percent |
| **산포형** | | |
|  | 산포형 | VPoint |
|  | 산포형 | HPoint |
| **원형** | | |
|  | 원형 | Pie |
|  | 분리된 원형 | Pie_Split |
|  | 입체 효과의 원형 | Pie_3D |
|  | 분리된 입체 효과의 원형 | Pie_Split_3D |
| **게이지** | | |
|  | 게이지 | Gauge |
| **분산형** | | |
|  | 분산형 | Radiant_NoPoint |

| | | |
|---|---|---|
|  | 누적형 | Radiant_Accumulative_NoPoint |
|  | 퍼센트형 | Radiant_Percent_NoPoint |
|  | 포인트 방사형 | Radiant |
|  | 누적형 | Radiant_Accumulative |
|  | 퍼센트형 | Radiant_Percent |
| **계단형** | | |
|  | 계단형 | VStep |
|  | 계단형 | HStep |
| **사용자** | | |
|  | 사용자형 | VCustom |
|  | 누적 사용자형 | VCustom_Accumulative |
|  | 100% 기준 누적 사용자형 | VCustom_Percent |
|  | 사용자형 | HCustom |
|  | 누적 사용자형 | HCustom_Accumulative |
|  | 100% 기준 누적 사용자형 | HCustom_Percent |

## ■ 간트 차트 컴포넌트 속성

간트 차트 속성들은 setattr 함수에서만 사용 가능합니다.

| 속성 | 값 유형 | 설명 |
|---|---|---|

| | | |
|---|---|---|
| Gantt_startdate | Auto / Parameter / 날짜 | 간트 차트를 그리는 전체 날짜 중 시작일(최소 날짜)<br>- AUTO : 데이터 값에 따라 자동 설정<br>- Parameter : gt_fromDate라는 이름의 패러미터 값을 받아서 설정<br>- 날짜 : yyyy-MM-dd 형태의 날짜 |
| Gantt_enddate | Auto / Parameter / 날짜 | 간트 차트를 그리는 전체 날짜 중 종료일(최대 날짜)<br>- AUTO : 데이터 값에 따라 자동 설정<br>- Parameter : gt_toDate라는 이름의 패러미터 값을 받아서 설정<br>- 날짜 : yyyy-MM-dd 형태의 날짜 |
| Gantt_taskbartype | Separated / United | 막대형식 설정<br>- Separated : 계획바와 실적바가 나누어진 형태<br>- United : 계획바와 실적바가 합쳐진 형태 |
| Gantt_timescale | Year / Quarter / Month / Week / Day | 계획 및 실적에 대한 기준 시간 유형을 년도별, 분기별, 월별, 주별, 일별로 설정 |
| Gantt_weektype | Sun-Mon / Mon-Sun | 주 표시 유형 설정<br>시간 유형이 주별일 경우만 적용되며 주를 일요일에서 월요일까지로 볼 것인지, 월요일에서 일요일까지로 볼 것인지 설정 |

■ **밴드 컴포넌트 속성**

| 속성 | 값 유형 | 설명 |
|---|---|---|
| autosize | true / false | 자동크기 조절<br>※ 주의사항 : 페이지 헤더/풋터, 테일 밴드에는 적용되지 않습니다. |
| enable | true / false | 밴드 활성화<br>※ 참고사항 : 값이 "false"인 경우에는 밴드 또는 밴드에 있는 컴포넌트들의 스크립트가 실행되지 않고 해당 밴드가 보이지 않습니다. |
| visible | true / false | 밴드 보이기<br>※ 참고사항 : 값이 "false"인 경우에는 밴드 또는 밴드에 있는 컴포넌트들의 스크립트는 실행되나, 해당 밴드는 보이지 않습니다 |
| height | 숫자 값 (단위:INT) | 밴드 높이<br>※ 주의사항 : 페이지 풋터, 테일, 사이드 밴드의 height는 스크립트상에서 변경할 수 없습니다. |

| | | |
|---|---|---|
| width | 숫자 값 (단위:INT) | 밴드 너비<br>※ 주의사항 : width는 익스팬더 밴드에만 유효합니다. |
| forcenewpage | true / false | 다음 페이지로 넘김<br>※ 주의사항 : 타이틀 밴드에는 적용되지 않습니다 |
| datasetname | 문자열 | 데이터셋 이름 |
| emptydataband | true / false | 데이터 밴드의 반복 횟수 값이 데이터 밴드에 연결된 데이터 건수보다 많을 경우 빈 밴드로 바인딩되었는지 여부<br>※ 주의사항 : getattr, getcompattr, getuserattr, getusercompattr만 가능합니다 |
| pageendstop | true / false | 설정한 반복 횟수만큼 밴드가 반복되다가 페이지 끝에서 멈춤<br>※ 주의사항 : 더미 밴드에서만 가능합니다. |
| repeatnum | 숫자 값 (단위:INT) | 밴드 반복 횟수<br>※ 주의사항 : 더미 밴드, 데이터 밴드, 그룹 밴드, 데이터 익스팬더 밴드에서만 가능합니다. |

## ■ 리전 컴포넌트 속성

| 속성 | 값 유형 | 설명 |
|---|---|---|
| Region_hgap | 숫자 값 (타입:FLOAT) | 리전 수평 간격 |
| Region_vgap | 숫자 값 (타입:FLOAT) | 리전 수직 간격 |

## ■ 도형 컴포넌트 속성

| 속성 | 값 유형 | 설명 |
|---|---|---|
| ROUND | 숫자 값 (타입:FLOAT) | 사각형 컴포넌트의 곡선 효과 |
| LINETYPE | Horizontal / Vertical / Slash / Backslash | 선과 화살표 컴포넌트의 선 종류 |
| THICKNESS | 숫자 값 (타입:FLOAT) | 선과 화살표 컴포넌트의 선 굵기 |
| DASH | 숫자 값 (타입:FLOAT) | 선과 화살표 컴포넌트의 점선(GAP과 함께 사용) |
| GAP | 숫자 값 (타입:FLOAT) | 선과 화살표 컴포넌트의 간격(DASH와 함께 사용) |
| ARROWARROWSIZE | 숫자 값 (타입:FLOAT) | 화살표 컴포넌트의 화살표 크기 |

| | | |
|---|---|---|
| ARROWTYPE | 1 / 2 / 3 / 4 / 5 / 6 | 화살표 컴포넌트의 화살표 모양<br>1 : 삼각형, 2 : 날카로운 삼각형<br>3 : 열린 삼각형, 4 : 마름모<br>5 : 원, 6 : 반원 |
| ARROWDIRECT | 1 / 2 / 3 | 화살표 컴포넌트의 화살표 방향<br>1 : 기본 방향, 2 : 역방향<br>3 : 양방향 |

■ **입력 컴포넌트 속성**

| 속성 | 값 유형 | 설명 |
|---|---|---|
| Protocol | POST / GET | 버튼 컴포넌트 고유 속성으로 폼 데이터의 전송 방식 |
| URL | 문자열 | 버튼 컴포넌트 고유 속성으로 데이터를 받는 프로그램(ASP, JSP)의 URL |
| Checked | true / false | 체크 박스 및 라디오 버튼 컴포넌트 고유 속성으로 체크 박스의 체크 유무 설정 |
| SelectedIdx | 정수 값 (타입:INT) | 콤보 박스 컴포넌트 고유 속성으로 보여질 인덱스 설정 |
| GroupName | 문자열 | 라디오 버튼 컴포넌트 고유 속성으로 여러 개의 라디오 버튼들을 하나의 그룹 이름으로 설정 |

## 크로스탭 관련 함수

아래 크로스탭 관련 함수의 depth 즉, 레벨은 0부터 시작합니다.

■ **string getColPivot(int depth)**

지정한 깊이의 열 피봇 값을 리턴

depth는 pivot의 깊이를 말하며, 가장 상위가 0이고 아래 단계로 1씩 증가합니다.

■ **string getRowPivot(int depth)**

지정한 깊이의 행 피봇 값을 리턴

depth는 pivot의 깊이를 말하며, 가장 상위가 0이고 아래 단계로 1씩 증가합니다.

■ **double getSummary("SUM"|"AVG"|"MIN"|"MAX", string rowpivot0, string rowpivot1,...,string colpivot0,string colpivot2,...)**

피봇 값에 해당하는 수식(sum, avg, min, max) 결과 값을 리턴

값이 문자열이나 이미지인 경우 크로스탭의 속성상 sum이 아닌 non, cnt만 표현되며, 스크립트 실행시 0 값이 반환됨

CTValueShape에서만 사용 가능하며 아규먼트는 행 피봇 값 순서대로, 열 피봇 값 순서대로 인식함

적절하지 않은 아규먼트가 설정되면 해당 아규먼트는 무시됨

ex) 행 피봇이 2개, 열 피봇이 2개인 크로스탭의 경우

string rp0, rp1, co0, cp1;

rp0=getRowPivot(0)

rp1=getRowPivot(1)

cp0=getColPivot(0)

cp1=getColPivot(1)

getSummary("SUM", rp0, rp1, cp0, cp1); → 해당 행, 열 피봇 값의 sum값

getSummary("AVG", rp0, rp1); → 해당 행 피봇 값의 AVG값

getSummary("MAX", "", "", cp0, cp1); → 해당 열 피봇 값의 MAX 값(행 피봇 값이 empty string이 있을 경우에는 해당 행, 열 피봇 값의 MAX값을 리턴)

getSummary("MIN"); → 크로스탭 전체 value 중 MIN 값

ex) 아래와 같은 크로스탭의 경우

| Sample | | 현대 | | | 대우 | | | 합계 |
|---|---|---|---|---|---|---|---|---|
| | | 소나타 | 그랜저 | 합계 | 누비라 | 레간자 | 합계 | |
| 전라 | 전주 | 9<br>2 | 10<br>3 | 24,0 | 6<br>5<br>3 | | 14,0 | 38,0 |
| | 광주 | | 2<br>4 | 6,0 | 4<br>6 | 6<br>7 | 23,0 | 29,0 |
| | 합계 | 11,0 | 19,0 | 30,0 | 24,0 | 13,0 | 37,0 | 67,0 |
| 충청 | 대전 | 3<br>4 | 4<br>3 | 14,0 | 1<br>8 | | 9,0 | 23,0 |
| | 충주 | 9 | 8<br>9<br>8 | 34,0 | 7<br>0 | 9<br>7 | 23,0 | 57,0 |
| | 합계 | 16,0 | 32,0 | 48,0 | 16,0 | 16,0 | 32,0 | 80,0 |
| 합계 | | 27,0 | 51,0 | 78,0 | 40,0 | 29,0 | 69,0 | 147,0 |

getSummary("SUM", "전라", "전주", "현대", "소나타") → 11.0

getSummary("SUM", "전라", "전주") → 38.0

getSummary("SUM", "전라", "서울") → 67.0 (행 피봇 값 "전라"의 하위 피봇 값 중 "서울"이 존재하지 않으므로 무시

하고 "전라"의 sum값을 리턴함)

getSummary("SUM", "", "", "현대", "그랜저") → 51.0

getSummary("SUM") → 147.0

- **void Pivot(int fromAxis, int fromDepth,int toAxis, int toDepth)**
- **void Pivot(String fromAxis,  int fromDepth,String toAxis, int toDepth)**

해당 피봇을 이동

Axis : 축(행/열), 0이면 행, 1이면 열

Depth : 피봇의 레벨, 0이면 최상위 피봇

크로스탭이 그려지기 전에 크로스탭에 함수를 이용해서 크로스탭의 피봇을 동적으로 변경할 수 있음

- **void removePivot(int Axis, int Depth)**
- **void removePivot(String Axis, int Depth)**

해당 피봇을 삭제

크로스탭이 그려지기 전에 크로스탭에 함수를 이용해서 크로스탭의 피봇을 동적으로 변경할 수 있음

Axis : 축(행/열), 0이면 행, 1이면 열

Depth : 피봇의 레벨, 0이면 최상위 피봇

## 편의 함수

- **boolean isSpaceRemained(double height)**

밴드에서만 사용할 수 있는 함수로 height 만큼의 밴드 공간이 현재 페이지에 들어갈 수 있는지를 체크하여 공간이 있으면 'true', 없으면 'false'를 리턴

만약 height 값이 0보다 같거나 작을 경우 현재 밴드의 기본 height 값으로 계산됨

이 함수를 이용하여 현재 밴드가 다음 페이지로 넘어갈 것인지 여부를 미리 알 수 있음

isSpaceRemained(0)가 false인 경우 현재 밴드가 다음 페이지로 넘어가서 출력됨

- **double getsystem("empty_space")**

밴드가 그려지기 전 현재 페이지에 남아있는 공간을 리턴

기본적으로 페이지 풋터와 테일 밴드를 위한 부분을 제외한 순수하게 사용 가능한 공간을

의미하며 해당 공간의 높이를 Pixel 단위로 리턴함
밴드에서만 사용할 수 있음

- **void msgBox("title", "msg")**
  메시지 박스를 표시
  title : 메시지 박스의 타이틀
  msg : 메시지 박스에 나타낼 메시지

- **encodeURI("value")**
  입력된 문자열을 URI 방식으로 인코딩하여 반환
  예) encodeURI("!@#$%^&*")

  ※ 참고사항 : encodeURI 함수는 특수 문자가 포함된 데이터를 서버로 전송 시 URI 방식으로 인코딩할 경우에 사용합니다.

## 보고서 링크 관련 함수

보고서 링크 함수 중 setLinkServer, setLinkServlet, setLinkURL는 함수가 여러 번 반복되더라도 가장 마지막에 설정된 보고서만 링크됩니다.
링크에 관련된 함수가 실행되면 setLinkParam, setLinkOption으로 링크 함수 앞에 설정한 값은 무시됩니다.

- **setLinkServer("host_name/host_IP", port_num, "report_file_name")**

- **setLinkServlet("servlet_url", "report_file_name")**
  보고서 미리보기 상태에서 특정 컴포넌트를 클릭하여 관계된 상세 보고서를 호출할 수 있는 보고서 링크 함수로, 오즈 서버 데몬 및 서블릿에 링크할 보고서를 요청
  오즈 서버 데몬일 경우 port_num을 명시하여야 함
  일반 라벨 컴포넌트의 경우 각 컴포넌트의 스크립트에 링크 함수를 작성할 수 있으며, 차트 컴포넌트의 경우 차트 컴포넌트를 클릭하여 보고서 링크를 구현하기를 원할 경우에는 차트 컴포넌트 속성 중 "오즈 스크립트(링크)"에 링크 함수를 작성해야 함
  최상위 보고서의 오즈 서버 주소를 그대로 따를 경우에는 report_file_name만 지정하면 됨

즉, 데몬 서버인 경우에는 host_name/host_IP를 ""로, port_num을 0으로 설정하고 서블릿 서버인 경우에는 servlet_url를 ""로 설정

■ **setLinkParam("OZFormParam_name", "value")**

링크할 보고서에 폼 패러미터 값을 넘김

반복해서 설정할 때마다 패러미터 리스트에 추가되고, 이름이 같을 경우 값만 수정됨

setLinkParam() 함수는 폼 패러미터 값을 설정하고, ODI 패러미터 값은 setLinkOption() 함수를 이용하여 설정함

■ **setLinkOption("option_tag", "value")**

보고서 링크시 오즈 뷰어 관련 옵션 또는 ODI 패러미터 값 등을 설정

ex) ODI 패러미터 값을 설정하는 경우

```
setLinkOption("odi.odinames", "sample1");
setLinkOption("odi.sample1.pcount", "2");
setLinkOption("odi.sample1.args1", "FromDate=2004-01-01");
setLinkOption("odi.sample1.args2", "ToDate=2004-01-31");
```

그외 설정 가능한 옵션은 아래 표를 참조하세요.

| option_tag | value | 설명 |
|---|---|---|
| refreshperiod | hh:mm:ss | 시간:분:초로 지정한 시간마다 보고서 데이터 자동 갱신 |
| refreshcache | true / false | 서버의 캐시를 갱신하도록 요청 (기본 값 : false) |
| compressedForm | true / false | 압축된 폼 요청 |
| odi.odi_name.usescheduleddata | 데이터 파일명 | 서버의 스케줄링 결과 데이터를 가져오도록 요청 |
| displayname | 리포트 이름 | 보고서 트리에 보여줄 리포트 이름을 지정 |
| openfile | 오즈 파일명 | 로컬 PC에서 오즈 파일 오픈 할때 사용 |
| presavedata | true / false | 서버로부터 가져온 데이터를 로컬 하드에 저장할지 여부 (기본 값 : false) |
| usesecurity | true / false | 보고서 링크 시에 사용자 아이디 및 패스워드를 입력하는 메시지 창 사용 여부 (기본 값 : false) |

| | | |
|---|---|---|
| largebundle | true / false | 여러 페이지의 보고서를 한 페이지로 보기<br>(기본 값 : false) |

■ **setLinkURL("url", "target")**

웹 브라우저 target 프레임의 URL을 호출하여 표시

url : 호출할 URL로, 프로토콜과 같이 입력 ex) http://www.forcs.com

target : 웹 브라우저 호출 옵션

| target | 뷰어를 Stand-Alone 형태로 띄운 경우 | 뷰어를 웹 브라우저 안에 띄운 경우 |
|---|---|---|
| _self | 보고서를 호출한 윈도우 또는 프레임에 해당 URL이 표시됨 | 보고서가 포함된 윈도우 또는 프레임에 해당 URL이 표시됨 |
| _parent | 보고서를 호출한 프레임의 부모 프레임에 해당 URL이 표시됨. 만약 부모 프레임이 없는 경우에는 _self를 사용한 것과 동일하게 표시됨 | 보고서가 포함된 프레임의 부모 프레임에 해당 URL이 표시됨. 만약 부모 프레임이 없는 경우에는 _self를 사용한 것과 동일하게 표시됨 |
| _top | 보고서를 호출한 윈도우 탑 레벨 프레임에 해당 URL이 표시됨. 만약 리포트를 호출한 윈도우가 탑 레벨 프레임이면 _self를 사용한 것과 동일하게 표시됨 | 보고서가 포함된 윈도우 탑 레벨 프레임에 해당 URL이 표시됨. 만약 리포트를 호출한 윈도우가 탑 레벨 프레임이면 _self를 사용한 것과 동일하게 표시됨 |
| _blank | 새로운 윈도우에 해당 URL이 표시됨 | |
| 윈도우명 | 지정된 윈도우에 해당 URL이 표시됨 | |

■ **SetLinkPostURL("url", "target", "postparam")**

웹 브라우저 target 프레임의 URL을 POST 방식으로 호출하여 표시

만일 뷰어가 StandAlone 형태로 실행된 경우에는 "target"이 "_blank"일 경우에만 사용 가능함

url : 호출할 URL로, 프로토콜과 같이 입력 ex) http://www.forcs.com

target : 웹 브라우저 호출 옵션

postparam : post 방식으로 전달할 패러미터

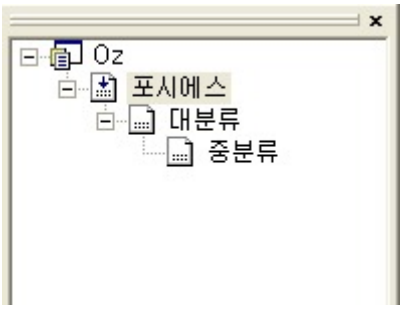
■ **setLinkToc("tocName")**

목차 라벨을 설정

목차 라벨은 보고서 트리에 표시되는 목차에 대해서만 링크가 허용됨

tocName : 목차 라벨의 경로. "/"를 구분자로 하여 "리포트템플릿의표시이름/단계1/단계2/.../단계n" 형태로 설정

ex) 아래 그림과 같은 보고서 트리에서




특정 라벨을 클릭했을 때 "중분류"로 이동할 수 있도록 목차 링크를 설정하려면 라벨에 다음과 같이 스크립트를 입력하여야 함

setLinkToc("포시에스/대분류/중분류"); //리포트템플릿의표시이름/단계1/단계2

■ **setLinkRepository("url", "mode")**

서버 리파지토리에 있는 파일을 실행 또는 저장

setLinkRepository 실행시 반드시 리파지토리 패러미터가 설정되어있어야 함

URL : 리파지토리 파일 경로 예) ozp://test.xls, ozp://category/test.xls

mode : 파일에 대한 동작

| mode | 설명 |
|---|---|
| run | 링크된 파일 실행 |
| save | 링크된 파일 저장 |
| choice | 링크된 파일 실행/저장/취소 선택 |

예) 서버 리파지토리에 있는 test.xls 파일을 실행시키는 예

setLinkRepository("ozp://test.xls", "run")

■ **setLinkExecute("cmd", "parameter")**

링크 선택시 설정한 명령문을 실행

cmd : 실행하고자 하는 명령어

parameter : 명령어 실행시 전달할 패러미터

예) test.txt 파일을 메모장으로 실행시키는 예

setLinkExecute("notepad.exe", "c://test.txt")

■ **setBrowserOption("option_tag", "value")**

보고서 링크 시 웹 브라우저가 실행될 때 웹 브라우저의 화면을 제어하는 옵션을 설정합니다. 설정 가능한 옵션은 다음과 같습니다.

| option_tag | value | 설명 |
|---|---|---|
| ShowToolbar | true / false | 웹 브라우저 실행 시 툴바의 표시 여부<br>(기본 값 : true) |
| ShowStatusBar | true / false | 웹 브라우저 실행 시 상태 표시줄의 표시 여부<br>(기본 값 : true) |
| ShowMenuBar | true / false | 웹 브라우저 실행 시 메뉴바의 표시 여부<br>(기본 값 : true) |
| ShowAddressBar | true / false | 웹 브라우저 실행 시 주소 입력창의 표시 여부<br>(기본 값 : true) |
| Left | "<INT>" | 웹 브라우저의 왼쪽 좌표 |
| Top | "<INT>" | 웹 브라우저의 상단 좌표 |
| Width | "<INT>" | 웹 브라우저의 너비 |
| Height | "<INT>" | 웹 브라우저의 높이 |

※ 주의사항 : setBrowserOption 함수를 호출하기 전에 "setLinkPostURL" 함수나 "setLinkURL" 함수가 먼저 호출되어야 동작합니다.

예) 웹브라우저의 툴바를 숨기는 경우

```
setLinkPostURL("www.forcs.com", "_blank");
setBrowserOption("ShowToolbar","false");
```

■ **setLinkEvent("UserTag", "UserValue")**

링크 이벤트를 발생시키고 외부 이벤트(OZLinkCommand, OZUserActionCommand)로 인자 값을 전송

UserTag : 사용자 정의 태그

UserValue : 사용자 정의 값

## 뷰어 옵션 설정 함수

■ **setReportOption("option_tag", "value")**

오즈 뷰어의 패러미터 값을 설정

주로 보안이 요구되는 옵션 값을 설정할 때 사용하며, 초기화 스크립트에서만 사용 가능함

오즈 뷰어 호출 HTML에 있는 패러미터 값보다 우선 순위가 높음

ex) setReportOption("connection.pcount","3")

리포트 뷰어 옵션 중 setReportOption()로 제어 가능한 옵션은 아래 표를 참조하세요.

| 패러미터 | 제어 가능한 옵션 |
|---|---|
| connection | args#, clientdmtype, compresseddatamodule, displayname, fetchtype, pageque, pcount, refreshcache, refreshperiod, serverdmtype, usescheduleddata |
| csv | exceptfirstpage, filename, pagetitle, pageline, pagestyle, path, removerange, savetointeger, separator |
| export | applyformat, confirmsave, filename, format, lockcontrols, mode, onlydata, path |
| excel | arrangestyle, exceptfirstpage, filename, matchmode, matchsubmode, numberformat, path, removeblank, removeoption, removerange, savetointeger |
| han97 | filename, path, saveastable |
| hml | filename, keeplabelsize, path, saveastable, savetobackground, savetoheaderfooter, verttoparagraph |
| hdm | indent, separator |
| html | filename, horizontal, imagepath, offsetx, offsety, path, savebypage, vertical |
| information | debug, turnoffdynamic |
| jpg | filename, path |
| memo | bordercolor, bottomline, caption, clip, effect, halign, fillcolor, fontcolor, fontname, fontsize, fontstyle, leftline, rightline, spacing, stretch, topline, valign, wordwrap, wrapspace |
| odi | odinames, odiname.args#, odiname.clientdmtype, diname.fetchtype, odiname.pcount, odiname.serverdmtype |
| ozd | filename, memoallowed, path, password, saveall |
| pdf | author, creator, filename, keyword, masterpassword, path, printable, saverange, title, userpassword, subject |
| ppt | filename, path |

| | |
|---|---|
| print | adjust, alldocument, close, collate, copies, ignoreerror, lockopt, mode, once, pageorder, pageorient, pagerange, pages, pagesinone, printbypage, printername, size, spoolpages, style, usedefaultpaper, usedialogopt |
| svg | filename, path |
| text | exceptfirstpage, filename, pagetitle, pageline, pagestyle, path, removerange, separator, savetointeger |
| tiff | encode, filename, path |
| toolbar | position, iconwidth, iconheight, iconposition, bgcolor, useseparator, all, file, open, print, save, option, showtree, refresh, addmemo, inputparameter, find, savednm pageselection, pagenavigator, top, up, bottom, down, leftmost, left, rightmost, right, zoom, zoomout, zoomin, zoomtype, zoomreset, about, close |
| viewer | addmarginleft, addmargintop, bgcolor, bgimage, editmodecolor, frameheight, framewidth, framex, framey, marginleft, margintop, progresscommand, showtree, smartframesize, useeditmode, useinborder, useoutborder, usestatusbar, zoom |
| word | filename, path |

## USL 서버 패러미터 관련 함수

- **getUSLServerParam("name")**

  USL 서버를 사용할 경우 서버에서 설정된 패러미터 값을 가져옴. 해당 패러미터가 없는 경우 "" 을 리턴함

  예) getUSLServerParam("OZUSL_SERVER_URL")

# Ⅳ. 제품별 활용 방법

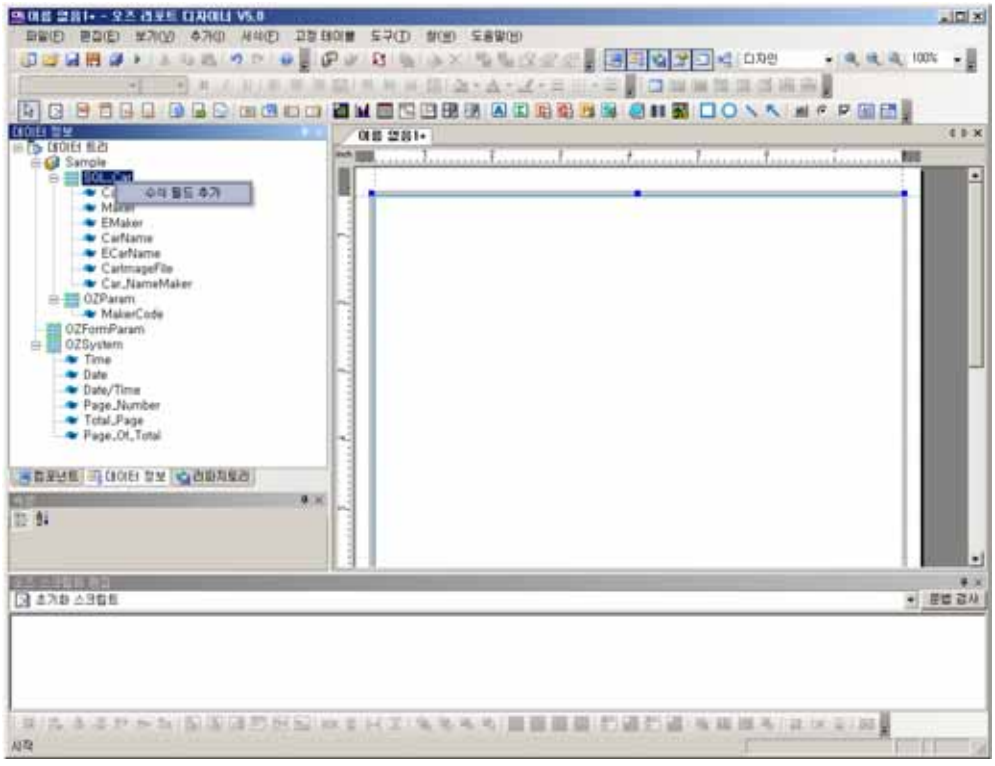
- 오즈 리포트 디자이너
- 오즈 쿼리 디자이너

# 오즈 리포트 디자이너

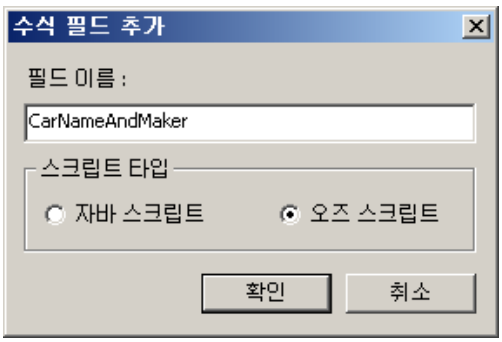
오즈 리포트 디자이너에서는 수식 필드를 추가하거나 또는 컴포넌트의 속성을 제어할 때 오즈 스크립트를 사용합니다.

## 수식 필드 스크립트

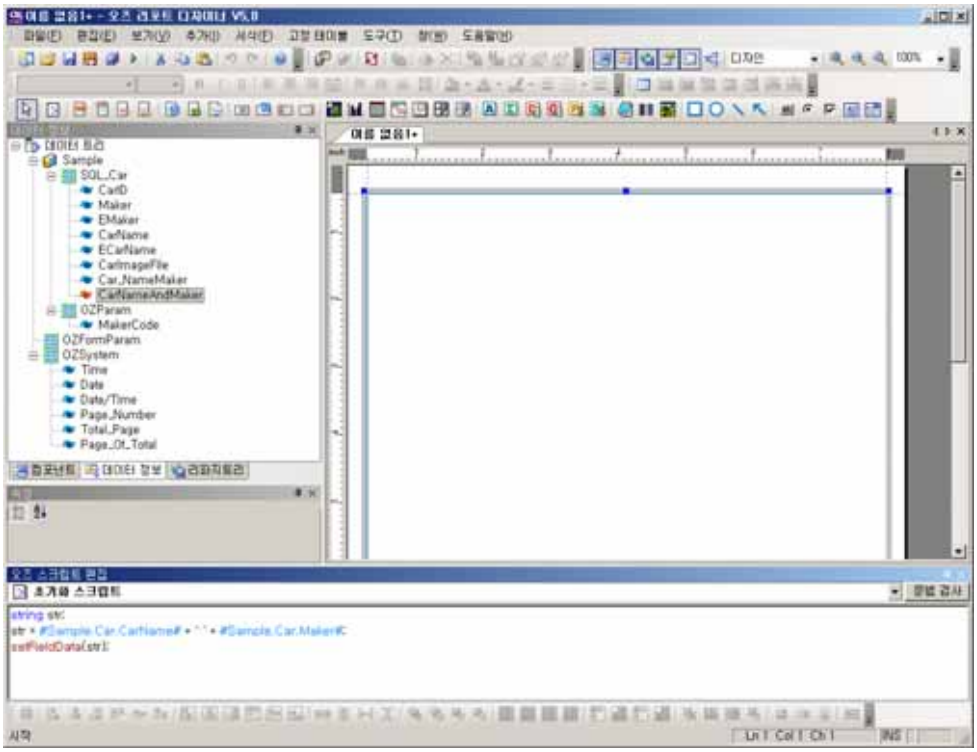
오즈 쿼리 디자이너와 마찬가지로 사용자가 필요한 필드를 추가할 경우에 사용합니다.
추가하고자 하는 데이터셋에서 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [수식 필드 추가] 메뉴를 클릭합니다.




수식 필드 추가 다이얼로그에서 필드 이름을 입력하고 스크립트 타입을 "오즈 스크립트"로 선택한 후 [확인] 버튼을 클릭합니다.




추가된 수식 필드를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [수식 필드 수정] 메뉴를 클릭하여 오즈 리포트 디자이너의 오즈 스크립트 편집 창에서 스크립트를 입력합니다.

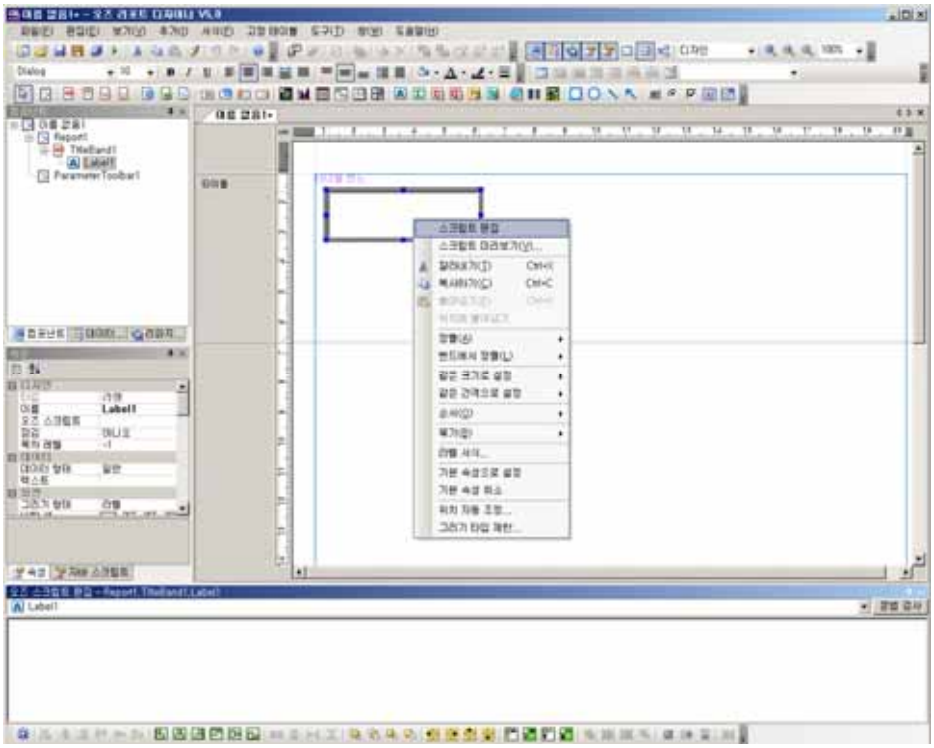


```
string str;
str = #Sample.Car.CarName# + " " + #Sample.Car.Maker#;
setFieldData(str);
```

스크립트 작성이 완료되면 [문법 검사] 버튼을 클릭하여 스크립트에 문법 에러가 없는지 확인합니다.

## 컴포넌트 스크립트

오즈 리포트 디자이너에서는 보고서, 밴드, 컴포넌트 각각에 스크립트를 추가할 수 있습니다.
밴드 또는 컴포넌트를 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [스크립트 편집] 메뉴를 클릭하고 오즈 스크립트 편집 창을 활성화한 후 오즈 스크립트 편집 창에 스크립트를 입력합니다.

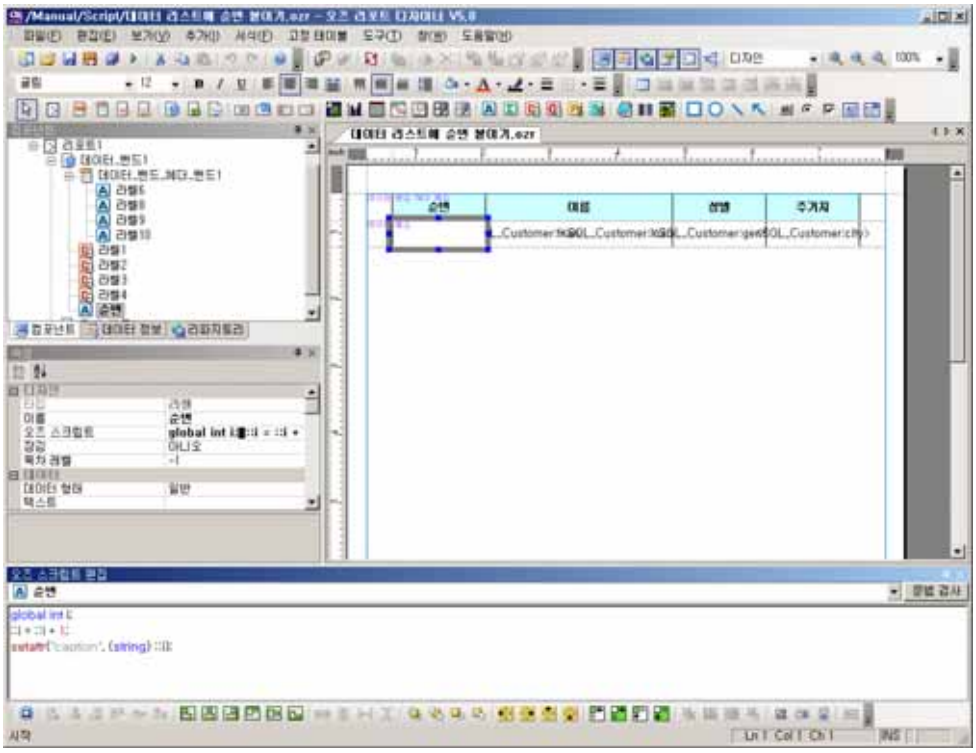


## 활용 예

■ **데이터 리스트에 순번 붙이기**

고객 리스트, 상품 리스트 등 다양한 데이터 리스트를 출력하는 보고서에서 첫 번째 컬럼으로 순번을 붙이기 원하는 경우가 많습니다. SQL문에서 순번을 붙여서 가져오기 힘든 경우 오즈 스크립트를 이용할 수 있습니다.

순번을 표시할 라벨을 추가한 후 스크립트를 입력합니다.



입력된 스크립트는 다음과 같습니다.

```
global int i;
::i = ::i + 1;
setattr("caption", (string) ::i);
```

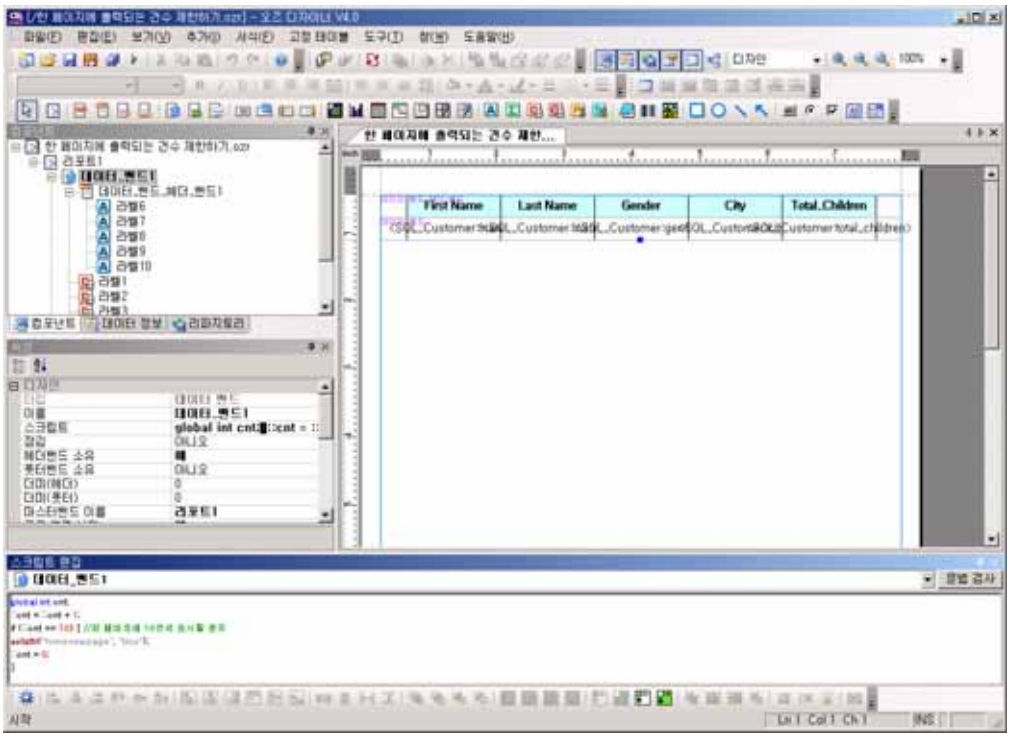
보고서를 미리보기하면 순번 라벨에 순번이 표시됩니다.




■ **한 페이지에 출력되는 건수 제한하기**

스크립트를 이용하여 한 페이지에 출력되는 데이터 건수를 제한할 수 있습니다.

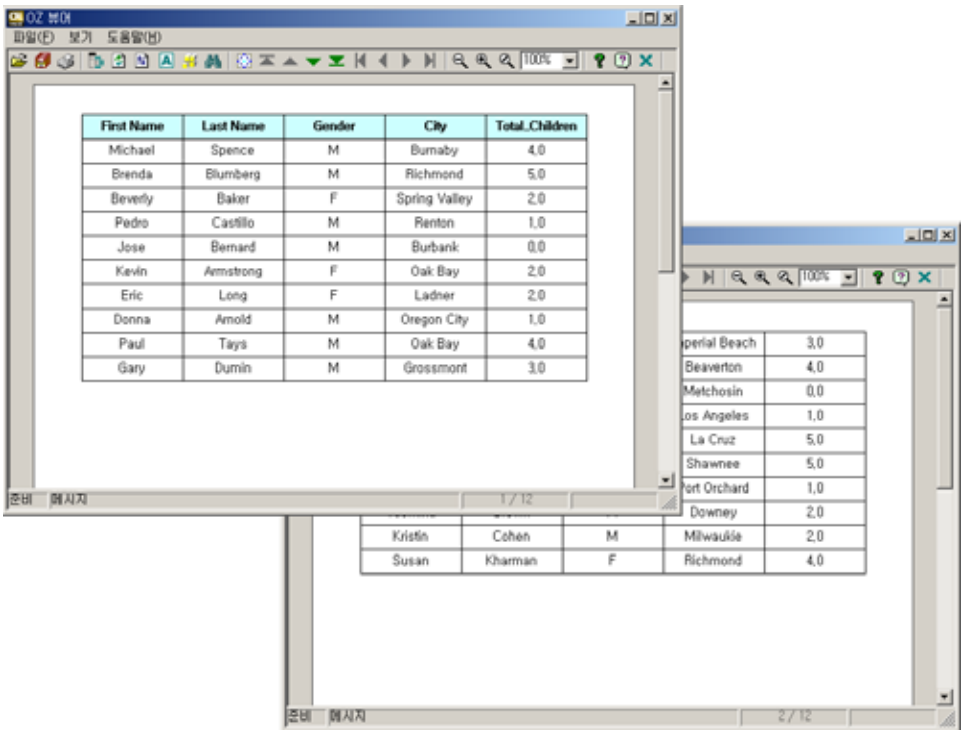
데이터가 표시되는 라벨이 추가되어있는 데이터 밴드에 스크립트를 입력합니다.



입력된 스크립트는 다음과 같습니다.

```
global int cnt;
::cnt = ::cnt + 1;
if (::cnt == 10) { //한 페이지에 10건씩 표시할 경우
   setattr("forcenewpage", "true");
   ::cnt = 0;
}
```

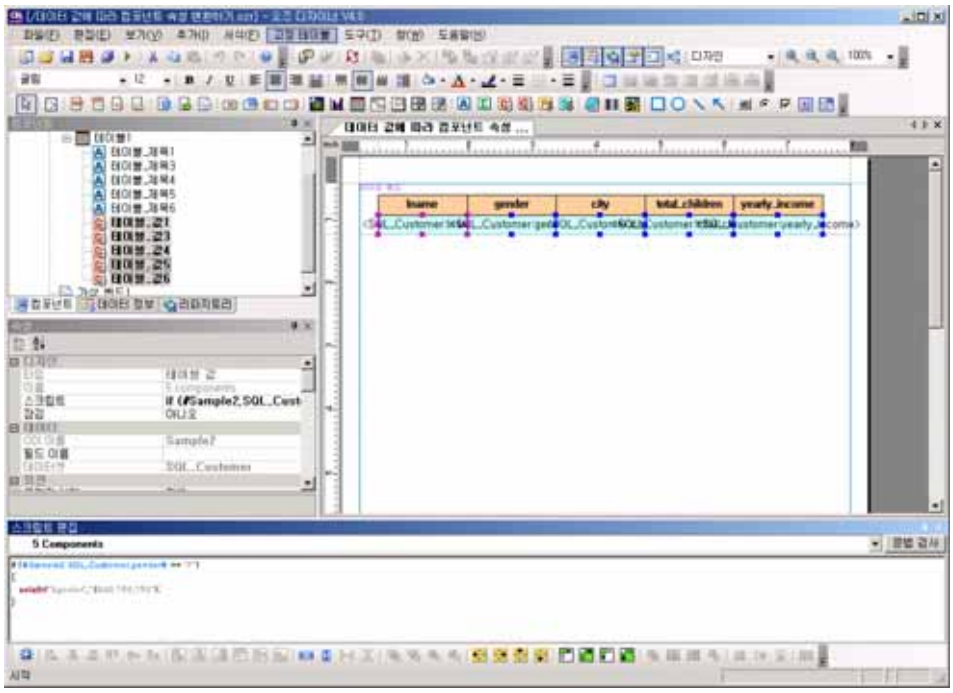
보고서를 미리보기하면 한 페이지에 10건씩 출력되는 것을 확인할 수 있습니다.




## ■ 데이터 값에 따라 컴포넌트 속성 변환하기

특정 필드의 데이터 값이 음수일 때 글자색을 빨간색으로 표시하거나 성별이 여자인 경우 배경색을 변경하는 등 데이터 값에 따라 컴포넌트의 속성을 변화시켜 좀 더 시각적인 보고서를 만들할 수 있습니다.

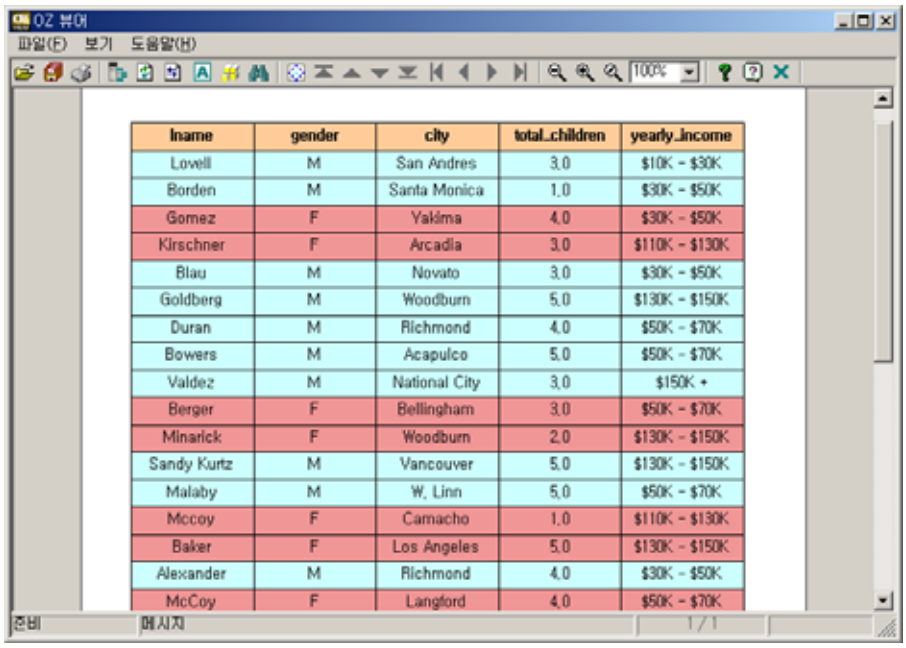
예를 들어, 테이블에 데이터를 표시할 때 성별이 여성인 경우 라벨의 배경색을 변경하고자 한다면 테이블 값 라벨을 선택한 후 속성을 변환하는 스크립트를 입력하여 구현할 수 있습니다.



성별이 여성인 경우 테이블의 행의 라벨을 모두 변경하려면 테이블 값 라벨을 모두 선택한 후 아래와 같이 스크립트를 입력합니다.

```
if (#Sample2.SQL_Customer.gender# == "F")
{
   setattr("bgcolor","$240,150,150");
}
```

보고서를 미리보기하면 성별이 여성인 경우 행의 모든 라벨의 배경색이 변경되는 것을 확인할 수 있습니다.

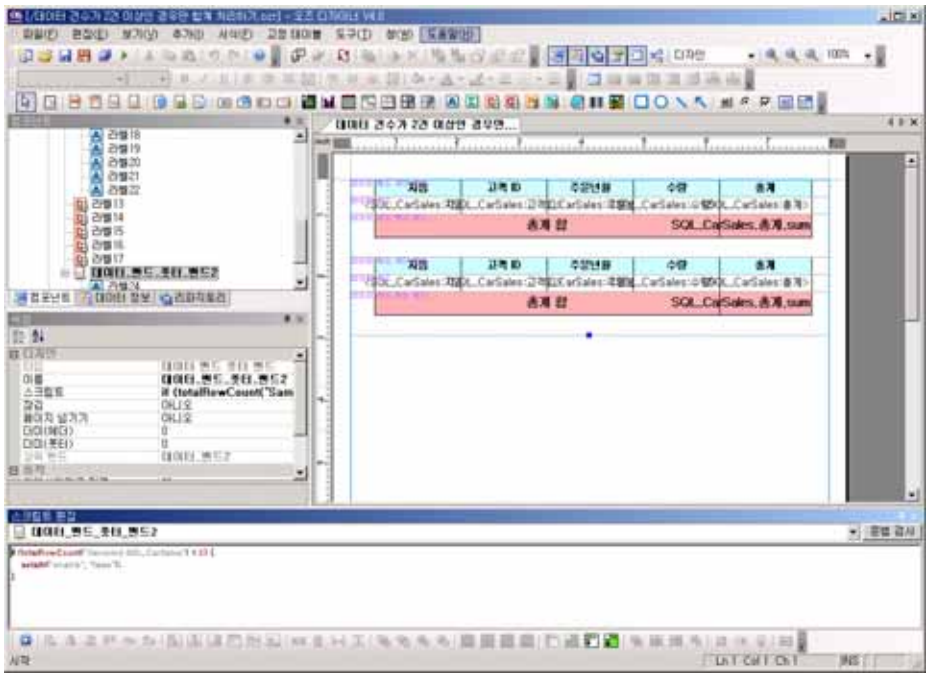



| lname | gender | city | total_children | yearly_income |
|---|---|---|---|---|
| Lovell | M | San Andres | 3.0 | $10K - $30K |
| Borden | M | Santa Monica | 1.0 | $30K - $50K |
| Gomez | F | Yakima | 4.0 | $30K - $50K |
| Kirschner | F | Arcadia | 3.0 | $110K - $130K |
| Blau | M | Novato | 3.0 | $30K - $50K |
| Goldberg | M | Woodburn | 5.0 | $130K - $150K |
| Duran | M | Richmond | 4.0 | $50K - $70K |
| Bowers | M | Acapulco | 5.0 | $50K - $70K |
| Valdez | M | National City | 3.0 | $150K + |
| Berger | F | Bellingham | 3.0 | $50K - $70K |
| Minarick | F | Woodburn | 2.0 | $130K - $150K |
| Sandy Kurtz | M | Vancouver | 5.0 | $130K - $150K |
| Malaby | M | W. Linn | 5.0 | $50K - $70K |
| Mccoy | F | Camacho | 1.0 | $110K - $130K |
| Baker | F | Los Angeles | 5.0 | $130K - $150K |
| Alexander | M | Richmond | 4.0 | $30K - $50K |
| McCoy | F | Langford | 4.0 | $50K - $70K |

■ **데이터 건수가 2건 이상인 경우만 합계 처리하기**

데이터에 대한 합계를 데이터 밴드의 풋터 밴드에서 수식 라벨 등을 이용하여 합계를 출력할 때 데이터 밴드에 설정된 데이터셋의 레코드 수를 확인하여 합계를 처리하고 있는 데이터 밴드의 풋터 밴드를 비활성화시키는 스크립트를 이용하여 구현할 수 있습니다.

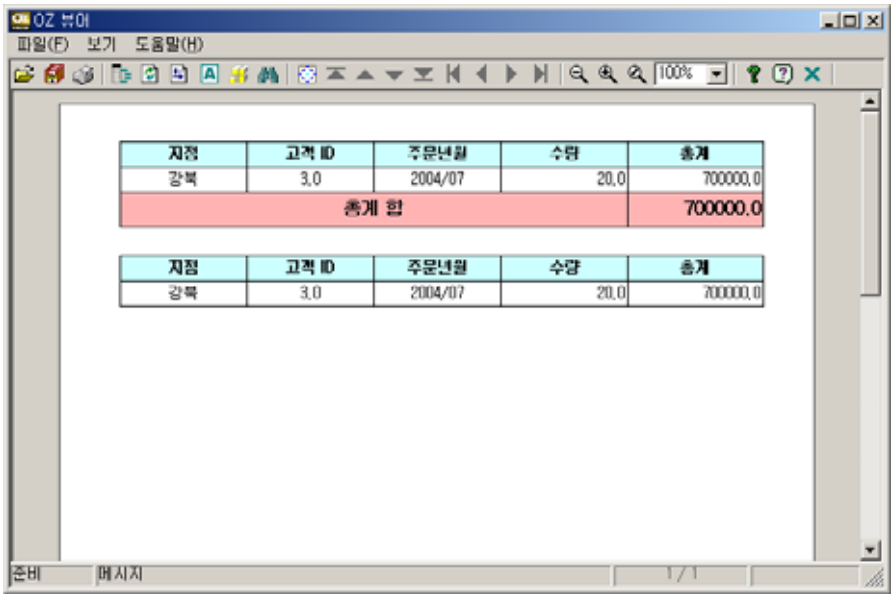
예를 들어, 헤더 밴드와 풋터 밴드를 포함하는 데이터 밴드를 두 개 추가한 후 같은 데이터셋의 데이터를 출력하게 설정한 보고서를 아래 그림과 같이 만든 후에



데이터_밴드_풋터_밴드2를 선택하여 레코드 수가 2건 보다 작으면 합계를 출력하는 데이터 풋터 밴드를 비활성화하도록 하는 스크립트를 아래와 같이 입력합니다.

```
if (totalRowCount("Sample4.SQL_CarSales") < 2) {
  setattr("enable", "false");
}
```

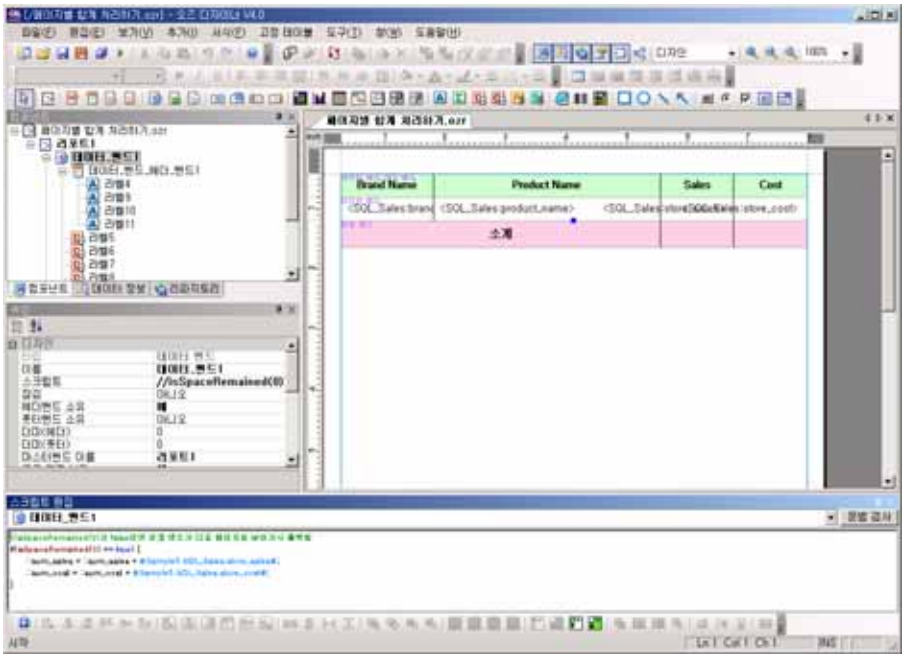
보고서를 미리보기하면 스크립트가 입력된 두 번째 데이터 밴드의 풋터 밴드는 출력되지 않는 것을 확인할 수 있습니다.




## ■ 페이지별 합계 처리하기

페이지별로 소합계를 출력하는 보고서를 만들 때에는 테일 밴드나 페이지 풋터 밴드를 이용하여 구현할 수 있습니다.

예를 들어, 판매액과 비용을 페이지별로 소계를 내어 출력하는 보고서를 만들 경우에는 헤더 밴드를 포함하는 데이터 밴드와 데이터 라벨을 추가한 후 테일 밴드를 추가하여 스크립트를 입력합니다.




리포트1을 선택하여 판매액과 비용을 설정할 전역 변수를 선언하는 스크립트를 아래와 같이 입력합니다.

```
global double sum_sales, sum_cost;
```

데이터 밴드1을 선택하여 현재 페이지에 출력된 판매액과 비용의 합계를 더하는 스크립트를 아래와 같이 입력합니다.

```
if(isSpaceRemained(0) == true) {
   ::sum_sales = ::sum_sales + #Sample5.SQL_Sales.store_sales#;
   ::sum_cost = ::sum_cost + #Sample5.SQL_Sales.store_cost#;
}
//isSpaceRemained(0)가 false이면 현재 밴드가 다음 페이지로 넘어가서 출력됨
```

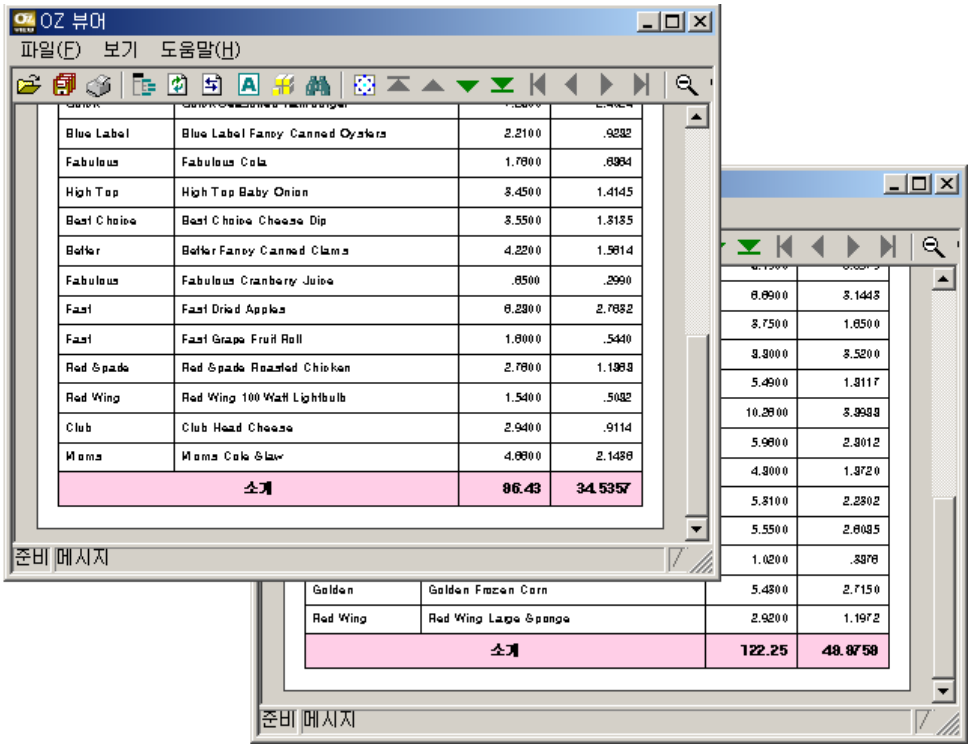
판매액의 소계를 출력하는 라벨을 선택하여 값을 출력하도록 하는 스크립트를 아래와 같이 입력합니다.

```
setattr("caption", formatNumber(::sum_sales, "###.####"));
::sum_sales = #Sample5.SQL_Sales.store_sales#;
```

비용의 소계를 출력하는 라벨을 선택하여 값을 출력하도록 하는 스크립트를 아래와 같이 입력합니다.

```
setattr("caption", formatNumber(::sum_cost, "###.####"));
::sum_cost = #Sample5.SQL_Sales.store_cost#;
```

보고서를 미리보기하면 각각 페이지별로 판매액과 비용의 소계가 출력되는 것을 확인할 수 있습니다.




## ■ 마스터 디테일 보고서에서 한 페이지당 디테일 반복 횟수 고정하기

고객별 주문정보를 출력하는 보고서를 예를 들어 보면 한 고객별로 주문정보를 출력하는데 한 페이지당 출력되는 주문 정보를 10건으로 제한합니다. 또 해당 페이지에 주문 정보가 10건이 되지 않으면 더미 밴드를 이용하여 빈칸으로 10건을 채워주고 한 고객 정보의 주문 정보가 여러 페이지로 넘어갈 경우 각 페이지마다 고객 정보를 출력합니다.

페이지마다 고객 정보를 출력하기 위해서는 주문 정보를 출력하는 밴드의 속성에서 고객 정보를 출력하는 밴드 즉 마스터 밴드를 고정하는 속성인 '마스터 고정'을 '예'로 설정합니다.

각 밴드별 다음과 같은 스크립트를 적용합니다.

```
/* 마스터 밴드(고객정보).... */
   global int i;     //디테일 밴드의 반복 횟수
   ::i = 0;           //새 마스터 밴드가 바인딩 될 때마다 0으로 재설정
```

```
/* 디테일 밴드(주문정보)....  */
:: i = ::i +1;           //디테일 밴드가 반복될 때마다 1씩 더해줍니다.
if(::i%10 == 0) setattr("forcenewpage","true");
// 한 페이지에 디테일 데이터를 10개 단위로 잘라서 출력하기 위해 디테일 반복 횟수가 10으로 나누어 나머지가 0이 될 때 페이지를 넘깁니다.
```

```
/* 디테일 밴드의 더미 풋터에서.... */
int repeatnum; // 더미밴드의 반복 횟수
if(::i%10 ==0)
   setattr("visible","false");
   //디테일 밴드의 개수가 10으로 나누어 떨어질때는 더미밴드가 나타나지 않도록 합니다.
}else {
  repeatnum = 10 -(::i%10);
  setattr("repeatnum",repeatnum);
  //디테일 밴드의 개수가 10으로 나누어  떨어지지 않을때 나머지 부분을 채워줄 더미밴드의 반복 횟수를 설정합니다.
}
setattr("forcenewpage","true");
//더미 풋터에 이 스크립트를 추가하거나 속성 창에서 [페이지 넘기기] 속성을 '예'로 선택하면 더미 풋터가 출력되고 페이지가 넘어갑니다.
```

### ■ 데이터 값에 따라 서식 바꾸기

데이터 값에 따라 보고서 서식을 바꾸기 위해서는 여러개의 더미 밴드를 사용하여 서식을 종류별로 그린 다음 더미 밴드의 스크립트에서 반복 횟수 혹은 Enable/Disable 속성을 설정합니다. 일단 여러 서식을 담고 있는 더미 밴드를 생성하고 반복 횟수를 1로 설정합니다. 스크립트에서 특정 조건이 만족할 경우는 반복 횟수를 적절한 값으로 변경하며 그렇지 않은 경우는 밴드를 비활성화시킵니다.

각각의 더미 밴드에 다음과 같은 스크립트를 추가합니다.

```
if (#ODI_name.DS.Value# == 1)
setattr("repeatnum", (string)::repeatnum);
else
setattr("enable", "false");
```

### ■ 보고서 링크하기

하나의 보고서에서 다른 보고서를 호출하는 보고서 링크 기능은 스크립트를 이용하여 구현됩니다. 다음은 고객이름을 출력하는 데이터 라벨 컴포넌트를 더블 클릭했을 때 해당 고객에 해당하는 "상세고객정보" 보고서를 호출하는 스크립트입니다. "상세고객정보" 보고서는 외부 패러미터로 고객아이디(CustID)를 받아서 해당 고객에 대한 상세 정보를 출력합니다. 현재 보고서의 고객 이름 컴포넌트에 다음의 스크립트를 작성합니다.

- 예) 오즈 서버가 데몬 형태일 경우

```
setLinkServer("127.0.0.1", 8003, "상세고객정보.ozr");
setLinkOption("displayname", #CustomerName# + " 상세정보");
setLinkOption("odi.odinames", "상세고객정보");
setLinkOption("odi.상세고객정보.pcount", "1");
setLinkOption("odi.상세고객정보.args1", "CustID="+(string)#CustomerID#);
```

- 예) 오즈 서버가 서블릿 형태일 경우

```
setLinkServlet("http://oz.forcs.com:8001/app/oz/server", "상세고객정보.ozr");
setLinkOption("displayname", #CustomerName# + " 상세정보");
setLinkOption("odi.odinames", "상세고객정보");
setLinkOption("odi.상세고객정보.pcount", "1");
setLinkOption("odi.상세고객정보.args1", "CustID="+(string)#CustomerID#);
```

- 예) 오즈 서버가 데몬 형태이고, 패러미터가 있을 경우

```
setLinkServer("211.116.251.115", 8003, "상세고객정보.ozr");
setLinkOption("displayname", #CustomerName# + " 상세정보");
setLinkOption("odi.odinames", "상세고객정보");
setLinkOption("odi.상세고객정보.pcount", "2");
setLinkOption("odi.상세고객정보.args1", "CustID="+(string)#CustomerID#);
setLinkOption("odi.상세고객정보.args2", "CustName="+#CustomerName#);
```

- 예) HyperLink를 하는 경우

```
setattr("caption", "http://www.forcs.com/");
setLinkURL("http://www.forcs.com", "_blank");
```

## ■ 이미지 그림 바꾸기

각 페이지마다 이미지가 바뀌어야 할 경우 사용합니다. 각 페이지마다 바뀌어야 하므로 페이지 헤더 밴드를 이용합니다.

페이지 헤더 밴드의 스크립트에 다음과 같이 추가합니다.

```
global int i;
::i = ::i + 1;
```

이미지 컴포넌트의 스크립트에 다음과 같이 추가합니다.

```
if(::i == 1){
setattr("caption","file:/c:/ozimage/1.jpg");
}
else if(::i == 2){
setattr("caption","file:/c:/ozimage/2.jpg");
}
else if(::i == 3){
setattr("caption","file://c:/ozimage/3.jpg");
}
else if(::i == 4){
setattr("caption","file://c:/ozimage/4.jpg");
}
```

### ■ 날짜 표기하기

수식 필드에서 현재부터 10일 단위로 증가시킵니다. 수식 필드의 type을 "Timestamp"로 설정합니다.

수식 필드 컴포넌트의 스크립트에 다음과 같이 추가합니다.

```
Date date;
date = dateAdd(d,10);
setFieldData(date);
```

### ■ 이미지 라벨에서 파일 링크하기

이미지 라벨에서 i값에 따라 이미지를 뿌려주거나, 파일을 다운로드 할 수 있도록 하는 스크립트입니다.

이미지 라벨 컴포넌트의 스크립트에 다음과 같이 추가합니다.

```
int i;
i=#OZFormParam.cond#;
if (i==1){
setattr("caption","http://localhost/web.gif");
}
else if (i==2) {
setattr("drawtype","Label");
setattr("caption","JVM Download file - Press double click this band");
setLinkURL("http://localhost/msjavx86_3802.exe","_self");
}
```

### ■ 데이터 밴드에서 특정 수만큼만 데이터 표현하기

샘플 데이터 등을 위해서 데이터베이스에 있는 모든 데이터가 아니라 특정 수의 데이터만큼만 가져오고 싶을 때 사용하는 스크립트입니다.

데이터 밴드 컴포넌트의 스크립트에 다음과 같이 추가합니다.

```
int i,t;
i=#tODI.DS.evalue#;
t = t+1;
if(t >i) setattr("enable","false");
```

### ■ 시스템 날짜 (현재 날짜) 표현하기

현재 사용 날짜를 사용자 정의에 맞게 표현하고 싶을 때 사용하는 스크립트입니다.

라벨 컴포넌트의 스크립트에 다음과 같이 추가합니다.

```
Date dd;
string str;
str = formatDate(dd,"yyyy-MM-dd");
setattr("caption",str);
```

## ■ DB안의 String 타입 날짜 데이터를 Date 타입으로 표현하기

특정 필드 값이 String 형식으로 날짜를 표현하고 있을 때 사용 날짜를 사용자 정의에 맞게 표현하고 싶을 때 사용하는 스크립트입니다.

데이터 라벨 컴포넌트의 스크립트에 다음과 같이 추가합니다.

```
Date dd;
string str;
dd = parseDate(getattr("caption"),"yyyyMMdd");
//스트링 데이터를 Date 타입으로 전환

str = formatDate(dd,"yyyy-MM-dd");
//Date 타입으로 전환된 데이터를 라벨에 입력하기 위하여 스트링으로 전환

setattr("caption",str);
```

## ■ 차트 옵션 동적으로 바꾸기

차트 스크립트를 이용한 동적 옵션 설정이 가능합니다.

차트 스크립트를 이용하여 Y축 옵션을 동적으로 설정할 수 있습니다.

- 주 눈금수 설정

```
setattr("Chart_unitofticks_y1" ,"5");
setattr("Chart_unitofticks_y2","20");
```

- 눈금단위 설정 (주 눈금수와 함께 설정될 경우 주 눈금수가 우선)

```
setattr("Chart_#ofticks_y1","100");
setattr("Chart_#ofticks_y2","100");
```

- 막대원점 설정

```
setattr("Chart_origin_y1","Max");
setattr("Chart_origin_y2","Min");
setattr("Chart_origin_y1","100");
```

- 최소 값 설정

```
setattr("Chart_min_y1","50");
setattr("Chart_min_y2","Auto");
```

- 최대 값 설정

```
setattr("Chart_max_y1","60");
setattr("Chart_max_y2","Auto");
```

- 최대 값 올림 설정 (value는 10진 단위입니다.)

```
setattr("Chart_roundup_y1","2"); → 10 단위에서 올림
setattr("Chart_roundup_y2","3"); → 100 단위에서 올림
```

- 최소 값 내림 설정

```
setattr("Chart_roundoff_y1","1"); → 1 단위에서 버림
setattr("Chart_roundoff_y2","2"); → 10 단위에서 버림
```

## ■ 차트에서 보고서 링크하기

디자이너의 차트 속성 창을 보면 '스크립트' 바로 아래에 '오즈 스크립트(링크)' 라는 항목이 하나 더 추가되어 있습니다. 여기에 설정된 스크립트는 차트(바, 라인, 점...)를 클릭할 때 마다 실행되며 일반 스크립트와 마찬가지로 링크 설정도 가능합니다.

다음과 같은 속성을 이용하여 해당 차트의 마우스가 위치한 곳의 정보를 가져올 수 있습니다.

| | |
|---|---|
| Chart_tooltip_caption | tooltip에 표시되는 캡션 |
| Chart_tooltip_contents | 컴포넌트의 Y축 값 |
| Chart_tooltip_rowpivot | X축 항목 값 |
| Chart_tooltip_colpivot | 계열 값 |

※ 주의사항 : 위 속성은 오즈 스크립트(링크)에서만 사용 가능하며, chart_tooltip_caption 을 제외하고는 getattr 함수에서만 사용할 수 있습니다.

- ex 1) 드릴다운 보고서

```
setLinkServer("211.123.456.789",8008,"sub_chart.xml");
setLinkParam("sub_chart_yvalue", getattr("Chart_tooltip_contents"));
setLinkParam("sub_chart_xvalue", getattr("Chart_tooltip_rowpivot"));
setLinkParam("sub_chart_lvalue", getattr("Chart_tooltip_colpivot"));
```

- ex 2) 차트 툴팁 설정

```
setattr("chart_tooltip_caption",                                    "Y값
="+getattr("chart_tooltip_contents")+"\n"                           +"x값
="getattr("chart_tooltip_rowpivot")+"\n"+"계열
="+getattr("chart_tooltip_colpivot"));
```

## ■ 간트 차트 옵션 동적으로 바꾸기

간트 차트 관련 스크립트를 이용한 동적 옵션 설정이 가능합니다.

- 시작일, 종료일

```
//시작일 설정
  setattr("Gantt_startdate","Parameter");
//종료일 설정
  setattr("Gantt_enddate","Parameter");
```

- 막대 형식

```
setattr("Gantt_taskbartype","separated");
```

- 시간 유형

```
setattr("Gantt_timescale","Week");
```

- 주표시 유형

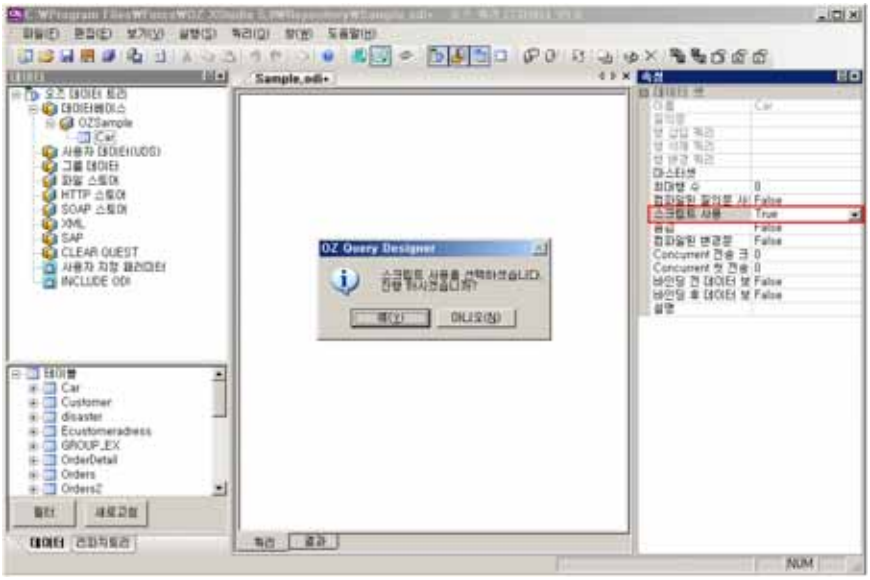
```
setattr("Gantt_weektype","Sun-Mon");
```

# 오즈 쿼리 디자이너

## 스크립트 쿼리 사용하기

스크립트 쿼리는 매뉴얼 모드에서 쿼리 데이터셋을 작성할 경우 사용할 수 있으며, 조건에 따라 동적으로 쿼리문을 생성하기 위해 데이터셋 SQL문 설정 함수인 setQueryString("SQL String")을 사용합니다.
스크립트 쿼리는 오즈 서버에서 데이터셋을 생성할 때 파싱 및 실행되며, 운영 상의 스크립트 오류는 서버 로그에 남는 에러 로그를 분석해야 합니다.

오즈 쿼리 디자이너에서 스크립트 쿼리를 사용하기 위해서는 매뉴얼 모드에서 데이터셋을 선택한 후 속성 창에서 "스크립트 사용" 속성 값을 "true"로 설정합니다.

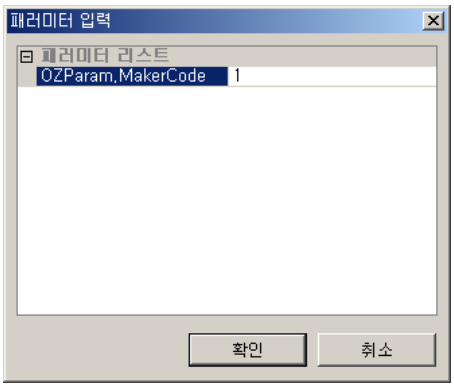



스크립트 사용 진행을 물어보는 메시지 박스에서 [예] 버튼을 선택하면 스크립트 쿼리를 사용할 수 있습니다.
스크립트 쿼리를 사용할 경우에는 오즈 스크립트를 이용하여 동적으로 SQL 문장을 만든 후 setQueryString 함수를 이용하여 작성된 SQL문을 설정합니다.
예를 들어, ODI 패러미터 MakerCode (필드 타입:INTEGER)에 따라 데이터를 가져오는 쿼리문의 경우 "스크립트 사용" 속성 값을 "예"로 설정한 후 쿼리문을 다음과 입력합니다.

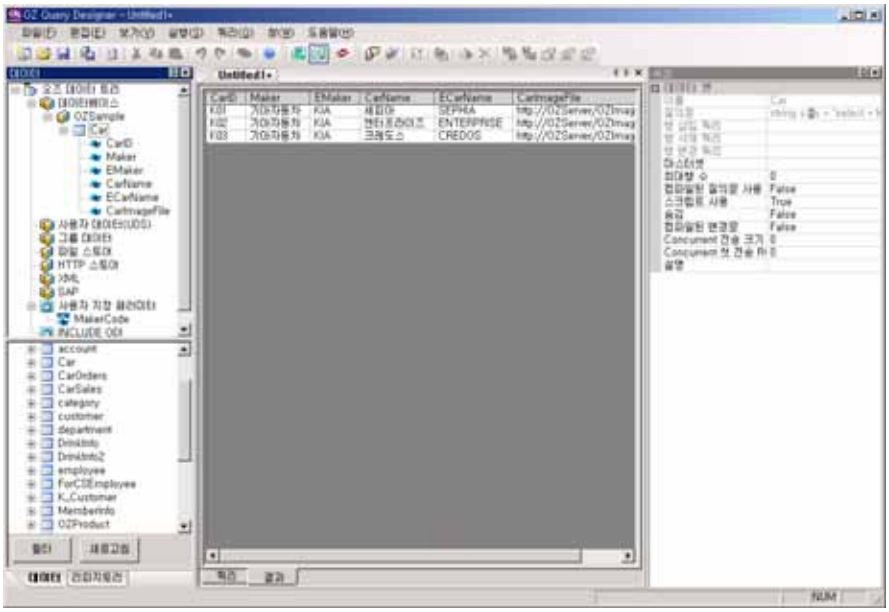
```
string s;
s = "select * from car";
```

```
if(#OZParam.MakerCode# == 1) {
   s = s + " where Maker = '기아자동차'";
} else if(#OZParam.MakerCode# == 2) {
   s = s + " where Maker = '대우자동차'";
} else {
   s = s + " where Maker = '현대자동차'";
}
setQueryString(s);
```

쿼리문 실행 아이콘( )을 클릭하면 ODI 패러미터 값을 입력하는 다이얼로그에 적절한 값을 입력합니다.



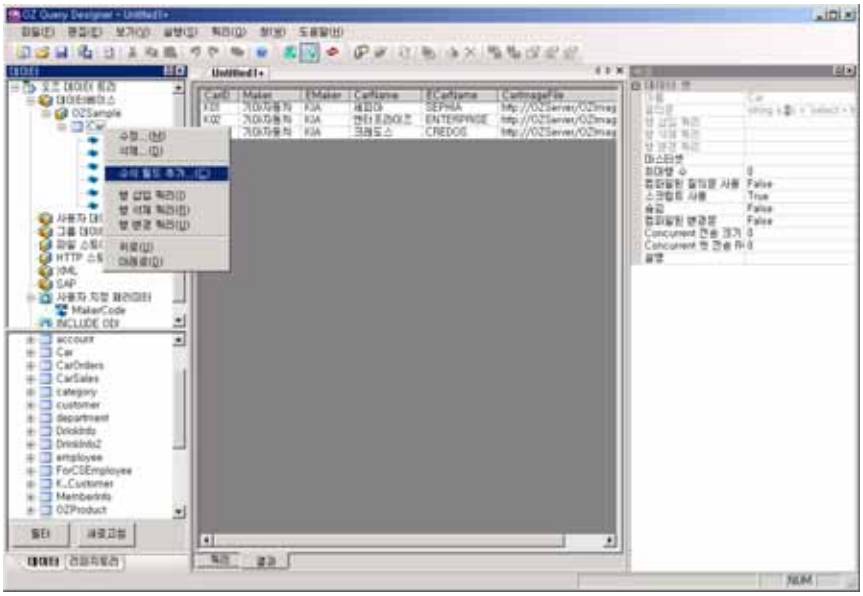

[확인] 버튼을 클릭하여 실행 결과를 확인합니다.



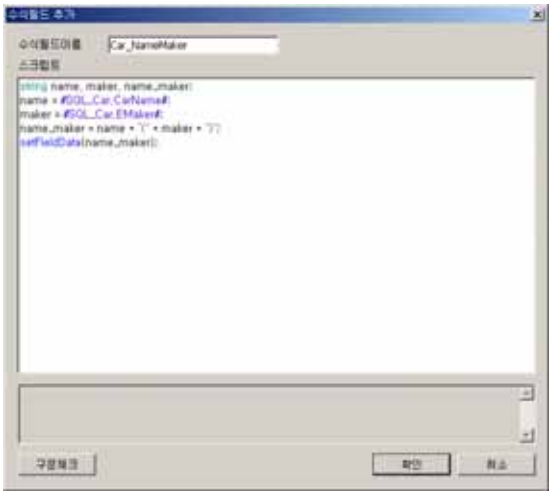
## 수식 필드 사용하기

수식 필드는 생성된 데이터셋에 사용자가 필요한 필드를 추가할 경우에 사용하며, 수식 필드 값 설정 함수인 setFieldData(<data>)를 사용합니다.

수식 필드를 추가하고자 하는 데이터셋을 선택하고, 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [수식 필드 추가] 메뉴를 클릭합니다.



수식 필드 추가 다이얼로그의 스크립트 편집창에서 오즈 스크립트를 이용하여 필드 값을 설정하는 수식을 스크립트로 작성합니다.



```
string name, maker, name_maker;
```

```
name = #SQL_Car.CarName#;
maker = #SQL_Car.EMaker#;
name_maker = name + "(" + maker + ")";
setFieldData(name_maker);
```

스크립트 작성이 완료되면 [구문체크] 버튼을 클릭하여 스크립트에 문법 에러가 없는지 확인한 후 [확인] 버튼을 클릭하면 수식 필드가 추가됩니다.

