

# HTML5 리포트 파일 정의서 버전 1.0



## 문서 이력

문서 버전	날짜	내용	작성자
1.0.0.1	2011-5-31	파일 정의서 초안 처음 배포	개발팀

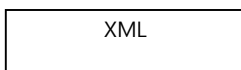
<b>HTML5 리포트 파일 정의서 버전 1.0</b> .....	1
문서 구조 .....	6
XML 정의 .....	8
글로벌 섹션(document/global).....	9
데이터셋 목록(document/global/datasets) .....	10
<b>연결 정보(document/global/datasets/connection)</b> .....	11
JDBC 연결(connection[@connection-type="jdbc"]) .....	12
속성 목록.....	12
XML 정의 .....	12
데이터셋(document/global/datasets/connection/dataset).....	14
SQL 데이터셋(dataset[@dataset-type="sql"]) .....	15
속성 목록.....	15
XML 정의 .....	15
스타일 목록(document/global/styles).....	18
기본 스타일(document/global/styles/default-style) .....	19
기본 스타일 목록 .....	19
XML 정의 .....	20
참고 사항.....	21
스타일(document/global/styles/style) .....	22
보임(style[@family="visible"]) .....	23
속성 목록.....	23
XML 정의 .....	23
참고 사항.....	23
라인(style[@family="line"]) .....	24
속성 목록.....	24
XML 정의 .....	25
참고 사항.....	25
테두리(style[@family="border"]).....	26
속성 목록.....	26
XML 정의 .....	28
참고 사항.....	28
배경(style[@family="background"]).....	30
속성 목록.....	30
XML 정의 .....	31
참고 사항.....	31
폰트(style[@family="font"]).....	33

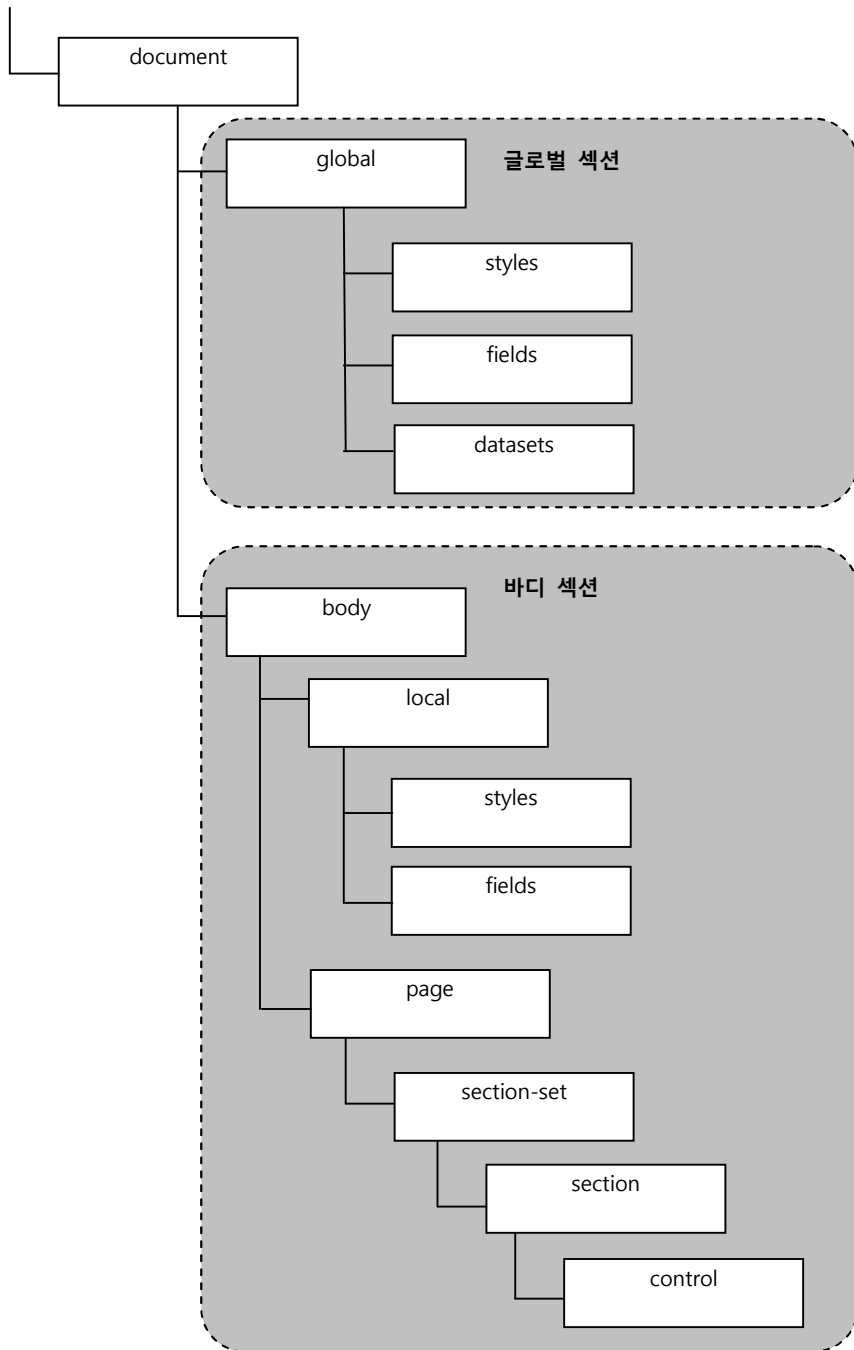
속성 목록.....	33
XML 정의.....	33
참고 사항.....	34
이미지(style[@family="image"]).....	35
속성 목록.....	35
XML 정의.....	35
참고사항.....	36
여백(style[@family="padding"]).....	38
속성 목록.....	38
XML 정의.....	38
참고 사항.....	39
페이지(style[@family="page"]).....	40
속성 목록.....	40
XML 정의.....	42
참고 사항.....	43
섹션(style[@family="section"]).....	44
속성 목록.....	44
XML 정의.....	44
참고 사항.....	45
필드 목록(document/global/fields).....	46
매개 변수 필드.....	47
속성 목록.....	47
XML 정의.....	47
시스템 필드.....	48
속성 목록.....	48
XML 정의.....	48
바디 섹션(document/body).....	49
스타일 목록(document/body/local/styles).....	50
기본 스타일(document/body/local/styles/default-style).....	51
기본 스타일 목록.....	51
스타일(document/body/local/styles/style).....	52
요약(style[@family="summary"]).....	53
속성 목록.....	53
XML 정의.....	53
컨트롤과 요약 필드.....	54
요약 함수(style[@summary-function='*']).....	55

요약 범위-리포트(style[@summary-scope='report']).....	56
요약 범위-그룹(style[@summary-scope='group']).....	56
요약 범위-섹션에의존(style[@summary-scope='dependon-section']) .	57
조건(style[@family="condition"]) .....	59
속성 목록.....	59
XML 정의.....	59
참고 사항.....	59
그룹핑(style[@family="grouping"]) .....	60
속성 목록.....	60
XML 정의.....	60
참고 사항.....	61
그룹 소트-안함(style[@group-sort='none']) .....	62
그룹 소트-오름차순(style[@group-sort='ascend']).....	62
그룹 소트-내림차순(style[@group-sort='descend']).....	62
필드 목록(document/body/local/fields) .....	63
데이터 베이스 필드.....	64
속성 목록.....	64
XML 정의.....	64
공식 필드.....	65
속성 목록.....	65
XML 정의.....	65
페이지.....	66
페이지 여백과 컨트롤의 좌표.....	67
섹션.....	69
섹션 묶음.....	71
보고서 머리글(section[@section-type="reportheader"]).....	73
속성 목록.....	73
XML 정의.....	73
페이지 머리글(section[@section-type="pageheader"]).....	74
속성 목록.....	74
XML 정의.....	74
본문(section[@section-type="detail"]).....	75
속성 목록.....	75
XML 정의.....	75
보고서 바닥글(section[@section-type="reportfooter"]).....	77
속성 목록.....	77

XML 정의 .....	77
페이지 바닥글(section[@section-type="pagefooter"]).....	78
속성 목록 .....	78
XML 정의 .....	78
그룹 머리글(section[@section-type="groupheader"]).....	80
속성 목록 .....	80
XML 정의 .....	80
그룹 바닥글(section[@section-type="groupfooter"]) .....	82
속성 목록 .....	82
XML 정의 .....	82
컨트롤 .....	84
라인(control[@type="line"]) .....	85
속성 목록 .....	85
XML 정의 .....	85
도형(control[@type="shape"]).....	87
속성 목록 .....	87
XML 정의 .....	87
글상자(control[@type="textbox"]).....	89
속성 목록 .....	89
XML 정의 .....	90
콘텐츠 타입(control[@content-type]) .....	90
보임 스타일 .....	92
요약 스타일 .....	93
조건 스타일 .....	93
테이블(control[@type="table"]).....	95
속성 목록 .....	95
XML 정의 .....	96
테이블 셀.....	98
속성 목록 .....	98
XML 정의 .....	98
콘텐츠 타입(control[@content-type]) .....	99

## 문서 구조



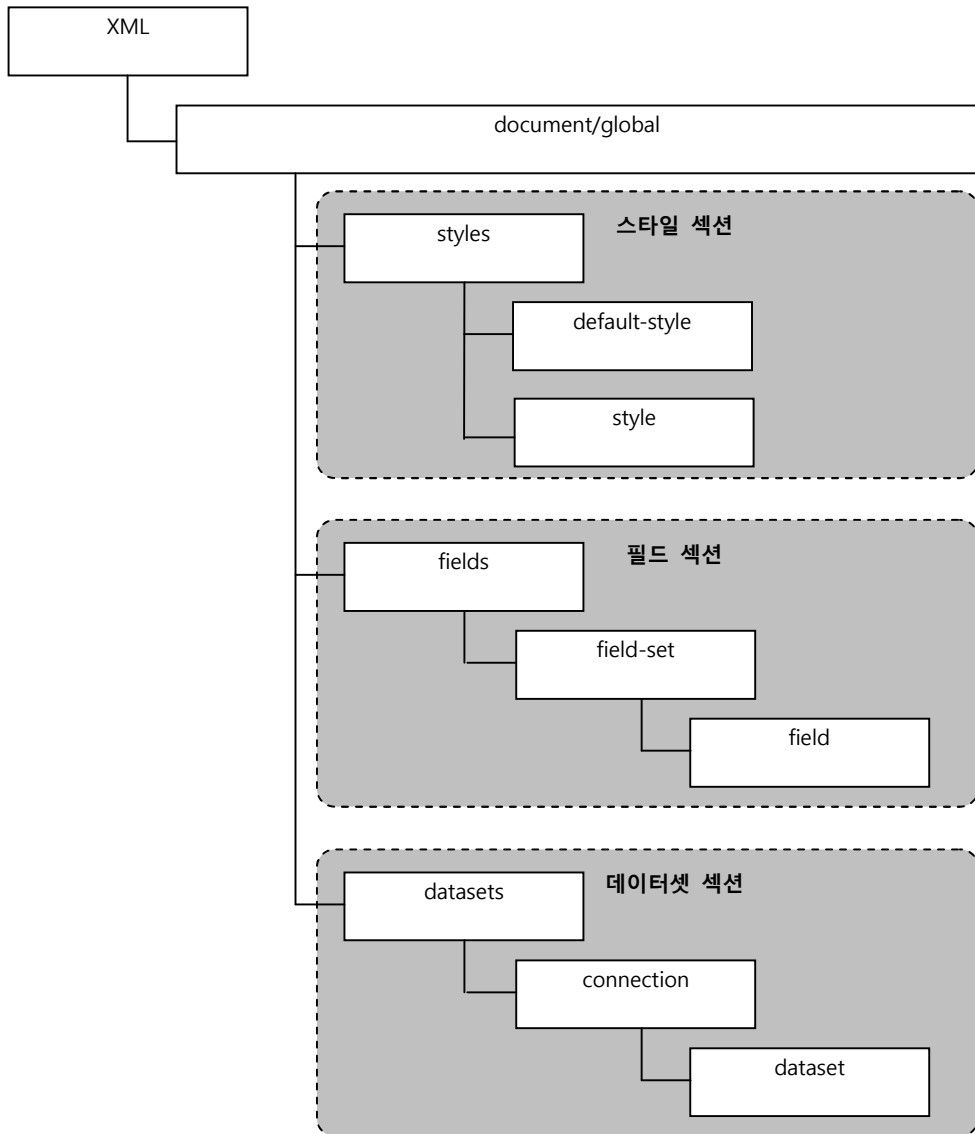


## XML 정의

```
<?xml version="1.0" encoding="utf-8" ?>
<document type="" version="1.0">
<global>
  <fields>
    <field-set field-type="parameter">
      <field name="id" />
    </field-set>
  </fields>
  <datasets>
    <connection connection-type="jdbc" name="connect1"
      jdbc-classname="oracle.jdbc.driver.OracleDriver "
      jdbc-connection="jdbc:oracle:thin:@124.137.28.174:1521:XE "
      jdbc-connection-username="username "
      jdbc-connection-password="pwd">
      <dataset dataset-type="sql" name="dataset1">
        <![CDATA[ select * from Customers  ]]></dataset>
      </connection>
    </datasets>
  <styles>
    <default-style family="background" background-color="rgb(255,255,255) "
      background-style="normal" background-fill-style="none"
      background-fill-color="rgb(0,0,0) " />
    <style family="background" name="배경 1"
      background-color="rgb(255,255,255) " />
  </styles>
</global>
<body>
  <local>
    <fields>
      <field-set field-type="database">
        <field name="name" />
        <field name="department" />
      </field-set>
    </fields>
    <styles>
      <default-style family="grouping"
        grouping-field="database.name" grouping-sort="none" />
      <style family="grouping" name="그룹핑 1"
        grouping-field="database.department" />
    </styles>
  </local>
  <page>
    <section-set section-type="detail">
      <section name="detail1">
        <control control-type="shape" background-style="배경 1" />
      </section>
    </section-set>
  </page>
</body>
</document>
```



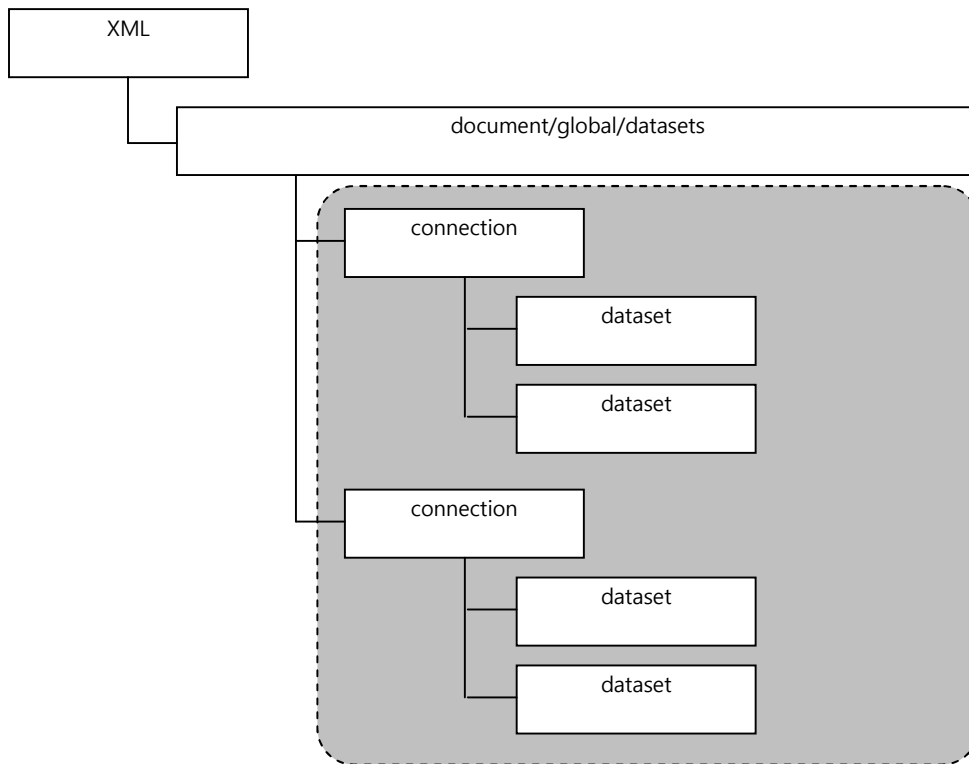
## 글로벌 섹션(document/global)



## 데이터셋 목록(document/global/datasets)

데이터셋은 일반적으로 보고서에서 사용하는 데이터를 일컫는다. 데이터 베이스 없이 출력 하는 보고서도 있지만 대다수의 보고서는 데이터 베이스를 기반으로 한다. 데이터 베이스가 될 수 있는 소스는 매우 다양하다. 데이터 베이스의 대명사인 Oracle을 비롯한 전문 DBMS 제품부터 XML이나 CSV 형태 등의 파일도 데이터 소스가 될 수 있다. 하지만, 본 리포팅 툴에서는 JDBC 기반의 데이터 베이스 연결만을 지원한다. 지속적인 업그레이드를 통해 다양한 데이터 소스를 지원할 예정이다.

보고서에서 사용하는 데이터셋은 document/global/datasets 노드 하위에 정의한다.



위 그림이 보여주듯이 보고서는 여러 개의 연결 정보를 포함할 수 있다. 각각의 연결 정보는 여러 개의 데이터셋을 포함할 수 있다. 구조적으로 여러 개의 데이터셋이 포함되지만 하나의 데이터셋만이 보고서에서 사용된다. 데이터셋 하나만 정의하면 되는데 굳이 복잡하게 여러 개의 데이터셋을 포함하는 구조로 설계한 이유는 서브 리포트<sup>1</sup> 형태의 컨트롤을 지원하기 위함이다. 현재 버전에서는 지원하지 않지만 향후 업그레이드 하면서 서브 리포트 형태의 컨트롤을 지원할 예정이다. 서브 리포트 컨트롤을 지원하면 보고서에서 여러 개의 데이터셋을 사용할 수 있다.

<sup>1</sup> 서브 리포트(Subreport) 는 보고서 안에 또 다른 보고서를 배치하여 복잡하고 여러 SQL 필요한 보고서를 만들 때 유용하게 사용할 수 있는 컨트롤의 일종이다.

## 연결 정보(document/global/datasets/connection)

연결 정보는 데이터셋을 생성하는 데 필요한 정보를 포함한다. 연결 정보의 타입에 따라서 연결 정보의 속성들이 서로 다르다. 현재는 JDBC 타입의 연결 정보만을 제공하며 앞으로 새로운 타입의 연결 정보가 추가 될 수 있다.

## JDBC 연결(connection[@connection-type="jdbc"])

JDBC는 자바 애플리케이션에서 데이터 베이스를 조작하기 위한 API 이다. 자바로 데이터 베이스를 다루는 프로그램을 개발한다면 JDBC의 사용은 필수적이다. 본 리포팅 서버는 연결 정보가 JDBC로 설정 되면 해당 노드에 있는 속성값을 사용하여 데이터 베이스에 연결 하고 보고서에 필요한 데이터를 만든다.

### 속성 목록

속성 목록			
속성	상수	버전	설명
connection-type	jdbc	1.0	연결 객체 타입을 설정한다.
name		1.0	연결 객체 이름을 설정한다.
jdbc-classname		1.0	JDBC Class 이름을 설정한다.
jdbc-connection		1.0	JDBC 연결 문자열을 설정한다.
jdbc-connection-username		1.0	JDBC 연결에 사용할 사용자명을 설정한다.
jdbc-connection-password		1.0	JDBC 연결에 사용할 사용자의 패스워드를 설정한다.

### XML 정의

```
<?xml version="1.0" encoding="utf-8" ?>
<document type="" version="1.0">
  <global>
    <datasets>
      <connection connection-type="jdbc" name="connect1"
        jdbc-classname="oracle.jdbc.driver.OracleDriver "
        jdbc-connection="jdbc:oracle:thin:@124.137.28.174:1521:XE "
        jdbc-connection-username="username "
        jdbc-connection-password="pwd">
        <dataset dataset-type="sql" name="dataset1">
          <![CDATA[ select * from Customers ]]> </dataset>
        <dataset dataset-type="sql" name="dataset2">
          <![CDATA[ select * from Customers ]]> </dataset>
        </connection>
      </datasets>
    </global>
  </body />
</document>
```

```
Connection conn = null;
try {
    Class.forName("oracle.jdbc.driver.OracleDriver");
} catch(ClassNotFoundException e) {
    e.printStackTrace();
}

try {
    conn = DriverManager.getConnection("jdbc:oracle:thin@127.0.0.1:1521:orcl",
    "scott",
    "tiger");
} catch(Exception e) {
    e.printStackTrace();
}
```

**데이터셋(document/global/datasets/connection/dataset)**

## SQL 데이터셋(dataset[@dataset-type="sql"])

### 속성 목록

속성 목록			
속성	상수	버전	설명
dataset-type	sql	1.0	데이터셋 이름을 설정합니다.
name		1.0	JDBC Class 이름을 설정한다.

### XML 정의

```
<?xml version="1.0" encoding="utf-8" ?>
<document type="" version="1.0">
  <global>
    <databases>
      <connection connection-type="jdbc" name="connect1"
        jdbc-classname="oracle.jdbc.driver.OracleDriver "
        jdbc-connection="jdbc:oracle:thin:@124.137.28.174:1521:XE "
        jdbc-connection-username="username"
        jdbc-connection-password="pwd">
        <dataset dataset-type="sql" name="dataset1">
          <![CDATA[ select * from Customers  ]]> </dataset>
          <dataset dataset-type="sql" name="dataset2">
          <![CDATA[ select * from Customers  ]]> </dataset>
        </connection>
      </databases>
    </global>
  <body />
</document>
```

```
Connection conn = null;
try {
    Class.forName("oracle.jdbc.driver.OracleDriver");
} catch(ClassNotFoundException e) {
    e.printStackTrace();
}

try {
    conn = DriverManager.getConnection("jdbc:oracle:thin@127.0.0.1:1521:orcl",
        "scott",
        "tiger");
} catch(Exception e) {
    e.printStackTrace();
}
```



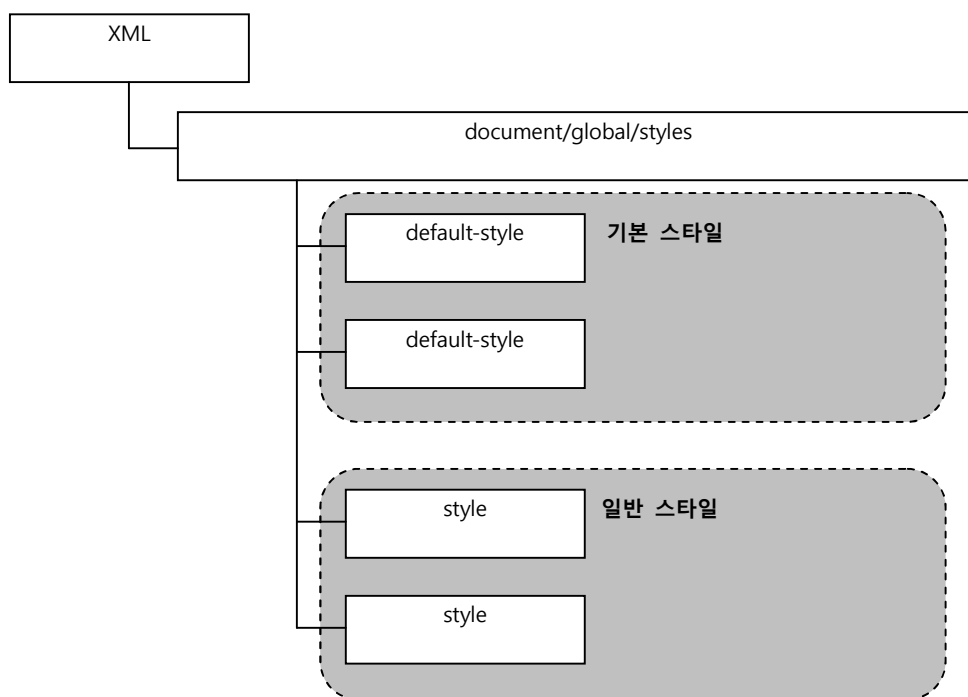
```

<?xml version="1.0" encoding="utf-8" ?>
<document type="" version="1.0">
  <global>
    <databases>
      <connection connection-type="jdbc" name="connect1"
        jdbc-classname="oracle.jdbc.driver.OracleDriver "
        jdbc-connection="jdbc:oracle:thin:@124.137.28.174:1521:XE "
        jdbc-connection-username="username"
        jdbc-connection-password="pwd">
        <dataset dataset-type="sql" name="dataset1">
        <![CDATA[ select * from Customers ]]> </dataset>
        <dataset dataset-type="sql" name="dataset2">
        <![CDATA[ select * from Customers ]]> </dataset>
      </connection>
      <connection connection-type="jdbc" name="connect2"
        jdbc-classname="com.microsoft.jdbc.sqlserver.SQLServerDriver "
        jdbc-connection="jdbc:microsoft:sqlserver://localhost:1433 "
        jdbc-connection-username="user"
        jdbc-connection-password="pwd">
        <dataset dataset-type="sql" name="dataset1">
        <![CDATA[ select * from Customers ]]></dataset>
      </connection>
    </databases>
  </global>
</body />
</document>

```

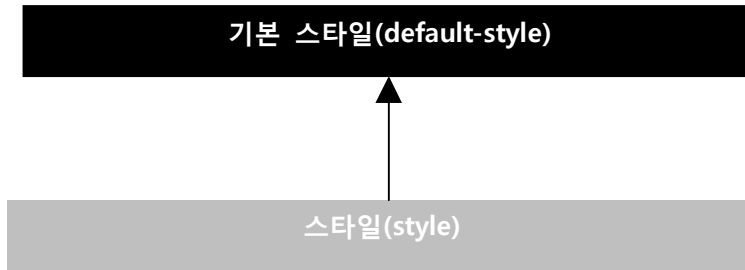
## 스타일 목록(document/global/styles)

보고서에 사용하는 모든 스타일은 document/global/styles 노드 하위에 정의한다. 대부분의 스타일은 화면을 그리는 데 필요한 정보를 포함한다. 컨트롤은 필요한 스타일을 이름으로 참조 한다. 리포트 디자이너 개발자는 스타일을 생성할 때 가독성 있으면서 이름이 겹치지 않도록 해야 한다. 하나의 스타일을 여러 컨트롤이 동시에 참조할 수 도 있다. 앞서 언급했듯이 스타일을 공유하는 방식은 메모리를 최적화할 수 있고 파일의 사이즈를 줄이는데 매우 효과적인 방법 중에 하나다.



## 기본 스타일(document/global/styles/default-style)

리포팅 툴에서 제공하는 스타일의 기본값을 미리 정의한 것을 기본 스타일이라고 한다. 스타일 종류별로 기본 스타일은 하나씩만 존재할 수 있다. 스타일의 속성 개수는 적게는 한 개에서 많게는 열 개 이상도 가질 수 있다. 기본 스타일은 의미 그대로 기본값을 설정하기 때문에 스타일의 모든 속성값을 설정해야 한다. 반면에, 스타일은 모든 속성값을 설정할 필요가 없으며 기본값과 다른 속성만 설정하면 된다. 이러한 기본 스타일이란 개념을 도입함으로써 리포트 파일의 사이즈를 줄이고 스타일의 속성값을 일괄로 변경 가능한 장점을 얻을 수 있다. 객체 지향 개념의 상속이라는 관점에서 보면 기본 스타일은 스타일의 부모가 되고 스타일은 기본 스타일의 모든 속성을 그대로 물려받는다.



### 기본 스타일 목록

스타일 목록			
스타일 종류	상수	버전	설명
visible		1.0	보임 여부를 설정
line		1.0	라인 스타일
border		1.0	테두리 스타일
background		1.0	배경 스타일
font		1.0	폰트 스타일
image		1.0	이미지 스타일
padding		1.0	여백 스타일
page		1.0	페이지 스타일
section		1.0	섹션 스타일

## XML 정의

```
<?xml version="1.0" encoding="utf-8" ?>
<document type="" version="1.0">
<global>
<styles>
  <default-style family="line"
    line-color="rgb(255,255,0)" line-type="solid" line-width="1" />
  <default-style family="border"
    border-right-color="rgb(0,0,0)" border-right-style="solid"
    border-right-width="10" border-left-color="rgb(0,0,0)"
    border-left-style="solid" border-left-width="10"
    border-top-color="rgb(0,0,0)" border-top-style="solid"
    border-top-width="10" border-bottom-color="rgb(0,0,0)"
    border-bottom-style="solid" border-bottom-width="10" />
  <default-style family="background"
    background-color="rgb(255,255,255)"
    background-style="normal"
    background-fill-style="none"
    background-fill-color="rgb(0,0,0)" />
  <default-style family="font"
    font-name="굴림" font-color="rgb(0,0,0)" font-size="10"
    font-strike="0" font-underline="0" font-italic="0" font-bold="0"
    font-wordwrap="0" font-horizontal="left" font-vertical="top" />
  <default-style family="image"
    image-type="gif" image-source="data" image-path=""
    image-data="ODlhEwA0AKIAAPr6+srKyv//9/f3zKoAAAAAAAAAAAAACH5BA
    AAAAAAIAAAATAA4AAAMuGLPc viGISaudUYDNuwfZJ3KhRxBjaaKfu
    rbSeMLaN9M2i4+iy50xnydCLBqPCQA7" image-fill="original"
    image-align-vertical="top" image-align-horizontal="left" />
  <default-style family="line"
    padding-right="10" padding-left="10" padding-top="10"
    padding-bottom="10" />
  <default-style family="page"
    page-type="a4" page-width="2100" page-height="297"
    page-direction="portrait" page-padding="10"
    page-padding-left="10" page-padding-top="10"
    page-padding-bottom="10" />
</styles>
</global>
<body />
</document>
```

## 참고 사항

기본 스타일과 스타일의 개념이 약간은 헷갈릴 수 있기 때문에 배경 스타일을 사용한 예를 알아 보겠다.

```
<?xml version="1.0" encoding="utf-8" ?>
<document>
  <global>
    <styles>
      <default-style family="background"
        background-color="rgb(255,255,255) "
        background-style="normal"
        background-fill-style="none"
        background-fill-color="rgb(0,0,0) " />

      <style family="background" name="배경 1"
        background-color="rgb(0,0,0) " />

    </styles>
  </global>
</body/>
</document>
```

배경 스타일은 4개의 속성<sup>2</sup>을 지원한다. 하지만, 위 XML에서 박스로 표시한 부분을 보면 스타일 노드에 배경색 속성만 정의하고 있다. 그렇다면, 배경 스타일 "배경1"을 참조하는 컨트롤은 배경을 그릴 때 어떤 식으로 그릴지 생각해보자.

```
<style family="background" name="background1" background-color="rgb(0,0,0) " />
```

스타일 노드에 명시하지 않은 속성은 해당 스타일의 기본 스타일 속성 값을 참조한다. 본 예에서는 스타일 노드에 배경색 속성만 정의했지만 나머지 속성은 기본 스타일 속성 값을 참조하기 때문에 아래와 같이 스타일 노드를 정의한 것과 동일하다.

```
<style family="background" name="background1"
  background-color="rgb(0,0,0) "
  background-style="normal"
  background-fill-style="none"
  background-fill-color="rgb(0,0,0) " />
```

<sup>2</sup> 배경 스타일은 특정 영역을 페인트하는 속성으로 background-color, background-style, background-fill-style, background-fill-color 속성을 지원한다.

스타일(document/global/styles/style)

## 보임(style[@family="visible"])

보임 스타일은 컨트롤의 보임 여부를 설정한다. 보임 스타일을 지원하는 컨트롤은 해당 스타일을 제어하여 컨트롤의 보임 여부를 설정할 수 있다. 리포팅 툴에서 제공하는 대부분의 컨트롤은 보임 속성을 지원한다.

### 속성 목록

속성 목록			
속성	상수	버전	설명
visible-viewer		1.0	보임 여부를 설정

### XML 정의






```
<?xml version="1.0" encoding="utf-8" ?>
<document>
  <global>
    <styles>
      <style family="visible" visible-viewer="0" />
      <style family="visible" visible-viewer="1" />
    </styles>
  </global>
</document>
```

### 참고 사항

## 라인(style[@family="line"])

**line**

### 속성 목록

속성 목록				
속성	상수		버전	설명
line-color	rgb(r,b,g)		1.0	
line-style	none	0	1.0	
	sold	1	1.0	
	dash	2	1.0	
	dot	3	1.0	
	dashdot	4	1.0	
	dashdotdot	5	1.0	
line-width			1.0	



## XML 정의

```
<?xml version="1.0" encoding="utf-8" ?>
<document>
  <global>
    <styles>
      <style family="line" name="line1" line-color="rgb(255,255,0)"
        line-type="solid" line-width="1" />
      <style family="line" name="line2" line-color="rgb(255,0,0)"
        line-type="dot" line-width="2" />
      <style family="line" name="line3" line-color="rgb(0,0,0)"
        line-type="dash" line-width="3" />
    </styles>
  </global>
</document >
```

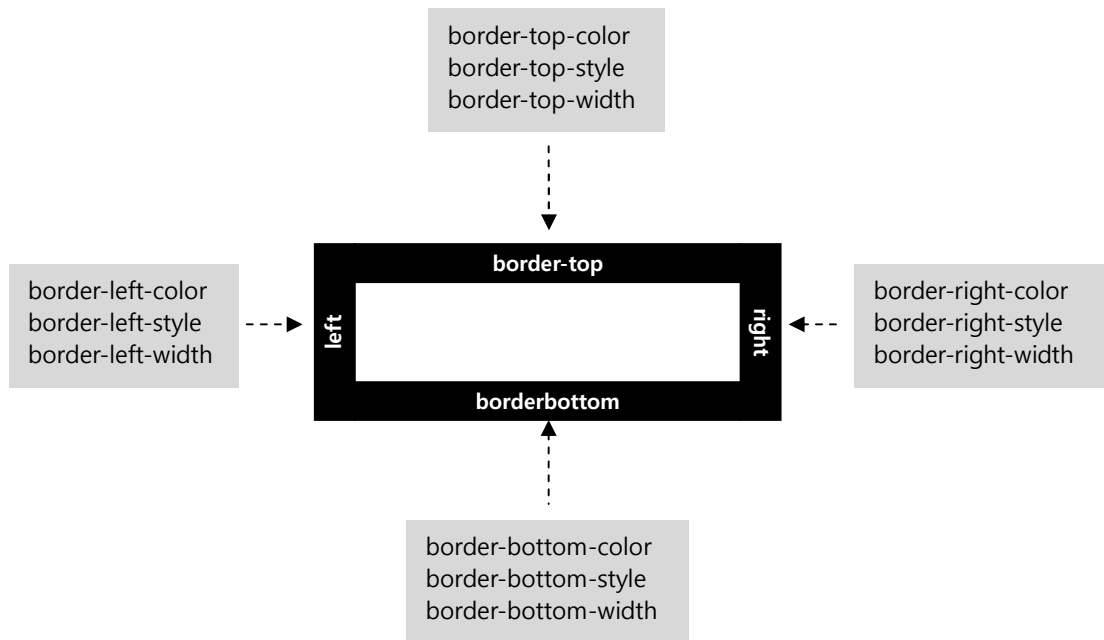
## 참고 사항

라인과 마찬가지로 테두리를  $x1, y1$  을 기준으로 중앙에 그리는 것을 원칙으로 한다.



## 테두리(style[@family="border"])

테두리 스타일은 컨트롤의 테두리 모양새를 결정한다. 테두리 스타일은 컨트롤의 왼쪽, 위쪽, 오른쪽, 아래쪽의 라인 스타일을 결정한다.



### 속성 목록

속성 목록				
속성	상수		버전	설명
border-right-color	RGB		1.0	
border-right-style	none	0	1.0	
	sold	1	1.0	
	dash	2	1.0	
	dot	3	1.0	
	dashdot	4	1.0	
	dashdotdot	5	1.0	
border-right-width	doublesolid	6	1.0	
			1.0	

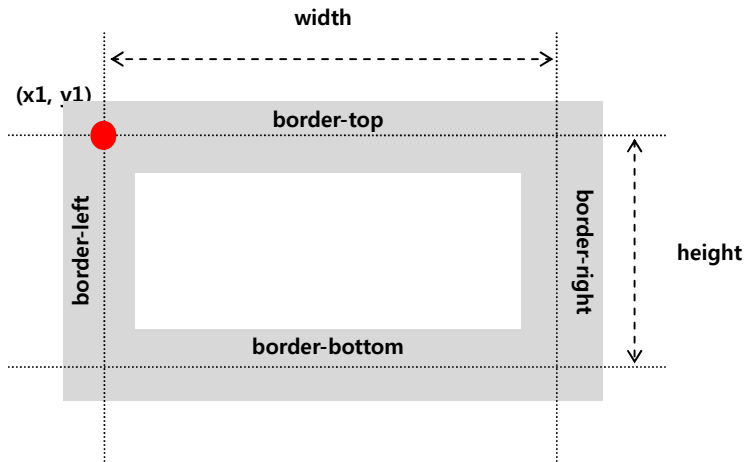
border-left-color			1.0	
border-left-style	none	0	1.0	
	solid	1	1.0	
	dash	2	1.0	
	dot	3	1.0	
	dashdot	4	1.0	
	dashdotdot	5	1.0	
	doublesolid	6	1.0	
border-left-width			1.0	
border-top-color			1.0	
border-top-style	none	0	1.0	
	solid	1	1.0	
	dash	2	1.0	
	dot	3	1.0	
	dashdot	4	1.0	
	dashdotdot	5	1.0	
	doublesolid	6	1.0	
border-top-width			1.0	
border-bottom-color			1.0	
border-bottom-style	none	0	1.0	
	solid	1	1.0	
	dash	2	1.0	
	dot	3	1.0	
	dashdot	4	1.0	
	dashdotdot	5	1.0	
	doublesolid	6	1.0	
border-bottom-width			1.0	

## XML 정의

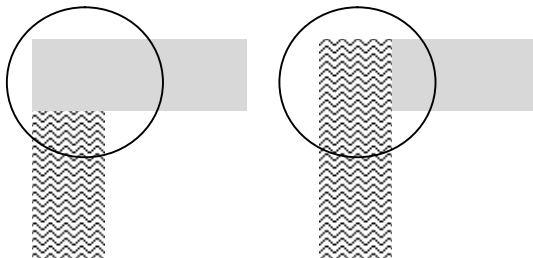
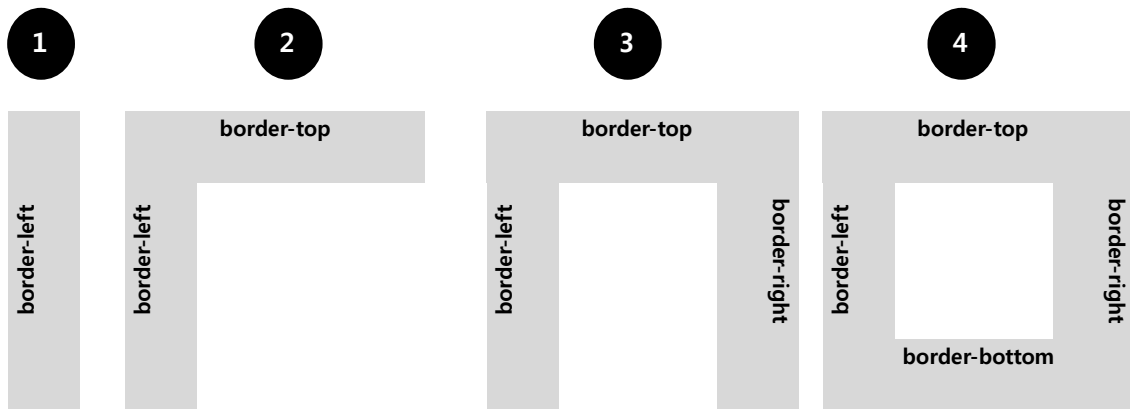
```
<?xml version="1.0" encoding="utf-8" ?>
<document>
  <global>
    <styles>
      <style family="border" name="border1"
        border-right-color="rgb(0,0,0)" border-right-style="solid"
        border-right-width="10"
        border-left-color="rgb(0,0,0)" border-left-style="solid"
        border-left-width="10"
        border-top-color="rgb(0,0,0)" border-top-style="solid"
        border-top-width="10"
        border-bottom-color="rgb(0,0,0)" border-bottom-style="solid"
        border-bottom-width="10" />
      <style family="border" name="border2"
        border-right-color="rgb(255,255,0)" border-right-style="dash"
        border-right-width="20"
        border-left-color="rgb(0,255,255)" border-left-style="dash"
        border-left-width="20"
        border-top-color="rgb(0,0,0)" border-top-style="dash"
        border-top-width="20"
        border-bottom-color="rgb(0,0,0)" border-bottom-style="dash"
        border-bottom-width="20" />
    </styles>
  </global>
</document>
```

## 참고 사항

라인과 마찬가지로 테두리를 x1, y1 을 기준으로 중앙에 그리는 것을 원칙으로 한다.

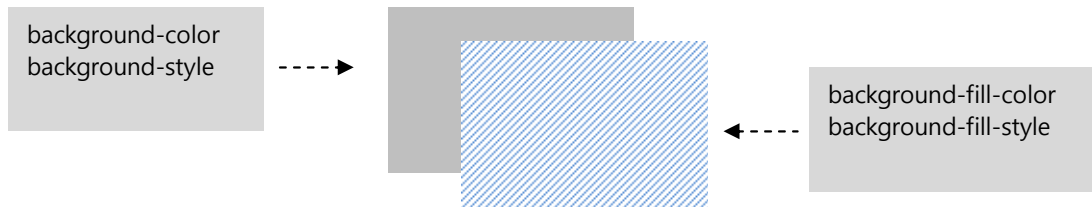


테두리 스타일은 스펙상으로 4면으로 다르게 설정할 수 있다. 제일 먼저 왼쪽 테두리를 그리고 시계 방향으로 나머지 테두리를 그리는 것을 기본 원칙으로 한다. 분 문서에서 테두리를 그리는 순서까지 정의하는 이유는 어플리케이션마다 그리는 순서를 달리하면 동일한 파일 포맷도 사용자가 사용하는 어플리케이션에 따라서 다르게 보일 수 있기 때문이다. 테두리 두께와 색이 동일하면 그리는 순서에 상관없이 동일하게 보인다. 하지만 테두리의 색상이 틀리고 두께가 크면 클수록 차이점은 명확해진다.

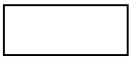
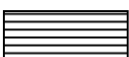



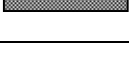


## 배경(style[@family="background"])

배경 스타일 타입은 컨트롤의 배경을 그리는데 필요한 정보를 포함한다.



### 속성 목록

속성 목록				
속성	상수		버전	설명
background-color	rgb(r,g,b)		1.0	배경 색을 지정한다.
background-style	normal	0	1.0	배경 색을 적용한다.
	transparent	1	1.0	배경 색을 적용하지 않는다.
background-fill-color	rgb(r,g,b)		1.0	채움 색을 지정한다.
background-fill-style	none	0	1.0	
	horizontalline	1	1.0	
	verticalline	2	1.0	
	crossline	3	1.0	
	frontdiagonalline	4	1.0	
	backdiagonalline	5	1.0	
	crossdiagonalline	6	1.0	
	dotdensed	7	1.0	
	dot	8	1.0	

	dotrare	9	1.0	

## XML 정의

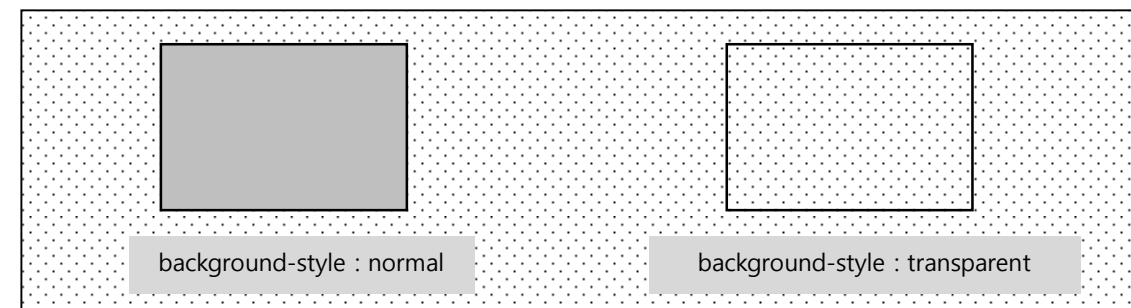
```

<?xml version="1.0" encoding="utf-8" ?>
<document>
  <global>
    <styles>
      <style family="background" name="background1"
        background-color="rgb(255,255,255)"
        background-style="normal"
        background-fill-style="none"
        background-fill-color="rgb(0,0,0)" />
      <style family="background" name="background2"
        background-color="rgb(255,0,255)"
        background-style="transparent"
        background-fill-style="none"
        background-fill-color="rgb(0,0,0)" />
    </styles>
  </global>
</document>

```

## 참고 사항

background-style 속성을 "normal" 로 설정하면 지정된 영역을 background-color에 지정한 색으로 배경을 칠한다. background-style 속성을 "transparent"로 설정하면 background-color 지정한 색은 무시한다. 지정된 영역을 그리지 않음으로써 컨트롤의 배경이 보이게 하여 투명한 효과를 내야 한다.







## 폰트(style[@family="font"])

### 속성 목록

속성 목록				
속성	상수		버전	설명
font-name	none		1.0	
font-color				
font-size				
font-strike				
font-underline				
font-italic				
font-bold				
font-wordwrap				
font-horizontal-align	left	0		
	middle	1		
	right	2		
font-vertical-align	top	0		
	center	1		
	bottom	2		

### XML 정의

```
<?xml version="1.0" encoding="utf-8" ?>
<document>
  <global>
    <styles>
      <style family="font" name="font1"
        font-name="굴림" font-color="rgb(0,0,0)" font-size="10"
        font-strike="0" font-underline="0" font-italic="0" font-bold="0"
        font-wordwrap="0" font-horizontal="left" font-vertical="top" />
      <style family="font" name="font2"
        font-name="명조" font-color="rgb(255,0,0)" font-size="7"
        font-strike="1" font-underline="1" font-italic="1" font-bold="1"
        font-wordwrap="1" font-horizontal="left" font-vertical="top" />
    </styles>
  </global>
</document>
```

## 참고 사항

## 이미지(style[@family="image"])

이미지 스타일은 컨트롤의 배경에 이미지를 그리는데 필요한 정보를 포함한다.



### 속성 목록

속성 목록				
속성	상수		버전	설명
image-type	"bmp", "jpg", "png"		1.0	이미지의 타입을 지정한다.
image-source	path	0	1.0	
	data	1	1.0	
image-path			1.0	이미지가 있는 전체 경로를 설정한다.
image-data			1.0	이미지 데이터를 설정한다.(Base64 형태)
image-fill	none	0	1.0	이미지의 채움 형태를 설정한다.
	original	1	1.0	
	stretch	2	1.0	
image-align-vertical	top	0	1.0	이미지의 세로 정렬을 설정한다.
	center	1	1.0	
	bottom		1.0	
image-align-horizontal	left		1.0	이미지의 가로 정렬을 설정한다.
	middle		1.0	
	right		1.0	

### XML 정의

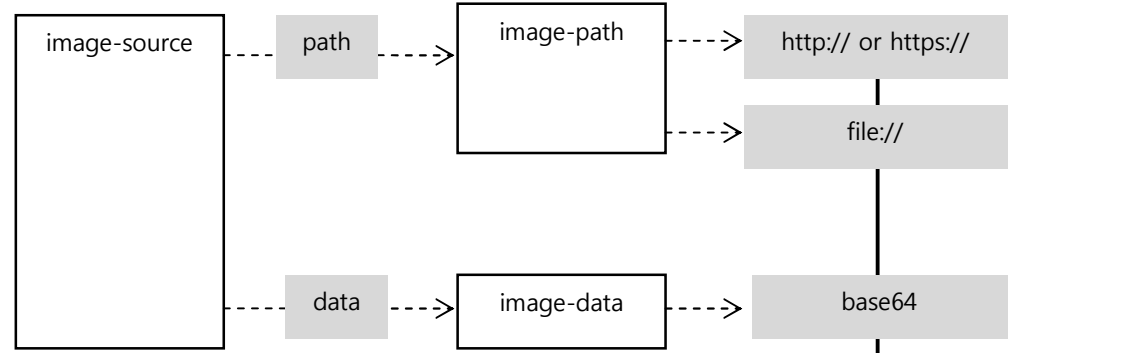
```

<?xml version="1.0" encoding="utf-8" ?>
<document>
  <global>
    <styles>
      <style family="image" name="image1"
        image-type="gif"
        image-source="data"
        image-path=""
        image-data="ODlhEwA0AKIAAPr6+s rKyv///9/f3zKoAAAAAAAAAAAAACH5BA
          AAAAAALAAAAATAA4AAAMuGLPc viGISaudUYDNuwfZJ3KhRxBjaaKfu
          rbSeMLaN9M2i4+iy50xnydCLBqPCQA7 "
        image-fill="original"
        image-align-vertical="top"
        image-align-horizontal="left" />
      <style family="image" name="image2"
        image-type="png"
        image-source="path"
        image-path="http://www.google.co.kr/logo.png "
        image-data=""
        image-fill="original"
        image-align-vertical="top"
        image-align-horizontal="left" />
    </styles>
  </global>
</document>

```

**참고사항**

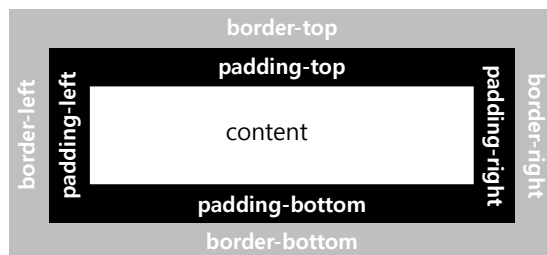
image-source 속성에서 이미지의 경로 정보를 얻는다. image-source 속성의 값이 "path" 이면 image-path 속성에서 값을 읽어 이미지를 내려 받아 생성한다. image-source 속성의 값이 "data" 이면 image-data 속성에서 값을 읽어 이미지를 생성한다. 문서의 포맷이 XML 이기 때문에 이미지의 내용을 바이너리 자체로 넣을 수 없기 때문에 베이스64(base64)로 인코딩하여 저장한다. 어플리케이션 개발자는 반대로 베이스64 로 디코딩하여 바이너리 이미지 데이터를 얻는다.





## 여백(style[@family="padding"])

여백 스타일은 컨트롤의 여백 정보를 포함한다.



### 속성 목록

속성 목록			
속성	상수	버전	설명
padding-right		1.0	오른쪽 여백을 설정한다.
padding-left		1.0	왼쪽 여백을 설정한다.
padding-top		1.0	위쪽 여백을 설정한다.
padding-bottom		1.0	아래쪽 여백을 설정한다.

여백 스타일에서 여백의 의미는 테두리와 텍스트 사이의 여백을 의미한다. 일반적으로 사각형 영역 안에 텍스트를 그릴 때 간격 없이 바로 텍스트를 그리면 답답하고 문서가 조잡해 보인다. 사각형 영역 안에서 텍스트를 그리는 시작 위치를 정할 수 있는 스타일이 바로 여백 스타일이다. 텍스트가 없는 컨트롤에서는 여백 스타일은 의미 없는 스타일이다. 어플리케이션 개발자는 텍스트를 그리는 시작 위치와 여백을 벗어나지 않도록 주의해서 프로그램을 구현해야 한다.

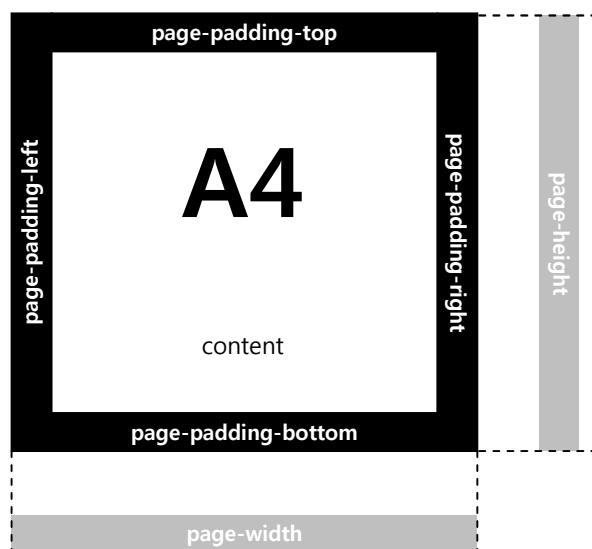
### XML 정의

```
<?xml version="1.0" encoding="utf-8" ?>
<document>
  <global>
    <styles>
      <style family="padding" name="padding1"
        padding-right="10" padding-left="10"
        padding-top="10" padding-bottom="10" />
      <style family="padding" name="padding2"
        padding-right="50" padding-left="50"
        padding-top="50" padding-bottom="50" />
    </styles>
  </global>
</document>
```

## 참고 사항

## 페이지(style[@family="page"])

페이지 스타일은 용지 종류 및 사이즈 등의 페이지 정보를 포함한다.



### 속성 목록

속성 목록				
속성	상수		버전	설명
page-type	a3transverse	0	1.0	
	a4	1	1.0	
	a5	2	1.0	
	b4	3	1.0	
	b4envelope	4	1.0	
	b5	5	1.0	
	b5envelope	6	1.0	
	b6envelope	7	1.0	
	c3envelope	8	1.0	
	c4envelope	9	1.0	



	c5envelope	10	1.0	
	c65envelope	11	1.0	
	c6envelope	12	1.0	
	csheet	13	1.0	
	dlenvelope	14	1.0	
	dsheet	15	1.0	
	esheet	16	1.0	
	executive	17	1.0	
	folio	18	1.0	
	germanlegal fanfold	19	1.0	
	germanstan dardfanfold	20	1.0	
	italyenvelop e	21	1.0	
	ledger	22	1.0	
	legal	23	1.0	
	lettersmall	24	1.0	
	monarchenv elope	25	1.0	
	papernote	26	1.0	
	papernumbe r10envelope	27	1.0	
	number11en velope	28	1.0	
	number12en velope	29	1.0	
	number14en velope	30	1.0	
	number9env elope	31	1.0	
	personalenv elope	32	1.0	
	quarto	33	1.0	
	standard10x	34	1.0	

	14			
	standard11x 17	35	1.0	
	statement	36	1.0	
	tabloid	37	1.0	
	usstandardfa nFold	38	1.0	
	userdefine	255	1.0	
page-width			1.0	
page-height			1.0	
page-direction	portrait	0	1.0	
	landscape	1	1.0	
page-padding-right			1.0	
page-padding-left			1.0	
page-padding-top			1.0	
page-padding-bottom			1.0	

## XML 정의

```

<?xml version="1.0" encoding="utf-8" ?>
<document>
  <global>
    <styles>
      <style family="page" name="pageA4"
        page-type="a4" page-width="2100"
        page-height="2970" page-direction="portrait"
        page-padding-right="10" page-padding-left="10"
        page-padding-top="10" page-padding-bottom="10" />
      <style family="page" name="pageB4"
        page-type="b4" page-width="2570"
        page-height="3640" page-direction="landscape"
        page-padding-right="10" page-padding-left="10"
        page-padding-top="10" page-padding-bottom="10" />
    </styles>
  </global>
</document>

```

## 참고 사항

## 섹션(style[@family="section"])

섹션 스타일은 섹션의 스타일을 정의한다.

### 속성 목록

속성 목록			
속성	상수	버전	설명
section-height			섹션의 높이를 설정한다.
section-new-page	none	0	
	before	1	섹션 출력 전에 페이지 넘김.
	after	2	섹션 출력 후에 페이지 넘김
	before-after	3	섹션 출력 전후에 페이지 넘김.

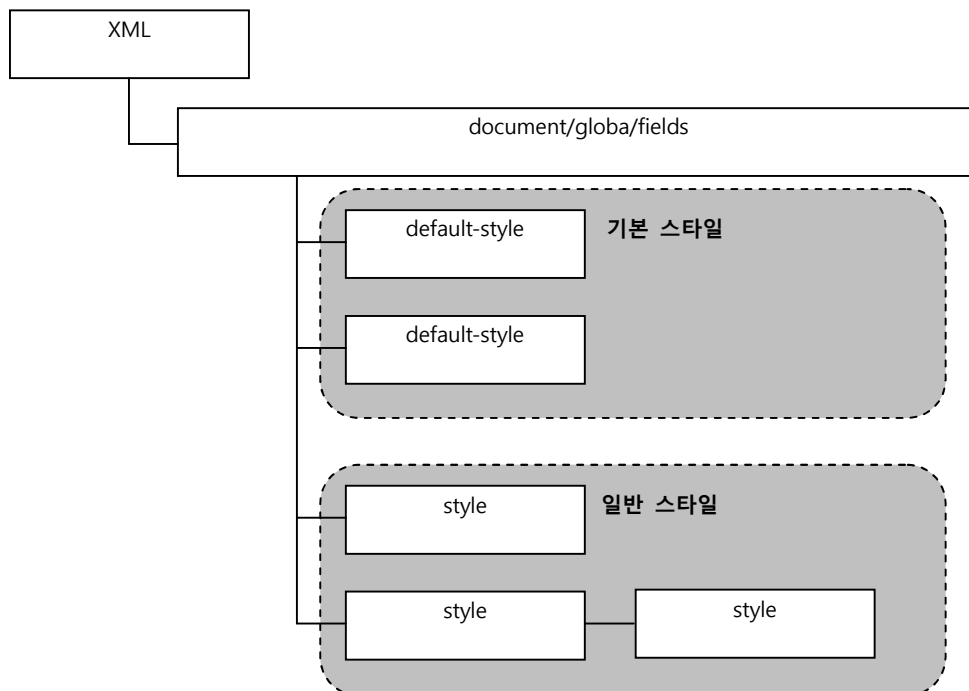
### XML 정의

```
<?xml version="1.0" encoding="utf-8" ?>
<document>
  <global>
    <styles>
      <style family="section" name="섹션 1"
        section-height="100"
        section-new-page="none" />
    </styles>
  </global>
  <body>
    <page>
      <section-set section-type="detail">
        <section name="detail1" section-style="섹션 1">
          <control control-type="shape" background-style="배경 1" />
        </section>
      </section-set>
    </page>
  </body>
</document>
```

## 참고 사항

## 필드 목록(document/global/fields)

글로벌 섹션에 있는 필드 목록은 모든 보고서에서 공통으로 참조할 수 있는 필드를 정의한다. 해당 섹션에 정의할 수 있는 필드는 매개 변수 필드 와 시스템 필드이다. 매개 변수 필드는 외부에서 값을 받아 보고서로 넘기는 역할을 하며 시스템 필드는 페이지 번호 및 개수 등의 특수한 목적으로 사용되는 필드이다.



## 매개 변수 필드

매개 변수 필드는 외부에서 값을 받을 수 있는 필드이다. 보고서를 작성하다 보면 예외적으로 데이터 베이스가 아닌 외부에서 넘긴 데이터를 보고서에 출력해야 하는 경우가 있는데 이때 사용할 수 있는 필드가 바로 매개 변수 필드다. 즉, 매개 변수 필드는 외부 값을 보고서로 넘기는 징검다리 역할을 한다. 매개 변수 필드는 외부의 값을 보고서에 출력하려 목적으로 사용할 수 도 있지만 조회 조건에 맞게 동적인 보고서 생성을 위한 SQL의 조건 값으로 도 사용할 수 있다.

## 속성 목록

속성 목록			
속성	상수	버전	설명
name		1.0	매개 변수 필드 이름을 설정한다.
CDATA 섹션		1.0	매개 변수 필드의 값을 설정한다.

## XML 정의

```
<?xml version="1.0" encoding="utf-8" ?>
<document>
  <global>
    <fields>
      <field-set field-type="parameter">
        <field name="parameter1"><![CDATA[ parameter1 data ]]></field>
        <field name="parameter2"><![CDATA[ parameter2 data]]></field>
        <field name="parameter3"><![CDATA[ parameter3 data]]></field>
      </field-set>
    </fields>
  </global>
</document>
```

## 시스템 필드

시스템 필드는 데이터 이외에 레코드 번호 및 개수 전체 페이지 수 및 현재 페이지 그리고 출력 날짜 등의 정보를 출력하는 필드이다. 앞서 언급한 정보들은 일반 개발자들은 알 수 없고 리포팅 틀에서 만 알 수 있기 때문에 시스템 필드를 통해서 제공한다.

### 속성 목록

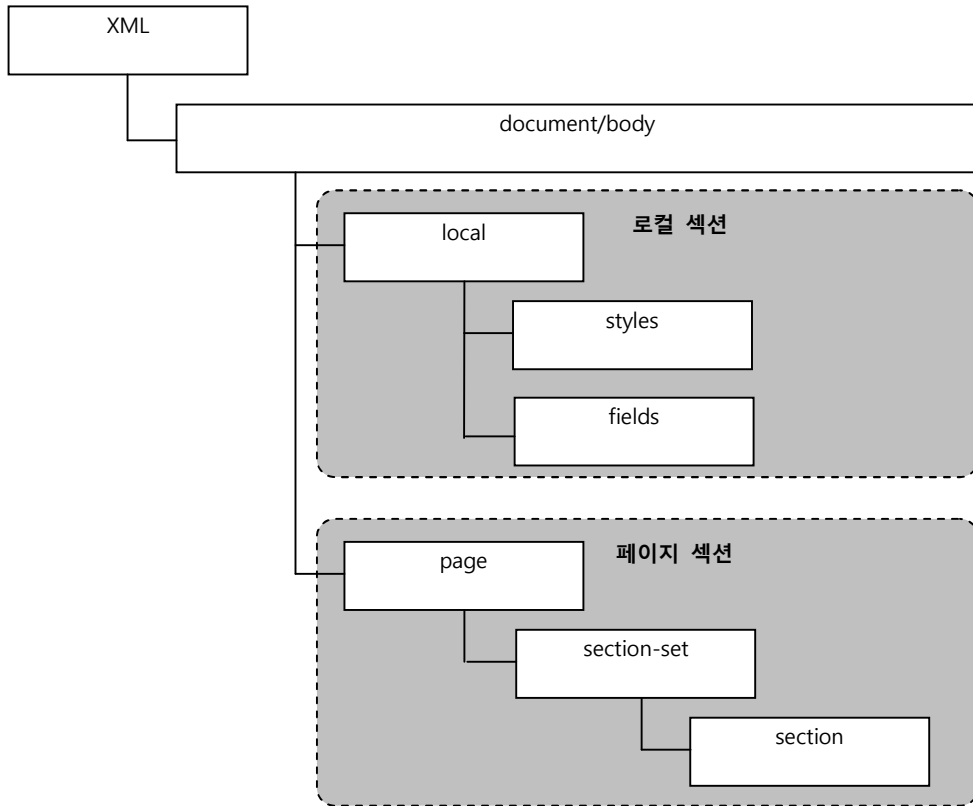
속성 목록			
속성	상수	버전	설명
name	pagenumer	1.0	페이지 번호를 출력한다.
	pagecount	1.0	페이지 개수를 출력한다.
	recordnumber	1.0	레코드 번호를 출력한다.
	recordcount	1.0	레코드 개수를 출력한다.

### XML 정의

```
<?xml version="1.0" encoding="utf-8" ?>
<document>
  <global>
    <fields>
      <field-set field-type="system">
        <field name="pagenumber"></field>
        <field name="pagecount"></field>
        <field name="recordnumber"></field>
        <field name="recordcount"></field>
      </field-set>
    </fields>
  </global>
</document>
```

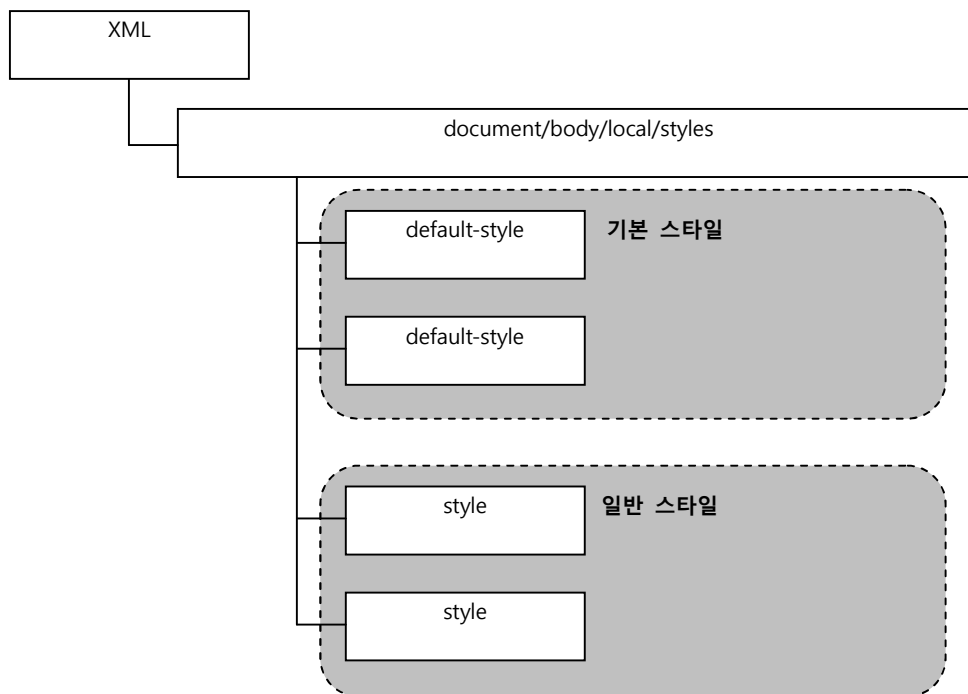


## 바디 섹션(document/body)



## 스타일 목록(document/body/local/styles)

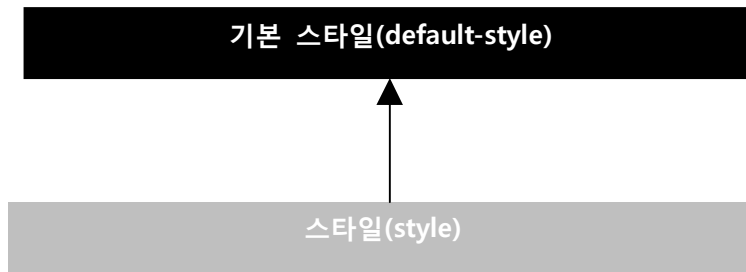
보고서에 공통<sup>3</sup>으로 사용하는 스타일은 글로벌 섹션(document/global/styles) 노드 하위에 정의한다. 하지만, 해당 보고서에서 만 사용할 수 있는 스타일은 document/body/local/styles 노드 하위에 정의한다. 앞으로 노드 'document/body/local/styles'를 로컬 섹션이라는 용어로 대체하겠다. 글로벌 섹션에 정의하는 스타일과 로컬 섹션에 정의하는 스타일의 종류가 서로 다르다. 글로벌 섹션에 정의하는 스타일은 여러 보고서 에서 동시에 참조할 수 있지만 로컬 섹션에 정의하는 스타일은 해당 보고서에서 만 사용할 수 있다. 글로벌 섹션에 있는 스타일 대부분은 모양새와 관련한 스타일이며 로컬 섹션에는 있는 스타일은 동작과 관련된 스타일이다. 동작과 관련된 스타일의 대부분이 특정 보고서의 데이터 베이스 필드를 참조하는 경우가 많기 때문에 글로벌 섹션으로 옮겨 다른 보고서와의 공유가 어렵다. 로컬 섹션에 정의하는 스타일도 글로벌 섹션과 마찬가지로 기본값을 설정하는 default-style 과 style로 구성된다.



<sup>3</sup> 서버 리포트 컨트롤을 지원하지 않는 현재 버전에서는 '보고서에서 공통'이라는 의미가 헷갈릴 수 있다. 현재 파일 포맷상으로는 보고서가 하나만 존재할 수 있기 때문에 글로벌 섹션과 로컬 섹션을 구분하는 것은 무의미할 수 있다. 하지만, 향후 업그레이드를 통해 서버 리포트를 컨트롤을 지원하면 보고서가 여러 개의 서버 보고서를 포함할 수 있다. 보고서가 여러 개 있을 수 있다는 전제로 '보고서에서 공통'이라는 표현을 사용했다.

## 기본 스타일(document/body/local/styles/default-style)

리포팅 툴에서 제공하는 스타일의 기본값을 미리 정의한 것을 기본 스타일이라고 한다. 스타일 종류별로 기본 스타일은 하나씩만 존재할 수 있다. 스타일의 속성 개수는 적게는 한 개에서 많게는 열 개 이상도 가질 수 있다. 기본 스타일은 의미 그대로 기본값을 설정하기 때문에 스타일의 모든 속성값을 설정해야 한다. 반면에, 스타일은 모든 속성값을 설정할 필요가 없으며 기본값과 다른 속성만 설정하면 된다. 이러한 기본 스타일이란 개념을 도입함으로써 리포트 파일의 사이즈를 줄이고 스타일의 속성값을 일괄로 변경 가능한 장점을 얻을 수 있다. 객체 지향 개념의 상속이라는 관점에서 보면 기본 스타일은 스타일의 부모가 되고 스타일은 기본 스타일의 모든 속성을 그대로 물려받는다.



### 기본 스타일 목록

스타일 목록			
스타일 종류	상수	버전	설명
summary		1.0	요약 스타일
condition		1.0	조건 스타일
grouping		1.0	그룹핑 스타일

**스타일(document/body/local/styles/style)**

## 요약(style[@family="summary"])



요약 스타일은 보고서에서 특정 필드의 값을 요약하는데 사용한다. 리포팅 툴에서 요약이라 함은 금액의 합계나 상품의 개수 등을 계산하는 통계를 의미한다. 데이터를 보기 좋게 출력할 수 있게 다양한 컨트롤을 지원하는 것 이외에도 통계를 지원하는 것도 리포팅 툴의 중요한 기능 중에 하나다.

### 속성 목록

속성 목록			
속성	상수	버전	설명
summary-function	none	0	
	sum	1	합계를 계산함.
	avg	2	평균을 계산함.
	count	3	개수를 계산함.
	min	4	최소값을 계산함
	max	5	최대값을 계산함.
summary-scope	report	0	요약 범위를 전체 리포트로 설정함.
	group	1	요약 범위를 그룹으로 설정함.
	dependon-section	2	요약 범위를 섹션에 따라 동적으로 변하게 설정함.
summary-group <sup>4</sup>			그룹 인덱스를 설정함.

### XML 정의

<sup>4</sup> summary-group 속성은 요약을 위한 그룹 인덱스(몇 번째 그룹)를 나타낸다. summary-scope 속성이 group 로 설정된 경우에만 참조한다.

```

<?xml version="1.0" encoding="utf-8" ?>
<document type="" version="1.0">
<body>
  <local>
    <styles>
      <style family="summary" name="가격합계 "
        summary-function="sum" summary-scope="report"
        summary-group="" />
      <style family="summary" name="세금합계 "
        summary-function="sum" summary-scope="report"
        summary-group="" />
    </styles>
  </local>
  <page>
    <section-set section-type="reportheader">
      <section name="보고서머리글 ">
        <control control-type="textbox" x1="0" y1="0" width="100" height="100"
          content-type="field" summary-style="가격합계 ">
          <![CDATA[database.price]]>
        </control>
        <control control-type="textbox" x1="200" y1="0" width="100" height="100"
          content-type="field" summary-style="세금합계 ">
          <![CDATA[database.tax]]>
        </control>
      </section>
    </section-set>
  </page>
</body>
</document>

```

## 컨트롤과 요약 필드

요약 스타일은 글상자 컨트롤과 테이블 셀에서 사용할 수 있다. 요약 스타일에는 요약에 필요한 가장 중요한 속성이 없다. 속성 목록을 다시 한 번 꼼꼼히 보면 요약 함수, 요약 범위 등의 정보는 있지만 정작 요약 대상이 되는 요약 필드에 대한 속성이 없다. 요약 스타일에 요약 필드에 대한 속성이 없는 이유는 컨트롤에 있는 필드 정보를 사용하기 때문이다. 글상자 컨트롤과 테이블 셀은 일반적으로 텍스트를 출력하지만 필드와 연결되면 데이터 베이스의 값이나 파라미터 필드의 값을 출력할 수 도 있다.

글상자 컨트롤과 테이블 셀 모두 content-type 속성이 있다. content-type 속성은 컨트롤의 CDATA 섹션이 포함한 데이터의 타입이 무엇인지를 나타내는 중요한 속성이다. content-type 속성이 "field" 이면 CDATA 섹션이 포함한 데이터는 출력할 필드 이름을 나타내기 때문에 리포트 엔진은 해당 필드 값을 읽어서 출력한다. 반면에, content-type 속성이 "text" 이면 CDATA 섹션이 포함된 데이터는 단순 텍스트

트를 나타내기 때문에 리포트 엔진은 텍스트를 출력한다.

```
<control control-type="textbox" content-type="field" summary-style="가격합계 ">  
  <![CDATA[database.price]]>  
</control>
```

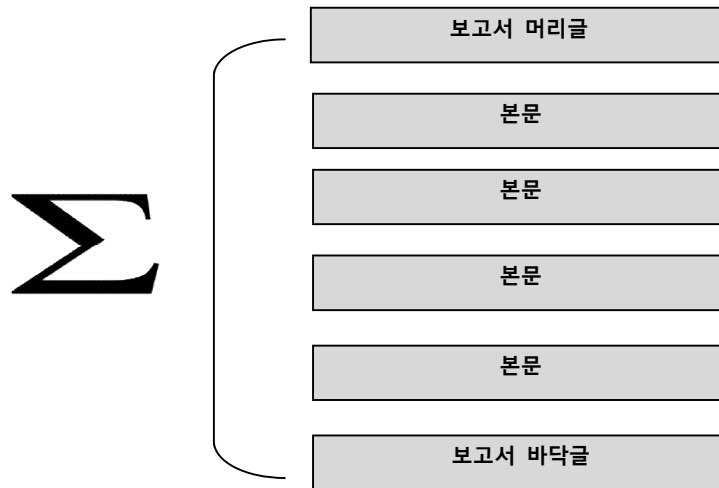
```
<control control-type="textbox" content-type="text" summary-style="가격합계 ">  
  <![CDATA[database.price]]>  
</control>
```

컨트롤에 요약 스타일을 설정했지만 content-type 속성의 값이 "text" 이기 때문에 리포트 엔진은 요약 수행을 하지 않고 CDATA 섹션에 있는 텍스트를 출력한다. 앞서 언급했지만 개발에 중요한 포인트이기 때문에 반복해서 설명하면 요약 스타일은 content-type 속성의 값이 "field" 일 때만 수행한다.

## 요약 함수(style[@summary-function='\*'])

## 요약 범위-리포트(style[@summary-scope='report'])

요약 범위 리포트(@summary-scope='report')는 전체 데이터에 대한 요약을 수행한다. 일반적으로 보고서 머리글 또는 바닥글 섹션에서 전체 합계를 출력할 때 사용할 수 있다.



## 요약 범위-그룹(style[@summary-scope='group'])

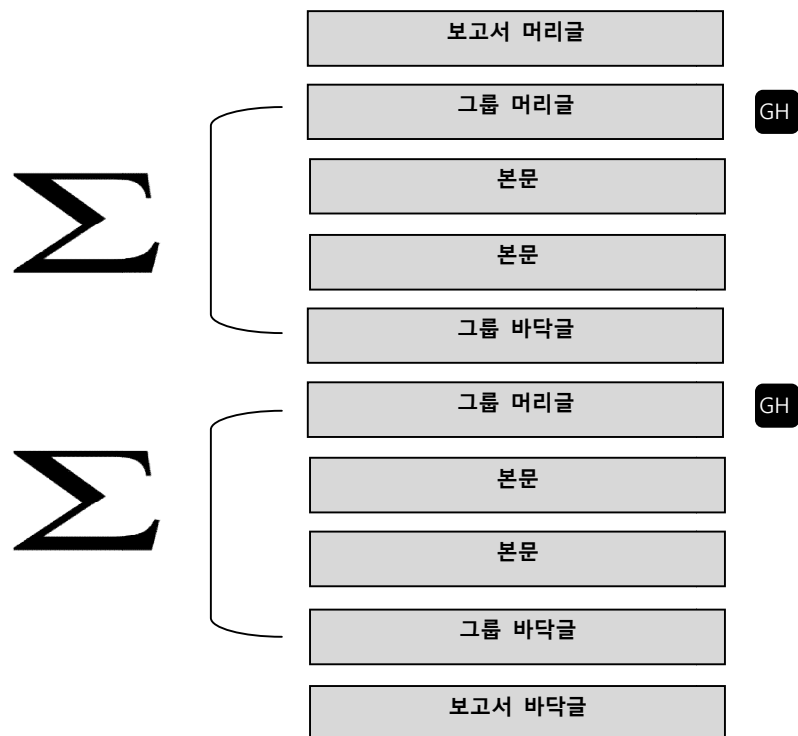
요약 범위 그룹(@summary-scope='group')은 그룹 데이터에 대한 요약을 수행한다. 요약 범위 그룹은 보고서가 그룹핑된 경우에만 사용할 수 있다. 보고서는 여러 단계로 그룹핑이 가능하기 때문에 몇 번째 그룹에 대한 요약인지 표시할 필요가 있다. 이때 몇 번째 그룹인지를 표시하는 속성이 summary-group 속성이다. 몇 번째 그룹에 대한 요약할지 결정한다. summary-group 속성은 요약 범위가 그룹으로 설정했을 때 만 사용되면 그 이외에는 사용되지 않는다.

```
<style family="summary" name="세금합계"
  summary-function="sum"
  summary-scope="group"
  summary-group="1"/>
```



때문에 몇 번째 그룹에 대한 요약은 수행할 지 명시할 필요성이 있다.

보고서를 그룹핑 하면 그룹핑 필드로 소트 옵션에 맞게 그룹핑 된다. 보고서에서 여러 그룹이 나올 수 있다. 이때 각 그룹에 대한 요약을 하고 싶을 때 요약 범위를 그룹을 설정하고 해당 그룹의 이름을 설정하면 된다. 본문 섹션에 배치하면 자신이 포함되는 그룹의 요약이 가능하다. 보고서 머리글과 바닥글 섹션은 그룹 섹션을 벗어 나기 때문에 약간은 다른 식을 작동한다. 보고서 머리글 섹션에서 요약 범위가 그룹으로 설정된 컨트롤은 제일 처음 나오는 그룹의 요약 값을 계산하며 보고서 바닥글 섹션에서 요약 범위가 그룹으로 설정된 컨트롤은 제일 마지막에 나오는 그룹의 요약 값을 계산한다.



### 요약 범위-섹션에의존(style[@summary-scope='dependon-section])

요약 범위를 섹션에 의존으로 설정하면 요약의 범위가 섹션의 타입에 따라 동적으로 바뀐다. 섹션 타입에 따라 요약 범위는 아래 표를 참조하면 된다.

섹션의 종류와 요약 범위	
섹션 종류	요약 범위
보고서 머리글	요약 범위가 전체 데이터로 설정됨.
보고서 바닥글	요약 범위가 전체 데이터로 설정됨.

페이지 머리글	지원하지 않음.
페이지 바닥글	지원하지 않음.
본문	요약 범위가 레코드 하나로 설정됨.
그룹 머리글	요약 범위가 그룹으로 설정됨.
그룹 바닥글	요약 범위가 그룹으로 설정됨.

섹션의 타입에 따라 요약 범위가 바뀐다는 의미는 요약을 설정할 때 보고서 개발자가 명시적으로 리포트, 그룹 등을 설정할 필요가 없다는 뜻이다. 전체 요약을 출력 하고 싶다면 보고서 머리글 또는 바닥글 섹션에 컨트롤을 배치하면 되고 그룹 요약을 출력하고 싶으면 그룹 머리글이나 바닥글 섹션에 컨트롤을 배치하면 된다. 섹션의 타입에 따라 요약 범위가 자동으로 바뀌면 보고서 디자인이 편리하다. 그룹 바닥글에 요약 값을 출력하기 위해서 이미 수십 개에 달하는 테이블 셀에 요약 설정을 한 상태에서 보고서 바닥글에 전체 요약 값을 출력해야 하는 경우가 발생했을 때 그룹 바닥글에 있는 테이블을 복사해서 보고서 바닥글에 복사하면 된다. 이러한 기능이 없다면 복사한 후에 셀의 요약 설정을 하나씩 변경하는 번거로움이 있다.

## 조건(style[@family="condition"])

페이지 스타일은 용지 종류 및 사이즈 등의 페이지 정보를 포함한다.

### 속성 목록

### XML 정의

```
<?xml version="1.0" encoding="utf-8" ?>
<document-style>
  <styles>
    <style family="page" name="pageA4"
      page-type="a4" page-width="2100"
      page-height="2970" page-direction="portrait"
      page-padding-right="10" page-padding-left="10"
      page-padding-top="10" page-padding-bottom="10" />
    <style family="page" name="pageB4"
      page-type="b4" page-width="2570"
      page-height="3640" page-direction="landscape"
      page-padding-right="10" page-padding-left="10"
      page-padding-top="10" page-padding-bottom="10" />
  </styles>
</document-style>
```

### 참고 사항

## 그룹핑(style[@family="grouping"])

페이지 스타일은 용지 종류 및 사이즈 등의 페이지 정보를 포함한다.

### 속성 목록

속성 목록				
속성	상수		버전	설명
grouping-field				
grouping-sort	none	0		
	ascend	1		
	descend	2		

### XML 정의

```

<?xml version="1.0" encoding="utf-8" ?>
<document type="" version="1.0">
<body>
  <local>
    <styles>
      <style family="grouping" name="부서별그룹핑 "
        grouping-field="database.department" grouping-sort="ascend" />
      <style family="grouping" name="직책별그룹핑 "
        grouping-field="database.position" grouping-sort="descend" />
    </styles>
  </local>
  <page>
    <section-set section-type="groupheader" grouping-style="부서별그룹핑 ">
      <section name="그룹머리글 1" />
    </section-set>
    <section-set section-type="groupheader" grouping-style="부서별그룹핑 ">
      <section name="그룹머리글 2" />
    </section-set>
    <section-set section-type="detail">
      <section name="본문 1" />
    </section-set>
    <section-set section-type="groupfooter">
      <section name="그룹바닥글 2" />
    </section-set>
    <section-set section-type="groupfooter">
      <section name="그룹바닥글 1" />
    </section-set>
  </page>
</body>
</document>

```

## 참고 사항

**그룹 소트-안함(style[@group-sort='none'])**

**그룹 소트-오름차순(style[@group-sort='ascend'])**

**그룹 소트-내림차순(style[@group-sort='descend'])**

## 필드 목록(document/body/local/fields)

## 데이터 베이스 필드

데이터 베이스 필드는 보고서에 연결된 데이터셋의 데이터를 출력하는 필드다. 여기서 데이터셋은 SQL 수행한 결과로 반환된 레코드셋 정도로 이해하면 된다. 무엇보다도 리포팅 툴의 사용 목적이 데이터를 출력하는 데 있기 때문에 보고서 작성시 가장 많이 사용하는 필드가 바로 데이터 베이스 필드다.

### 속성 목록

속성 목록			
속성	상수	버전	설명
name		1.0	데이터 베이스 필드명을 설정한다.

### XML 정의

```
<?xml version="1.0" encoding="utf-8" ?>
<document>
  <global>
    <fields>
      <field-set field-type="database">
        <field name="name" />
        <field name="department" />
        <field name="position" />
        <field name="phone" />
        <field name="address" />
      </field-set>
    </fields>
  </global>
</document>
```



## 공식 필드

공식 필드는 자바 스크립트로 데이터를 변형하거나 복잡한 연산이 필요할 때 사용하는 필드이다. 공식 필드에서 사용하는 스크립트 언어는 웹 개발에서 가장 대중적으로 사용되는 자바 스크립트이다. 자바 스크립트는 대부분의 개발자들이 친숙하기 때문에 특별한 교육 없이도 사용할 수 있으며 인터넷에 있는 수 많은 자료를 그대로 활용할 수 있는 장점이 있다. 공식 필드는 정형화 되지 않고 조건에 따라서 보고서의 모양이 변경되거나 데이터가 바뀌는 등의 복잡하고 가변성이 많은 보고서를 만드는데 반드시 필요하다.

## 속성 목록

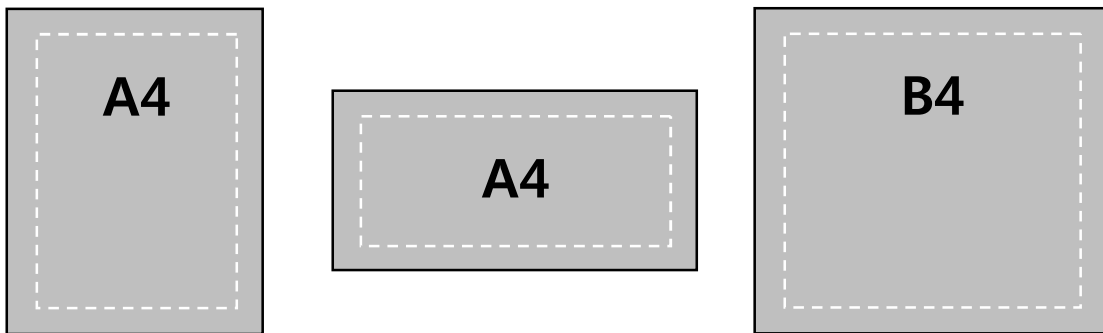
속성 목록			
속성	상수	버전	설명
name		1.0	공식 필드명을 설정한다.
CDATA 섹션		1.0	공식 필드의 스크립트를 작성한다.

## XML 정의

```
<?xml version="1.0" encoding="utf-8" ?>
<document type="" version="1.0">
  <global>
    <fields>
      <field-set field-type="formula">
        <field name="formula1"><![CDATA[
          return "test";
        ]]></field>
        <field name="formula2"><![CDATA[
          var a = 10;
          var b = 20;
          return a + b;]]></field>
        <field name="formula3"><![CDATA[
          var a = report.field("database.number1");
          var b = report.field("database.number2");
          return a + "-" + b; ]]> </field>
      </field-set>
    </fields>
  </global>
</document>
```

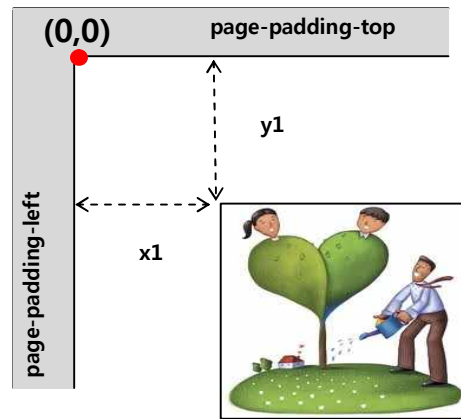
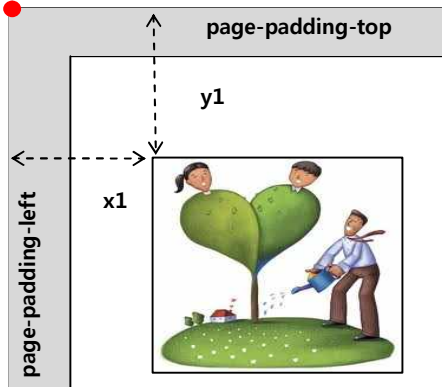
## 페이지

본 문서에서 페이지는 쪽을 구분하거나 컨트롤을 포함하는 컨테이너(container)의 역할도 하면서 프린터 출력 용지에 대한 정보를 담고 있다. 용지에 대한 정보가 body 나 document 속성에 없고 매 페이지마다 용지 정보가 포함되는 이유는 하나의 문서 안에 서로 다른 용지가 포함될 수 있기 때문이다. 즉, 즉 첫 페이지는 A4 세로 방향 용지, 두 번째 페이지는 A4 가로 방향 용지, 세 번째 페이지는 B4 용지가 될 수 있다.

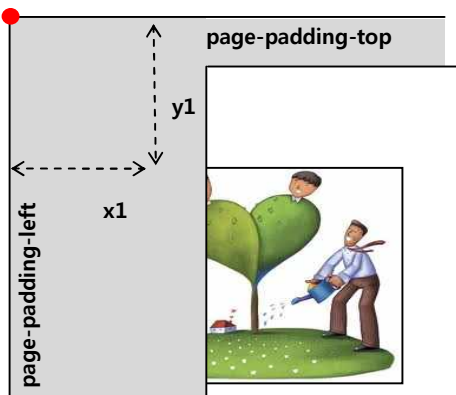


## 페이지 여백과 컨트롤의 좌표

(0,0)



페이지에는 오른쪽, 왼쪽, 위쪽, 아래쪽 여백을 설정하는 속성이 있다. 페이지 여백은 제본 등의 특별한 이유로 적당히 비워둔 페이지의 가장자리 공간 정도로 이해하면 된다.



페이지의 왼쪽 여백과 위쪽 여백은 컨트롤의 시작 위치를 결정하는 중요한 속성이다. 일반적으로 개발자나 사용자들은 왼쪽 여백을 페이지에 포함된 내용을 전체적으로 오른쪽으로 이동시키고 싶을 때 사용하고 오른쪽 여백은 페이지 내용을 전체적으로 아래쪽으로 이동시키고 싶을 때 사용한다. 애플리케이션 개발자는 이러한 일반적인 사람들의 의도에 맞게 컨트롤을 그릴 때 페이지의 왼쪽 여백과 위쪽 여백을 컨트롤의 베이스 좌표로 사용해야 한다

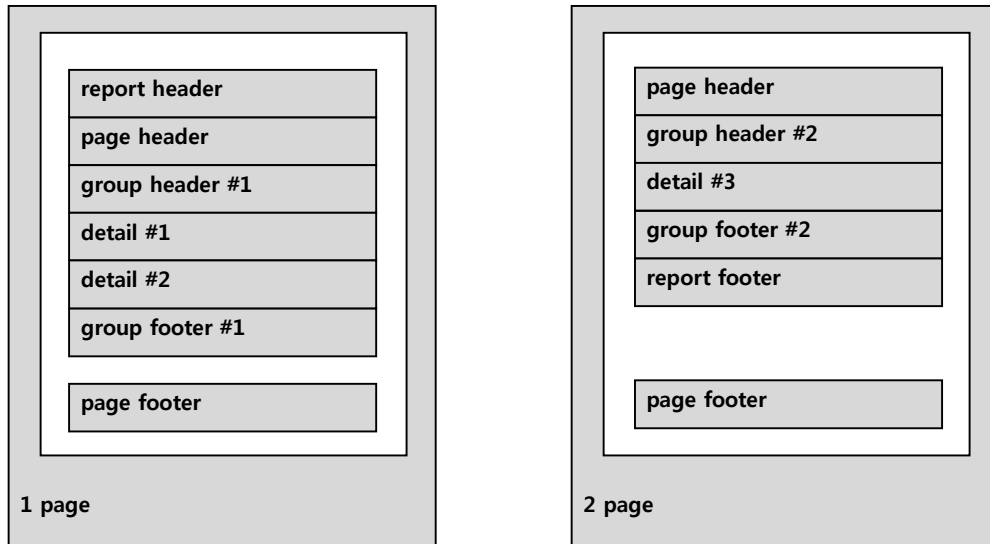
페이지 안에 포함된 모든 컨트롤은 페이지 여백 속성에 의해 그리는 위치가 영향 받는다. 페이지 여백의 의미대로 x1 좌표에 페이지 왼쪽 여백 사이즈를 더하고 y1 좌표에 페이지 위쪽 여백을 더해야 한

다. 페이지 위쪽 여백을 늘리면 전체 컨트롤이 아래로 밀려야 한다. 그리고, 페이지의 왼쪽 여백을 늘리면 컨트롤이 오른쪽으로 밀려야 한다.

실제 x1 좌표 :  $\text{page.page-padding-left} + \text{control.x1}$

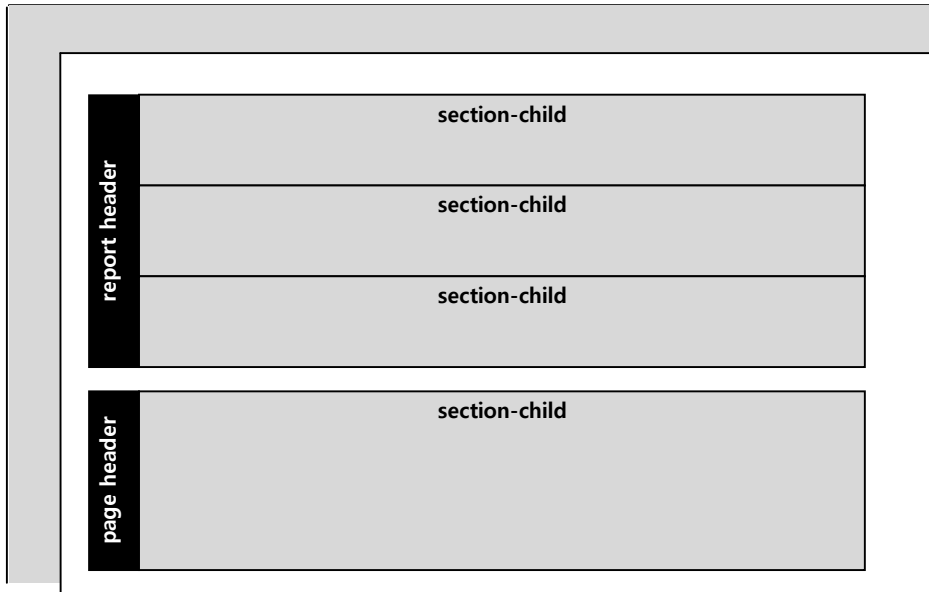
실제 y1 좌표 :  $\text{page.page-padding-top} + \text{control.y1}$

## 섹션



대부분의 보고서는 출력 용지가 정해져 있기 때문에 워드프로세서처럼 용지의 좌측 상단을 기준으로 한 절대 좌표에 컨트롤을 배치하는 개념은 아마도 대부분의 사람들이 쉽게 이해할 수 있고 툴을 직감적으로 사용할 수 있다. 직감적인 사용이라는 장점이 있음에도 리포팅 툴의 사용자 인터페이스를 워드프로세서처럼 만들면 리스트 형태의 단순한 보고서는 쉽게 작성할 수 있겠지만 복잡한 보고서는 작성이 어려운 문제점이 있다. 예를 들면, 특정 키 값으로 그룹핑을 했을 때 그룹이 시작하는 시점에서 타이틀을 출력하고 끝나는 시점에 소계를 출력하는 등의 복잡한 보고서를 작성하기는 어렵다. 앞서 예를 든 보고서를 만들기 어려운 이유 중에 하나는 바로 데이터에 따라서 찍히는 위치가 동적으로 바뀌기 때문이다. 이러한 문제점을 해결하고자 리포팅 툴에서는 출력 용지를 기준으로 한 절대 좌표를 사용하지 않고 개념적으로 출력되는 영역을 몇 가지로 구분을 하고 그 영역을 "섹션(section)" 이라 부른다.

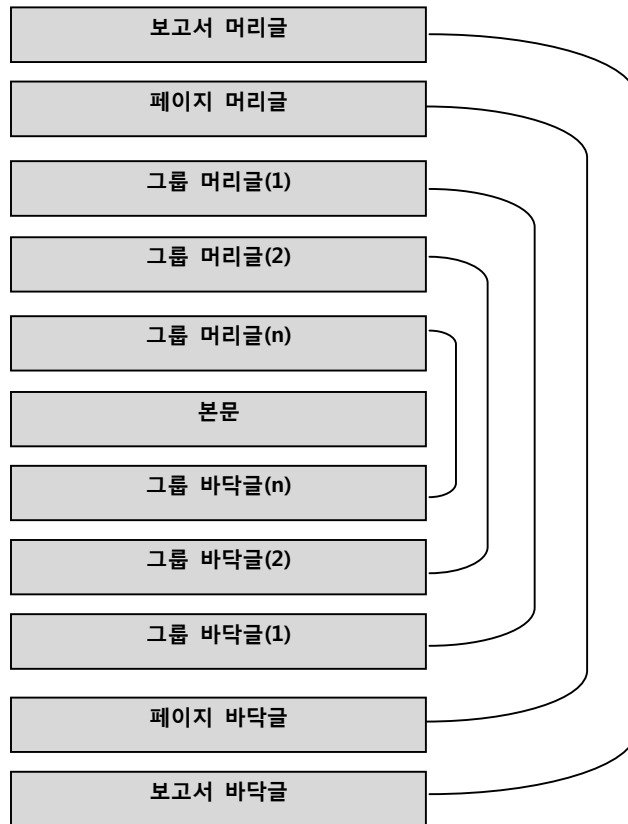
섹션의 종류	
섹션 종류	설명
보고서 머리글	보고서의 맨 처음 한 번만 출력되는 섹션
보고서 바닥글	보고서의 맨 마지막 한 번만 출력되는 섹션
페이지 머리글	매 페이지 상단에 출력되는 섹션
페이지 바닥글	매 페이지 하단에 출력되는 섹션
본문	레코드 개수만큼 반복되는 섹션
그룹 머리글	그룹핑이 시작할 때 출력되는 섹션
그룹 바닥글	그룹핑이 끝날 때 출력되는 섹션



속성 목록				
속성	상수		버전	설명
section-type	reporthead	0		
	reportfooter	1		
	pageheader	2		
	pagefooter	3		
	detail	4		
	groupheader	5		
	groupfooter	6		

## 섹션 묶음

page 노드 하위로 setction-set 노드가 온다. 한 보고서 안에 여러 섹션이 올 수 있다. 섹션의 배치 순서는 매우 중요하다. 보고서 머리글, 보고서 바닥글, 페이지 머리글, 페이지 바닥글, 본문 섹션은 페이지 내에서 하나씩만 있어야 한다. 페이지 노드 하위로 최소한 본문 섹션은 포함해야 한다. 즉, 본문 섹션은 필수다. 그리고 보고서에 그룹핑이 추가될 때 마다 그룹 머리글과 그룹 바닥글이 한 쌍으로 존재한다. 보고서에 그룹핑이 추가된 만큼 그룹 머리글과 바닥글이 추가되어야 하면 그 쌍이 한쌍 맞아야 한다. 만일에 그룹핑의 쌍이 안 맞으면 보고서 생성을 하지 못한다. 그래서 아래와 같은 순서로 섹션을 배치하기를 권장한다.



```
<?xml version="1.0" encoding="utf-8" ?>
<document type="" version="1.0">
<global />
<body>
  <page>
    <section-set section-type="reportheader" />
    <section-set section-type="pageheader" />
    <section-set section-type="groupheader" />
    <section-set section-type="groupheader" />
    <section-set section-type="detail" />
    <section-set section-type="groupfooter" />
    <section-set section-type="groupfooter" />
    <section-set section-type="pagefooter" />
    <section-set section-type="reportfooter" />
  </page>
</body>
</document>
```



## 보고서 머리글(section[@section-type="reportheader"])

보고서 머리글은 보고서에서 처음 한 번만 출력되는 섹션이다. 보고서 머리글에 무엇을 작성하라는 규칙은 없지만 일반적으로 보고서의 제목이나 첫 장에 만 출력할 필요가 있는 경우 보고서 머리글에 작성한다. 아주 특별한 경우로 첫 장에 보고서 전체에 대한 합계를 출력하는 경우도 보고서 머리글에 작성한다.

### 속성 목록

section 노드 속성 목록			
속성	상수	버전	설명
name		1.0	섹션의 이름을 설정한다.
visible-style		1.0	보임 스타일을 설정한다.
section-style		1.0	섹션 스타일을 설정한다.
condition-style		1.0	조건 스타일을 설정한다.
background-style		1.0	배경 스타일을 설정한다.

### XML 정의

```
<?xml version="1.0" encoding="utf-8" ?>
<document type="" version="1.0">
<global />
<body>
  <page>
    <section-set section-type="reportheader">
      <section name="보고서머리글 1" height="100"
        section-style="section1"
        condition-style="condition1"
        background-style="background1" />
      <section name="보고서머리글 2" height="200"
        section-style="section2"
        condition-style="condition1"
        background-style="background1" />
    </section-set>
  </page>
</body>
</document>
```

## 페이지 머리글(section[@section-type="pageheader"])

페이지 머리글은 매 페이지 상단에 출력되는 섹션이다. 매 페이지 상단에 무엇인가를 출력하고 싶다면 페이지 머리글에 디자인하면 된다. 일반적으로 페이지 번호나 전체 페이지 수 등을 페이지 머리글에 디자인 하는 경우가 많다.

### 속성 목록

section 노드 속성 목록			
속성	상수	버전	설명
name		1.0	섹션의 이름을 설정한다.
height		1.0	섹션의 높이를 설정한다.
section-style		1.0	섹션 스타일을 설정한다.
condition-style		1.0	조건 스타일을 설정한다.
background-style			배경 스타일을 설정한다.

### XML 정의

```
<?xml version="1.0" encoding="utf-8" ?>
<document type="" version="1.0">
<global />
<body>
  <page>
    <section-set section-type="pageheader">
      <section name="페이지 머리글 1" height="100" />
      <section name="페이지 머리글 2" height="200" />
    </section-set>
  </page>
</body>
</document>
```

## 본문(section[@section-type="detail"])

본문은 리포팅 틀에서 제일 중요한 섹션이자 가장 많이 사용하는 섹션이다. 리포팅 틀을 사용하는 목적은 데이터를 일정한 모양새를 갖춰서 보여주는 데 있다. 이러한 기본적인 목적에 맞게 보고서는 데이터와 연결되고 데이터 개수만큼 반복되는 섹션이 본문이다. 데이터를 리스트 형태로 출력하고 싶다면 본문 섹션에 디자인을 하면 된다. 본문을 제외한 다른 섹션들은 필요에 따라서 삭제할 수 있지만 본문 섹션은 절대로 삭제할 수 없다.

※ 본문 섹션에 해당하는 **section-set** 노드가 없는 보고서 포맷은 잘 못 만들어진 포맷으로 처리하여 보고서 생성을 하지 않는다.

## 속성 목록

section 노드 속성 목록			
속성	상수	버전	설명
name		1.0	섹션의 이름을 설정한다.
height		1.0	섹션의 높이를 설정한다.
section-style		1.0	섹션 스타일을 설정한다.
condition-style		1.0	조건 스타일을 설정한다.
background-style			배경 스타일을 설정한다.

## XML 정의

```
<?xml version="1.0" encoding="utf-8" ?>
<document type="" version="1.0">
<global />
<body>
  <page>
    <section-set section-type="detail">
      <section name="본문 1" height="100" />
      <section name="본문 2" height="200" />
    </section-set>
  </page>
</body>
</document>
```

## 보고서 바닥글(section[@section-type="reportfooter"])

보고서 바닥글은 보고서에서 제일 마지막 한 번만 출력되는 섹션이다. 데이터의 총계나 마지막 장에만 출력할 부분은 보고서 바닥글에 디자인하면 된다.

### 속성 목록

section 노드 속성 목록			
속성	상수	버전	설명
name		1.0	섹션의 이름을 설정한다.
height		1.0	섹션의 높이를 설정한다.
section-style		1.0	섹션 스타일을 설정한다.
condition-style		1.0	조건 스타일을 설정한다.
background-style			배경 스타일을 설정한다.

### XML 정의

```
<?xml version="1.0" encoding="utf-8" ?>
<document type="" version="1.0">
<global />
<body>
  <page>
    <section-set section-type="reportfooter">
      <section name="보고서바닥글 1" height="100" />
      <section name="보고서바닥글 2" height="200" />
    </section-set>
  </page>
</body>
</document>
```

## 페이지 바닥글(section[@section-type="pagefooter"])

페이지 바닥글은 매 페이지 하단에 출력되는 섹션이다. 매 페이지 하단에 무엇인가를 출력하고 싶다면 페이지 바닥글에 디자인하면 된다. 페이지 머리글과 마찬가지로 페이지 번호나 전체 페이지 수 등을 페이지 바닥글에 디자인 하는 경우가 많다.

### 속성 목록

section 노드 속성 목록			
속성	상수	버전	설명
name		1.0	섹션의 이름을 설정한다.
height		1.0	섹션의 높이를 설정한다.
section-style		1.0	섹션 스타일을 설정한다.
condition-style		1.0	조건 스타일을 설정한다.
background-style			배경 스타일을 설정한다.

### XML 정의

```
<?xml version="1.0" encoding="utf-8" ?>
<document type="" version="1.0">
<global />
<body>
  <page>
    <section-set section-type="pagefooter">
      <section name="페이지바닥글 1" height="100" />
      <section name="페이지바닥글 2" height="200" />
    </section-set>
  </page>
</body>
</document>
```



## 그룹 머리글(section[@section-type="groupheader"])

통계를 목적으로 보고서를 만든다면 데이터 그룹핑은 거의 필수나 다름없다. 리포팅 툴에서 그룹핑을 하면 그룹 머리글과 그룹 바닥글이 본문 섹션을 둘러싼 모양새로 생성된다. 본문 위쪽에 만들어지는 섹션이 그룹 머리글이다. 이러한 그룹 머리글은 그룹이 시작하는 시점, 즉 그룹에 포함되는 본문 섹션이 출력되기 바로 전에 출력되는 섹션이다. 일반적으로 그룹별로 타이틀을 출력할 때 그룹 머리글 섹션을 사용한다.

※ 그룹 머리글 과 그룹 바닥글 섹션은 절대 분리할 수 없는 섹션으로 그 개수가 항상 똑같아야 한다. 예를 들어 그룹핑이 2개이면 그룹 머리글 섹션 2개와 그룹 바닥글 섹션 2개가 있어야 한다. 만일에 그룹 머리글과 바닥글 섹션의 개수가 서로 틀린 보고서 포맷은 잘 못 만들어진 포맷으로 처리하여 보고서 생성을 하지 않는다.

### 속성 목록

section 노드 속성 목록			
속성	상수	버전	설명
name		1.0	섹션의 이름을 설정한다.
height		1.0	섹션의 높이를 설정한다.
section-style		1.0	섹션 스타일을 설정한다.
condition-style		1.0	조건 스타일을 설정한다.
background-style		1.0	배경 스타일을 설정한다.
grouping-style		1.0	그룹핑 스타일을 설정한다.

### XML 정의



```
<?xml version="1.0" encoding="utf-8" ?>
<document type="" version="1.0">
<global />
<body>
  <page>
    <groupheader section-type="groupheader">
      <section name="그룹머리글 1" height="100" />
      <section name="그룹머리글 2" height="200" />
    </section-set>
  </page>
</body>
</document>
```

## 그룹 바닥글(section[@section-type="groupfooter"])

통계를 목적으로 보고서를 만든다면 데이터 그룹핑은 거의 필수나 다름없다. 리포팅 툴에서 그룹핑을 하면 그룹 머리글과 그룹 바닥글이 본문 섹션을 둘러싼 모양새로 생성된다. 본문 아래쪽에 만들어지는 섹션이 그룹 바닥글이다. 이러한 그룹 바닥글은 그룹이 끝나는 시점 즉, 그룹에 포함되는 본문 섹션이 모두 출력된 후에 나오는 섹션이다. 일반적으로 그룹별로 소계를 출력할 때 그룹 바닥글 섹션을 사용한다.

※ 그룹 머리글 과 그룹 바닥글 섹션은 절대 분리할 수 없는 섹션으로 그 개수가 항상 똑같아야 한다. 예를 들어 그룹핑이 2개이면 그룹 머리글 섹션 2개와 그룹 바닥글 섹션 2개가 있어야 한다. 만일에 그룹 머리글과 바닥글 섹션의 개수가 서로 틀린 보고서 포맷은 잘 못 만들어진 포맷으로 처리하여 보고서 생성을 하지 않는다.

### 속성 목록

section 노드 속성 목록			
속성	상수	버전	설명
name		1.0	섹션의 이름을 설정한다.
height		1.0	섹션의 높이를 설정한다.
section-style		1.0	섹션 스타일을 설정한다.
condition-style		1.0	조건 스타일을 설정한다.
background-style			배경 스타일을 설정한다.

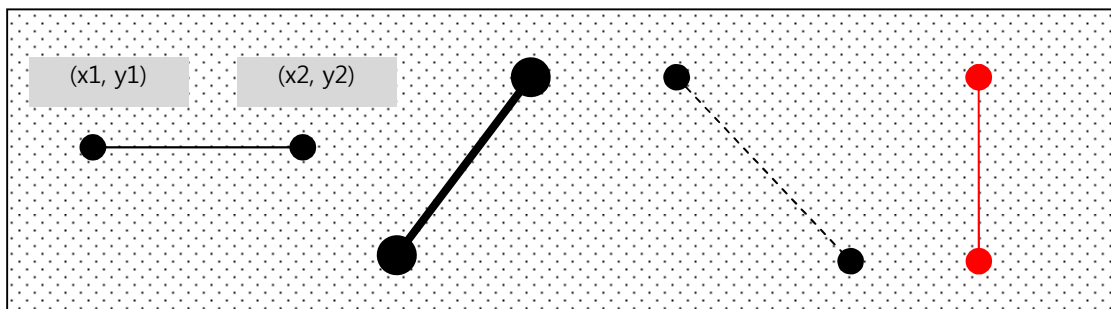
### XML 정의

```
<?xml version="1.0" encoding="utf-8" ?>
<document type="" version="1.0">
<global />
<body>
  <page>
    <section-set section-type="groupfooter">
      <section name="그룹바닥글 1" height="100" />
      <section name="그룹바닥글 2" height="200" />
    </section-set>
  </page>
</body>
</document>
```

## 컨트롤

## 라인(control["@type="line"])

라인 컨트롤은 시작점(x1, y1)과 끝점(x2, y2)을 지정하면 특정 라인 스타일과 색으로 라인을 그리는 컨트롤이다.



라인 컨트롤에서 참조하는 스타일은 "라인" 스타일만이다. "라인" 스타일 이외의 스타일은 적용하더라도 무시한다. 라인 스타일을 지정하지 않으면 기본 라인 스타일을 적용한다.

### 속성 목록

속성 목록		
속성	상수	설명
x1		라인의 X 시작 좌표
y1		라인의 Y 시작 좌표
x2		라인의 X 끝 좌표
y2		라인의 Y 끝 좌표
line-style		라인의 스타일

모든 좌표 값은 0 이상의 양수이다.

### XML 정의

```
<?xml version="1.0" encoding="utf-8" ?>
<document type="" version="1.0">
<stylesheet>
  <style type="line"
    id="line1"
    style="line-color:rgb(255,255,0);line-type:solid;line-width:1" />
</stylesheet>
<body>
<page>
  <control type="line" style="line:line1" x1="1" y1="1" x2="4" y2="4" />
</page>
</body>
</document>
```

## 도형(control[@type="shape"])

도형 컨트롤은 시작점과 끝점을 지정하면 지정한 형태의 도형을 그리는 컨트롤이다.

### 속성 목록

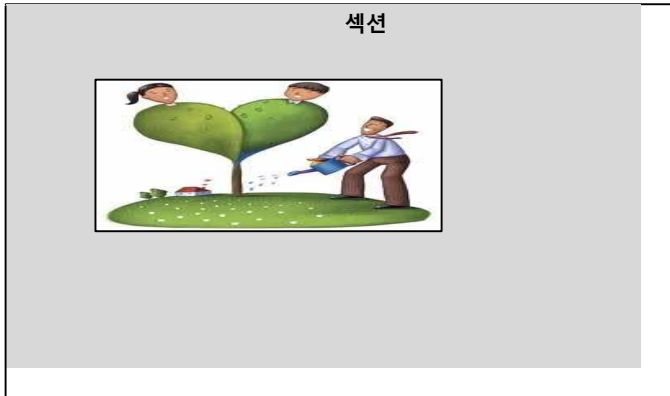
속성 목록		
속성	상수	설명
x1		도형 컨트롤의 시작 X 좌표
y1		도형 컨트롤의 시작 Y 좌표
width		도형 컨트롤의 넓이
height		도형 컨트롤의 높이
background-style		도형 컨트롤의 스타일
border-style		

### XML 정의

```
<?xml version="1.0" encoding="utf-8" ?>
<document type="" version="1.0">
<stylesheet>
<style type="border"
  id="border1"
  style="border-right-color:rgb(0,0,0);border-right-style:solid;border-right-
width:10;border-left-color:rgb(0,0,0);border-left-style:solid;border-left-
width:10;border-top-color:rgb(0,0,0);border-top-style:solid;border-top-width:10;border-
bottom-color:rgb(0,0,0);border-bottom-style:solid;border-bottom-width:10" />
<style type="background"
  id="background1"
  style="background-color:rgb(255,255,255);background-style:normal;background-
fill-style:none;background-fill-color:rgb(0,0,0)" />
</stylesheet>
<body>
<page>
  <control type="shape"
    class="border: border1; background: background1"
    x1="100"
    y1="100"
    width="200"
    height="200" />
</page>
</body>
</document>
```



## 글상자(control[@type="textbox"])



글상자 컨트롤은 보고서 제목과 같은 텍스트를 출력하거나 데이터 베이스 내용을 출력할 때 사용하는 대표적인 컨트롤이다. 도형 컨트롤과 비슷하지만 텍스트를 출력할 수 있다는 점에서 차이가 있다. 텍스트를 출력하기 때문에 이에 따른 폰트, 여백 등의 추가적인 속성들을 포함한다. 보고서에 데이터의 값이나 텍스트를 출력할 때 사용한다. 본 리포팅 툴에서 이미지 컨트롤이 없는 대신에 글상자 컨트롤이 이미지 스타일을 지원함으로써 이미지 컨트롤을 대신한다. 보고서에서 이미지를 추가하고 싶다면 글상자 컨트롤이나 테이블 셀의 이미지 스타일을 설정하면 된다. 글상자 컨트롤의 기능을 요약하면 다음과 같다.

- 보고서 제목과 같은 텍스트를 출력한다.
- 데이터 베이스 및 매개 변수 필드의 내용을 출력한다.
- 이미지를 출력한다.
- 요약 값을 출력한다.

### 속성 목록

속성 목록			
속성	상수	버전	설명
x1		1.0	글상자 컨트롤의 x1 좌표를 설정.
y1		1.0	글상자 컨트롤의 y1 좌표를 설정.
width		1.0	글상자 컨트롤의 너비를 설정.
height		1.0	글상자 컨트롤의 높이를 설정.

content-type	text	0	1.0	글상자 컨트롤의 내용을 일반 텍스트로 설정.
	field	1	1.0	글상자 컨트롤의 내용을 필드로 설정.
visible-style			1.0	글상자의 보임 스타일을 설정
border-style			1.0	글상자의 테두리 스타일을 설정.
background-style			1.0	글상자의 배경 스타일을 설정.
font-style			1.0	글상자의 폰트 스타일을 설정.
image-style			1.0	글상자의 이미지 스타일을 설정.
padding-style			1.0	글상자의 여백 스타일을 설정.
summary-style			1.0	글상자의 요약 스타일을 설정.
condition-style			1..0	글상자의 조건 스타일을 설정.

## XML 정의

```

<?xml version="1.0" encoding="utf-8" ?>
<document type="" version="1.0">
<body>
<page>
<section-set>
<section>

<control control-type="textbox" x1="0" y1="0" width="100" height="100"
content-type="text" visible-style="보임 1" border-style="테두리 1"
background-style="배경 1" font-style="폰트 1" image-style="이미지 1"
padding-style="여백 1" summary-style="" condition-style="">
<![CDATA[ 안녕하세요 ~ ]]></control>

<control control-type="textbox" x1="0" y1="0" width="100" height="100"
content-type="field" visible-style="보임 1" border-style="테두리 1"
background-style="배경 1" font-style="폰트 1" image-style="이미지 1"
padding-style="여백 1" summary-style="" condition-style="">
<![CDATA[database.name]]></control>

</section>
</section-set>
</page>
</body>
</document>

```

## 콘텐츠 타입(control[@content-type])

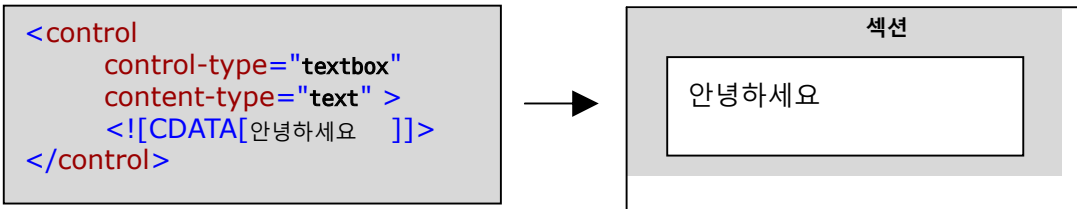
글상자 컨트롤의 가장 큰 특징은 텍스트를 출력하는 데 있다. 텍스트에는 보고서의 타이틀이나 각 항

목의 타이틀과 같이 고정 텍스트가 될 수도 있고 데이터 베이스나 매개 변수 필드의 값이 될 수도 있다. 즉, 글상자 컨트롤이 고정 텍스트를 출력할 것인지 아니면 특정 필드의 값을 출력할 것인지 구분할 필요가 있는데 콘텐츠 타입 속성이 이런 역할을 하는 속성이다. 콘텐츠 타입 속성이 "문자"로 설정됐으면 CDATA 섹션의 내용은 일반 텍스트가 된다. 콘텐츠 타입 속성이 "필드"로 설정됐으면 필드 값을 출력하라는 의미이므로 CDATA 섹션의 내용은 필드 이름이 된다. 글상자 컨트롤은 해당 필드 값을 레코드에서 읽어 화면에 출력하면 된다. 간단하다.

```
<control control-type="textbox" x1="0" y1="0" width="100" height="100"
  content-type="text" >
  <![CDATA[안녕하세요 ]]>
</control>
```

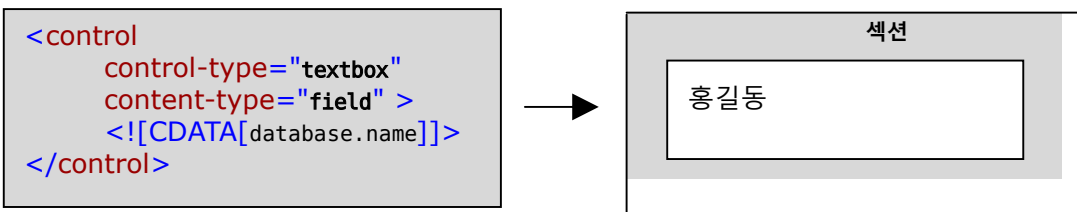
### 콘텐츠 타입(control[@content-type="text"])

콘텐츠 타입 "문자(text)"는 CDATA 섹션의 내용이 고정 텍스트임을 나타낸다. 글상자 컨트롤은 CDATA 섹션이 포함한 텍스트를 그대로 출력하면 된다.



### 콘텐츠 타입(control[@content-type="field"])

콘텐츠 타입 "필드(field)"는 CDATA 섹션의 내용이 필드 이름임을 나타낸다. 글상자 컨트롤은 CDATA 섹션이 포함한 필드 이름의 값을 레코드로부터 읽어 출력하면 된다. 해당 콘텐츠 타입은 "문자" 타입 처럼 고정적이지 않고 레코드마다 내용이 동적으로 바뀐다. 글상자 컨트롤이 본문 섹션에 위치하고 콘텐츠 타입이 필드라면 매번 다른 데이터를 출력한다.



CDATA 섹션에 필드 이름을 명시하는 방법은 아래처럼 필드 타입과 필드 이름을 묶은 형태이다. 필드

이름은 사용자가 임의로 작성 가능하지만 필드 타입은 아래처럼 표처럼 영문으로 이미 정의되어 있다. 개발자는 참조하려는 필드의 타입이 무엇인지 명확히 알고 있어야 한다.

**<![CDATA[필드타입.필드이름]]>**

필드			
필드 타입 한글명	필드 타입 영문명	필드 이름(예)	예시
매개 변수 필드	parameter	number	parameter.number
시스템 필드	system	pagenumber	system.pagenumber
데이터 베이스 필드	database	firstname	database.firstname
공식 필드	formula	fullname	formula.fullname

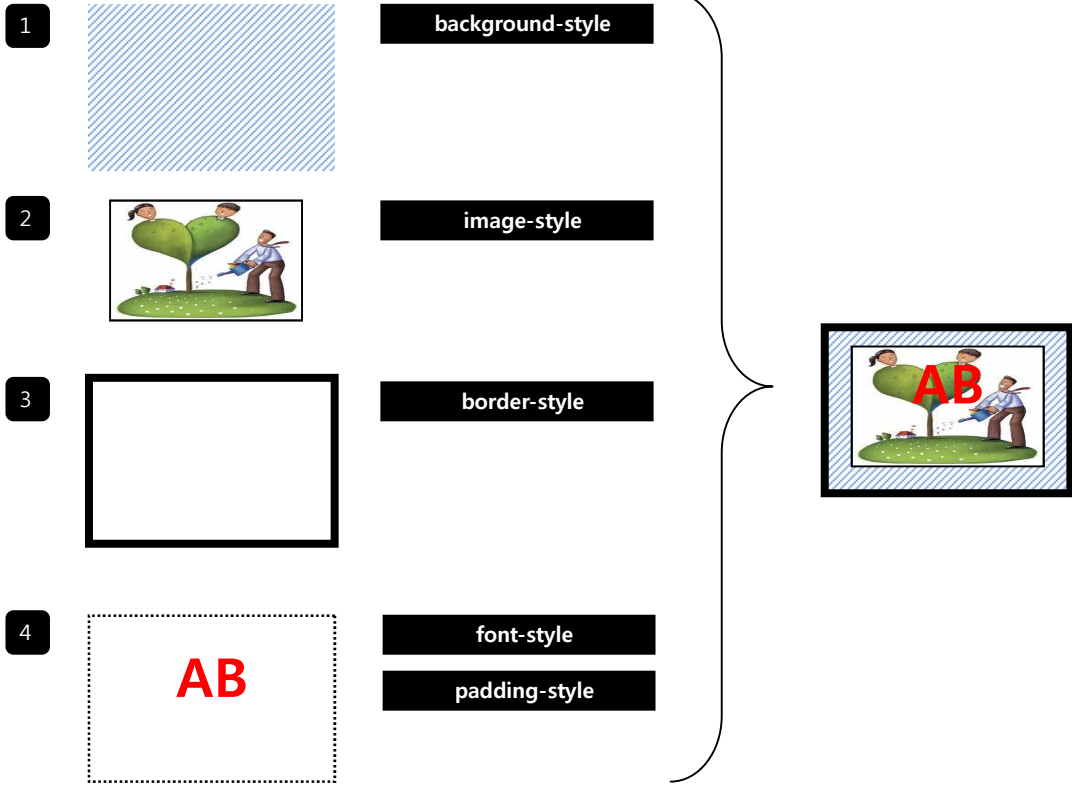
```

<?xml version="1.0" encoding="utf-8" ?>
<document type="" version="1.0">
<body>
  <local>
    <fields>
      <field-set field-type="database">
        <field name="name" />
        <field name="department" />
      </field-set>
    </fields>
  </local>
  <page>
    <section-set>
      <section>
        <control control-type="textbox" x1="0" y1="0" width="100" height="100"
          content-type="field" >![CDATA[database.name]]>
        </control>
      </section>
    </section-set>
  </page>
</body>
</document>
  
```

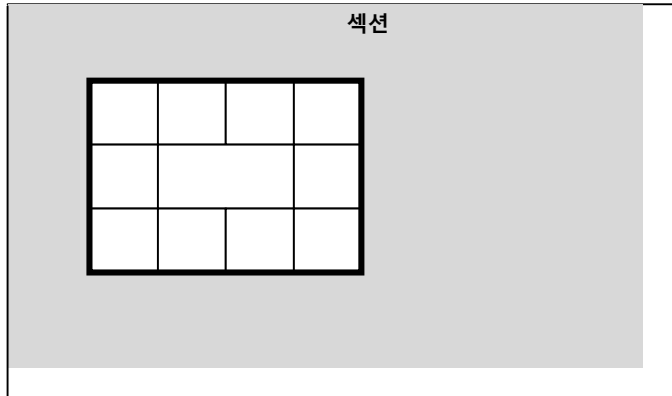
### 보임 스타일

요약 스타일

조건 스타일



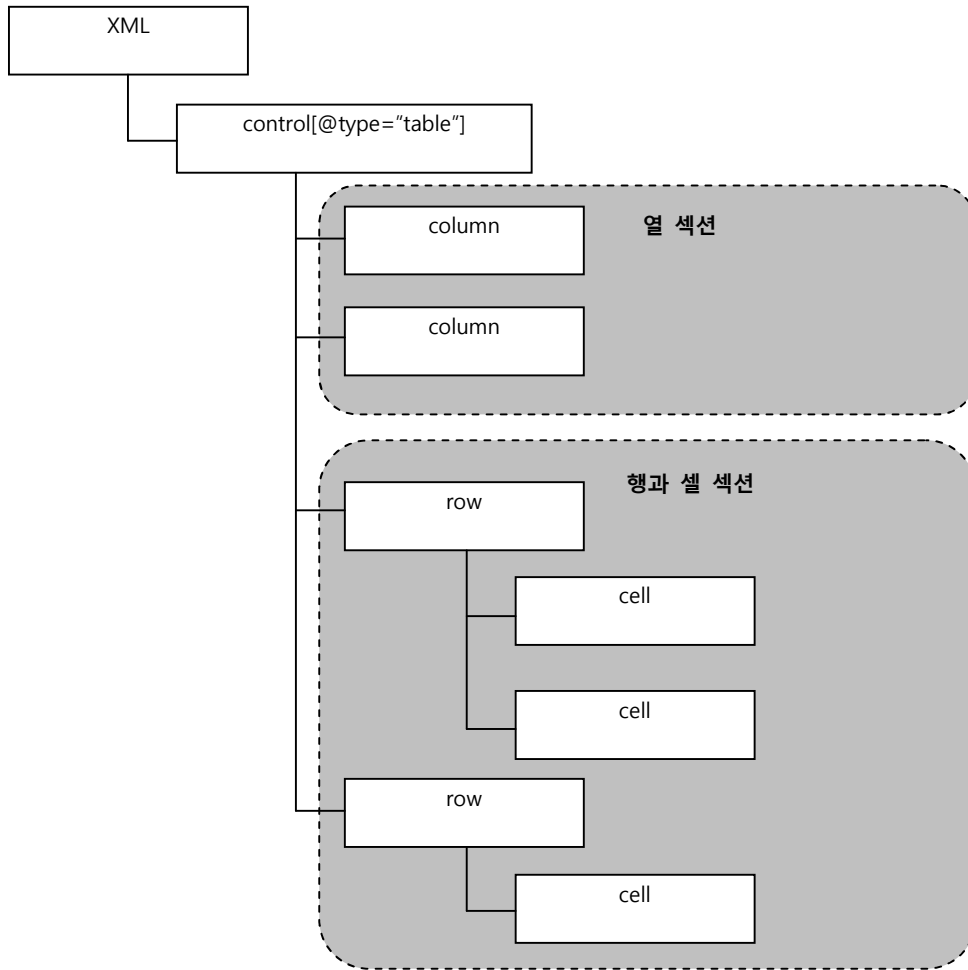
## 테이블(control[@type="table"])



테이블 컨트롤은 다른 컨트롤과 비교하면 매우 복잡한 컨트롤에 속한다. 테이블 컨트롤은 행(Row), 열(Column), 셀(Cell)로 구성된 복합 컨트롤이다. 테이블 컨트롤만 보면 행, 열, 셀을 포함하는 단순한 컨테이너(Container)에 불과하다. 글상자 컨트롤을 사용해서 보고서를 만들 수 있지만 일반적으로 테이블 컨트롤을 많이 사용한다. 테이블 컨트롤이 많이 사용되는 이유에는 직감적인 디자인이 가능하고 편리성에 있다. 특히, 한국처럼 장표 형식의 보고서를 만들 때 테이블 컨트롤의 사용은 거의 필수나 다름 없다. 테이블 컨트롤 없이 글상자 컨트롤만으로 보고서를 디자인하는 것은 반복적인 작업을 계속하는 지루하고도 힘든 작업이다. 테이블 컨트롤이 없는 외국 리포팅 툴을 써본 개발자라면 누구나 뼈저리게 느끼는 부분이다.

### 속성 목록

속성 목록				
속성	상수		버전	설명
x1			1.0	
y1			1.0	
visible-style				
condition-style				



## XML 정의



```

<?xml version="1.0" encoding="utf-8" ?>
<document type="" version="1.0">
<body>
  <page>
    <section-set>
      <section>
        <control control-type="table" visible-style="보임 1" condition-style="">
          <column width="100" />
          <column width="200" />
          <row height="100">
            <cell content-type="text" border-style="테두리 1"
              background-style="배경 1" font-style="폰트 1"
              image-style="" padding-style="여백 1"
              summary-style="" condition-style="">
              <![CDATA[셀 (1,1)]]></cell>
            <cell content-type="text" border-style="테두리 1"
              background-style="배경 1" font-style="폰트 1"
              image-style="" padding-style="여백 1"
              summary-style="" condition-style="">
              <![CDATA[ 셀 (1,2) ]]> </cell>
          </row>
          <row height="100">
            <cell content-type="field" border-style="테두리 1"
              background-style="배경 1" font-style="폰트 1"
              image-style="" padding-style="여백 1"
              summary-style="" condition-style="">
              <![CDATA[ database.name]]> </cell>
            <cell content-type="field" border-style="테두리 1"
              background-style="배경 1" font-style="폰트 1"
              image-style="" padding-style="여백 1"
              summary-style="" condition-style="">
              <![CDATA[database.phone]]></cell>
          </row>
        </control>
      </section>
    </section-set>
  </page>
</body>
</document>

```

## 테이블 셀

테이블 셀은 테이블 컨트롤의 핵심 요소이다. 앞서 설명했듯이 테이블 컨트롤 자체는 테이블 셀을 포함하는 컨테이너에 불과하며 텍스트를 출력하고 필드 값을 출력하는 것은 테이블 셀이다. 테이블 셀의 기능은 기본적으로 글상자 컨트롤과 같다. 글상자 컨트롤과 반복되는 설명이지만 테이블 셀의 기능을 요약하면 다음과 같다.

- 보고서 제목과 같은 텍스트를 출력한다.
- 데이터 베이스 및 매개 변수 필드의 내용을 출력한다.
- 이미지를 출력한다.
- 요약 값을 출력한다.

## 속성 목록

속성 목록				
속성	상수		버전	설명
content-type	text	0	1.0	셀의 내용을 일반 텍스트로 설정.
	field	1	1.0	셀의 내용을 필드로 설정.
border-style			1.0	셀의 테두리 스타일을 설정.
background-style			1.0	셀의 배경 스타일을 설정.
font-style			1.0	셀의 폰트 스타일을 설정.
image-style			1.0	셀의 이미지 스타일을 설정.
padding-style			1.0	셀의 여백 스타일을 설정.
summary-style			1.0	셀의 요약 스타일을 설정.
condition-style			1..0	셀의 조건 스타일을 설정.

## XML 정의

```

<?xml version="1.0" encoding="utf-8" ?>
<document type="" version="1.0">
<body>
  <page>
    <section-set>
      <section>
        <control control-type="table" visible-style="보임 1" condition-style="">
          <column width="100" />
          <column width="200" />
          <row height="100">
            <cell content-type="text" border-style="테두리 1"
              background-style="배경 1" font-style="폰트 1"
              image-style="" padding-style="여백 1"
              summary-style="" condition-style="">
              <![CDATA[셀 (1,1)]]></cell>
            <cell content-type="field" border-style="테두리 1"
              background-style="배경 1" font-style="폰트 1"
              image-style="" padding-style="여백 1"
              summary-style="" condition-style="">
              <![CDATA[database.name]]></cell>
          </row>
        </control>
      </section>
    </section-set>
  </page>
</body>
</document>

```

## 콘텐츠 타입(control[@content-type])

테이블 셀의 가장 큰 특징은 글상자 컨트롤과 마찬가지로 텍스트를 출력하는 데 있다. 텍스트에는 보고서의 타이틀이나 각 항목의 타이틀과 같이 고정 텍스트가 될 수도 있고 데이터 베이스나 매개 변수 필드의 값이 될 수도 있다. 즉, 글상자 컨트롤이 고정 텍스트를 출력할 것인지 아니면 특정 필드의 값을 출력할 것인지 구분할 필요가 있는데 콘텐츠 타입 속성이 이런 역할을 하는 속성이다. 콘텐츠 타입 속성이 "문자"로 설정됐으면 CDATA 섹션의 내용은 일반 텍스트가 된다. 콘텐츠 타입 속성이 "필드"로 설정됐으면 필드 값을 출력하라는 의미이므로 CDATA 섹션의 내용은 필드 이름이 된다. 글상자 컨트롤은 해당 필드 값을 레코드에서 읽어 화면에 출력하면 된다. 간단하다.

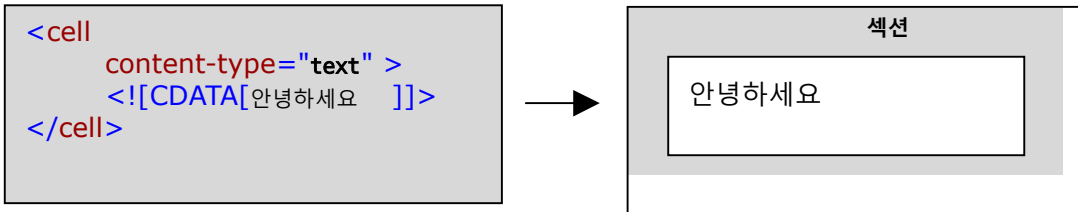
```

<cell content-type="text" >
  <![CDATA[안녕하세요 ]]>
</control>

```

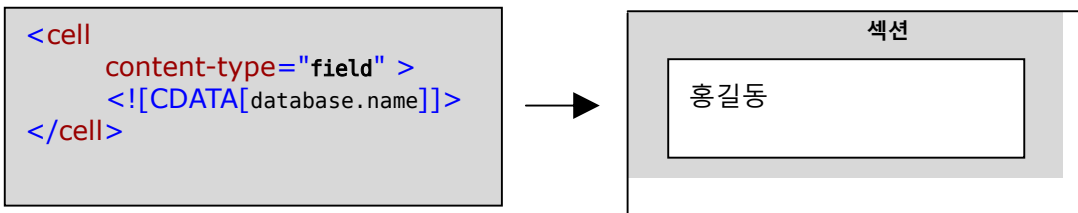
## 콘텐츠 타입(control[@content-type="text"])

콘텐츠 타입 "문자(text)"는 CDATA 섹션의 내용이 고정 텍스트임을 나타낸다. 글상자 컨트롤은 CDATA 섹션이 포함된 텍스트를 그대로 출력하면 된다.



### 콘텐츠 타입(control[@content-type="field"])

콘텐츠 타입 "필드(field)"는 CDATA 섹션의 내용이 필드 이름임을 나타낸다. 글상자 컨트롤은 CDATA 섹션이 포함된 필드 이름의 값을 레코드로부터 읽어 출력하면 된다. 해당 콘텐츠 타입은 "문자" 타입 처럼 고정적이지 않고 레코드마다 내용이 동적으로 바뀐다. 글상자 컨트롤이 본문 섹션에 위치하고 콘텐츠 타입이 필드라면 매번 다른 데이터를 출력한다.



CDATA 섹션에 필드 이름을 명시하는 방법은 아래처럼 필드 타입과 필드 이름을 묶은 형태이다. 필드 이름은 사용자가 임의로 작성 가능하지만 필드 타입은 아래처럼 표처럼 영문으로 이미 정의되어 있다. 개발자는 참조하려는 필드의 타입이 무엇인지 명확히 알고 있어야 한다.

```
<![CDATA[필드타입.필드이름]]>
```

필드			
필드 타입 한글명	필드 타입 영문명	필드 이름(예)	예시
매개 변수 필드	parameter	number	parameter.number
시스템 필드	system	pagenumber	system.pagenumber
데이터 베이스 필드	database	firstname	database.firstname
공식 필드	formula	fullname	formula.fullname

```
<?xml version="1.0" encoding="utf-8" ?>
<document type="" version="1.0">
<body>
  <local>
    <fields>
      <field-set field-type="database">
        <field name="name" />
        <field name="department" />
      </field-set>
    </fields>
  </local>
  <page>
    <section-set>
      <section>
        <control control-type="textbox" x1="0" y1="0" width="100" height="100"
          content-type="field" ><![CDATA[database.name]]>
        </control>
      </section>
    </section-set>
  </page>
</body>
</document>
```