

---

# rMate Grid 사용 설명서

**Version 3.3**



---

정품을 구입하신 고객에게는 기술상담 및 지원을 제공합니다  
(Tel : 02-2655-9767, [riamore@riamore.net](mailto:riamore@riamore.net))

**(주)리아모어소프트**

# 목 차

1. 개요 .....	4
1.1. rMate Grid 의 주요 특징.....	4
1.2. 시스템 요구사항.....	4
1.3. 그리드 시스템 구성 환경 설명.....	4
2. 기본적인 rMate Grid 생성 .....	5
2.1. rMate Grid 라이선스 등록하기.....	5
2.2. 로컬 환경에서 테스트하기 위한 준비 사항 .....	6
2.3. 제공된 샘플을 토대로 그리드 생성하기 .....	7
3. 그리드 데이터 형식 .....	9
3.1. XML 형식의 데이터 작성 및 그리드에 적용하기.....	9
3.2. JSON 형식의 데이터 작성 및 그리드에 적용하기 .....	11
3.3. 자바스크립트 배열 형식의 데이터 작성 및 그리드에 삽입하기 .....	12
3.4. 날짜 형식의 자료 .....	15
4. 그리드 레이아웃 .....	15
4.1. rMate Grid 에서 레이아웃의 역할 .....	15
4.2. 레이아웃 작성 방법에 대하여.....	16
4.3. 레이아웃의 rMateGrid 노드 설정하기 .....	17
4.4. 레이아웃의 UI 요소 설정하기 .....	18
4.5. 레이아웃의 Style 노드(CSS 적용) 설정하기 .....	18
5. 그리드의 기본 구조 및 용어설명 .....	21
5.1. 그리드 기본구조 설명 .....	21
5.2. 용어 설명.....	21
6. 그리드 설정 및 자바스크립트와의 연동 .....	23
6.1. flashVars 설정 .....	23
6.2. FABridge 에 대하여 .....	24
6.3. 연동을 위한 기본 설정 및 변수 .....	25
6.4. rMate Grid 컴포넌트의 속성 연동 .....	26
6.5. rMate Grid 컴포넌트의 스타일 연동 .....	27
6.6. rMate Grid 컴포넌트의 이벤트 연동 .....	28
7. 에디팅.....	29
7.1. 자바스크립트를 통한 데이터 수정.....	29
7.2. 그리드의 편집기능을 통한 데이터 수정 .....	32
8. 행, 셀 속성 제어.....	34
8.1. 행 속성 .....	35
8.2. 셀 속성 .....	35

8.3.	자바스크립트를 이용한 행, 셀 속성지정 .....	36
8.4.	데이터에 행, 셀 속성지정 .....	38
9.	컴포넌트별 속성, 스타일, 함수 및 이벤트 .....	42
9.1.	GridApp .....	42
9.2.	GridRoot .....	44
9.3.	DataGrid .....	50
9.4.	DataGridColumnGroup .....	61
9.5.	DataGridColumn .....	63
9.6.	DataGridSelectorColumn .....	69
9.7.	DataGridFooter .....	72
9.8.	DataGridFooterColumn .....	73
9.9.	CurrencyFormatter .....	75
9.10.	DateFormatter .....	76
9.11.	NumberFormatter .....	78
9.12.	NumberMaskFormatter .....	79
9.13.	PercentFormatter .....	79
9.14.	ContextMenu .....	80
9.15.	ContextMenuItem .....	81
9.16.	SpanCellAttribute .....	82
9.17.	SpanRowAttribute .....	84
9.18.	SpanArrayCollection .....	84
9.19.	SpanXMLListCollection .....	88
9.20.	SpanSummaryCollection .....	92
9.21.	SpanMergingField .....	95
9.22.	SpanSummaryRow .....	96
9.23.	SpanSummaryField .....	97
9.24.	HierarchicalData .....	98
9.25.	GroupingCollection .....	98
9.26.	Grouping .....	100
9.27.	GroupingField .....	100
9.28.	SummaryRow .....	101
9.29.	SummaryField .....	101
9.30.	ArrayCollection .....	102
9.31.	XMLListCollection .....	103
9.32.	Box .....	104
9.33.	Label .....	105

## 1. 개요

### 1.1. rMate Grid 의 주요 특징

rMate Grid 는 Adobe® Flex™ 제품을 사용해 개발된 어플리케이션으로 사용자에게 데이터를 시각적으로 알기 쉽게 제시할 수 있는 솔루션을 제공합니다.

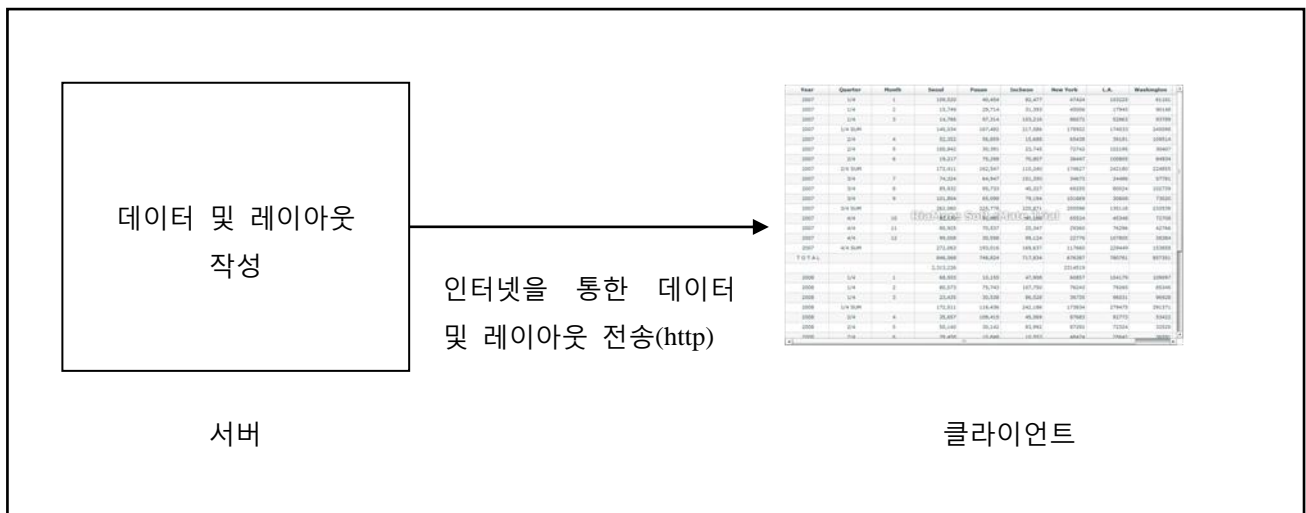
HTML, .NET, JSP, PHP, ASP, ColdFusion 등과 같은 웹 스크립트 언어에 구애 받지 않고, 그리드를 생성할 수 있습니다.

<http://www.riamore.net> 의 데모 메뉴에서 강력하고 쉬운 그리드를 직접 체험하실 수 있습니다.

### 1.2. 시스템 요구사항

- 서버 : rMate Grid 는 톰캣, IIS, 웹로직, 웹스피어, 제우스, LCDS 등등 모든 WAS 서버에서 작동하며, 웹 스크립트 언어에 의존적이지 않습니다.
- 클라이언트 : **Adobe Flash Player 9 버전 이상이 설치된 웹 브라우저 환경**이 필요합니다.  
(단, 엑셀 export, import 는 10 버전 이상에서 가능합니다.)

### 1.3. 그리드 시스템 구성 환경 설명



<그림 1 서버와 클라이언트 간의 구성도>

rMate Grid 를 생성하기 위해서는 그리드 레이아웃과 그리드에 표현하고자 하는 데이터가 필요합니다. 그리드 레이아웃은 그리드를 어떻게 표현할지를 나타내는 서식을 결정하게 됩니다. 예를 들면 그리드의 형태, 셀의 색상 및 병합 유무 등등 세밀한 부분을 제어하는 XML 파일입니다.

rMate 그리드 데이터 연동은 다음 5 가지 형태를 지원합니다.

- URL 을 호출하여 XML 데이터를 받아 처리하는 형태
- URL 을 호출하여 JSON 데이터를 받아 처리하는 형태
- XML 문자열을 자바스크립트를 통해 전달하는 형태
- JSON 문자열을 자바스크립트를 통해 전달하는 형태
- 자바스크립트에서 배열을 선언하고 값을 넣어 array 객체를 전달하는 형태.

이에 대한 자세한 설명은 제공된 CD 의 샘플을 실행시켜 다음 항목에서 확인할 수 있습니다.

주어진 CD의 index.html 실행 → Grid Type 선택 → 데이터 연동 선택

## 2. 기본적인 rMate Grid 생성

### 2.1. rMate Grid 라이선스 등록하기

정상적인 그리드를 생성하기 위해서는 rMate Grid 라이선스를 등록하셔야 합니다. 제공된 시디의 다음경로에 있는 파일이 그리드에 삽입할 라이선스입니다.

- /LicenseKey/rMateGridLicense.js

위 자바스크립트 파일을 <head> 태그 안에 삽입하시면 됩니다. 다른 작업을 하실 필요는 없습니다.

```

<html>
<head>

...

...

...

<!--rMate Grid 라이선스 등록 완료 모습 -->
<script src="rMateGridLicense.js" language="javascript"></script>

<!-- 사용자 정의 설정 시작 -->
<script language="JavaScript" type="text/javascript">

...

...

...
  
```

<예제 1 rMate Grid 라이선스 등록 예제>

## 2.2. 로컬 환경에서 테스트하기 위한 준비 사항

rMate 그리드를 Local 환경에서 테스트하기 위해서는 다음 작업이 선행되어야 합니다. 이는 Adobe Flash Player 보안정책에 해당됩니다. (서버에 올리셔서 사용하는 경우에는 아래작업은 불필요합니다)

Step1. 아래 해당되는 경로를 따라 폴더로 이동하십시오.

- **Windows XP:** Application Data\Macromedia\Flash Player\#Security\FlashPlayerTrust  
 (for example, C:\Documents and Settings\User1\Application Data\Macromedia\Flash Player\#Security\FlashPlayerTrust)
- **Vista :**  
 appData\Roaming\Macromedia\FlashPlayer\#Security\FlashPlayerTrust  
 (for example, C:\Users\User1\AppData\Roaming\Macromedia\Flash Player\#Security\FlashPlayerTrust)
- **Windows7:**  
 C:\Users\User1\AppData\Roaming\Macromedia\Flash Player\#Security\FlashPlayerTrust

**\* Application Data** 폴더는 숨겨진 폴더입니다. 탐색기 메뉴 중 도구-> 폴더옵션-> 보기-> 숨긴 파일 및 폴더 표시를 체크하십시오.

Step2. FlashPlayerTrust 폴더 안에 **rMateTrust.cfg** 라는 텍스트 형식의 파일을 만들고 메모장을 이용하여 편집하기 위해 여십시오.

Step3. 제공된 rMate 그리드 저장 폴더 경로를 rMateTrust.cfg 에 넣어주십시오.

예를 들어 제공된 샘플의 저장 위치가 C:\Users\User1\Desktop\WrMateGrid 입니다.

위 경로를 그대로 rMateTrust.cfg 파일에 추가하세요. (필요에 따라 여러줄이 들어가도 무방하며, 다른 rMate 제품의 경로가 들어가 있어도 됩니다)

최종적인 결과는 다음과 같습니다.

C:\Users\User1\Desktop\WrMateGrid

<예제 2 rMateTrust.cfg 파일의 내용>

위 과정은 현재 등록된 폴더 안의 플래시 파일들은 신뢰할 수 있다는 의미입니다.

### 2.3. 제공된 샘플을 토대로 그리드 생성하기

그리드를 생성하기 위해서는 제공된 CD 의 다음 경로를 따라 각각의 파일을 복사하여 작업 디렉토리에 복사하십시오.

- 데이터 파일 Samples/DataXml/DataOneDepthRevenues.xml
- 레이아웃 파일 : Samples/LayoutXml/LayoutDataGridProperty.xml
- 그리드 swf 파일 : Component/rMateGridWeb.swf
- 자바스크립트 파일 Samples/JS/AC\_OETags.js  
Samples/JS/rMateGrid.js
- 라이선스 파일 : Samples/ rMateGridLicense.js

위 파일을 다음과 같이 html 파일에 설정합니다.

```
<html>
<head>
<title>rMate Flash Grid</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<script type="text/javascript" src="./JS/AC_OETags.js"></script>
<script type="text/javascript" src="./JS/rMateGrid.js"></script>
<script type="text/javascript" src="./JS/rMateGridLicense.js"></script>

<!-- 사용자 정의 설정 시작 -->
<script type="text/javascript">

// ----- rMateGrid flashVars 설정 시작 -----

// 그리드 레이아웃 URL 경로. 반드시 설정하십시오.
var layoutURL = encodeURIComponent("./LayoutXml/LayoutDataGridProperty.xml");
var flashVars = "layoutURL="+layoutURL;

// 데이터를 URL 경로를 통해 가져올 경우 설정하십시오.
// 배열형태로 데이터를 삽입할 경우 주석처리나 삭제하십시오.
// 배열형태와 같이 사용할 경우, 우선순위는 배열형태 데이터 삽입입니다.
var dataURL = encodeURIComponent("./DataXml/DataOneDepthRevenues.xml");
flashVars += "&dataURL="+dataURL;

// 데이터가 JSON형일 경우 넣어줍니다. (설정을 안할경우 xml로 인식하여 오류가 발생합니다)
//flashVars += "&dataType=json";

// ----- rMateGrid flashVars 설정 끝 -----

</script>
<!-- 사용자 정의 설정 끝 -->
</head>

<body onLoad="rMateGridInit()">
```

```
<table border="0" cellpadding="0" cellspacing="0" width="100%" height="100%">
<tr>
<td>
<script type="text/javascript">
<!--
// 파라미터 설명입니다.
// 1. ID(임의로 정하십시오)
// 2. rMateGrid 컴포넌트 swf 파일 URL(확장자는 생략하십시오)
// 3. 컴포넌트 환경 변수
// 4. 가로 사이즈
// 5. 세로 사이즈
// 6. rMateGrid 로딩 전 배경화면색
rMateGridCreate("grid1", "../Component/rMateGridWeb", flashVars, "100%",
"100%", "#FFFFFF");
// -->
</script>
</td>
</tr>
</table>
</body>
</html>
```

<예제 3 HTML 페이지에 그리드 삽입하기>

위 처럼 정의 한 후 저장 후 실행시키면 다음과 같은 화면을 볼 수 있습니다.

Year	Quarter	Month	Seoul	Pusan	Incheon	NewYork	LA	Washington
2007	1/4	1	109520	40454	82477	47424	103225	61161
2007	1/4	2	15749	29714	31393	45006	17945	90148
2007	1/4	3	14766	97314	103216	86072	52863	93789
2007	2/4	4	52352	56859	15688	65438	39181	109514
2007	2/4	5	100842	30391	23745	72742	102195	30407
2007	2/4	6	19217	75298	70807	36447	100805	84934
2007	3/4	7	74324	64947	101350	34673	24486	57781
2007	3/4	8	85932	95733	40327	69255	80024	102739
2007	3/4	9	101804	65098	79194	101669	30608	73020
2007	4/4	10	92130	91881	45166	65524	45348	72708
2007	4/4	11	80925	70537	25347	29360	76296	42766
2007	4/4	12	99008	30598	99124	22776	107805	38384
2008	1/4	1	68503	10155	47908	60857	104179	109097
2008	1/4	2	80573	75743	107750	76243	79265	85345
2008	1/4	3	23435	30538	86528	36735	96031	96928
2008	2/4	4	35657	109415	45569	87683	92773	53422
2008	2/4	5	50140	30142	83992	87292	72324	32520
2008	2/4	6	39458	10848	10553	48474	25642	36591

<그림 2 HTML 페이지에 그리드 삽입 성공 화면>

rMate 그리드와 스크립트간의 동기화 확인을 위한 함수인 rMateGridInit() 는 반드시 처음 스크립트가 로드 될 때 호출되어야 합니다. 제공된 모든 샘플은 html 문서 안의 body 태그에 onload 함수를 이용한 <body onload="rMateGridInit()"> 로 호출됩니다.



onload 시 호출해야 할 함수가 더 있을 경우 아래를 참고하십시오.

```
<html>
  <head>          //헤더 부분에 아래와 같이 window.onload 삽입하시고
  ...             // onload 할 때 호출해야 할 함수를 넣으십시오.
  ...
  <script type="text/javascript">
    window.onload = function ()
    {
      rMateGridInit();
      .....
    }
  </script>
</head>
...
<html>
```

<예제 4 rMateGridInit() 동기화 예제>

### 3. 그리드 데이터 형식

rMate Grid 의 수치 데이터는 **XML 형식과 JSON 형식, 자바스크립트 배열 형식**을 지원합니다. 사용자는 몇 가지 규칙에 따라 그리드 데이터를 작성할 필요가 있습니다. 규칙에 의거하여 데이터 XML 형식과 JSON 형식, 배열형식으로 각각 변환하여 어떻게 그리드에 삽입하는지를 이번 장에서 설명합니다.

#### 3.1. XML 형식의 데이터 작성 및 그리드에 적용하기

XML 형식으로 데이터를 작성 할 때 반드시 준수해야 할 점은 맨 처음의 XML root 가 존재해야 한다는 점 입니다. 따라서 rMate Grid root 밑의 children 을 가지고 데이터로 인식하여 출력하게 됩니다. 데이터 XML 파일이 여러 root 를 가지게 되면 그리드는 데이터를 표현하지 않습니다.

매출(Revenue), 비용(Cost), 이윤(Profit)을 바탕으로 월간 보고서를 그리드로 표현하기 위해 데이터를 작성해 보도록 하겠습니다.

월(Month)	매출(Revenue)	비용(Cost)	이윤(Profit)
Jan.	10,000	5,000	5,000
Feb.	15,000	7,000	8,000
Mar.	12,000	6,000	6,000
Apr.	30,200	4,000	26,200

May.	28,000	10,000	18,000
Jun.	12,000	5,000	7,000
Jul.	22,000	10000	12,000
Aug.	13,000	6,000	7,000
Sep	22,000	10,000	12,000
Oct.	29,000	8,000	11,000
Nov.	18,000	7,500	10,500
Dec.	30,000	12,000	28,000

### <XML 형식의 데이터>

```

<Sales>
  <item>
    <Month>Jan</Month>
    <Revenue>10000</ Revenue >
    <Cost>5000</Cost>
    <Profit>5000</Profit>
  </item>
  <item>
    <Month>Feb</Month>
    <Revenue>15000</Revenue>
    <Cost>7000</Cost>
    <Profit>8000</Profit>
  </item>
    :
    :
  <item>
    <Month>Dec</Month>
    <Revenue>30000</Revenue>
    <Cost>12000</Cost>
    <Profit>18000</Profit>
  </item>
</Sales>
  
```

### <예제 5 XML 형식으로 데이터 변환>

위와 같이 데이터를 XML 유형으로 표현할 수 있습니다.

예를 들어 위에 작성한 XML 데이터를 "singleData.xml" 로 저장하였다면 그리드에 singleData.xml 의 경로를 넘겨줘야 합니다. 이에 대한 예제는 <예제 3 HTML 페이지에 그리드 삽입하기> 를 참고하세요.

\* XML 로 데이터를 작성할 경우 소팅시 숫자에 의한 소팅이 아닌 문자열에 의한 소팅이 되므로 숫자 컬럼에 대한 소팅기능을 사용해야 할 경우에는 컬럼의 sortCompareFunction 에 numericSort 를 설정해 줘야 합니다.(샘플 -> 컬럼정렬(XML) 참조)

### 3.2. JSON 형식의 데이터 작성 및 그리드에 적용하기

JSON 형식으로 데이터를 작성 할 때 파일의 형식은 <http://www.json.org/> 에서 정의한 형식을 반드시 준수해야 하며 데이터 JSON 파일이 형식에 안 맞을 경우에는 그리드는 데이터를 표현하지 않습니다.

위의 3.1 에서 정의한 매출(Revenue), 비용(Cost), 이윤(Profit)을 바탕으로 월간 보고서를 그리드로 표현하기 위해 데이터를 작성해 보도록 하겠습니다.

#### <JSON 형식의 데이터>

```
[
  {
    "Month": "Jan",
    "Revenue": 10000,
    "Cost": 5000,
    "Profit": 5000
  },
  {
    "Month": "Feb",
    "Revenue": 15000,
    "Cost": 7000,
    "Profit": 8000
  },
  .
  .
  .
  {
    "Month": "Dec",
    "Revenue": 30000,
    "Cost": 12000,
    "Profit": 18000
  }
]
```

<예제 6 JSON 형식으로 데이터 변환>

위와 같이 데이터를 JSON 유형으로 표현할 수 있습니다.

### 3.3. 자바스크립트 배열 형식의 데이터 작성 및 그리드에 삽입하기

rMate Grid 는 사용자의 편의를 위하여 배열 형식의 데이터를 그리드에 삽입할 수 있도록 하였습니다. 그 방법에 대한 설명입니다.

매월 이윤에 대해 자바스크립트에서 배열 형식으로 데이터를 작성해 보도록 하겠습니다.

월(Month)	이윤(Profit)
Jan.	10,000
Feb.	15,000
Mar.	12,000
Apr.	30,200
May.	28,000
Jun.	12,000
Jul.	22,000
Aug.	13,000
Sep	22,000
Oct.	29,000
Nov.	18,000
Dec.	30,000

```

var gridData = [{"Month": "Jan", "Profit": 10000},
  {"Month": "Feb", "Profit": 15000},
  {"Month": "Mar", "Profit": 12000},
  {"Month": "Apr", "Profit": 30200},
  {"Month": "May", "Profit": 28000},
  {"Month": "Jun", "Profit": 12000},
  {"Month": "Jul", "Profit": 22000},
  {"Month": "Aug", "Profit": 13000},
  {"Month": "Sep", "Profit": 22000},
  {"Month": "Oct", "Profit": 29000},
  {"Month": "Nov", "Profit": 18000},
  {"Month": "Dec", "Profit": 30000} ];

```

<예제 7 배열형식으로 데이터 변환>

위와 같이 배열 형식으로 변환할 수 있습니다.

3.1 장에서 작성된 매출(Revenue), 비용(Cost), 이윤(Profit)을 바탕으로 작성된 다중 데이터를 배열형식으로 변환하면 다음과 같습니다.

```

var gridData = [{"Month": "Jan", "Revenue": 10000, "Cost": 5000, "Profit": 5000},
  {"Month": "Feb", "Revenue": 15000, "Cost": 7000, "Profit": 8000},
  {"Month": "Mar", "Revenue": 12000, "Cost": 6000, "Profit": 6000},
  {"Month": "Apr", "Revenue": 30200, "Cost": 4000, "Profit": 26200},
  {"Month": "May", "Revenue": 28000, "Cost": 10000, "Profit": 18000},
  {"Month": "Jun", "Revenue": 12000, "Cost": 5000, "Profit": 7000},
  {"Month": "Jul", "Revenue": 22000, "Cost": 10000, "Profit": 12000},

```

```
{ "Month": "Aug", "Revenue": 13000, "Cost": 6000, "Profit": 7000 },
{ "Month": "Sep", "Revenue": 22000, "Cost": 10000, "Profit": 12000 },
{ "Month": "Oct", "Revenue": 29000, "Cost": 8000, "Profit": 21000 },
{ "Month": "Nov", "Revenue": 18000, "Cost": 7500, "Profit": 10500 },
{ "Month": "Dec", "Revenue": 30000, "Cost": 12000, "Profit": 18000 } ];
```

#### <예제 8 배열형식으로 다중 데이터 변환>

배열형식의 데이터를 그리드에 삽입하는 방법은 아래와 같습니다.

```
<html>
<head>
<script src="AC_OETags.js" language="javascript"></script>
<script src="rMateGrid.js" language="javascript"></script>
<script src="rMateGridLicense.js" language="javascript"></script>

<!-- 사용자 정의 설정 시작 -->
<script language="JavaScript" type="text/javascript">

// ----- rMateGrid flashVars 설정 시작-----

// rMateGrid 레이아웃 URL 경로. 반드시 설정하십시오.
var layoutURL = encodeURIComponent("LayoutSimple.xml");
var flashVars = "layoutURL="+layoutURL;

// 배열형태로 데이터를 삽입할 경우 주석처리나 삭제하십시오. 1

// ----- rMateGrid flashVars 설정 끝 -----

// rMateGrid 관련 객체
var gridApp; // 플래시 기본 객체

// rMateGrid가 로딩된 후 불러지는 함수 3
var rMateGridOnLoad = function() {
    gridApp = FABridge.gridBridge.root();

    gridApp.setData(gridData); 5
}

// rMateGrid가 로딩된 후 불러질 함수를 설정합니다.
FABridge.addInitializationCallback("gridBridge", rMateGridOnLoad); 2

var gridData = [{"Month": "Jan", "Profit": 10000},
                {"Month": "Feb", "Profit": 15000}, 4
                {"Month": "Mar", "Profit": 12000},
                {"Month": "Apr", "Profit": 30200},
                {"Month": "May", "Profit": 28000},
                {"Month": "Jun", "Profit": 12000},
                {"Month": "Jul", "Profit": 22000},
                {"Month": "Aug", "Profit": 13000},
```

```

        {"Month": "Sep", "Profit": 22000},
        {"Month": "Oct", "Profit": 29000},
        {"Month": "Nov", "Profit": 18000},
        {"Month": "Dec", "Profit": 30000} ];

<!-- 사용자 정의 설정 끝 -->
</script>
</head>

<body onload="rMateGridInit()">
<table>
    <tr>
    <td>
        <script>
        <!--
            // rMateGrid를 생성하고자 하는 위치에서 다음과 같이 함수 호출하십시오.
            // 파라미터 설명 (순서대로)
            // 1. ID (임의로 정의하십시오)
            // 2. swf 파일 URL (확장자는 생략)
            // 3. URL 경로 등을 정의한 변수명.
            // 4. rMateGrid 가로 사이즈.
            // 5. rMateGrid 세로 사이즈.
            // 6. rMateGrid 배경색.

                rMateGridCreate("grid1", "../Component/rMateGridWeb",
flashVars, "100%", "100%", "#FFFFFF")
            // -->
        </script>
    </td>
</tr>
</table>

</body>
</html>

```

#### <예제 9 배열형식으로 데이터 삽입하는 HTML 문서>

위의 그리드를 삽입한 html 문서를 보면 배열로 데이터를 삽입하기 위해 다음 과정을 밟아야 합니다.

첫째로, XML 경로를 통한 데이터 삽입부분을 주석처리 하거나 삭제하십시오.

둘째, rMate Grid 와 스크립트간 동기화가 완료된 후 rMate Grid 가 호출할 함수를 지시하십시오. 이는 **FABridge** 라는 자바스크립트 변수를 통하여 지시합니다. 지시법은 아래와 같습니다.

```

FABridge.addInitializationCallback("gridBridge", rMateGridOnLoad);
                          변경 불가          사용자에 의해 정의

```

그리드와 웹브라우저 동기화가 완료된 후 사용자가 정의한 함수(rMateGridOnLoad 으로 설정한 함수)를 호출합니다.

셋째, FABridge.addInitializationCallback 에 할당된 함수 명에 맞는 함수를 직접 정의하십시오.

위 예에서 함수명은 **"rMateGridOnLoad"** 이므로 **function rMateGridOnLoad()** 를 정의하였습니다.

넷째, 그리드 데이터를 배열 형태로 작성해 주십시오. 작성방법은 자바 스크립트에서 일반적으로 생성하는 배열형식과 동일합니다.

다섯째, `function rMateGridOnLoad()` 함수를 통해 그리드에 데이터를 배열로 삽입하십시오. 위 예에서 `gridApp.setData(gridData);` 부분입니다. 그리드에 데이터를 배열형태로 삽입하기 위해서 그리드에 `setData` 라는 함수를 만들어 놓았습니다. 이 함수는 배열형태 객체, XML 문자열, JSON 문자열을 파라미터로 받을수 있으며, 전달된 파라미터의 형식과 `flashVars` 에 설정된 값을 비교하여 XML, JSON,array 여부를 판별하여 데이터를 설정합니다

### 3.4. 날짜 형식의 자료

데이터의 형식에 관계없이 날짜 데이터를 넣는 경우에는 문자열로 놓여지며, 날짜 객체로 변환시 오류가 발생할 수 있습니다.

컴포넌트중 날짜와 관련된 `DataFormatter` 나 `DateEditor` 등은 전달된 데이터를 날짜형식으로 변환합니다. 이때 날짜의 형식은 반드시 아래 형식중의 하나여야 정상적으로 Date 객체로 변환되며, 형식 오류가 있을 경우 기능에 오류가 발생할 수 있습니다.

```
MM/DD/YYYY HH:MM:SS TZD
HH:MM:SS TZD Day Mon/DD/YYYY
Mon DD YYYY HH:MM:SS TZD
Day Mon DD HH:MM:SS TZD YYYY
Day DD Mon HH:MM:SS TZD YYYY
Mon/DD/YYYY HH:MM:SS TZD
YYYY/MM/DD HH:MM:SS TZD
```

TZD(시간대 지정)는 항상 GMT-HHMM 또는 UTC-HHMM 형식입니다. 이는 GMT(그리니치 표준시)를 기준으로 한 시간과 분 오프셋을 나타냅니다. GMT 를 현재는 UTC(표준시)라고도 합니다. 년, 월, 일을 나타내는 부분은 슬래시(/)나 공백을 사용하여 구분할 수 있지만 대시(-)는 사용할 수 없습니다.

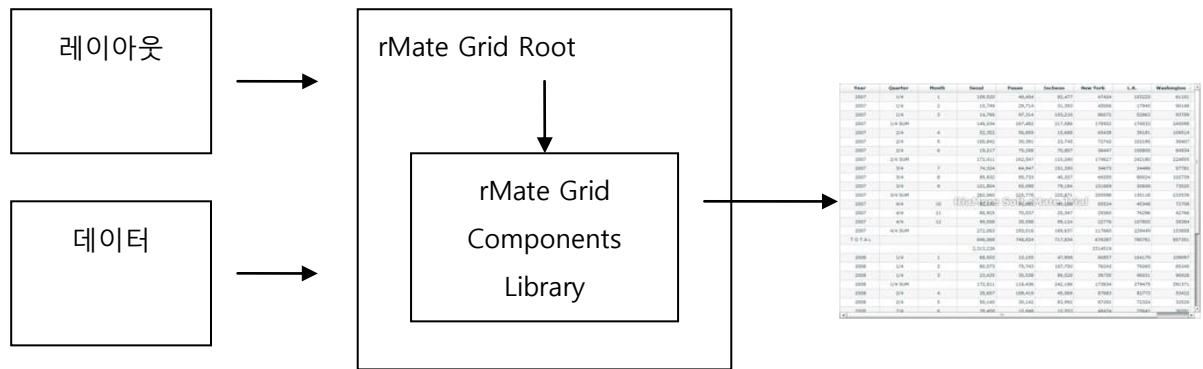
따라서 데이터에 날짜를 넣을 경우에는 YYYY/MM/DD 형식을 권장합니다.

## 4. 그리드 레이아웃

### 4.1. rMate Grid 에서 레이아웃의 역할

rMate Grid 는 미리 작성된 레이아웃의 XML 을 파싱하여 동적으로 생성됩니다. 사용자가 작성한 레이아웃을 rMate Grid 의 swf 파일에 입력하면 그리드는 미리 내포하고 있는 rMate Grid Component 클래스를 기반으로 레이아웃에 작성된 그리드를 생성하고 데이터를 출력하게 됩니다.

다음은 이런 과정을 다이어그램으로 표현한 것입니다.



<그림 3 rMate Grid 동작 다이어그램>

## 4.2. 레이아웃 작성 방법에 대하여

rMate Grid 의 모든 레이아웃은 <rMateGrid>태그로 시작하여 </rMateGrid>태그로 끝납니다. 레이아웃은 크게 그리드 UI 요소와 그리드, Style 세 부분으로 구분됩니다. 그리드 UI 요소는 옵션적으로 필요시 정의되는 요소이며, Style 또한 그리드의 스타일 제어를 위해 필요시 정의 하면 됩니다.

레이아웃을 작성하는 방법은 기본적으로 XML 형식입니다.

rMate Grid 는 ID 에 의한 객체 접근을 지원합니다. 예를 들어 DataGridView 객체의 속성으로 formatter 가 존재합니다. 이 formatter 속성의 속성값으로 그리드 UI 요소에서 정의한 NumberFormatter 인스턴스를 할당합니다. 이를 표현한 일반적인 방법은 아래와 같습니다.

```
<rMateGrid>
  <DataGrid>
    .
    .
    .
    <DataGridColumn dataField="Profit" textAlign="right">
      <formatter>
        <NumberFormatter useThousandsSeparator="true"/>
      </formatter>
    </DataGridColumn>
  </DataGrid>
</rMateGrid>
```

<예제 10 일반적인 속성 값 할당 법>

그러나 사용자의 편의를 위해 <예제 11 ID 에 의한 속성 값 할당 법>과 같이 NumberFormatter 인스턴스를 미리 생성한 후 그 ID 를 통해 NumberFormatter 인스턴스를 활용할 수 있습니다. 이는 여러 객체들이 하나의 NumberFormatter 인스턴스에 접근하고자 할 때 불필요하게 NumberFormatter



인스턴스를 생성 하지 않고 미리 만들어진 하나의 NumberFormatter 에 접근하기 때문에 효율적입니다. 또한 어느 특정 객체를 반드시 참조해야 하는 경우에도 효율적입니다.

```
<rMateGrid>
  <NumberFormatter id="numfmt" useThousandsSeparator="true"/>
  <DataGrid>
    .
    .
    .
    //객체 ID인 경우엔 시작과 끝을 중괄호({,})로 묶음
    <DataGridColumn dataField="Profit" textAlign="right" formatter="{numfmt}"/>
  </DataGrid>
</rMateGrid>
```

<예제 11 ID 에 의한 속성 값 할당 법>

### 4.3. 레이아웃의 rMateGrid 노드 설정하기

rMateGrid 노드는 레이아웃의 시작과 끝을 나타냅니다. 또한 이 노드의 속성은 그리드 전체의 꾸미기를 담당합니다.

속성 명	유효 값	설명
backgroundColor	RGB(기본값: 0xFFFFFF)	바탕 색을 나타냅니다.
backgroundAlpha	0.0 ~ 1.0(기본값:1.0)	바탕 투명도를 나타냅니다.
borderStyle	none   solid   inset   outset (기본값:none)	테두리 선을 정의합니다.
borderColor	RGB(기본값: 0xB7BABC)	테두리 색을 정의합니다.
borderThickness	Number (기본값:1)	테두리 두께를 정의합니다.
cornerRadius	Number(기본값:0)	테두리의 둥글게 처리 정도를 나타냅니다.

<표 1 rMateGrid 노드 속성과 유효값 설명>

아래는 rMateGrid 노드의 속성 사용 예를 나타낸 것 입니다.

```
<rMateGrid cornerRadius="12" borderStyle="solid" backgroundColor="0xFFFF77">
  <DataGrid>
    .....
  </DataGrid>
</rMateGrid>
```

<예제 12 rMateGrid 노드의 속성 사용 예>

rMateGrid 에 대한 속성은 그리드의 바탕을 표현해야 하는 경우에만 사용하면 되며, 기본적으로 별도의 속성을 주실 필요는 없습니다.

#### 4.4. 레이아웃의 UI 요소 설정하기

그리드 UI 요소는 레이아웃의 필수사항이 아닌 선택사항인 컴포넌트를 선언하는 것을 뜻합니다. 그리드에 사용될 ContextMenu(오른쪽 마우스 클릭시 나타나는 메뉴), 숫자나 날짜를 표현하는데 쓰이는 포맷터, 셀이나 행의 속성을 정의하는데 쓰이는 속성정의 컴포넌트 (SpanCellAttribute, SpanRowAttribute)등을 정의하면 됩니다.

```
<rMateGrid>
    <ContextMenu id="cMenu">
        <ContextMenuItem id="cMenuInsert" caption="Insert Row"/>
        <ContextMenuItem id="cMenuDelete" caption="Delete Row"/>
    </ContextMenu>
    <NumberFormatter id="numfmt" useThousandsSeparator="true"/>
    <DataGrid id="dg1" contextMenu="{cMenu}" horizontalScrollPolicy="auto"
horizontalGridLines="true" horizontalGridLineColor="0xCCCCCC"
verticalAlign="middle" selectionMode="multipleRows">
        <columns>
            <DataGridColumn dataField="Selected"/>
            <DataGridColumn dataField="From"/>
            <DataGridColumn dataField="Subject" />
            <DataGridColumn dataField="Date"/>
            <DataGridColumn dataField="CC"/>
            <DataGridColumn dataField="Profit" textAlign="right"
formatter="{numfmt}"/>
        </columns>
    </DataGrid>
</rMateGrid>
```

<예제 13 그리드 UI 요소 사용 예>

#### 4.5. 레이아웃의 Style 노드(CSS 적용) 설정하기

<Style> 노드는 전반적인 스타일을 미리 정의하는 스타일시트이며, html 에서 사용하는 CSS 와 동일한 정의를 사용합니다. <Style/>을 선언하는 방법에는 몇 가지 규칙이 있습니다.

- 첫째, <Style/> 노드는 반드시 <rMateGrid> 노드의 자식 위치에 정의하십시오.
- 둘째, 스타일네임을 정의 할 때 시작은 도트(.)을 찍고 소문자로 시작하는 영문자로 표기하십시오.
- 셋째, 스타일네임을 정한 후 그에 따른 구체적인 스타일을 정의 할 때는 반드시 중괄호({,})로 시작과 끝을 표시하십시오.
- 넷째, 속성명과 속성값(value)의 구분은 콜론(:)이며 끝은 세미콜론으로 나타냅니다.

다음은 올바른 예와 잘못된 예를 나타낸 것입니다.

| 올바른 예                                           | 잘못 작성 된 예                                       |
|-------------------------------------------------|-------------------------------------------------|
| <pre>&lt;rMateGrid&gt;   &lt;DataGrid&gt;</pre> | <pre>&lt;rMateGrid&gt;   &lt;DataGrid&gt;</pre> |

<pre> ...  &lt;Style&gt;   .rMateGridStyle   {     backgroundColor:0xFFFFFE;     borderColor:0x77EE9E;     borderThickness:3;     borderStyle:solid;   } &lt;/Style&gt; &lt;/rMateGrid&gt; </pre>	<pre> ...  &lt;Style&gt;   GridStyle   [     backgroundColor:0xFFFFFE;     borderColor:"0x77EE9E";     borderThickness:3;     borderStyle:'solid';   ] &lt;/Style&gt; &lt;/rMateGrid&gt; </pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<표 2 Style 작성 예>

위와 같이 rMateGridStyle 스타일을 정의하였습니다. 그렇다면 이 스타일을 적용하는 방법에 대해 보도록 하겠습니다.

<pre> &lt;rMateGrid styleName="rMateGridStyle"&gt;   &lt;DataGrid&gt;    .....    &lt;Style&gt;     .rMateGridStyle     {       backgroundColor:0xFFFFFE;       borderColor:0x77EE9E;       cornerRadius:12;       borderThickness:3;       borderStyle:solid;     }   &lt;/Style&gt; &lt;/rMateGrid&gt; </pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<예제 14 Style 적용 예>

<예제 14 Style 적용 예>를 보시면 rMateGridStyle 은 <rMateGrid/> 노드의 스타일을 정의 한 것이고 적용법은 styleName="정의한 스타일이름" 즉, styleName="rMateGridStyle" 과 같습니다.

<Style/>에서 정의 가능한 각각의 속성은 이 문서의 <컴포넌트별 속성, 스타일, 함수 및 이벤트>에서 각 컴포넌트의 스타일 부분을 참조하시면 됩니다.

스타일을 활용한 예제와 출력 모습은 아래와 같습니다.

<pre> &lt;rMateGrid&gt;   &lt;DataGrid id="dg1" styleName="gridStyle"&gt;     &lt;columns&gt;       &lt;DataGridColumn id="dg1col1" dataField="Year"/&gt;       &lt;DataGridColumn id="dg1col2" dataField="Quarter"/&gt;       &lt;DataGridColumn id="dg1col3" dataField="Month"/&gt;       &lt;DataGridColumn id="dg1col4" dataField="Seoul"/&gt;     &lt;/columns&gt;   &lt;/DataGrid&gt; &lt;/rMateGrid&gt; </pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

```

        <DataGridColumn id="dg1col5" dataField="Pusan"/>
        <DataGridColumn id="dg1col6" dataField="Incheon"/>
        <DataGridColumn id="dg1col7" dataField="NewYork"/>
        <DataGridColumn id="dg1col8" dataField="LA"/>
        <DataGridColumn id="dg1col9" dataField="Washington"/>
        <DataGridColumn id="dg1col10" dataField="Revenue"/>
        <DataGridColumn id="dg1col11" dataField="Percent"/>
    </columns>
</DataGrid>
<Style>
    .gridStyle {
        color:0xFF0000;
        alternatingItemColors:0xFFCC33,0xFFFF33;
        headerColors:0x888888,0xFFFFFFFF;
        horizontalGridLines:true;
        horizontalGridLineColor:0xFF0000;
        fontWeight:bold;
        textDecoration:underline;
        fontSize:12;
        verticalAlign:middle;
        verticalGridLineColor:0x0000FF;
    }
</Style>
</rMateGrid>

```

<예제 15 Style 적용한 예제>

Year	Quarter	Month	Seoul	Pusan	Incheon	NewYork	LA	Washin...	Revenue	Percent
2007	1/4	1	109520	40454	82477	47424	103225	61161	444260	40
2007	1/4	2	15749	29714	31393	45006	17945	90148	229956	20
2007	1/4	3	14766	97314	103216	86072	52863	93789	448020	40
2007	2/4	4	52352	56859	15688	65438	39181	109514	339031	31
2007	2/4	5	100842	30391	23745	72742	102195	30407	360322	33
2007	2/4	6	19217	75298	70807	36447	100805	84934	387508	36
2007	3/4	7	74324	64947	101350	34673	24486	57781	357561	28
2007	3/4	8	85932	95733	40327	69255	80024	102739	474011	37
2007	3/4	9	101804	65098	79194	101669	30608	73020	451393	35
2007	4/4	10	92130	91881	45166	65524	45348	72708	412757	36
2007	4/4	11	80925	70337	25347	29860	76296	42766	325230	29
2007	4/4	12	99008	30598	99124	22776	107805	38384	397696	35
2008	1/4	1	68503	10155	47908	60857	104179	109097	400699	31
2008	1/4	2	80573	75743	107750	76243	79265	85345	504918	40
2008	1/4	3	23435	30538	86528	36735	96031	96928	370196	29
2008	2/4	4	35657	109415	45569	87683	92773	53422	424520	45
2008	2/4	5	50140	30142	83992	87292	72324	32520	356410	37
2008	2/4	6	39458	10848	10553	48474	25642	36591	171565	18
2008	3/4	7	33761	49046	31351	46829	97148	42630	300765	31
2008	3/4	8	89645	72565	23678	78847	62559	87722	415017	42
2008	3/4	9	14844	30709	83037	23130	65006	48367	265093	27
2008	4/4	10	30598	55523	90576	79997	71346	63569	391608	34

<그림 4 Style 적용 결과>

<예제 15 Style 적용 예>에서 alternatingItemColors 같이 색을 배열로 넣을 경우에는 <Style/> 안에서는 "[와 "]" 필요없이 ","로만 연결하여 표시하면 됩니다. 하지만 <DataGrid alternatingItemColors="[0xFFCC33,0xFFFF33]"> 와 같이 콤포넌트 태그에서 처리하실 경우에는 "[와 "]"를 붙여서 배열처리를 해줘야 합니다.

## 5. 그리드의 기본 구조 및 용어설명

### 5.1. 그리드 기본구조 설명

Year	Quarter	Month	Region				
			Korea			U.S.A.	
			Seoul	Pusan	Incheon	New York	L.A.
2007	1/4	1	109,520	40,454	82,477	47424	10322
2007	1/4	2	15,749	29,714	31,393	45006	1794
2007	1/4	3	14,766	97,314	103,216	86072	5286
2007	2/4	4	52,352	56,859	15,688	65438	3918
2007	2/4	5	100,842	30,391	23,745	72742	10219
2007	2/4	6	19217	75,298	70,807	36447	10080
2007	3/4	7	74,324	64,947	101,350	34673	2448
2007	3/4	8	85,932	95,733	40,327	69255	8002
2007	3/4	9	101,804	65,098	79,194	101669	3060
2007	4/4	10	92,130	91,881	45,166	65524	4534
2007	4/4	11	80,925	70,537	25,347	29360	7629
2007	4/4	12	99,008	30,598	99,124	22776	10780
2008	1/4	1	68,503	10,155	47,908	60857	10417
2008	1/4	2	80,573	75,743	107,750	76243	7926
2008	1/4	3	23,435	30,538	86,528	36735	9603
2008	2/4	4	35,657	109,415	45,569	87683	9277
2008	2/4	5	50,140	30,142	83,992	87292	7232
2008	2/4	6	39,458	10,848	10,553	48474	2564
2008	3/4	7	33,761	49,046	31,351	46829	9714
2008	3/4	8	89,645	72,565	23,678	78847	6255

<그림 5 그리드 기본 구조 설명>

기본 구조는 그리드 안에 컬럼이 존재하며 컬럼안에는 헤더 렌더러로 구성된 헤더부분과 아이템 렌더러로 구성된 셀이 있고 셀이 합쳐져 행을 이루는 데이터 부분으로 구성됩니다.

필요에 따라 컬럼을 여러 합쳐 그룹 컬럼을 만들 수 있습니다.

레이아웃에 의해 설명한다면 <DataGrid> 노드에 의해 그리드가 만들어 지고 <DataGridColumn> 노드로 컬럼에 대한 정의를 설정하면, 그리드에서는 컬럼정보에 의해 헤더렌더러를 만들어 헤더부분을 채우고 데이터를 받아들여 각 컬럼정보에 따라 아이템렌더러로 개별 행들을 만들어 컬럼헤더 밑에 붙여 나가게 되는 것입니다.

### 5.2. 용어 설명

- 그리드(DataGrid)

그리드 또는 데이터그리드로 행,열을 표시하는 컴포넌트.

레이아웃에서 <DataGrid> 노드에서 정의하여 생겨나는 그리드 객체로 rMate Grid 의 가장 중심

컴포넌트입니다.

- 컬럼(column)

그리드의 개별 열을 뜻합니다.

레이아웃에서 컬럼을 정의하는 것으로 <DataGridColumn> 또는 <DataGridSelectorColumn>이 있으며, 각 컴포넌트는 실제 컬럼의 내용을 포함하거나 작업하는 것은 아니고 컬럼에 대한 정보만을 가지고 있게 됩니다.

- 컬럼그룹(ColumnGroup)

개별 컬럼을 모아 계층구조를 나타내기 위한 컬럼

실제 컬럼과는 달리 하위에 반드시 컬럼을 포함해야 하며 헤더를 표시하는 속성과 관련된 기능을 가집니다.

레이아웃에서 <DataGridColumnGroup>노드로 정의하며, <DataGrid>의 groupedColumns 에 지정해 줍니다.

- 헤더(Header)

그리드에서 맨위의 컬럼정보를 나타내는 부분을 말합니다.

- 헤더 렌더러(headerRenderer)

그리드에서 헤더를 표현하기 위해 생성하는 컴포넌트로 해당 컬럼의 headerRenderer 속성에서 사용할 컴포넌트명을 명시하며 그리드에 의해 생성 관리됩니다.

headerRenderer 에 명시를 안하면 일반 텍스트가 표시되는 헤더 렌더러가 사용되고, 그외에 HtmlHeader, ComboBoxHeader 가 있습니다.

- 아이템 렌더러(itemRenderer)

그리드에서 셀을 표현하기 위해 생성하는 컴포넌트로 해당 컬럼의 itemRenderer 속성에서 사용할 컴포넌트명을 명시하며, 그리드에 의해 생성 관리됩니다.

itemRenderer 에 명시를 안하면 일반 텍스트가 표시되는 아이템 렌더러가 사용되고, 그외에는 HtmlItem, IconItem, ImageItem 이 있습니다.

- 아이템 에디터(itemEditor)

그리드에서 셀을 편집하기 위해 생성하는 컴포넌트로 해당 컬럼의 itemEditor 속성에서 사용할 컴포넌트명을 명시하며, 그리드에 의해 생성 관리됩니다.

itemEditor 에 명시를 안하면 일반 텍스트를 에디팅하는 아이템 에디터가 사용되고, 그외에는 CheckBoxEditor, ComboBoxEditor, DateEditor 가 있습니다.

- 셀렉터(selector)

그리드에서 행의 선택을 표현하기 위해 쓰이는 컴포넌트로 컬럼과 동일한 레벨에서 정의하고 사용되지만 차이점은 데이터의 특정 필드와 연결되지는 않는다는 점입니다.

- 컬렉션(collection)

데이터를 저장하는 컴포넌트들로 배열과 XML 등을 담을 수 있습니다.

사용자가 넣은 데이터는 rMate Grid 내에서 컬렉션 객체로 변환되는데 변환후에는 데이터에 대한 조작을 할 수 있는 함수 및 이벤트를 지원하는 기능을 제공하게 됩니다. 따라서 데이터에 조작은 반드시 컬렉션 객체를 통해 이루어져야 합니다.

rMate Grid 에서 제공되는 컬렉션은 ArrayCollection, XMLListCollection, SpanArrayCollection,

SpanXMLListCollection, SpanSummaryCollection, HierachicalData, GroupingCollection 이 있으며 레이아웃에서 dataProvider 에 특별히 컬렉션을 정의하지 않은 경우 데이터의 종류에 따라 배열이면 ArrayCollection, XML 이면 XMLListCollection 이 사용됩니다.

## 6. 그리드 설정 및 자바스크립트와의 연동

플래시로 작성된 rMate Grid 와 웹브라우저와의 연동을 위하여 플래시 속성인 flashVars 와 어도브에서 제공하는 FABridge 모듈을 사용합니다. 이의 사용법 및 기본 변수에 대해서 설명하겠습니다.

### 6.1. flashVars 설정

flashVars 변수는 rMate Grid 의 데이터, 레이아웃 등을 설정할 수 있는 가장 기본적인 변수입니다. 이 변수는 스크립트 단에서 그리드를 임베딩할 때 함께 파라미터로 함께 삽입하게 됩니다.

flashVars 변수 설정은 다음과 같은 규칙을 따릅니다.

- flashVars 변수는 스트링입니다.
- 두 개 이상의 설정을 하기 위해 구분자 & 를 사용합니다. 예를 들어 데이터 XML URL 과 레이아웃 XML URL 을 설정하는 방법은 아래와 같습니다

```
var layoutURL = encodeURIComponent("./Layout.xml");
var flashVars = "layoutURL="+layoutURL;

var dataURL =encodeURIComponent("./singleData.xml");
flashVars += "&dataURL="+dataURL;
```

<예제 16 flashVars 적용 예제>

flashVars 로 설정 가능한 속성값은 다음과 같습니다.

1. 그리드 레이아웃 XML URL 경로
2. 그리드 데이터 XML URL 경로
3. EUC-KR 로 인코딩 된 레이아웃의 한글지원

```
flashVars += "&flash.system.System.useCodePage=true";
```

4. 읽어들이는 데이터의 유형이 JSON 형식일 경우

```
flashVars += "&dataType=json";
```

5. useExternalInterface 사용여부 (자바스크립트연동 여부 - XPlatform 연동이 사용)

```
flashVars += "&useExternalInterface=false";
```

## 6. 다국어 설정

```
flashVars += "&localeChain=en_US";
```

위와 같이 flashVars 를 설정한 후 실제로 그리드를 생성하는 함수에 설정한 flashVars 를 파라미터로 넣어줍니다.

```
rMateGridCreate("grid1", "../Component/rMateGridWeb", flashVars, "100%", "100%", "#FFFFFF");
```

이때 rMateGridCreate 함수의 파라미터는 다음과 같습니다. (순서대로)

1. ID (임의로 정의하십시오)
2. swf 파일 URL(확장자는 생략)
3. URL 경로 등을 정의한 flashVars 변수.
4. rMateGrid 가로 사이즈.
5. rMateGrid 세로 사이즈.
6. rMateGrid 배경색.
7. wmode(안 넣을 경우 "window"이며 "transparent"나 "opaque"가 가능)

## 6.2. FABridge 에 대하여

- FABridge(Flex-Ajax Bridge)는 어도브에서 제공하는 브라우저의 자바스크립트와의 통신을 위한 기능을 제공하는 모듈입니다. 플래시에서 제공하는 기본적인 연동방식은 ExternalInterface 방식이 입습니다만 제약사항이 많고 코딩에 불편한 점이 많아 어도브에서 새로 제공하는 연동방식이 FABridge 인 것입니다.
- FABridge 를 사용하면 ActionScript 로 작성된 플래시내의 컴포넌트에 대해 속성값의 지정 및 조회, 함수 호출, 스타일 지정등이 가능하며, 컴포넌트에서 발생하는 이벤트를 자바스크립트 함수와 연동시키는 것이 가능합니다. 그러나 언어간의 차이때문에 Number, Boolean, String, Array 등의 기본적인 요소만 서로 교환이 가능하며, 파생된 개체에 대한 작업은 제약이 따릅니다.
- FABridge 에서는 여러 swf 와의 연동도 지원하며, 이를 지원하기 위해 개별 swf 의 flashVars 속성을 통해 서로를 구분할 수 있도록 bridgeName 을 설정하는 기능을 제공합니다.  
한 페이지내에 한 개의 그리드만이 존재하는 경우에는 bridgeName 은 기본적으로 "gridBridge" 로 지정되므로 신경 쓸 필요가 없지만, 여러 개의 그리드가 존재할 경우에는 개별 swf 를 선언시에 flashVars 속성을 지정할 때 &bridgeName=??? 형식으로 개별 swf 의 bridgeName 을 지정하고 자바스크립트에서 swf 를 불러올 때 지정한 bridgeName 을 활용하여 swf 에 접근하도록 해야 합니다. 이에 대한 예제는 제공되는 CD 의 샘플중 "여러 그리드 생성" 메뉴에서 예시하고 있으니 참고하시기 바랍니다.
- FABridge 와 연동되어 실행되는 자바스크립트 함수는 오류시에도 브라우저의 자바스크립트 오류 표시로 나타나지 않습니다. 이를 표시하기 위해 rMateGrid.js 에 rMateGridAlertFABridgeError 변수를 두고 true 이면 오류시 alert 로 해당 오류를 표시하도록 하였습니다. 만일 오류 감지 기능을



없애려면 해당 변수의 값을 false 로 조정하시기 바랍니다.

- FABridge 에 대한 자세한 사항은

[http://www.adobe.com/kr/devnet/flex/articles/framework\\_beta\\_fabridge.html](http://www.adobe.com/kr/devnet/flex/articles/framework_beta_fabridge.html) 를 통해 조회해 보실 수 있습니다.

### 6.3. 연동을 위한 기본 설정 및 변수

자바스크립트 연동을 위해 기본적으로 사용하는 변수와 사용 예를 보면 다음과 같습니다.

```
<html>
<head>
<script src="AC_OETags.js" language="javascript"></script>
<script src="rMateGrid.js" language="javascript"></script>
<script src="rMateGridLicense.js" language="javascript"></script>

<!-- 사용자 정의 설정 시작 -->
<script language="JavaScript" type="text/javascript">

// ----- rMateGrid flashVars 설정 시작-----

// rMateGrid 레이아웃 URL 경로. 반드시 설정하십시오.
var layoutURL = encodeURIComponent("LayoutSimple.xml");
var flashVars = "layoutURL="+layoutURL;

// 데이터를 URL 경로를 통해 가져올 경우 설정하십시오.
// 배열형태로 데이터를 삽입할 경우 주석처리나 삭제하십시오.
// 배열형태와 같이 사용할 경우, 우선순위는 배열형태 데이터 삽입입니다.
var dataURL = encodeURIComponent("./DataXml/DataOneDepthRevenues.xml");
flashVars += "&dataURL="+dataURL;

// ----- rMateGrid flashVars 설정 끝 -----

// rMateGrid 관련 객체
var gridApp; // 플래시 기본 객체
var gridRoot; // 데이터와 그리드를 포함하는 객체
var dataGrid; // 그리드

// rMateGrid가 로딩된 후 불려지는 함수
var rMateGridOnLoad = function() {
    gridApp = FABridge.gridBridge.root();
    gridRoot = gridApp.getGridRoot();

    var layoutCompleteHandler = function(event) {
        dataGrid = gridApp.getDataGrid();
    }

    gridRoot.addEventListener("layoutComplete", layoutCompleteHandler);
}

// rMateGrid가 로딩된 후 불려질 함수를 설정합니다.
FABridge.addInitializationCallback("gridBridge", rMateGridOnLoad);
```

1

3

4

2

```

</script>
<!-- 사용자 정의 설정 끝 -->
</head>

<body onload="rMateGridInit()">
<table>
  <tr>
    <td>
      <script>
      <!--
        rMateGridCreate("grid1", "../Component/rMateGridWeb",
flashVars, "100%", "100%", "#FFFFFF")
      // -->
      </script>
    </td>
  </tr>
</table>

</body>
</html>

```

<예제 17 자바스크립트와 연동 변수 예>

1. 자바스크립트를 통해 rMate Grid 에서 접근하는 기본 객체로 3 개를 선언하였습니다.
  - gridApp : 플래시 객체로 라이선스 검사 및 자바스크립트와 연동을 매개합니다.
  - gridRoot : 그리드와 데이터를 포함하는 객체로 레이아웃과 데이터를 설정한 내용에 따라 로딩하고 그리드를 생성하는 객체입니다. 플래시가 로딩된 후 데이터를 읽어 들이게 되면 "dataComplete" 이벤트가 발생하고 layout 에 의해 그리드가 만들어 지면 "layoutComplete" 이벤트가 발생하게 됩니다.
  - dataGrid : 설정된 레이아웃에 따라 그리드가 생성되면 넣어지는 변수
2. rMate Grid 와 스크립트간 동기화가 완료된 후 rMate Grid 가 호출할 함수를 설정합니다. 이는 FABridge 라는 자바스크립트 변수를 통하여 지시합니다. 위의 예에서는 로딩 완료 후 rMateGridOnLoad 함수를 호출하도록 지정하였습니다.
3. rMateGridOnLoad 함수에서 1 번에서 선언한 변수에 객체를 세팅합니다.
4. dataGrid 변수는 로딩과 동시에 만들어지는 것이 아니므로 이벤트를 통해 그리드가 만들어진 후에 세팅되도록 합니다. 또한 생성된 객체들에 이벤트를 걸어야 하는 경우 해당 객체에 적합한 이벤트를 찾아 연동할 함수와 함께 addEventListener 를 통하여 지정하도록 합니다. 객체별로 발생하는 이벤트는 아래 <컴포넌트별 속성, 스타일, 함수 및 이벤트>에서 이벤트 부분을 찾아 보시면 됩니다.

## 6.4. rMate Grid 컴포넌트의 속성 연동

rMate Grid 의 컴포넌트에 접근할 때 컴포넌트의 속성값(Property)을 지정하거나 조회하는 방법을 설명하겠습니다.

자바스크립트에서 플래시내의 객체 속성에 접근하는 방법은 반드시 함수를 통해서 해야 합니다.

```
var selectionChangeHandler = function(event) {
    var rowIndex = event.getRowIndex();
    var columnIndex = event.getColumnIndex();
    var dataRow = gridRoot.getItemAt(rowIndex);
    inputForm.itemXMLData.value = dataRow;
    inputForm.rowIndex.value = rowIndex;
    var column = dataGrid.getColumns()[columnIndex];
    var dataField = column.getDataField();
}

function selectedUp() {
    inputForm.selectedIndex.value = inputForm.selectedIndex.value - 1;
    dataGrid.setSelectedIndex(inputForm.selectedIndex.value);
}
```

<예제 18 컴포넌트 속성 연동 예>

위의 예에서 event 의 속성으로 rowIndex 가 있으나 이의 값을 가져오는 방법은 직접 event.rowIndex 형식으로 하는 것이 아닌 event.getRowIndex() 함수를 호출하여 가져오는 것입니다. 이것은 일반적인 getter, setter 방식으로 하는 것을 의미하며, 속성에 값을 지정하는 경우에는 event.setRowIndex(value) 형식으로 작성하면 됩니다. 마찬가지로 dataGrid.getColumns() 는 dataGrid 객체의 columns 속성을 가져오는 뜻이 되며, 반환 객체가 배열이므로 배열에서 columnIndex 번째 객체를 가져오는 문장이 되겠습니다.

세번째 예는 dataGrid 의 selectedIndex 속성에 값을 지정하는 예로 setSelectedIndex()함수를 사용하여 값을 지정하고 있습니다.

## 6.5. rMate Grid 컴포넌트의 스타일 연동

rMate Grid 에서 스타일을 설정할 수 있는 방법은 레이아웃에서 개별 컴포넌트 선언시 스타일을 지정하는 방법과 <Style>에서 CSS 형식으로 스타일을 만들고 해당 CSS 를 원하는 컴포넌트에 styleName 에 의거하여 지정하는 방법, 자바스크립트를 통해 setStyle, getStyle 함수를 사용하여 지정하는 방법이 있습니다. 레이아웃에 의거한 스타일 설정방법은 4 장에서 설명하였으므로, 여기서는 자바스크립트를 이용하는 방법에 대해 설명하겠습니다.

자바스크립트에서 플래시 객체의 스타일을 지정하거나 조회하는 경우에는 객체의 setStyle, getStyle 함수를 사용합니다.

```
function changeStyle() {
    var paddingTop = dataGrid.getStyle("paddingTop");
    dataGrid.setStyle("paddingTop", paddingTop + 1);
}
```

<예제 19 컴포넌트 스타일 연동 예>

위의 예에서는 dataGrid 의 스타일 속성인 paddingTop 을 가져와서 1 을 더해 다시 지정하는 예 입니다.

getStyle 함수를 통해 paddingTop 의 값을 가져와서 setStyle 함수를 사용하여 값을 지정하고 있습니다. 여기서 주의할 점은 모든 객체가 스타일을 가지고 있는 것은 아니며, 스타일 속성이 무엇이 있는지를 확인해야 한다는 점입니다. 이를 확인하기 위해서는 아래 <컴포넌트별 속성, 스타일, 함수 및 이벤트>에서 원하는 객체의 함수에 getStyle, setStyle 함수의 존재 여부를 확인하고 조정할 수 있는 스타일을 찾아 조정하는 작업이 필요합니다.

만일 레이아웃에 스타일을 미리 지정해 놓았고 자바스크립트에서 해당 스타일을 적용하고자 한다면 styleName 속성을 이용하여 객체.setStyleName(스타일명) 함수를 사용하여 지정하면 됩니다.(단 해당 컴포넌트가 styleName 속성이 있어야 합니다)

## 6.6. rMate Grid 컴포넌트의 이벤트 연동

rMate Grid 내의 컴포넌트에 대해 이벤트 발생시에 자바스크립트 함수를 호출할 수 있도록 이벤트 연동을 할 수 있습니다.

```
// rMateGrid가 로딩된 후 불러지는 함수
var rMateGridOnLoad = function() {
    gridApp = FABridge.gridBridge.root();
    gridRoot = gridApp.getGridRoot();

    var selectionChangeHandler = function(event) {
        var rowIndex = event.getRowIndex();
        var columnIndex = event.getColumnIndex();
        inputForm.selectedIndex.value = rowIndex;
    }

    var layoutCompleteHandler = function(event) {
        dataGrid = gridApp.getDataGrid();
        dataGrid.addEventListener("change", selectionChangeHandler);
        // 미리 선택된 행을 설정
        dataGrid.setSelectedIndex(6);
    }

    gridRoot.addEventListener("layoutComplete", layoutCompleteHandler);
}

// rMateGrid가 로딩된 후 불러질 함수를 설정합니다.
FABridge.addInitializationCallback("gridBridge", rMateGridOnLoad);
```

<예제 20 컴포넌트 이벤트 연동 예>

위의 예에서는 dataGrid 에서 발생하는 "change" 이벤트(사용자가 마우스를 클릭하여 행이나 셀을 선택한 경우에 발생)에 대해 selectionChangeHandler 함수를 호출하도록 설정한 경우입니다.

이벤트가 발생하여 selectionChangeHandler 함수가 호출되면 발생한 이벤트의 내용이 event 파라미터로 전달되고 event 변수에서 상세한 내용을 받아 볼 수 있습니다.

rMate Grid 의 컴포넌트중 이벤트를 발행하는 객체와 객체별로 발생하는 이벤트의 종류, 이벤트 종류에 따른 이벤트 상세내역은 아래 <컴포넌트별 속성, 스타일, 함수 및 이벤트>에서 자세히 살펴보실 수 있으며, 컴포넌트에 addEventListener, removeEventListener 함수가 없는 컴포넌트는 이벤트를 받을 수 없는 객체입니다.

## 7. 에디팅

rMate Grid 에서 데이터를 에디팅하는 방법은 크게 2 종류로 자바스크립트를 이용하는 방법과 그리드내의 편집기능을 이용하여 수정하는 방법입니다.

(그리드의 데이터가 계층형일 경우에는 에디팅은 지원되지 않습니다)

### 7.1. 자바스크립트를 통한 데이터 수정

자바스크립트를 통해 데이터를 수정하기 위해 자료를 수정하는 예는 다음과 같습니다.

```
// rMateGrid 관련 객체
var gridApp;    // 플래시 기본 객체
var gridRoot;   // 데이터와 그리드를 포함하는 객체
var dataGrid;   // 그리드

// rMateGrid가 로딩된 후 불려지는 함수
var rMateGridOnLoad = function() {
    gridApp = FABridge.gridBridge.root();
    gridRoot = gridApp.getGridRoot();

    var selectionChangeHandler = function(event) {
        var rowIndex = event.getRowIndex();
        var columnIndex = event.getColumnIndex();
        var dataRow = gridRoot.getItemAt(rowIndex);
        inputForm.itemXMLData.value = dataRow;
        inputForm.rowIndex.value = rowIndex;
        var column = dataGrid.getColumns()[columnIndex];
        var dataField = column.getDataField();
    }

    var layoutCompleteHandler = function(event) {
```

```
        dataGrid = gridApp.getDataGrid();
        dataGrid.addEventListener("change", selectionChangeHandler);
    }

    gridRoot.addEventListener("layoutComplete", layoutCompleteHandler);
}

// rMateGrid가 로딩된 후 불러질 함수를 설정합니다.
FABridge.addInitializationCallback("gridBridge", rMateGridOnLoad);

// 전체 삭제
function removeAll() {
    gridRoot.removeAll();
}

// 행 삭제
function removeItemAt() {
    // 현재 선택된 행을 삭제 합니다.
    gridRoot.removeItemAt();
    // 원하는 행의 번호를 통해 행을 삭제합니다.
    // gridRoot.removeItemAt(inputForm.rowIndex.value);
}

// 행 수정
function setItemAt() {
    var item = inputForm.itemXMLData.value;
    gridRoot.setItemAt(item, inputForm.rowIndex.value);
}

// 행 추가
function addItemAt() {
    var item = inputForm.itemXMLData.value;
    gridRoot.addItemAt(item, inputForm.rowIndex.value);
}

// 맨끝에 행 추가
function addItem() {
    var item = inputForm.itemXMLData.value;
    gridRoot.addItemAt(item);
}
```

```

// 수정 작업 결과 가져오기 (그리드에서 작업된 입력,수정,삭제 내용을 가져옵니다)
// 데이터는 배열 형태로
//     idx: 행번호
//     job: 수행 작업 (I:입력, U:수정, D:삭제)
//     data: 행의 자료
// 를 가지고 있으며 삭제가 먼저 오도록 정렬되어 있습니다.
function getChangedData() {
    var changedData = gridRoot.getChangedData();
    if (changedData.length == 0)
        alert("변경된 자료가 없습니다");
    else {
        for (var i = 0; i < changedData.length; i++)

            alert("index:"+changedData[i].idx+"\n"+"job:"+changedData[i].job+"\n"+"data:"+changedData[i].
data);
    }
}

// 수정 작업 결과 XML로 가져오기 (그리드에서 작업된 입력,수정,삭제 내용을 XML로 가져옵니다)
// 데이터는 XML 형태로
//     idx: 행번호
//     job: 수행 작업 (I:입력, U:수정, D:삭제)
//     data: 행의 자료
// node를 가지고 있으며 삭제가 먼저 오도록 정렬되어 있습니다.
function getChangedDataXML() {
    var changedDataXML = gridRoot.getChangedDataXML();
    if (changedDataXML == null)
        alert("변경된 자료가 없습니다");
    else
        alert(changedDataXML);
}

```

<예제 21 자바스크립트를 이용한 데이터 수정 예>

위의 예에서는 사용자가 그리드에서 한 개의 행을 선택하면 자바스크립에서 선택된 행에 대한 정보(행번호 등)를 받아 선택된 행의 data 를 gridRoot.getItemAt 함수를 호출하여 읽어오고, 이후 사용자의 수정이 진행되면 gridRoot.setItemAt 함수를 사용하여 그리드의 data 를 수정하거나, gridRoot.removeItemAt 함수를 사용하여 해당 행을 삭제하는 기능을 설명하였습니다.

또한 전체삭제는 gridRoot.removeAll 함수를 호출하여 작업할 수 있으며, 새로운 행을 추가하는 경우에는

gridRoot.addItemAt 함수를 사용하여 원하는 행 또는 맨 마지막 행에 새로운 data(행의 index 번호 파라미터를 안주거나 -1 을 세팅)를 등록할 수 있습니다.

매번 변경사항 생길 때마다 서버와의 작업으로 서버의 데이터도 변경하는 것도 가능하나, 사용자가 변경 작업을 모두 처리하고 한꺼번에 수정사항을 모두 반영하는 경우를 위해, 그리드에서는 위의 함수 호출에 대해 자료를 저장하는 기능이 제공됩니다.

이런 기능을 제공하기 위해 gridRoot.getChangedData, gridRoot.getChangedDataXML 함수를 제공하는데 데이터가 배열일 경우와 XML 일 경우에 따라 불러 쓰면 되며 getChangedData 함수의 반환값은 다음 형식의 배열입니다.

idx : 행번호

job : 수행된 작업을 나타내며 다음 값을 가집니다.

I:등록, D:삭제, U:수정

data : 사용자가 등록 또는 수정한 내용(행의 모든 데이터가 저장됨)

getChangedDataXML 함수의 반환값은 위 데이터를 XML 로 표현하여 문자열로 반환하게 됩니다.

이때 배열의 순서는 사용자의 에디팅 순서이나 서버작업의 편의를 위해 job 이 "D"인 것 즉 삭제된 자료가 맨 먼저 나타나도록 하였습니다.

자바스크립트를 이용하여 행을 수정하는 방법은 그리드의 데이터가 배열이나 XML 이냐에 따라 약간 달라지는데, 이유는 개별 행의 자료를 넘겨 받아 처리하는 방식에서 차이가 나기때문입니다. 배열의 경우 전달받은 행의 data 는 자바스크립트에서 직접인식이 가능한 객체이기 때문에 일반적인 자바스크립트 코딩방식을 써서 조작이 쉬운 반면, XML 로 행의 data 를 받게되면 XML 문자열을 받은 것이기 때문에 데이터를 조작하기 위해서는 XML 을 파싱하고, 재생성하는 작업이 별도로 필요하게 됩니다. XML 의 파싱과정의 예는 샘플에서 "XML 데이터 파싱" 메뉴에서 간략히 코딩되어있으니 참고하시기 바랍니다.

행을 수정하는 방법외에 개별 필드를 수정하는 방법으로 gridRoot.getItemFieldAt 와 gridRoot.setItemFieldAt 함수가 제공됩니다. 행의 수정이 XML 여부에 따라 바뀌는 것과 달리 데이터의 형식에 구애되지 않으므로 편리하며 사용 예는 샘플에서 "필드 데이터 제어" 메뉴를 참고하시기 바랍니다.

## 7.2. 그리드의 편집기능을 통한 데이터 수정

그리드의 편집기능을 이용하기 위해서는 레이아웃을 구성할 때 다음과 같은 속성을 구성합니다.

```
<rMateGrid>
  <DataGrid id="dg1" editable="true" doubleClickEnabled="true"
    itemEditBeginningJsFunction="itemEditBeginningChecker"
    itemEditEndJsFunction="itemEditEndChecker"
    selectionMode="singleRow">
    <columns>
      <DataGridColumn dataField="Region" editable="false"/>
    </columns>
  </DataGrid>
</rMateGrid>
```



```

        <DataGridColumn dataField="Territory" width="150"/>
        <DataGridColumn dataField="Territory_Rep"
headerText="Territory Rep" width="150"/>
        <DataGridColumn dataField="Actual"/>
        <DataGridColumn dataField="Estimate"/>
        <DataGridColumn dataField="Real"/>
        <DataGridColumn dataField="Price"/>
    </columns>
</DataGrid>
</rMateGrid>

```

<예제 22 편집용 레이아웃 예>

위의 예에서 editable 속성은 그리드의 수정모드로 반드시 true 를 주어야하며, doubleClickEnabled 는 더블 클릭 가능여부로 더블클릭에 의거해서 셀 편집모드로 전환할 지를 지정할 수 있습니다.

itemEditBeginningJsFunction 는 자바스크립트 함수를 통해 해당 셀의 수정가능여부를 판별하기 위한 기능으로 사용자 특정 셀을 클릭 또는 더블클릭으로 수정 모드로 들어가려하면 itemEditBeginningJsFunctiond 에서 지정한 자바스크립 함수가 수행되어 결과 true 일 경우에는 셀 수정모드로 전환되고, false 일 경우에는 무반응이 됩니다.

itemEditEndJsFunction 는 자바스크립트 함수를 통해 셀의 수정 값의 반영여부를 판별하는 기능으로 사용자가 셀의 수정을 마치게 되면 itemEditEndJsFunction 에서 지정한 자바스크립트 함수가 수행되어 정상적인 값이 입력되었다면 null, 오류가 있을 경우에는 오류메세지를 반환하게 하여 수정내용의 반영여부를 조작할 수 있게 됩니다.

Region	Territory	Territory Rep	Actual	Estimate	Real	Price
Southwest	Arizona	Arizona	38865	40000	30000	3001
Southwest	Arizona	Arizona	29885	30000	30000	3002
Southwest	Central California	Central California	29134	30000	30000	3003
Southwest	Nevada	Nevada	52888	45000	30000	3004
Southwest	Northern California	Northern California	38805	40000	30000	3005
Southwest	Northern California	Northern California	55498	40000	30000	3006
Southwest	Southern California3	Southern California	44985	202		
Southwest	Southern California	Southern California	44913	45000	30000	3008
Nor	Arizona	Arizona	38865	40000	30000	3009
Nor	Arizona	Arizona	29885	30000	30000	3000
Nor	Central California	Central California	29134	30000	30000	3001
Nor	Nevada	Nevada	52888	45000	30000	3002
Nor	Northern California	Northern California	38805	40000	30000	3003
Nor	Northern California	Northern California	55498	40000	30000	3004
Nor	Southern California	Southern California	44985	45000	30000	3005
Nor	Southern California	Southern California	44913	45000	30000	3006
Southeast	Arizona	Arizona	38865	40000	30000	3007
Southeast	Arizona	Arizona	29885	30000	30000	3008
Southeast	Central California	Central California	29134	30000	30000	3009
Southeast	Nevada	Nevada	52888	45000	30000	3000

<그림 6 그리드 편집 오류 메세지표시>

또한 전체 그리드에 대해 수정모드로 하였어도 개별 컬럼에 대해 editable 속성을 주어 수정이 안되도록 지정하실 수 있습니다.

다음은 그리드에서 편집을 위해 제공되는 컴포넌트(아이템에디터)를 설명하겠습니다.

그리드에서 제공되는 아이템에디터에는 텍스트, 체크박스, 콤보박스, 날짜선택창등이 있으며 기본으로는 텍스트가 사용되고 필요시 지정할 수 있으며 레이아웃에서 다음과 같이 지정하시면 됩니다.

```
<rMateGrid>
  <DataGrid id="dg1" editable="true" doubleClickEnabled="true"
verticalAlign="middle">
    <columns>
      <DataGridColumn dataField="Selected" width="80"
itemEditor="CheckBoxEditor" editorDataField="selected"
labelJsFunction="convertYesNo"/>
      <DataGridColumn dataField="From" width="80"/>
      <DataGridColumn dataField="Subject" width="150"
itemEditor="ComboBoxEditor"
itemRendererDataProvider="[{ 'label': 'GMC-1
Release1' }, { 'label': 'GMC-1 Release2' }, { 'label': 'GMC-1 Release3' }, { 'label': 'GMC-1
Release4' }, { 'label': 'GMC-1 Release5' }, { 'label': 'GMC-1 Release6' } ]"/>
      <DataGridColumn dataField="Date" itemEditor="DateEditor"/>
      <DataGridColumn dataField="CC"/>
      <DataGridColumn dataField="Profit" textAlign="right"
formatter="{numfmt}"/>
    </columns>
  </DataGrid>
</rMateGrid>
```

<예제 23 편집용 컴포넌트 사용 예>

아이템에디터를 지정하지 않으면 일반 문자열을 넣을 수 있는 기능의 텍스트 아이템에디터가 사용되며 itemEditor 속성에 원하는 아이템에디터의 이름을 넣어주면 됩니다.

아이템에디터를 CheckBoxEditor 를 사용하실 경우에는 반드시 editorDataField 속성에 selected 를 넣어주셔야 합니다.

ComboBoxEditor 를 사용하실 경우, 콤보박스에 고정된 내용을 보여주는 경우 itemRendererDataProvider 속성을 이용하여 콤보박스의 데이터를 JSON 형식으로 넣어주면 되나, 행마다 내용이 바뀌는 경우에는 itemRendererDataProviderField 속성을 이용하여 데이터에 넣어진 특정 필드에서 데이터를 가져오도록 설정할 수 있습니다.

## 8. 행, 셀 속성 제어

rMate Grid 에서 행에 대해서 줄 수 있는 속성은 배경색, 스타일(폰트관련이나 정렬등), 포맷, 잠금(에디팅 불가), 행의높이이며, 셀에 대해서 줄 수 있는 속성은 배경색 스타일, 포맷, 병합정보이며, 행, 셀에 속성을 지정하는 방법은 함수를 이용하는 방법과 데이터에 속성정보를 넣어서 표현하는 방법이 있습니다.

주의할 점은 행, 셀속성을 지정할 수 있으려면 그리드에 주는 데이터의 형식이 행, 셀속성을 넣을 수 있는 컬렉션이어야 한다는 것입니다. 그리드 자체에는 행, 셀의 속성을 저장하는 기능은 없으며, 관련정보는 데이터를 포함하는 컬렉션에서 저장, 관리됩니다. rMate Grid 에서 이를 지원하는 컬렉션은

SpanArrayCollection, SpanXMLListCollection, SpanSummaryCollection 이 있습니다.

## 8.1. 행 속성

행 속성을 저장하는 컴포넌트는 SpanRowAttribute 로 행에 줄수 있는 속성은 다음과 같습니다.

속성 명	유효 값(기본값)	설명
backgroundColor	uint	행의 배경색 0xFFFFFFFF 형식으로 표현합니다.
editable	Boolean(true)	행 수정 가능여부 그리드의 editable속성에 의한 편집 모드에서 클릭이나 더블클릭시에도 에디팅모드로 변환되지 않도록 합니다.
formatString	String	마스크 패턴 패턴적용이 실패하는 경우에는 원래의 데이터가 그대로 표시됩니다. 위의 <표 6 SpanCellAttribute, SpanRowAttribute의 마스크 패턴>를 참조하세요
rowHeight	Number	행의 높이(픽셀단위)
styleName	String	행에 적용할 스타일명 레이아웃의 <Style>노드에서 선언된 스타일명을 적어줍니다.

<표 3 행에 지정할 수 있는 속성>

## 8.2. 셀 속성

셀 속성을 저장하는 컴포넌트는 SpanCellAttribute 로 행에 줄수 있는 속성은 다음과 같습니다.

속성 명	유효 값(기본값)	설명
backgroundColor	uint	셀의 배경색 0xFFFFFFFF 형식으로 표현합니다.
colNo	int	컬럼의 index번호
colSpan	int(1)	병합할 열의 수
formatString	String	마스크 패턴 패턴적용이 실패하는 경우에는 원래의 데이터가 그대로 표시됩니다. 아래 <표 6 SpanCellAttribute, SpanRowAttribute의 마스크 패턴>를 참조하세요

rowSpan	int(1)	병합할 행의 수
styleName	String	셀에 적용할 스타일명 레이아웃의 <Style>노드에서 선언된 스타일명을 적어줍니다.

<표 4 셀에 지정할 수 있는 속성>

### 8.3. 자바스크립트를 이용한 행, 셀 속성지정

자바스크립트에서 행, 셀에 속성을 지정하는 예는 다음과 같습니다.

```
// rMateGrid 관련 객체
var gridApp;    // 플래시 기본 객체
var gridRoot;   // 데이터와 그리드를 포함하는 객체

// rMateGrid가 로딩된 후 불러지는 함수
var rMateGridOnLoad = function() {
    gridApp = FABridge.gridBridge.root();
    gridRoot = gridApp.getGridRoot();

    var dataCompletedHandler = function(event) {
        setSpanAttributes();
    }

    gridRoot.addEventListener("dataComplete", dataCompletedHandler);
}

// rMateGrid가 로딩된 후 불러질 함수를 설정합니다.
FABridge.addInitializationCallback("gridBridge", rMateGridOnLoad);

var collection;    // 그리드의 데이터 객체

function setSpanAttributes() {
    if (collection == null)
        collection = gridRoot.getCollection();
    if (collection == null) {
        alert("collection 객체를 찾을 수 없습니다");
        return;
    }
}
```

```

    }

    collection.addRowAttributeDetailAt(4, "subTotalStyle", 0xDDDDDD, "#,##0원", true, 40);

    collection.addCellAttributeDetailAt(0, 0, 12, 1);
    collection.addCellAttributeDetailAt(12, 0, 12, 1, "subTotalStyle", 0xFFCC00);
    for (i = 0; i < collection.getLength(); i++) {
        if (i % 3 == 0)
            collection.addCellAttributeDetailAt(i, 1, 3, 1);
    }
}

function clearRowAttr() {
    if (collection == null)
        collection = gridRoot.getCollection();
    collection.removeRowAttributeAt(4);
    collection.removeCellAttributeAt(0, 0);
}

function clearAllAttr() {
    if (collection == null)
        collection = gridRoot.getCollection();
    collection.removeAllAttribute();
}

```

<예제 24 자바스크립트에서 행, 셀 속성 적용 예>

위 예를 보면 맨 먼저 rMate Grid 에서 collection 컴포넌트의 인스턴스를 받아오고, 받아온 collection 객체에 addRowAttributeDetailAt 함수, addCellAttributeDetailAt 함수를 사용하여 원하는 곳에 속성을 지정합니다. 또한 지정된 속성의 삭제는 removeRowAttributeAt 함수와 removeCellAttributeAt 함수를 사용하여 각 함수에 대한 설명은 다음과 같습니다.

행 속성을 등록하는 addRowAttributeDetailAt 함수의 파라미터는 다음과 같습니다.

```
function      addRowAttributeDetailAt(rowNo:int,      styleName:String,      backgroundColor:Number,
formatString:String, editable:Boolean, rowHeight:Number)
```

rowNo : 행의 index 번호

styleName : 적용할 style 명 (layout 에서 정의한 스타일명('.')은 제외한 부분)

backgroundColor : 배경색 (RGB 값으로 적용)

formatString : 행에 적용할 formatString (매뉴얼 참조)

editable : 행의 수정가능여부 (true/false)

rowHeight : 행의 높이 (단위:pixel)

셀 속성을 등록하는 addCellAttributeDetailAt 함수의 파라미터는 다음과 같습니다.

```
function addCellAttributeDetailAt(rowNo:int, colNo:int, rowSpan:int, colSpan:int, styleName:String,
background-color:Color, formatString:String)
```

rowNo : 행의 index 번호

colNo : 열의 index 번호

rowSpan : 병합할 행의 갯수 (기본 1 개)

colSpan : 병합할 열의 갯수 (기본 1 개)

styleName : 적용할 style 명 (layout 에서 정의한 스타일명('.')은 제외한 부분)

background-color : 배경색 (RGB 값으로 적용)

formatString : 셀에 적용할 formatString (매뉴얼 참조)

행 속성을 삭제하는 removeRowAttributeAt 함수의 파라미터는 다음과 같습니다.

```
function removeRowAttributeAt(rowNo:int)
```

rowNo : 행의 index 번호

셀 속성을 삭제하는 removeCellAttributeAt 함수의 파라미터는 다음과 같습니다.

```
function removeCellAttributeAt(rowNo:int, colNo:int)
```

rowNo : 행의 index 번호

colNo : 열의 index 번호

모든 속성을 삭제하는 removeAllAttribute 함수는 다음 구조로 되어 있습니다.

```
function removeAllAttribute():Boolean
```

행, 셀의 속성을 데이터에 넣지 않고 모두 자바스크립트 함수로 설정하는 경우, 데이터에서 속성을 추출하는 작업이 필요없기 때문에 SpanArrayCollection, SpanXMLListCollection 콜렉션에는 extractable 속성이 있으며 이 extractable 속성을 false 로 하여 추출작업을 안하도록 하여야 합니다.

## 8.4. 데이터에 행, 셀 속성지정

rMate Grid에서는 데이터에 직접 행 또는 셀의 속성을 지정하는 방식을 지원합니다.

개별적인 함수에 의한 제어가 아닌 서버측에서 데이터를 가공할 때 표현속성을 넣어 줌으로써 작업의 편의성을 도모하고, 특히 데이터가 XML 형식일 때 마치 html 에서 table tag 에 속성을 넣어 제어하는 방식과 유사한 방식으로 제어할 수 있는 것입니다. 데이터가 XML 일 경우의 예를 들면 다음과 같습니다.

```
<?xml version="1.0" encoding="utf-8"?>
<RevenuesAnalysis>
  <RegionalRevenue>
```

```

    <Year>2007</Year>
    <Quarter>4/4</Quarter>
    <Month>12</Month>
    <Seoul>99008</Seoul>
    <Pusan>30598</Pusan>
    <Incheon>99124</Incheon>
    <NewYork>22776</NewYork>
    <LA>107805</LA>
    <Washington>38384</Washington>
    <Revenue>397696</Revenue>
    <Percent>35</Percent>
  </RegionalRevenue>

```

```

    <RegionalRevenue styleName="allTotalStyle" backgroundColor="0xFFFF00" editable="false"
    formatString="#,##0원" rowHeigh="50">

```

```

      <Year rowSpan="2" colSpan="3" styleName="allTotalHeaderStyle"
      backgroundColor="0xDDDDDD" formatString="#,##0">T O T A L</Year>

```

```

      <Seoul>846568</Seoul>
      <Pusan>748824</Pusan>
      <Incheon>717834</Incheon>
      <NewYork>676387</NewYork>
      <LA>780781</LA>
      <Washington>857351</Washington>
      <Revenue>4627745</Revenue>
      <Percent></Percent>
    </RegionalRevenue>

```

```

    :
    :

```

```

</RevenuesAnalysis>

```

<예제 25 XML 데이터에 행, 셀 속성 넣은 예>

```

<rMateGrid>
  <DataGrid id="dg1" editable="true" doubleClickEnabled="true"
  verticalAlign="middle">
    <columns>
      <DataGridColumn dataField="From" width="80"/>
      <DataGridColumn dataField="Subject" width="150"/>
      <DataGridColumn dataField="Date"/>
      <DataGridColumn dataField="CC"/>
      <DataGridColumn dataField="Profit" textAlign="right"/>
    </columns>
    <dataProvider>
      <!--
      그리드의 자료를 일반 data형식 (배열, XML) 이 아닌 별도의 컴포넌트에

```

입력해야 할 경우 아래와 같이 \$gridData를 넣어줍니다

```
-->
    <SpanXMLListCollection source="{ $gridData }"/>
  </dataProvider>
</DataGrid>
</rMateGrid>
```

#### <예제 26 SpanXMLListCollection 컴포넌트 사용 예>

이 작업에 사용되는 컴포넌트는 SpanXMLListCollection 으로 행을 나타내는 RegionalRevenue 노드에서 행의 속성을 XML attribute 에 넣음으로써 간단하게 행 속성을 넣어 줄 수 있고, 셀을 나타내는 Year 노드에서 셀의 속성을 XML attribute 에 넣어주게 됩니다. 셀에 대한 속성을 넣어줄 때 주의점은 원래의 셀 속성 중 colNo 는 넣어주실 필요가 없다는 점입니다. colNo 는 레이아웃상에서 정의 될 때 정해지기 때문에 내부적으로 자동 계산됩니다.

배열에 행 또는 셀의 속성을 지정하려면 SpanArrayCollection 컴포넌트를 사용하며, SpanArrayCollection 의 cellAttributeFieldSuffix 속성을 통하여 Cell 속성이 저장된 dataField 를 찾아 개별 Cell 의 속성을 설정하게 됩니다. 또한 Row 에 대한 속성은 rowAttributeField 속성을 통해 저장된 dataField 를 찾게 됩니다.

```
private var dpArray:SpanArrayCollection = new SpanArrayCollection([
    {Region:"Southwest3", Region_attr:{rowSpan:2, colSpan:3, styleName:"gridSummaryStyle",
    backgroundColor:"#00FF00"}, Territory:"Central California",
    Territory_Rep:"Joe Smith",
    Revenues:{Actual:29134, Estimate:30000, Real:30000, Price:3003}},
    {Region:"Southwest", Territory:"Nevada",
    Territory_Rep:"Bethany Pittman",
    Revenues:{Actual:52888, Estimate:45000, Real:30000, Price:3004}},
    {Region:"Nor", Territory:"Arizona",
    Territory_Rep:"Dana Binn",
    Revenues:{Actual:29885, Estimate:30000, Estimate_attr:{rowSpan:2,
    _ratr:{styleName:"allTotalStyle", backgroundColor:"0xFFFF00"}},
    {Region:"Nor", Territory:"Central California",
    Territory_Rep:"Joe Smith",
    Revenues:{Actual:29134, Estimate:30000, Real:30000, Price:3001}},
    {Region:"Northwest", Territory:"Southern California",
    Territory_Rep:"Alice Treu",
    Revenues:{Actual:44985, Estimate:45000, Real:30000, Price:3009}}
]);
```

#### <예제 27 배열에 행, 셀 속성 넣은 예>

```
<rMateGrid>
```



```

<DataGrid id="dg1" editable="true" doubleClickEnabled="true"
verticalAlign="middle">
  <columns>
    <DataGridColumn dataField="From" width="80"/>
    <DataGridColumn dataField="Subject" width="150"/>
    <DataGridColumn dataField="Date"/>
    <DataGridColumn dataField="CC"/>
    <DataGridColumn dataField="Profit" textAlign="right"/>
  </columns>
  <dataProvider>
    <!--
      그리드의 자료를 일반 data형식 (배열, XML) 이 아닌 별도의 컴포넌트에
      입력해야 할 경우 아래와 같이 $gridData를 넣어줍니다
    -->
    <SpanArrayCollection source="{ $gridData }"/>
  </dataProvider>
</DataGrid>
</rMateGrid>

```

<예제 28 SpanArrayCollection 컴포넌트 사용 예>

예에서 Region 필드에 대한 Cell 속성 필드명은 Region\_attr 이 되며 "\_attr"은 SpanArrayCollection 의 cellAttributeFieldSuffix 속성을 통해 변경할 수 있으며, 기본값은 "\_attr"입니다. 이때 속성 필드는 array 로 표현되어 여러 개의 값을 가질 수 있도록 합니다.(1 개의 속성만 표시하더라도 array 형식이어야 함) 또한 Row 속성 필드명은 "\_rattr"이 되어 해당 Row 에 대한 속성을 정의 하고 있습니다.

데이터에 행, 셀속성을 넣어 처리할 때 컬렉션 컴포넌트에서는 데이터에서 속성을 추출하는 작업이 일어 납니다. 그래서 약간의 지연시간이 발생할 수 있으니 유의하시기 바랍니다.

## 9. 컴포넌트별 속성, 스타일, 함수 및 이벤트

### 9.1. GridApp

자바스크립트에서 gridApp로 선언하는 컴포넌트로 가장 기본이 되는 컴포넌트입니다.  
스타일이나 이벤트에 대한 설정은 없습니다.

- 속성 및 유효 값 설명

속성 명	유효 값(기본값)	설명
bridgeName	String	자바스크립트 연동을 위한 bridgeName 기본적으로는 flashVars 에서 설정합니다.
dataURL	String	데이터를 읽어올 URL 기본적으로는 flashVars 에서 설정합니다. 변경을 원할 경우 직접 수정하지 마시고 setDataURLEx 함수를 사용하십시오
dataType	String	데이터를 URL 로 읽어오는 경우 데이터의 타입 xml,json 중 택일 기본적으로는 flashVars 에서 설정합니다. 변경을 원할 경우 setDataType 함수를 사용하십시오
layoutURL	String	레이아웃을 읽어올 URL 기본적으로는 flashVars 에서 설정합니다. 변경을 원할 경우 직접 수정하지 마시고 setLayoutURLEx 함수를 사용하십시오
showLoadingBar	Boolean	파일 로딩시 로딩바 표시 여부

- 함수 명 및 설명

메소드 명	설명
setData()	function setData(value:*)void  데이터를 자바스크립트를 통해 직접 등록합니다. value가 문자열일 경우에는 dataType에 따라 XML, JSON 데이터로 처리하고, 배

	<p>열일 경우에는 직접 데이터로 사용합니다.</p> <p><b>Parameters</b></p> <p>- value:* 데이터</p>
setDataURLEx()	<p>function setDataURLEx(value:String):void</p> <p>데이터URL을 자바스크립트를 통해 직접 등록합니다.</p> <p><b>Parameters</b></p> <p>- value:String 데이터URL</p>
setDataType()	<p>function setDataType(value:String):void</p> <p>데이터의 타입을 지정합니다. value로 넣어질 수 있는 값은 "xml"과 "json"입니다.</p> <p><b>Parameters</b></p> <p>- value:String 데이터 타입L</p>
setLayout()	<p>function setLayout(value:String):void</p> <p>레이아웃의 내용을 자바스크립트에서 직접 등록합니다. value가 문자열은 XML 형식이어야 합니다.</p> <p><b>Parameters</b></p> <p>- value:String 레이아웃 문자열</p>
setLayoutURLEx()	<p>function setLayoutURLEx(value:String):void</p> <p>레이아웃URL을 자바스크립트를 통해 직접 등록합니다.</p> <p><b>Parameters</b></p> <p>- value:String 레이아웃URL</p>
addLoadingBar()	<p>function addLoadingBar():void</p> <p>화면에 로딩표시를 나타냅니다.</p>
removeLoadingBar()	<p>function removeLoadingBar():void</p> <p>화면에 나타난 로딩표시를 지웁니다.</p>

## 9.2. GridRoot

자바스크립트에서 gridRoot로 선언하는 컴포넌트로 데이터와 그리드를 포함하고 있는 컴포넌트입니다. 스타일에 대한 설정은 없습니다.

- 속성 및 유효 값 설명

속성 명	유효 값(기본값)	설명
collection	실 데이터	그리드에 놓어진 데이터입니다.
filterFunction	Function	<p>데이터에 필터링을 수행할 필터링함수를 지정합니다.</p> <p>filterFunction 함수의 파라미터는 다음과 같습니다.</p> <pre>function filterFunction(item:Object):Boolean</pre> <p>item : 해당 행의 data 객체</p> <p>함수의 결과 true이면 보여지고 false를 반환하면 숨겨지게 됩니다.</p>
found	Boolean	(read-only)검색결과 search 함수 수행 결과를 가져옵니다.
gridData	Object	dataURL이나 setData()에 의해 읽혀진 데이터로 그리드에 저장하기 전 데이터 입니다.
importFileName	String	엑셀 import기능에 의거해 읽어들이는 파일의 파일명입니다. (디렉토리는 포함되어 있지 않고 파일명만 들어있음)
removeAllData	Boolean	removeAll() 명령에 의해 전체 데이터가 삭제된 경우 getChangedData() 나 getChangedDataXML() 명령으로는 아무것도 안나오게 되며 이 속성을 검사하여 전체삭제를 했는지 알 수 있게 됩니다.
searchString	String	검색할 문자열을 지정합니다.

- 함수 명 및 설명

메소드 명	설명
addEventListener()	function addEventListener(type:String, listener:Function):void

	<p>type에 선언된 이벤트가 발생할 경우 이벤트를 받을 수 있는 listener함수를 등록합니다.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- type:<b>String</b> – 이벤트 종류를 표현하는 문자열</li> <li>- listener:<b>Function</b> – 이벤트 발생시 불러질 함수</li> </ul>
addItemAt()	<p>function addItemAt(item:Object, index:int = -1, editMode:Boolean = true):void</p> <p>데이터의 특정 행에 item을 등록합니다. (합산, 계층형 데이터일 경우 작동 안함, 소팅, 필터링 상태에서는 불가 메시지가 표시되며 작동 안함)</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- item:<b>Object</b> – 등록할 행의 data객체</li> <li>- index:<b>int</b> – 등록할 행의 index번호, 넣지 않거나 -1을 넣어주면 맨마지막에 등록합니다.</li> <li>- editMode:<b>Boolean</b> – 등록을 수행한 후 해당 행의 수정모드로 전환할지 여부</li> </ul>
clearStyle()	<p>function clearStyle(styleProp:<b>String</b>):void</p> <p>컴포넌트에 등록되어 있는 스타일을 지웁니다.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- styleProp:<b>String</b> 스타일 속성명</li> </ul>
excelCSVImport()	<p>public function excelCSVImport(option:int = 0):void</p> <p>엑셀 CSV파일 가져오기를 수행합니다. 가져올때 option에 따라 레이아웃 변경 여부를 정하게 됩니다.</p> <p>사용자의 선택이 정상적으로 이루어져 수행이 완료된후에 importComplete 이벤트가 발생합니다.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- option:<b>int</b> 가져오기 옵션(0:사용자에게 질의, 1: 현재 레이아웃에 데이터만 가져오기, 2 : 헤더나 데이터에 따라 레이아웃을 재설정하고 데이터 가져오기)</li> </ul>
excelExport()	<p>function excelExport():void</p> <p>엑셀 내보내기를 수행합니다.</p> <p>export시 엑셀의 기능과 SpanDataGrid의 기능 차이에 의해 몇몇 기능은 반영이 안 될 수 있습니다.</p> <p>- StyleFunction에 의한 스타일 설정</p>

	<ul style="list-style-type: none"> <li>- 개별 renderer에 의해 설정된 스타일 또는 label을 제외한 내용 - 콤보나 이미 지등등</li> <li>- 개별 Formatter에 의한 설정된 일반적이지 않은 포맷</li> <li>- html - br태그만 처리</li> </ul>
expandItem	<p>function expandItem(index:int, open:Boolean, animate:Boolean = false, dispatchEvent:Boolean = false):void</p> <p>데이터가 계층형 또는 그룹핑일 경우 해당 index의 행의 하위 노드를 열거나 닫습니다.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- index:<b>int</b> – 작업할 행의 index번호</li> <li>- open:<b>Boolean</b> – true이면 하위 노드를 열고 false이면 닫음</li> <li>- animate:<b>Boolean</b> – 동작시 효과를 줄 지 여부</li> <li>- dispatchEvent:<b>Boolean</b> – 동작후 open 이벤트를 발생시킬지 여부</li> </ul>
getChangedData()	<p>function getChangedData():Array</p> <p>자바스크립트 또는 그리드 내에서 수정된 내용을 반환합니다.</p> <p>반환되는 값은 다음 형식으로 저장된 배열입니다.          idx : 행번호          job : 수행된 작업을 나타내며 다음 값을 가집니다.                  I:등록, D:삭제, U:수정          data : 사용자가 등록 또는 수정한 내용(행의 모든 데이터가 저장됨)</p> <p>배열은 작업순서대로 저장되나 서버작업의 편의를 위해 삭제된 것부터 나타납니다.</p>
getChangedDataXML()	<p>function getChangedDataXML():String</p> <p>자바스크립트 또는 그리드 내에서 수정된 내용을 XML형식의 문자열로 반환합니다.</p> <p>getChangedData함수의 결과를 XML형식으로 변환하니 getChangedData의 결과 값을 참고하십시오.</p>
getItemAt()	<p>function getItemAt(index:int):Object</p> <p>데이터의 특정 행을 반환합니다.</p> <p><b>Parameters</b></p>

	<p>- index:<b>int</b> – 가져올 행의 index번호</p> <p><b>Returns</b></p> <p>- Object: 해당 행의 데이터, 없을시 null</p>
getItemFieldAt()	<p>function getItemFieldAt(index:int, fieldName:String):Object</p> <p>데이터의 특정 행의 필드를 반환합니다.</p> <p><b>Parameters</b></p> <p>- index:<b>int</b> – 가져올 행의 index번호</p> <p>- fieldName:<b>String</b> – 필드명</p> <p><b>Returns</b></p> <p>- Object: 해당 필드의 데이터, 없을시 null</p>
getObjectById()	<p>function getObjectById(id:<b>String</b>):Object</p> <p>레이아웃에서 id로 지정된 컴포넌트를 반환합니다.</p> <p>레이아웃에 속성 id에 값을 넣을 컴포넌트에 대하여 전달된 id로 된 컴포넌트의 인스턴스를 반환합니다.</p> <p><b>Parameters</b></p> <p>- id:<b>String</b> 조회하고자 하는 컴포넌트의 id</p> <p><b>Returns</b></p> <p>- Object: id로 등록된 컴포넌트 인스턴스, 없을시 null</p>
getStyle()	<p>function getStyle(styleProp:<b>String</b>):*</p> <p>컴포넌트에 등록되어 있는 스타일을 반환합니다.</p> <p><b>Parameters</b></p> <p>- styleProp:<b>String</b> 스타일 속성명</p> <p><b>Returns</b></p> <p>- *: 스타일에 등록되어 있는 값, 없을시 null</p>
isItemOpen()	<p>function isItemOpen(index:<b>int</b>):Boolean</p> <p>데이터가 계층형이나 그룹핑일 경우 해당 index의 행이 열려 있는지 여부를 반환합니다.</p>

	<b>Parameters</b> - index: <b>int</b> 검사할 행의 index번호  <b>Returns</b> - Boolean: 열려있으면 true, 닫혀있으면 false
removeAll()	function removeAll():void  전체 데이터를 삭제합니다.
removeEventListener()	function removeEventListener(type:String, listener:Function):void  type에 선언된 이벤트에 연결된 함수 listener를 제거합니다. <b>Parameters</b> - type: <b>String</b> – 이벤트 종류를 표현하는 문자열 - listener: <b>Function</b> – 이벤트 발생시 불러질 함수
removeItemAt()	function removeItemAt(index:int = -1):void  데이터의 특정 행을 삭제합니다. (합산, 계층형 데이터일 경우 작동 안함)  <b>Parameters</b> - index: <b>int</b> – 삭제할 행의 index번호, 넣지 않거나 -1을 넣어주면 현재의 selectedIndex에 있는 행을 삭제합니다.
resetChangedData	function resetChangedData():void  내부에서 관리하는(getChangedData 또는 getChangedDataXML에서 사용) 수정 내역을 모두 지워 초기화 시킵니다.
search()	function search(searchStr:String = null, caseInsensitive:Boolean = true):Boolean  검색을 실행합니다. searchStr을 안넣거나 null일 경우 속성 searchString에 저장된 값을 사용하여 검색합니다.  <b>Parameters</b> - searchStr: <b>String</b> – 검색할 문자열 - caseInsensitive: <b>int</b> – 대소문자 구별여부  <b>Returns</b> - Boolean: 검색결과가 있으면 true, 없으면 false 반환



searchNext()	function searchNext():Boolean  다음찾기를 실행합니다.  <b>Returns</b> - Boolean: 검색결과가 있으면 true, 없으면 false 반환
searchPrevious()	function searchPrevious():Boolean  이전찾기를 실행합니다.  <b>Returns</b> - Boolean: 검색결과가 있으면 true, 없으면 false 반환
setItemAt()	function setItemAt(item:Object, index:int, editMode:Boolean = false):void  데이터의 특정 행을 item으로 수정합니다. (합산, 계층형 데이터일 경우 작동 안 함)  <b>Parameters</b> - item: <b>Object</b> – 수정할 행의 데이터 - index: <b>int</b> – 수정할 행의 index번호 - editMode: <b>Boolean</b> – 수정을 수행한 후 해당 행의 수정모드로 전환할지 여부
setItemFieldAt()	function setItemFieldAt(value:Object, index:int, fieldName:String):void  데이터의 특정 행의 필드값을 수정합니다. (합산, 계층형 데이터일 경우 작동 안 함)  <b>Parameters</b> - value: <b>Object</b> – 수정할 필드의 데이터 - index: <b>int</b> – 가져올 행의 index번호 - fieldName: <b>String</b> – 필드명
setStyle()	function setStyle(styleProp:String, newValue:*) :void  컴포넌트에 스타일을 등록합니다.  <b>Parameters</b> - styleProp: <b>String</b> 스타일 속성명 - newValue:* 스타일에 등록할 값

- 이벤트 및 설명

이벤트	설명
dataComplete	데이터 읽어오기가 완료되면 발생합니다.  이벤트 상세내역 없음
importComplete	엑셀 CSV파일 가져오기가 완료되면 발생합니다.  이벤트 상세내역 없음
itemDataChanged	사용자가 셀을 에디팅하면(에디팅이 끝나면) 발생합니다.  이벤트 상세내역 - columnIndex : 클릭한 열의 index번호 - rowIndex : 클릭한 행의 index번호 - dataField : 수정된 열의 필드명 - dataRow : 수정된 데이터 - oldValue : 수정되기 전의 값 - newValue : 수정된 후의 값
layoutComplete	레이아웃을 읽어와 그리드 생성을 완료하면 발생합니다.  이벤트 상세내역 없음
searchResultChanged	search, searchNext, searchPrevious등의 함수를 수행한 후 발생합니다. 함수의 반환값으로 검색 성공여부를 알수있으며, 이벤트를 통해 found속성을 조사하는 방법으로도 알수 있습니다.  이벤트 상세내역 없음

### 9.3. DataGrid

실제 그리드를 표시하는 컴포넌트입니다.

- 속성 및 유효 값 설명

속성 명	유효 값(기본값)	설명
alpha	Number(0~1)(기본값:1)	투명도(alpha) 값을 결정합니다.
columns	Array	각각의 칼럼에 표시될 DataGridColumn 객체의

		배열을 나타냅니다. 그리드에서 최대 표현할 수 있는 컬럼의 수는 255 개 입니다.
columnWidth	Number(50)	컬럼들의 기본 가로 사이즈를 픽셀 단위로 지정합니다.
dataProvider	Object	표시할 데이터세트입니다
dataTipJsFunction	Function	셀에서 표시할 데이터팁 라벨의 콜백함수를 지정합니다. 이는 데이터팁 내용을 사용자 정의할 수 있게합니다.  dataTipJsFunction 함수의 파라미터는 다음과 같습니다. function dataTipJsFunction(item:Object) item : 해당 행의 data 객체
displayDisclosureIcon	Boolean(true)	계층형 네비게이션에서 disclosure icons 의 유무를 나타냅니다. 만약 false 인 경우 disclosure icons 는 나타나지 않습니다.
displayItemsExpanded	Boolean(false)	만약 true 로 설정하였다면, 계층형 네비게이션에서 모든 아이템을 보이도록 확장합니다. True 로 설정한 상태에서 새로운 브랜치를 추가한 경우, 그 브랜치는 펼쳐진 형태로 출력됩니다.
doubleClickEnabled	Boolean(false)	더블클릭 가능 여부 (true,false 중 택일, 기본 false, true 의 경우 더블클릭하면 해당 셀을 수정할수 있게 됩니다)
draggableColumns	Boolean(true)	컬럼들의 순서를 재조정하게 허용할지 여부를 나타냅니다.
dragSelectable	Boolean(false)	마우스를 드래그하여 선택 영역을 지정하는 기능의 사용 여부. 이 속성은 selectionMode 가 multipleCells 이나 multipleRows 일 경우에만 작동합니다.
editable	Boolean(false)	editable 수정모드 지정 (true,false 중 택일, 기본 false)
enabled	Boolean(true)	사용자 반응을 받아들일지 여부를 나타냅니다. Enable 을 false 로 설정한 상태는 잠금상태가 되어 사용자 반응에 어떤 행위도 응하지 않습니다.
exportColumns	Array	Export 할 컬럼들의 배열 columns 이나 groupsColumns 에 설정한 모든 컬럼이 아닌, Excel export 할 컬럼만을 지정할

		<p>경우에 사용합니다.</p> <p>배열의 내용은 columns 상의 숫자로 가령 [0,1,3]로 설정한 경우, columns 에서 첫번째, 두번째, 네번째 컬럼을 export 하게 됩니다.</p>
exportFileName	String(Default export.xls)	<p>Export 할 excel 파일명</p> <p>이 문자열에는 다음 문자가 포함되면 안됩니다 / \ : * ? " &lt; &gt;   %</p>
exportFooter	String	Excel export 시 내용에 덧붙일 꼬릿말
exportFooterHeight	int(20)	Excel export 시 내용에 덧붙일 꼬릿말의 높이
exportOnlyData	Boolean(false)	Excel export 시 데이터만 export 할지 여부
exportTitle	String	Excel export 시 내용에 덧붙일 제목
exportTitleHeight	Int(Default 50)	Excel export 시 내용에 덧붙일 제목행의 높이
footers	Array	푸터에 붙일 객체의 배열
groupedColumns	Array	<p>컬럼을 그룹핑할 때 DataGridViewColumn 인스턴스 계층(hierarchy)를 정의하는 배열입니다.</p> <p>사용자가 만약 columns 와 groupedColumns 속성 모두를 지정한다면 columns 속성은 무시되고 groupedColumns 속성이 사용됩니다.</p>
groupItemRenderer	아이템렌더러 객체	계층형 네비게이션에서 브랜치 노드를 표현하기 위해 사용되는 아이템렌더러를 지정합니다.
groupRowHeight	Number(픽셀단위)	Grouped row 의 높이입니다.(픽셀단위)
headerHeight	Number	컬럼에서 셀의 헤더를 픽셀단위로 지정합니다.
headerWordWrap	Boolean(false)	만약 true 로 설정하면, 헤더에서 텍스트의 길이가 길어 한라인에 표시하지 못할 경우 다음 라인에 출력시킵니다.
height	Number	세로 사이즈를 픽셀단위로 지정합니다.
horizontalScrollPolicy	"auto", "on", "off"(off)	가로 스크롤바 표시 여부 및 정책을 결정합니다.
horizontalScrollPosition	Number	현재 가장 왼쪽 컬럼의 번호
itemEditBeginningJsFunction	Function	<p>editable 속성이 true 일 때 사용자가 셀을 클릭하여 셀의 수정 모드로 들어가면 불려지며, 수정가능이면 true 를, 수정 불가이면 false 를 반환하여 수정여부를 조작할 수 있습니다.</p> <p>itemEditBeginningJsFunction 함수의 파라미터는 다음과 같습니다.</p> <pre>function itemEditBeginningJsFunction     (rowIndex:int, columnIndex:int,     item:Object, dataField:String)     rowIndex : 해당 행의 index 번호</pre>

[illegible]

selectedIndex	int	선택된 행의 데이터내의 index 입니다.  기본 값은 -1 이며, 아무런 선택이 없을 경우 -1 이 지정됩니다.
selectedIndices	Array	선택된 행들의 데이터내의 index 배열입니다.  기본 값은 [ ]이며, 아무런 선택이 없을 경우 [ ]이 지정됩니다.
selectedItem	Object	선택된 행의 데이터의 참조입니다.
selectedItems	Array	선택된 행들의 데이터의 참조 배열입니다. 행의 선택 역순으로 배열에 들어갑니다.
selectionMode	"singleCell", "singleRow", "multipleCells", "multipleRows" , "none" (singleRow)	그리드의 선택모드를 설정합니다. 다중선택일 경우 ctrl 키와 마우스를 이용하여 여러 개를 선택할 수 있습니다.
showDataTips	Boolean(false)	그리드의 칼럼에 데이터팁을 표시할지 여부를 나타냅니다.
showFooters	Boolean(true)	푸터 표시 여부
showScrollTips	Boolean(false)	스크롤 썸네일 옆에 스크롤 툴팁을 표시할지 여부
sortableColumns	Boolean(true)	사용자가 헤더를 클릭함으로써 해당 칼럼의 데이터를 정렬시킬 수 있을지 여부를 결정합니다.
sortExpertMode	Boolean(true)	숙련 정렬 아이콘 표시 여부 조정을 나타냅니다. 화면에 정렬표시아이콘이 나타나지 않고, Ctrl Key 를 통해 정렬여부를 조정하게 됩니다.
styleName	String	스타일 이름으로 스타일 지정
variableRowHeight	Boolean	각각의 행이 다른 높이를 갖도록 할지 여부를 나타내는 플래그입니다.
verticalScrollPolicy	"on", "off", "auto"(auto)	스크롤바 생성 정책을 결정합니다. auto 인 경우 내용물이 현재 가로, 세로보다 클 경우 자동 생성이됩니다.
verticalScrollPosition	Number	현재의 화면에 나타난 첫번째 행 번호
visible	Boolean(true)	가시화 할지 여부를 나타냅니다.
width	Number	가로 사이즈를 픽셀 단위로 결정합니다.
wordWrap	Boolean(false)	워드랩 여부를 결정합니다.

● 스타일 및 유효 값 설명

스타일 명	유효 값(기본값)	설명
alternatingItemColors	Array	행들의 색을 결정합니다. 많은 컬러의 배열입니다. 첫번째 요소의 색이 한 행에 결정되었다면 다음 행은 두번째 색으로 결정됩니다. Alternating pattern 을 갖고 색이 결정됩니다.
backgroundAlpha	Number	배경의 투명도를 나타냅니다.
backgroundColor	RGB	배경색을 나타냅니다.
backgroundDisabledColor	RGB	잠긴상태(enabled=false) 인 경우 색을 나타냅니다.
borderColor	RGB	테두리 색을 지정합니다.
borderStyle	"none","solid","inset","outset" (inset)	테두리 선의 모양을 지정합니다.( none,solid,inset,outset)
borderThickness	Number	테두리 선의 두께를 나타냅니다.
color	RGB	텍스트의 색깔을 나타냅니다.
depthColors	Array	계층형 네비게이션의 각 레벨(depth 에 맞는) 행들의 컬러 배열을 나타냅니다.
disabledColor	RGB	잠긴상태의 텍스트 컬러를 나타냅니다.
disclosureClosedIcon	Boolean(true)	계층형 네비게이션에서 닫힌 브랜치 노드의 disclosure 아이콘의 유무를 나타냅니다. 만약 false 인 경우 disclosure 아이콘은 표시되지 않습니다.
disclosureOpenIcon	Boolean(true)	계층형 네비게이션에서 열린 브랜치 노드의 disclosure 아이콘의 유무를 나타냅니다. 만약 false 인 경우 disclosure 아이콘은 표시되지 않습니다.
dropShadowColor	RGB	그림자 효과의 그림자 컬러를 결정합니다.
dropShadowEnabled	Boolean(기본값:true)	그림자 효과를 나타낼지 여부를 나타냅니다.
exportFooterAlign	"right" ,"left","center"(right)	Excel export 시 내용에 덧붙일 꼬릿말의 수평 정렬
exportFooterColor	RGB	Excel export 시 내용에 덧붙일 꼬릿말의 글씨 색깔
exportFooterFontSize	Number(Default 10)	Excel export 시 내용에 덧붙일 꼬릿말의 font 크기
exportTitleAlign	"center","left","right"(center)	Excel export 시 내용에 덧붙일 제목행의 수평 정렬
exportTitleColor	RGB	Excel export 시 내용에 덧붙일 제목행의 글씨 색깔

exportTitleFontSize	Number(Default 18)	Excel export 시 내용에 덧붙일 제목행의 font 크기
fontFamily	폰트 이름	사용할 폰트를 결정합니다.
fontSize	Number(픽셀단위)	폰트 사이즈를 픽셀단위로 결정합니다.
fontStyle	"normal","italic"	폰트 이텔렉체를 사용할지를 나타냅니다.
fontWeight	"normal","bold"	폰트 볼드체를 사용할지를 나타냅니다.
headerColors	Array	헤더 배경에 그라디언트 색을 배열로 정의합니다. 배열로 정의된 색이 배경에 그라디언트로 표현됩니다.
headerStyleName	스타일 이름	칼럼 헤더의 스타일 이름을 나타냅니다.
horizontalGridLineColor	RGB	수평 그리드 라인의 색 (기본값: 0x666666.)
horizontalGridLines	Boolean	행과 행 사이의 수평 그리드 라인 표시 여부를 나타냅니다.
indentation	Number(17)	계층형 네비게이션의 각 노드의 인덴테이션을 픽셀단위로 나타냅니다.
openDuration	Number(250)	계층형 네비게이션에서 브랜치 노드의 열고 닫히는 애니메이션의 속도를 밀리섹컨드 단위로 나타냅니다.
paddingBottom	Number	아래쪽 여백을 나타냅니다.
paddingTop	Number	위쪽 여백을 나타냅니다.
paddingLeft	Number	왼쪽 여백을 나타냅니다.
paddingRight	Number	오른쪽 여백을 나타냅니다.
rollOverColor	RGB	마우스로 롤-오버 시 행의 배경색을 나타냅니다.
selectionColor	RGB	선택한 아이템(들)의 배경색을 나타냅니다.
selectionDisabledColor	RGB	잠긴 상태에서의 선택한 아이템(들)의 색을 나타냅니다.(기본값 : 0xDDDDDD)
shadowDirection	"center","left","right"	그림자 효과의 방향을 나타냅니다.
shadowDistance	Number	그림자 효과의 거리(distance)를 픽셀단위로 나타냅니다.
textAlign	"left","right","center"(left)	텍스트 정렬 방식을 나타냅니다.
textDecoration	"none","underline"	텍스트에 밑줄을 그을지 나타냅니다.
textIndent	Number	텍스트의 시작 들여쓰기를 픽셀단위로 지정합니다.
textRollOverColor	RGB	롤-오버 시 텍스트의 색을 나타냅니다. (기본값:0x2B333C)
textSelectedColor	RGB	선택 시 텍스트의 색을 나타냅니다.



		(기본값:0x2B333C)
verticalGridLineColor	RGB	수직 그리드 라인의 색을 결정합니다. (기본값: 0x666666.)
verticalGridLines	Boolean(기본값:true)	열과 열 사이의 수직 그리드라인을 표현할지 여부를 나타냅니다.(기본값 : true)

● 함수 명 및 설명

메소드 명	설명
addEventListener()	function addEventListener(type:String, listener:Function):void  type에 선언된 이벤트가 발생할 경우 이벤트를 받을 수 있는 listener함수를 등록합니다. <b>Parameters</b> - type: <b>String</b> – 이벤트 종류를 표현하는 문자열 - listener: <b>Function</b> – 이벤트 발생시 불러질 함수
clearStyle()	function clearStyle(styleProp: <b>String</b> ):void  컴포넌트에 등록되어 있는 스타일을 지웁니다. <b>Parameters</b> - styleProp: <b>String</b> 스타일 속성명
collapseAll()	function collapseAll():void  계층형 네비게이션의 모든 노드들의 자식을 표시하지 않도록 닫힌 형태로 표현합니다. 즉, 모든 폴더가 닫힌 형태로 만듭니다.
drawFoucs()	function drawFocus(isFocused: <b>Boolean</b> ):void  컴포넌트의 focus를 숨기거나 표시합니다 <b>Parameters</b> - isFocused: <b>Boolean</b> – true면 표시하고 false면 숨김
expandAll()	function expandAll():void  계층형 네비게이션의 모든 노드들의 자식을 표시하도록 모두 열린 상태로 표현합니다. 즉, 모든 폴더가 열린 형태로 만듭니다.
getStyle()	function getStyle(styleProp: <b>String</b> ):*

	<p>컴포넌트에 등록되어 있는 스타일을 반환합니다.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- styleProp:<b>String</b> 스타일 속성명</li> </ul> <p><b>Returns</b></p> <ul style="list-style-type: none"> <li>- *: 스타일에 등록되어 있는 값, 없을시 null</li> </ul>
invalidateList()	<p>public function invalidateList():void</p> <p>모든 행을 refresh합니다.</p>
selectionToClipboard()	<p>public function selectionToClipboard():void</p> <p>현재 선택된 내역을 클립보드에 넣어줍니다.          선택된 내용을 TSV(Tab-Separated Values)형식으로 클립보드에 넣어주게 됩니다.</p> <p>클립보드에 넣어주는 내용은 다음 설명에 따릅니다</p> <ul style="list-style-type: none"> <li>- 기본적으로 데이터의 원형이 놓여지며 포맷이나 스타일은 제외됩니다.</li> <li>- 날짜형식일 경우 포맷을 적용한 문자열로 넣어집니다.</li> <li>- labelFunction이 지정된 열에 대해서는 labelFunction이 적용된 값을 넣어줍니다.</li> <li>- 병합셀의 경우 한번만 포함되기 때문에 복사시 셀이 빠지는 현상이 일어날 수 있습니다.</li> </ul>
setStyle()	<p>function setStyle(styleProp:<b>String</b>, newValue:*) :void</p> <p>컴포넌트에 스타일을 등록합니다.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- styleProp:<b>String</b> 스타일 속성명</li> <li>- newValue:* 스타일에 등록할 값</li> </ul>
removeEventListener()	<p>function removeEventListener(type:String, listener:Function):void</p> <p>type에 선언된 이벤트에 연결된 함수 listener를 제거합니다.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- type:<b>String</b> – 이벤트 종류를 표현하는 문자열</li> <li>- listener:<b>Function</b> – 이벤트 발생시 불러질 함수</li> </ul>

● 이벤트 및 설명

이벤트	설명
change	<p>사용자가 마우스를 클릭하여 선택사항을 바꿨을 때 발생합니다.</p> <p>이벤트 상세내역</p> <ul style="list-style-type: none"> <li>- columnIndex : 클릭한 열의 index번호</li> <li>- rowIndex : 클릭한 행의 index번호</li> </ul>
click	<p>사용자가 마우스를 클릭하면 발생합니다.</p> <p>이벤트 상세내역</p> <ul style="list-style-type: none"> <li>- altKey : alter 키 클릭여부</li> <li>- controlKey : 컨트롤 키 클릭여부</li> <li>- ctrlKey : 컨트롤 키 클릭여부 (Mac일 경우)</li> <li>- shiftKey : Shift 키 클릭여부</li> <li>- stageX : X 좌표</li> <li>- stageY : Y 좌표</li> </ul>
doubleClick	<p>사용자가 마우스를 더블클릭하면 발생합니다.</p> <p>이벤트 상세내역</p> <ul style="list-style-type: none"> <li>- altKey : alter 키 클릭여부</li> <li>- controlKey : 컨트롤 키 클릭여부</li> <li>- ctrlKey : 컨트롤 키 클릭여부 (Mac일 경우)</li> <li>- shiftKey : Shift 키 클릭여부</li> <li>- stageX : X 좌표</li> <li>- stageY : Y 좌표</li> </ul>
gridMenuItemSelect	<p>사용자가 오른쪽 마우스를 눌러 ContextMenu를 선택하면 발생합니다.</p> <p>이벤트 상세내역</p> <ul style="list-style-type: none"> <li>- menuItemCaption : 선택한 메뉴의 메뉴명</li> </ul>
headerRelease	<p>사용자가 헤더를 눌렀다 떼는 경우 발생합니다. (정렬이 수행되기 전)</p> <p>이벤트 상세내역</p> <ul style="list-style-type: none"> <li>- columnIndex : 클릭한 헤더의 컬럼 index번호</li> <li>- dataField : 클릭한 헤더의 필드명</li> </ul>
headerShift	<p>컬럼이 이동되면 발생합니다.</p> <p>이벤트 상세내역</p>

	<ul style="list-style-type: none"> <li>- newIndex : 컬럼이 옮겨진 index번호</li> <li>- oldIndex : 컬럼의 이전 index번호</li> </ul>
itemClick	<p>사용자가 셀을 마우스로 클릭하면 발생합니다.</p> <p>이벤트 상세내역</p> <ul style="list-style-type: none"> <li>- columnIndex : 클릭한 열의 index번호</li> <li>- rowIndex : 클릭한 행의 index번호</li> </ul>
itemDoubleClick	<p>사용자가 셀을 마우스로 더블클릭하면 발생합니다.</p> <p>이벤트 상세내역</p> <ul style="list-style-type: none"> <li>- columnIndex : 클릭한 열의 index번호</li> <li>- rowIndex : 클릭한 행의 index번호</li> </ul>
itemRollOver	<p>사용자가 셀을 마우스로 롤오버하면 발생합니다.</p> <p>이벤트 상세내역</p> <ul style="list-style-type: none"> <li>- columnIndex : 롤오버한 열의 index번호</li> <li>- rowIndex : 롤오버한 행의 index번호</li> </ul>
keyDown	<p>사용자가 키보드를 눌렀을때 발생합니다.</p> <p>이벤트 상세내역</p> <ul style="list-style-type: none"> <li>- charCode : 눌러진 키보드의 문자 코드</li> <li>- controlKey : 컨트롤 키 클릭여부</li> <li>- ctrlKey : 컨트롤 키 클릭여부 (Mac일 경우)</li> <li>- keyCode : 눌러진 키보드의 key 코드</li> <li>- keyLocation : 눌러진 키보드의 key location</li> <li>- shiftKey : Shift 키 클릭여부</li> </ul>
keyUp	<p>사용자가 키보드를 눌렀다 떼었을때 발생합니다.</p> <p>이벤트 상세내역</p> <ul style="list-style-type: none"> <li>- charCode : 눌러진 키보드의 문자 코드</li> <li>- controlKey : 컨트롤 키 클릭여부</li> <li>- ctrlKey : 컨트롤 키 클릭여부 (Mac일 경우)</li> <li>- keyCode : 눌러진 키보드의 key 코드</li> <li>- keyLocation : 눌러진 키보드의 key location</li> <li>- shiftKey : Shift 키 클릭여부</li> </ul>
scroll	<p>스크롤이 수행될때 발생합니다.</p> <p>이벤트 상세내역</p>

	- delta : 변화량(양수이면 아래나 오른쪽이고 음수이면 왼쪽이나 위로 이동하는 것임) - direction : 스크롤 방향(좌우스크롤인지 상하스크롤인지 여부) - position : 이벤트 반영후 스크롤 위치
sort	컬럼의 정렬이 수행될때 발생합니다.  이벤트 상세내역 - columnIndex : 정렬을 수행하는 컬럼의 index번호 - dataField : 정렬이 수행되는 필드명 - multiColumnSort : 다중정렬 여부

#### 9.4. DataGridColumnGroup

그리드에서 컬럼 그룹을 설정할 때 사용하는 컴포넌트입니다.

##### ● 속명 및 유효 값 설명

속성 명	유효 값(기본값)	설명
children	Array	컬럼그룹의 컬럼들을 정의하는 DataGridColumn의 인스턴스들의 Array 입니다.
childrenDragEnabled	Boolean(true)	그룹에서 그래그로 자식 컬럼들을 재배치 시킬 수 있을지 여부를 나타냅니다.
dataField	실 데이터의 필드명	데이터에서 해당 컬럼이 표현할 데이터의 필드명을 지정합니다.
dataTipField	실 데이터의 필드명	데이터팁(툴팁)으로 표시할 데이터의 필드명을 지정합니다.
editable	Boolean(true)	그룹컬럼의 셀에 대한 수정모드 지정
headerText	String	컬럼의 헤더 텍스트입니다.
headerWordWrap	Boolean(false)	만약 true 로 설정하면, 헤더에서 텍스트의 길이가 길어 한라인에 표시하지 못할 경우 다음 라인에 출력시킵니다.
minWidth	Number	컬럼의 가로 사이즈 최소값을 지정합니다.
resizable	Boolean(true)	True 로 설정한다면, 사용자는 드래그로 컬럼의 가로 사이즈를 재조정할 수 있습니다.
visible	Boolean(true)	컬럼의 가시성 여부를 나타냅니다.
width	Number	가로 사이즈를 결정합니다.
wordWrap	Boolean	워드랩 여부를 결정합니다.

● 스타일 및 유효 값 설명

스타일 명	유효 값(기본값)	설명
backgroundColor	RGB	배경색을 나타냅니다.
color	RGB	텍스트의 색깔을 나타냅니다.
disabledColor	RGB	잠긴상태의 텍스트 컬러를 나타냅니다.
fontFamily	폰트 이름	사용할 폰트를 결정합니다.
fontSize	Number(픽셀단위)	폰트 사이즈를 픽셀단위로 결정합니다.
fontStyle	"normal", "italic"	폰트 이탤릭체를 사용할지를 나타냅니다.
fontWeight	"normal", "bold"	폰트 볼드체를 사용할지를 나타냅니다.
headerStyleName	스타일 이름	칼럼 헤더의 스타일 이름을 나타냅니다.
letterSpacing	Number(Default 0)	각각의 글자(character) 사이의 추가적인 공간을 픽셀단위로 지정합니다.
paddingLeft	Number	왼쪽 여백을 나타냅니다.
paddingRight	Number	오른쪽 여백을 나타냅니다.
textAlign	"left", "right", "center"	텍스트 정렬 방식을 나타냅니다.
textDecoration	"none", "underline"	텍스트에 밑줄을 그을지 나타냅니다.
textIndent	Number	텍스트의 시작 들여쓰기를 픽셀단위로 지정합니다.

● 함수 명 및 설명

메소드 명	설명
clearStyle()	function clearStyle(styleProp:String):*  컴포넌트에 등록되어 있는 스타일을 지웁니다.  <b>Parameters</b> - styleProp:String 스타일 속성명
getStyle()	function getStyle(styleProp:String):*  컴포넌트에 등록되어 있는 스타일을 반환합니다.  <b>Parameters</b> - styleProp:String 스타일 속성명

	<b>Returns</b> - *: 스타일에 등록되어 있는 값, 없을시 null
setStyle()	function setStyle(styleProp:String, newValue:*) : void  컴포넌트에 스타일을 등록합니다.  <b>Parameters</b> - styleProp:String 스타일 속성명 - newValue:* 스타일에 등록할 값

## 9.5. DataGridColumn

그리드에서 컬럼을 설정할 때 사용하는 컴포넌트입니다.

- 속성 및 유효 값 설명

속성 명	유효 값(기본값)	설명
dataField	실 데이터의 필드명	데이터에서 해당 컬럼이 표현할 데이터의 필드명을 지정합니다.
dataTipField	실 데이터의 필드명	데이터팁(툴팁)으로 표시할 데이터의 필드명을 지정합니다.
dataTipJsFunction	Function	<p>셀에서 표시할 데이터팁 라벨의 콜백함수를 지정합니다. 이는 데이터팁 내용을 사용자 정의할 수 있게합니다.</p> <p>dataTipJsFunction 함수의 파라미터는 다음과 같습니다.</p> <pre>function dataTipJsFunction(item:Object, value:Object, column:Column)</pre> <p>item : 해당 행의 data 객체  value : 해당 셀의 라벨  column : 해당 셀의 열을 정의한 Column 객체</p>
editable	Boolean(true)	컬럼 수정모드 지정
editorDataField	String(text)	아이템 에디터 사용시 수정이 완료된 경우 아이템 에디터에서 결과를 받아 올 속성명.
formatter	Formatter 객체	숫자나 통화, 날짜 등의 포매터 객체를 지정하여

		셀의 표현시 변환하도록 합니다.
headerText	String	컬럼의 헤더 텍스트입니다.
headerRenderer	헤더 렌더러 객체	<p>컬럼의 헤더에 컴포박스나 html을 표현하고자 하는 경우 해당 객체의 컴포넌트명을 삽입하십시오.</p> <p>가능한 컴포넌트명은 다음과 같습니다.          HtmlHeader : html을 표현하는 헤더 렌더러          ComboBoxHeader : 콤보박스로 선택할 수 있는 기능을 가진 헤더 렌더러</p>
headerRendererDataProvider	Object	이 열의 headerRenderer에 적용할 dataProvider
headerRendererLabelField	String	이 열의 headerRenderer에 headerRendererDataProvider를 적용할 경우 라벨 필드명
headerWordWrap	Boolean(false)	만약 true 로 설정하면, 헤더에서 텍스트의 길이가 길어 한라인에 표시하지 못할 경우 다음 라인에 출력시킵니다.
itemEditor	아이템 에디터 객체	<p>컬럼에 표현되는 셀을 수정할 경우 사용할 에디터를 지정할 경우 해당 객체의 컴포넌트명을 삽입하십시오.</p> <p>가능한 컴포넌트명은 다음과 같습니다.          CheckBoxEditor : 체크박스를 표현하는 에디터          ComboBoxEditor : 콤보박스를 표현하는 에디터          DateEditor : 날짜를 수정하는 에디터          DynamicComboBoxEditor : 표시되는 list를 선택 가능한 콤보박스를 표현하는 에디터</p>
itemRenderer	아이템 렌더러 객체	<p>컬럼에 표현되는 셀에 이미지나 아이콘등을 표현하고자 하는 경우 해당 객체의 컴포넌트명을 삽입하십시오.</p> <p>가능한 컴포넌트명은 다음과 같습니다.          HtmlItem : html을 표현하는 렌더러          DataProviderItem : itemRendererDataProvider에 설정된 값에서 표시할 라벨을 찾아 표현하는 렌더러          IconItem : icon을 포함한 문자열을 표현하는 렌더러          ImageItem : 이미지를 표현하는 렌더러</p>



		IndexNoItem : 행 번호를 표현하는 렌더러 HandCursorItem : 문자열위에 마우스를 올리면 손모양 커서가 나타나는 렌더러
itemRenderDataField	String	이 열의 ItemRenderer에 itemRenderDataProvider나 itemRenderDataProviderField를 적용할 경우 데이터가 라벨과 코드데이터형식으로 되어있을때 코드를 나타내는 데이터 필드명
itemRenderDataProvider	Object	이 열의 itemRenderer에 dataProvider속성이 있을 경우 dataProvider에 적용할 데이터
itemRenderDataProviderField	String	이 열의 itemRenderer에 dataProvider속성이 있고 dataProvider에 적용할 데이터가 그리드 데이터의 특정 필드로 들어 있을 경우 해당 필드명
itemRenderLabelField	String	이 열의 ItemRenderer에 itemRenderDataProvider나 itemRenderDataProviderField를 적용할 경우 해당 데이터의 라벨 필드명
labelJsFunction	Function	셀에서 표시할 라벨의 콜백함수를 지정합니다. 이는 라벨을 사용자 정의 할 수 있게합니다.  labelJsFunction 함수의 파라미터는 다음과 같습니다. function labelJsFunction(item:Object, value:Object, column:Column) item : 해당 행의 data 객체 value : 해당 셀의 라벨 column : 해당 셀의 열을 정의한 Column 객체
minWidth	Number(20)	칼럼의 가로 사이즈 최소값을 지정합니다.
resizable	Boolean(true)	true 로 설정한다면, 사용자는 드래그로 칼럼의 가로 사이즈를 재조정할 수 있습니다.
secondDataField	String	컬럼 표현에 필요한 두번째 데이터 필드 지정. itemRenderer에 IconItem, HandCursorItem을 사용하면서, 조건에 따라 아이콘 또는 손모양 커서를 표시하는 옵션을 사용할 경우 조건을 나타내는 데이터 필드를 지정하게 됩니다.
showDataTips	Boolean(false)	칼럼에서 데이터팁을 표시할지 여부를 나타냅니다.
sortCompareFunction	String	컬럼 정렬시 사용할 비교함수를 설정합니다. 적용가능한 함수는 다음과 같습니다.

		<p>numericSort : 데이터가 XML이면서 숫자인 경우 문자열 정렬이 되므로 이를 피해야 할 경우 적용하여 숫자에 의한 정렬이 수행되도록 합니다.</p> <p>firstGroupingSort : 그룹핑 기능을 사용하며 정렬을 할 경우, 합산결과를 맨위에 표시하도록 합니다.</p> <p>lastGroupingSort : 그룹핑 기능을 사용하며 정렬을 할 경우, 합산결과를 맨아래에 표시하도록 합니다.</p> <p>firstNumericGroupingSort : 그룹핑 기능을 사용하며 정렬을 할 경우, 합산결과를 맨위에 표시하도록 합니다. (데이터가 XML일 경우)</p> <p>lastNumericGroupingSort : 그룹핑 기능을 사용하며 정렬을 할 경우, 합산결과를 맨아래에 표시하도록 합니다. (데이터가 XML일 경우)</p>
sortable	Boolean(true)	사용자가 헤더를 클릭함으로써 해당 칼럼의 데이터를 정렬시킬 수 있을지 여부를 결정합니다.
sortDescending	Boolean(false)	현재 칼럼의 정렬이 내림차순인지 오름차순인지를 나타냅니다. false일 경우 오름차순, true일 경우 내림차순입니다.
spanable	Boolean(true)	이 컬럼에 span(병합)기능의 적용여부
styleJsFunction	Function	<p>표현되는 각각의 셀의 스타일(서식)을 사용자 정의 할 수 있는 콜백함수를 설정합니다.</p> <p>styleJsFunction 함수의 파라미터는 다음과 같습니다.</p> <p>function styleJsFunction(item:Object, value:Object, column:Column)</p> <p>item : 해당 행의 data 객체</p> <p>value : 해당 셀의 라벨</p> <p>column : 해당 셀의 열을 정의한 Column 객체</p>
truncateToFit	Boolean(false)	true이면 셀의 크기가 텍스트보다 작은 경우, 텍스트를 적당한 크기로 자르고 "..." 를 붙이게 됩니다. 적용되는 아이템 렌더러에 따라 적용이 안되는 경우가 있습니다.
visible	Boolean(true)	칼럼의 가시성 여부를 나타냅니다.
width	Number(100)	컬럼의 가로 크기(픽셀단위)
wordWrap	Boolean	워드랩 여부를 결정합니다.

● 스타일 및 유효 값 설명

스타일 명	유효 값(기본값)	설명
backgroundColor	RGB	배경색을 나타냅니다.
color	RGB	텍스트의 색깔을 나타냅니다.
disabledColor	RGB	잠긴상태의 텍스트 컬러를 나타냅니다.
fontFamily	폰트 이름	사용할 폰트를 결정합니다.
fontSize	Number(픽셀단위)	폰트 사이즈를 픽셀단위로 결정합니다.
fontStyle	"normal","italic"	폰트 이탤릭체를 사용할지를 나타냅니다.
fontWeight	"normal","bold"	폰트 볼드체를 사용할지를 나타냅니다.
headerStyleName	스타일 이름	칼럼 헤더의 스타일 이름을 나타냅니다.
icon	String	<p>itemRender가 IconItem일 경우 렌더러에서 보여줄 icon을 지정합니다.</p> <p>선택 가능한 아이콘은 다음과 같습니다.            Magnifier : 돋보기 (기본)            Plus : 플러스 기호            FolderOpen : 열린 폴더 모양            FolderClosed : 닫힌 폴더 모양            Arrow : 우측 방향 화살표            Required : 빨간색 별표시            Diskette : 디스켓 모양</p>
iconPlacement	"left","right"	itemRender가 IconItem일 경우 icon의 위치를 지정합니다.
letterSpacing	Number(Default 0)	각각의 글자(character) 사이의 추가적인 공간을 픽셀단위로 지정합니다.
paddingLeft	Number	왼쪽 여백을 나타냅니다.
paddingRight	Number	오른쪽 여백을 나타냅니다.
textAlign	"left","right","center"	텍스트 정렬 방식을 나타냅니다.
textDecoration	"none","underline"	텍스트에 밑줄을 그을지 나타냅니다.
textIndent	Number	텍스트의 시작 들여쓰기를 픽셀단위로 지정합니다.

● 함수 명 및 설명

메소드 명	설명
addEventListener()	function addEventListener(type:String, listener:Function):void

	<p>type에 선언된 이벤트가 발생할 경우 이벤트를 받을 수 있는 listener함수를 등록합니다.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- type:<b>String</b> – 이벤트 종류를 표현하는 문자열</li> <li>- listener:<b>Function</b> – 이벤트 발생시 불러질 함수</li> </ul>
clearStyle()	<p>function clearStyle(styleProp:<b>String</b>):*</p> <p>컴포넌트에 등록되어 있는 스타일을 지웁니다.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- styleProp:<b>String</b> 스타일 속성명</li> </ul>
getStyle()	<p>function getStyle(styleProp:<b>String</b>):*</p> <p>컴포넌트에 등록되어 있는 스타일을 반환합니다.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- styleProp:<b>String</b> 스타일 속성명</li> </ul> <p><b>Returns</b></p> <ul style="list-style-type: none"> <li>- *: 스타일에 등록되어 있는 값, 없을시 null</li> </ul>
setStyle()	<p>function setStyle(styleProp:<b>String</b>, newValue:*) :void</p> <p>컴포넌트에 스타일을 등록합니다.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- styleProp:<b>String</b> 스타일 속성명</li> <li>- newValue:* 스타일에 등록할 값</li> </ul>
removeEventListener()	<p>function removeEventListener(type:String, listener:Function):void</p> <p>type에 선언된 이벤트에 연결된 함수 listener를 제거합니다.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- type:<b>String</b> – 이벤트 종류를 표현하는 문자열</li> <li>- listener:<b>Function</b> – 이벤트 발생시 불러질 함수</li> </ul>

● 이벤트 및 설명

이벤트	설명
headerListChange	<p>컬럼의 headerRender가 ComboBoxHeader인 경우 사용자가 콤보박스의 내용을 바꾸면 발생합니다.</p> <p>이벤트 상세내역</p> <ul style="list-style-type: none"> <li>- columnIndex : 클릭한 헤더의 컬럼 index번호</li> <li>- itemRenderer : ComboBoxHeader 객체</li> <li>- rowIndex : ComboBox내에서 사용자가 선택한 행의 index번호</li> </ul>
itemIconClick	<p>컬럼의 itemRender가 IconItem인 경우 사용자가 아이콘을 클릭하면 발생합니다.</p> <p>이벤트 상세내역</p> <ul style="list-style-type: none"> <li>- columnIndex : 클릭한 열의 index번호</li> <li>- rowIndex : 클릭한 행의 index번호</li> </ul>
itemImageClick	<p>컬럼의 itemRender가 ImageItem인 경우 사용자가 이미지를 클릭하면 발생합니다.</p> <p>이벤트 상세내역</p> <ul style="list-style-type: none"> <li>- columnIndex : 클릭한 열의 index번호</li> <li>- rowIndex : 클릭한 행의 index번호</li> </ul>

## 9.6. DataGridSelectorColumn

DataGrid의 열표시 내용을 설명하는 클래스로 행의 선택을 표시하는 기능을 가지고 있습니다.

### ● 속명 및 유효 값 설명

속성 명	유효 값(기본값)	설명
allItemSelected	Boolean(false)	전체 선택 여부.
allowAllSelection	Boolean(true)	<p>헤더의 CheckBox를 통해 전체를 선택 또는 해제할 수 있는 기능을 제공할 지 여부.</p> <p>true일 경우 헤더에 CheckBox가 나타납니다.</p>
allowMultipleSelection	Boolean(true)	<p>사용자가 여러개의 행을 선택할수 있는지 여부.</p> <p>true일 경우 CheckBox로 선택을 하게되고 false일 경우 RadioButton으로 선택을 하게</p>

		됩니다.
headerText	String	컬럼의 헤더 텍스트입니다.
headerWordWrap	Boolean(false)	만약 true 로 설정하면, 헤더에서 텍스트의 길이가 길어 한라인에 표시하지 못할 경우 다음 라인에 출력시킵니다.
minWidth	Number	컬럼의 가로 사이즈 최소값을 지정합니다.
resizable	Boolean(true)	True 로 설정한다면, 사용자는 드래그로 컬럼의 가로 사이즈를 재조정할 수 있습니다.
showDataTips	Boolean(false)	컬럼에서 데이터팁을 표시할지 여부를 나타냅니다.
selectedIndex	int	선택된 행의 데이터내의 index입니다.  기본 값은 -1이며, 아무런 선택이 없을 경우 -1이 지정됩니다.
selectedIndices	Array	선택된 행들의 데이터내의 index 배열입니다.  기본 값은 []이며, 아무런 선택이 없을 경우 []이 지정됩니다.
selectedItem	Object	선택된 행의 데이터의 참조입니다.
selectedItems	Array	선택된 행들의 데이터의 참조 배열입니다. 행의 선택 역순으로 배열에 들어갑니다.
visible	Boolean(true)	컬럼의 가시성 여부를 나타냅니다.
width	Number	가로 사이즈를 결정합니다.

● 스타일 및 유효 값 설명

스타일 명	유효 값(기본값)	설명
backgroundColor	RGB	배경색을 나타냅니다.
color	RGB	텍스트의 색깔을 나타냅니다.
disabledColor	RGB	잠긴상태의 텍스트 컬러를 나타냅니다.
fontFamily	폰트 이름	사용할 폰트를 결정합니다.
fontSize	Number(픽셀단위)	폰트 사이즈를 픽셀단위로 결정합니다.
fontStyle	"normal", "italic"	폰트 이텔렉체를 사용할지를 나타냅니다.
fontWeight	"normal", "bold"	폰트 볼드체를 사용할지를 나타냅니다.
headerStyleName	스타일 이름	컬럼 헤더의 스타일 이름을 나타냅니다.
letterSpacing	Number(Default 0)	각각의 글자(character) 사이의 추가적인 공간을 픽셀단위로 지정합니다.
paddingLeft	Number	왼쪽 여백을 나타냅니다.

paddingRight	Number	오른쪽 여백을 나타냅니다.
textAlign	"left","right","center"	텍스트 정렬 방식을 나타냅니다.
textDecoration	"none","underline"	텍스트에 밑줄을 그을지 나타냅니다.
textIndent	Number	텍스트의 시작 들여쓰기를 픽셀단위로 지정합니다.

● 함수 명 및 설명

메소드 명	설명
clearSelection()	function clearSelection():void  행 선택 내용을 초기화합니다.
clearStyle()	function clearStyle(styleProp: <b>String</b> ):*  컴포넌트에 등록되어 있는 스타일을 지웁니다.  <b>Parameters</b> - styleProp: <b>String</b> 스타일 속성명
getStyle()	function getStyle(styleProp: <b>String</b> ):*  컴포넌트에 등록되어 있는 스타일을 반환합니다.  <b>Parameters</b> - styleProp: <b>String</b> 스타일 속성명  <b>Returns</b> - *: 스타일에 등록되어 있는 값, 없을시 null
setStyle()	function setStyle(styleProp: <b>String</b> , newValue:*) :void  컴포넌트에 스타일을 등록합니다.  <b>Parameters</b> - styleProp: <b>String</b> 스타일 속성명 - newValue:* 스타일에 등록할 값

● 이벤트 및 설명

이벤트	설명
change	<p>사용자가 마우스를 클릭하여 선택사항을 바꿨을 때 발생합니다.</p> <p>이벤트 상세내역</p> <ul style="list-style-type: none"> <li>- columnIndex : 클릭한 열의 index번호</li> <li>- rowIndex : 클릭한 행의 index번호</li> </ul>

## 9.7. DataGridFooter

DataGrid의 푸터를 생성하는 클래스로 푸터의 한행을 표시하는 기능을 가지고 있습니다.

### ● 속명 및 유효 값 설명

속성 명	유효 값(기본값)	설명
columns	Array	푸터의 칼럼을 정의하는 DataGridFooterColumn의 Array 입니다.
height	Number	푸터의 높이를 지정합니다.
summaryOnlyLeaf	Boolean(false)	useDataSource가 true이고 데이터가 계층형일 경우 합산시 맨 말단만 사용하여 계산할지 여부
useDataSource	Boolean(false)	데이터가 계층형이나 그룹핑 경우 합산시 source(원본)를 사용하여 계산할지 여부
visible	Boolean(true)	푸터의 가시성 여부를 나타냅니다.

### ● 스타일 및 유효 값 설명

스타일 명	유효 값(기본값)	설명
backgroundColor	RGB	배경색을 나타냅니다.
borderColor	RGB	테두리 색
color	RGB	텍스트의 색깔을 나타냅니다.
fontFamily	폰트 이름	사용할 폰트를 결정합니다.
fontSize	Number(픽셀단위)	폰트 사이즈를 픽셀단위로 결정합니다.
fontStyle	"normal", "italic"	폰트 이텔렉체를 사용할지를 나타냅니다.
fontWeight	"normal", "bold"	폰트 볼드체를 사용할지를 나타냅니다.
letterSpacing	Number(Default 0)	각각의 글자(character) 사이의 추가적인 공간을 픽셀단위로 지정합니다.



paddingBottom	Number	아래쪽 여백을 나타냅니다.
paddingTop	Number	위쪽 여백을 나타냅니다
textAlign	"left","right","center"	텍스트 정렬 방식을 나타냅니다.
textDecoration	"none","underline"	텍스트에 밑줄을 그을지 나타냅니다.
textIndent	Number	텍스트의 시작 들여쓰기를 픽셀단위로 지정합니다.
verticalAlign	"top","middle","bottom"	텍스트의 수직 정렬

● 함수 명 및 설명

메소드 명	설명
clearStyle()	<p>function clearStyle(styleProp:<b>String</b>):*</p> <p>컴포넌트에 등록되어 있는 스타일을 지웁니다.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- styleProp:<b>String</b> 스타일 속성명</li> </ul>
getStyle()	<p>function getStyle(styleProp:<b>String</b>):*</p> <p>컴포넌트에 등록되어 있는 스타일을 반환합니다.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- styleProp:<b>String</b> 스타일 속성명</li> </ul> <p><b>Returns</b></p> <ul style="list-style-type: none"> <li>- *: 스타일에 등록되어 있는 값, 없을시 null</li> </ul>
setStyle()	<p>function setStyle(styleProp:<b>String</b>, newValue:*) :void</p> <p>컴포넌트에 스타일을 등록합니다.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- styleProp:<b>String</b> 스타일 속성명</li> <li>- newValue:* 스타일에 등록할 값</li> </ul>

## 9.8. DataGridFooterColumn

DataGrid의 푸터의 컬럼을 설정하는 클래스입니다.

● 속명 및 유효 값 설명

속성 명	유효 값(기본값)	설명
dataColumn	Object	합산등의 계산에 연관된 그리드의 컬럼.
formatter	Formatter 객체	숫자나 통화, 날짜 등의 포매터 객체를 지정하여 셀의 표현시 변환하도록 합니다.
label	String	컬럼에 표시될 라벨.
labelJsFunction	Function	<p>셀에서 표시할 라벨의 콜백함수를 지정합니다. 이는 라벨을 사용자 정의 할 수 있게합니다.</p> <p>labelJsFunction 함수의 파라미터는 다음과 같습니다.</p> <pre>function labelJsFunction(column:Column)     column : 해당 셀의 열을 정의한 DataGridFooterColumn 객체</pre>
summaryOperation	"SUM","MIN","MAX","AVG", "COUNT"	<p>수행해야할 합산. 이 속성은 다음 값중 하나를 가질 수 있습니다.</p> <p>SUM : 합계 MIN : 최소값 MAX : 최대값 AVG : 평균 COUNT : 건수</p>
wordWrap	Boolean	워드랩 여부를 결정합니다.

● 스타일 및 유효 값 설명

스타일 명	유효 값(기본값)	설명
backgroundColor	RGB	배경색을 나타냅니다.
color	RGB	텍스트의 색깔을 나타냅니다.
fontFamily	폰트 이름	사용할 폰트를 결정합니다.
fontSize	Number(픽셀단위)	폰트 사이즈를 픽셀단위로 결정합니다.
fontStyle	"normal","italic"	폰트 이텔렉체를 사용할지를 나타냅니다.
fontWeight	"normal","bold"	폰트 볼드체를 사용할지를 나타냅니다.
letterSpacing	Number(Default 0)	각각의 글자(character) 사이의 추가적인 공간을 픽셀단위로 지정합니다.
paddingLeft	Number	왼쪽 여백을 나타냅니다.

paddingRight	Number	오른쪽 여백을 나타냅니다.
textAlign	"left","right","center"	텍스트 정렬 방식을 나타냅니다.
textDecoration	"none","underline"	텍스트에 밑줄을 그을지 나타냅니다.
textIndent	Number	텍스트의 시작 들여쓰기를 픽셀단위로 지정합니다.

● 함수 명 및 설명

메소드 명	설명
clearStyle()	<p>function clearStyle(styleProp:<b>String</b>):*</p> <p>컴포넌트에 등록되어 있는 스타일을 지웁니다.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- styleProp:<b>String</b> 스타일 속성명</li> </ul>
getStyle()	<p>function getStyle(styleProp:<b>String</b>):*</p> <p>컴포넌트에 등록되어 있는 스타일을 반환합니다.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- styleProp:<b>String</b> 스타일 속성명</li> </ul> <p><b>Returns</b></p> <ul style="list-style-type: none"> <li>- *: 스타일에 등록되어 있는 값, 없을시 null</li> </ul>
setStyle()	<p>function setStyle(styleProp:<b>String</b>, newValue:*)void</p> <p>컴포넌트에 스타일을 등록합니다.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- styleProp:<b>String</b> 스타일 속성명</li> <li>- newValue:* 스타일에 등록할 값</li> </ul>

## 9.9. CurrencyFormatter

화폐단위를 표시하는 포맷터입니다.

- 속성 및 유효 값 설명

속성 명	유효 값(기본값)	설명
alignSymbol	"left","right"(left)	화폐표시 심볼의 위치
currencySymbol	String	화폐표시 심볼
decimalSeparatorFrom	String(.)	변환할 문자열에서 소수점 구분 기호 문자
decimalSeparatorTo	String(.)	변환결과 문자열에서 소수점 구분 기호 문자
precision	int(-1)	정밀도 -1이면 소수점 이하표시를 변경없이 처리하고 양수이면 해당 숫자만큼 소수점 아래를 표시합니다.
rounding	"none","up","down","nearest" (none)	반올림 처리방식 precision에 의거하여 처리할 경우 처리방식을 지정 none : 처리안함 up : 올림 down : 버림 nearest : 반올림
thousandsSeparatorFrom	String(.)	변환할 문자열에서 천단위 구분 기호 문자
thousandsSeparatorTo	String(.)	변환결과 문자열에서 천단위 구분 기호 문자
useNegativeSign	Boolean(true)	마이너스 기호 사용여부 true일 경우 '-'기호를 사용하고 false일 경우에는 괄호를 붙여줍니다.
useThousandsSeparator	Boolean(true)	천단위 기호 사용여부 true경우 매 천단위마다 천단위기호를 넣어줍니다.

## 9.10. DateFormatter

날짜를 표시하는 포맷터입니다.

- 속성 및 유효 값 설명

속성 명	유효 값(기본값)	설명
formatString	String(MM/DD/YYYY)	표시형식을 나타내는 문자열

		아래 <표 5 DateFormatter의 마스크 패턴>를 참조하세요
--	--	---------------------------------------

- DateFormatter의 날짜 포맷 마스크 패턴

패턴 문자	설명
Y	<p>년을 나타냅니다. 패턴 캐릭터의 수가 2의 경우, 년은 2 자리수에 절약할 수 있습니다. 그 이외의 경우, 4 자리수로 표시됩니다.</p> <p>다음의 예의 3번째에서는 지정된 자리수를 맞추기 위해 0이 추가됩니다.</p> <p>YY = 05          YYYY = 2005          YYYYY = 02005</p>
M	<p>월을 나타냅니다. 형식은 다음의 조건에 의해 정해집니다.</p> <ul style="list-style-type: none"> <li>패턴 문자의 수가 1개의 경우, 1 자리수 또는 2 자리수의 수치로서 해석됩니다.</li> <li>패턴 문자의 수가 2개의 경우, 2 자리수의 수치로서 해석됩니다.</li> <li>패턴 문자의 수가 3개의 경우, 생략한 텍스트로서 해석됩니다.</li> <li>패턴 문자의 수가 4개의 경우, 생략되지 않는 텍스트로 해석됩니다.</li> </ul> <p>예:          M = 7          MM = 07          MMM = Jul          MMMM = July</p>
D	<p>일을 나타냅니다. 1글자로 표현하는 것도 가능하지만 통상은 2글자 문자열로 사용합니다.</p> <p>예:          D = 4          DD = 04          DD = 10</p>
E	<p>요일을 나타냅니다. 형식은 다음의 조건에 의해 정해집니다.</p> <ul style="list-style-type: none"> <li>패턴 문자의 수가 1개의 경우, 1 자리수 또는 2 자리수의 수치로서 해석됩니다.</li> <li>패턴 문자의 수가 2개의 경우, 2 자리수의 수치로서 해석됩니다.</li> <li>패턴 문자의 수가 3개의 경우, 생략한 텍스트로서 해석됩니다.</li> <li>패턴 문자의 수가 4개의 경우, 생략되지 않는 텍스트로 해석됩니다.</li> </ul> <p>예:          E = 1          EE = 01          EEE = Mon</p>

	EEEE = Monday
A	오전/오후를 나타냅니다.
J	시각 (0 ~ 23)입니다.
H	시각 (1 ~ 24)입니다.
K	오전/오후에 단락지었을 때 각(0 ~ 11)입니다.
L	오전/오후에 단락지었을 때 각(1 ~ 12)입니다.
N	분을 나타냅니다. 예: N = 3 NN = 03
S	초를 나타냅니다. 예: SS = 30
그외	그 외의 텍스트를 패턴 문자열에 추가해 설정할 수가 있습니다. 구두점이나 숫자, 임의의 소문자를 사용할 수 있습니다. 대문자는 패턴 캐릭터로서 해석되는 경우가 있기때문에, 대문자는 사용하지 말아 주시기 바랍니다. 예: EEEE, MMM.D, YYYY at H:NN A = Tuesday, Sept. 8, 2003 at 1:26 PM

<표 5 DateFormatter 의 마스크 패턴>

## 9.11. NumberFormatter

숫자를 표시하는 포맷터입니다.

- 속성 및 유효 값 설명

속성 명	유효 값(기본값)	설명
decimalSeparatorFrom	String(.)	변환할 문자열에서 소수점 구분 기호 문자
decimalSeparatorTo	String(.)	변환결과 문자열에서 소수점 구분 기호 문자
precision	int(-1)	정밀도 -1이면 소수점 이하표시를 변경없이 처리하고 양수이면 해당 숫자만큼 소수점 아래를 표시합니다.
rounding	"none", "up", "down", "nearest" (none)	반올림 처리방식 precision에 의거하여 처리할 경우 처리방식을 지정 none : 처리안함

		up : 올림 down : 버림 nearest : 반올림
thousandsSeparatorFrom	String(,)	변환할 문자열에서 천단위 구분 기호 문자
thousandsSeparatorTo	String(,)	변환결과 문자열에서 천단위 구분 기호 문자
useNegativeSign	Boolean(true)	마이너스 기호 사용여부 true일 경우 '-'기호를 사용하고 false일 경우에는 괄호를 붙여줍니다.
useThousandsSeparator	Boolean(true)	천단위 기호 사용여부 true경우 매 천단위마다 천단위기호를 넣어줍니다.

## 9.12. NumberMaskFormatter

숫자를 주어진 형식에 따라 표시하는 포맷터입니다.

- 속성 및 유효 값 설명

속성 명	유효 값(기본값)	설명
align	"left", "right"(left)	마스크 패턴 문자열보다 데이터가 적을 경우 문자열의 정렬 기준을 지정합니다. 가령 마스크 패턴이 "###-###" 이고 데이터로 "1234"로 넣을 경우 이 속성이 "right"이면 결과는 "1-234"가 되며 "left"일 경우 "123-4"가 됩니다.
formatString	String	마스크 패턴. 입력되는 숫자를 원하는 포맷으로 변형하여 출력하게 됩니다.
numberSymbol	String(#)	마스크 패턴에서 숫자를 나타내는 문자

## 9.13. PercentFormatter

백분율을 표시하는 포맷터입니다.

- 속성 및 유효 값 설명

속성 명	유효 값(기본값)	설명
alignSymbol	"left","right"(right)	백분율표시 심볼의 위치
decimalSeparatorFrom	String(.)	변환할 문자열에서 소수점 구분 기호 문자
decimalSeparatorTo	String(.)	변환결과 문자열에서 소수점 구분 기호 문자
divideHundred	Boolean(false)	변환할 값을 100으로 나누어 계산할지 여부
multiplyHundred	Boolean(false)	변환할 값에 100을 곱하여 계산할지 여부
percentSymbol	String(%)	백분율표시 심볼
precision	int(-1)	정밀도 -1이면 소수점 이하표시를 변경없이 처리하고 양수이면 해당 숫자만큼 소수점 아래를 표시합니다.
rounding	"none","up","down","nearest" (none)	반올림 처리방식 precision에 의거하여 처리할 경우 처리방식을 지정 none : 처리안함 up : 올림 down : 버림 nearest : 반올림
thousandsSeparatorFrom	String(.)	변환할 문자열에서 천단위 구분 기호 문자
thousandsSeparatorTo	String(.)	변환결과 문자열에서 천단위 구분 기호 문자
useNegativeSign	Boolean(true)	마이너스 기호 사용여부 true일 경우 '-'기호를 사용하고 false일 경우에는 괄호를 붙여줍니다.
useThousandsSeparator	Boolean(true)	천단위 기호 사용여부 true경우 매 천단위마다 천단위기호를 넣어줍니다.

## 9.14. ContextMenu

오른쪽 마우스버튼을 눌렀을 때 나타나는 컨텍스트 메뉴를 표시합니다.

플래시에서 제공하는 기본 컨텍스트 메뉴에 사용자가 정의한 메뉴를 추가할 수 있습니다.

- 속성 및 유효 값 설명

속성 명	유효 값(기본값)	설명
clipboardMenu	Boolean(false)	클립보드 관련 메뉴 표시여부



customItems	Array	메뉴에 표시될 ContextMenuItem 객체의 배열
-------------	-------	--------------------------------

- 함수 명 및 설명

메소드 명	설명
addEventListener()	function addEventListener(type:String, listener:Function):void  type에 선언된 이벤트가 발생할 경우 이벤트를 받을 수 있는 listener함수를 등록합니다. <b>Parameters</b> - type: <b>String</b> – 이벤트 종류를 표현하는 문자열 - listener: <b>Function</b> – 이벤트 발생시 불러질 함수
removeEventListener()	function removeEventListener(type:String, listener:Function):void  type에 선언된 이벤트에 연결된 함수 listener를 제거합니다. <b>Parameters</b> - type: <b>String</b> – 이벤트 종류를 표현하는 문자열 - listener: <b>Function</b> – 이벤트 발생시 불러질 함수

- 이벤트 및 설명

이벤트	설명
menuSelect	사용자가 오른쪽 마우스를 클릭하여 메뉴를 표시하려고 할 때 발생합니다. 메뉴항목의 표시여부를 조정하여 메뉴내용을 조작할 수 있습니다.  이벤트 상세내역 없음

## 9.15. ContextMenuItem

컨텍스트 메뉴에 나타나는 메뉴항목을 표시합니다.

- 속성 및 유효 값 설명

속성 명	유효 값(기본값)	설명
------	-----------	----

caption	String	메뉴에 나타나는 메뉴명
separatorBefore	Boolean(false)	메뉴표시전에 구분라인을 표시할지 여부
visible	Boolean(true)	메뉴항목 표시 여부

- 함수 명 및 설명

메소드 명	설명
addEventListener()	function addEventListener(type:String, listener:Function):void  type에 선언된 이벤트가 발생할 경우 이벤트를 받을 수 있는 listener함수를 등록합니다. <b>Parameters</b> - type: <b>String</b> – 이벤트 종류를 표현하는 문자열 - listener: <b>Function</b> – 이벤트 발생시 불러질 함수
removeEventListener()	function removeEventListener(type:String, listener:Function):void  type에 선언된 이벤트에 연결된 함수 listener를 제거합니다. <b>Parameters</b> - type: <b>String</b> – 이벤트 종류를 표현하는 문자열 - listener: <b>Function</b> – 이벤트 발생시 불러질 함수

- 이벤트 및 설명

이벤트	설명
menuItemSelect	컨텍스트 메뉴에서 사용자가 메뉴를 선택하면 발생합니다.  이벤트 상세내역 - target : 선택된 ContextMenuItem의 인스턴스

## 9.16. SpanCellAttribute

셀에 대한 병합, 스타일, 포맷, 배경색 정보를 저장하는 컴포넌트입니다.

- 속성 및 유효 값 설명

속성 명	유효 값(기본값)	설명
backgroundColor	uint	셀의 배경색 0xFFFFFFFF 형식으로 표현합니다.
colNo	int	컬럼의 index번호
colSpan	int(1)	병합할 열의 수
formatString	String	마스크 패턴 패턴적용이 실패하는 경우에는 원래의 데이터가 그대로 표시됩니다. 아래 <표 6 SpanCellAttribute, SpanRowAttribute의 마스크 패턴>를 참조하세요
rowSpan	int(1)	병합할 행의 수
styleName	String	셀에 적용할 스타일명 레이아웃의 <Style>노드에 선언된 스타일명을 적어줍니다.

- SpanCellAttribute, SpanRowAttribute 의 마스크 패턴

종류	마스크패턴 형식
날짜	YYYY/MM/DD YYYY-MM-DD YY/MM/DD YY-MM-DD YYYY년 MM월 DD일 -기타 DateFormatter참조
숫자	0 0.0 0.00 #,##0 #,##0.00
통화	\$#,##0 (USD)#,##0.00 #,##0.0원 -통화표시는 교체가능
백분율	0% 0.0% 0.00%
해제	none

-적용된 포맷을 모두 해제합니다

<표 6 SpanCellAttribute, SpanRowAttribute 의 마스크 패턴>

## 9.17. SpanRowAttribute

행에 대한 스타일, 포맷, 배경색 정보를 저장하는 컴포넌트입니다.

- 속성 및 유효 값 설명

속성 명	유효 값(기본값)	설명
backgroundColor	uint	행의 배경색 0xFFFFFFFF 형식으로 표현합니다.
editable	Boolean(true)	행 수정 가능여부 그리드의 editable속성에 의한 편집 모드에서 클릭이나 더블클릭시에도 에디팅모드로 변환되지 안도록 합니다.
formatString	String	마스크 패턴 패턴적용이 실패하는 경우에는 원래의 데이터가 그대로 표시됩니다. 위의 <표 6 SpanCellAttribute, SpanRowAttribute의 마스크 패턴>를 참조하세요
rowHeight	Number	행의 높이(픽셀단위)
styleName	String	행에 적용할 스타일명 레이아웃의 <Style>노드에 선언된 스타일명을 적어줍니다.

## 9.18. SpanArrayCollection

병합정보를 포함하는 배열정보를 저장하는 컬렉션 컴포넌트입니다.

- 속성 및 유효 값 설명

속성 명	유효 값(기본값)	설명
autoResetRowSpan	Boolean(true)	자동 rowSpan 해제 여부 소팅이나 필터링이 적용된 경우 해당라인의 cell속성의 rowSpan정보를 1로 지정하고,

		나머지 라인의 cell 속성을 없애도록 하여 모든 데이터가 보이도록 합니다.
cellAttributeFieldSuffix	String(_attr)	<p>cell속성 필드 접두사.</p> <p>dataProvider가 array형식일때 아이템 가운데 cell 속성을 저장하고 있는 필드의 이름을 정의하기 위한 속성으로, 해당 아이템에 대해 속성을 저장하는 아이템을 계산할때 사용됩니다.</p> <p>가령 아이템의 dataField가 "price"이고, 이 설정에서 "_attr"을 줬다면, 셀 속성을 가지고 있는 아이템의 dataField는 "price_attr"이 되게 됩니다. 기본값은 _attr입니다.</p>
extractable	Boolean(true)	<p>전체 데이터를 조사하여 행,열에 대한 병합 정보를 추출할지 여부.</p> <p>데이터에 병합정보를 넣지 않고, 속성 저장 함수를 수행하여 병합 정보를 설정할 경우에는 이 속성을 이용하여 정보수집을 막아 필요없는 작업을 안하도록 합니다.</p>
length	int	(read-only)데이터의 개수
rowAttributeField	String(_ratrr)	<p>행 속성 필드명.</p> <p>SpanArrayCollection가 object array형식일때 아이템 가운데 행 속성을 저장하고 있는 필드의 이름을 정의하기 위한 속성으로, 해당 아이템에 대해 속성을 저장하는 아이템을 계산할때 사용됩니다.</p> <p>가령 이 설정에서 "_ratrr"을 줬다면, 행 속성을 가지고 있는 필드의 이름은 "_ratrr"이 되게 됩니다. 기본값은 _ratrr입니다.</p>
source	Array	SpanArrayCollection의 원천 배열.

● 함수 명 및 설명

메소드 명	설명
addCellAttributeDetailAt()	<p>function addCellAttributeDetailAt(rowNo:int, colNo:int, rowSpan:int, colSpan:int, styleName:String = null, backgroundColor:Number, formatString:String = null, editable:Boolean = true):Boolean</p> <p>셀 속성 추가.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- rowNo:<b>int</b> – 행의 index번호</li> <li>- colNo:<b>int</b> – 열의 index번호(DataGrid의 columns기준)</li> <li>- rowSpan:<b>int</b> – 병합하려는 행의 갯수</li> <li>- colSpan:<b>int</b> – 병합하려는 열의 갯수</li> <li>- styleName:<b>String</b> – 셀에 적용하려는 스타일명</li> <li>- backgroundColor:<b>Number</b> – 배경색</li> <li>- formatString:<b>String</b> – 셀에 적용할 마스크 패턴</li> </ul>
addEventListener()	<p>function addEventListener(type:String, listener:Function):void</p> <p>type에 선언된 이벤트가 발생할 경우 이벤트를 받을 수 있는 listener함수를 등록합니다.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- type:<b>String</b> – 이벤트 종류를 표현하는 문자열</li> <li>- listener:<b>Function</b> – 이벤트 발생시 불러질 함수</li> </ul>
addRowAttributeDetailAt()	<p>function addRowAttributeDetailAt(rowNo:int, styleName:String = null, backgroundColor:Number, formatString:String = null, editable:Boolean = true, rowHeight:Number):Boolean</p> <p>행 속성 추가.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- rowNo:<b>int</b> – 행의 index번호</li> <li>- styleName:<b>String</b> – 셀에 적용하려는 스타일명</li> <li>- backgroundColor:<b>Number</b> – 배경색</li> <li>- formatString:<b>String</b> – 셀에 적용할 마스크 패턴</li> <li>- editable:<b>Boolean</b> – 행의 수정가능 여부</li> <li>- rowHeight:<b>Number</b> – 행의 높이</li> </ul>
getCellAttributeAt()	<p>function getCellAttributeAt(rowNo:int, colNo:int):SpanCellAttribute</p>

	<p>셀 속성 반환</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- rowNo:<b>int</b> – 행의 index번호</li> <li>- colNo:<b>int</b> – 열의 index번호(DataGrid의 columns기준)</li> </ul> <p><b>Returns</b></p> <ul style="list-style-type: none"> <li>- SpanCellAttribute : 셀 속성 객체, 없을시 null</li> </ul>
getRowAttributeAt()	<p>function getRowAttributeAt(rowNo:int):SpanRowAttribute</p> <p>행 속성 반환</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- rowNo:<b>int</b> – 행의 index번호</li> </ul> <p><b>Returns</b></p> <ul style="list-style-type: none"> <li>- SpanRowAttribute : 행 속성 객체, 없을시 null</li> </ul>
refresh()	<p>function refresh():Boolean</p> <p>정렬이나 필터링에 의해 재계산을 수행합니다.</p>
removeAllAttribute()	<p>function removeAllAttribute():Boolean</p> <p>모든 셀, 행 속성 삭제</p>
removeCellAttributeAt()	<p>function removeCellAttributeAt(rowNo:int, colNo:int):Boolean</p> <p>셀 속성 삭제</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- rowNo:<b>int</b> – 행의 index번호</li> <li>- colNo:<b>int</b> – 열의 index번호(DataGrid의 columns기준)</li> </ul> <p><b>Returns</b></p> <ul style="list-style-type: none"> <li>- Boolean : 성공시 true, 실패시 false</li> </ul>
removeRowAttributeAt()	<p>function removeRowAttributeAt(rowNo:int):Boolean</p> <p>셀 속성 삭제</p> <p><b>Parameters</b></p>

	- rowNo: <b>int</b> – 행의 index번호  <b>Returns</b> - Boolean : 성공시 true, 실패시 false
removeEventListener()	function removeEventListener(type:String, listener:Function):void  type에 선언된 이벤트에 연결된 함수 listener를 제거합니다. <b>Parameters</b> - type: <b>String</b> – 이벤트 종류를 표현하는 문자열 - listener: <b>Function</b> – 이벤트 발생시 불러질 함수

● 이벤트 및 설명

이벤트	설명
collectionChange	컬렉션의 데이터에 변경이 일어났을때 발생합니다. (그리드의 에디팅 기능을 이용할 경우에는 발생하지 않음)  이벤트 상세내역 - kind : 변경의 내용. 다음 중 한 개가 됩니다. add : 데이터 행 추가 remove : 데이터 행 삭제 replace : 데이터 행 수정 reset : 컬렉션 자체가 수정되어 reset이 필요한 상태 update : 데이터의 여러행의 수정이 일어난 경우 - location : 데이터가 추가, 삭제, 수정된 경우 해당 행의 번호 - items : 데이터가 추가, 삭제된 경우 해당 행이 들어있는 배열

## 9.19. SpanXMLListCollection

병합정보를 포함하는 XML정보를 저장하는 컬렉션 컴포넌트입니다.

● 속성 및 유효 값 설명

속성 명	유효 값(기본값)	설명
autoResetRowSpan	Boolean(true)	자동 rowSpan 해제 여부 소팅이나 필터링이 적용된 경우 해당라인의



		cell속성의 rowSpan정보를 1로 지정하고, 나머지 라인의 cell 속성을 없애도록 하여 모든 데이터가 보이도록 합니다.
extractable	Boolean(true)	전체 데이터를 조사하여 행,열에 대한 병합 정보를 추출할지 여부.  데이터에 병합정보를 넣지 않고, 속성 저장 함수를 수행하여 병합 정보를 설정할 경우에는 이 속성을 이용하여 정보수집을 막아 필요없는 작업을 안하도록 합니다.
length	int	(read-only)데이터의 개수
source	XMLList	SpanXMLListCollection의 원천 XMLList. 자바스크립트를 통해서 가져올 경우 toXMLString()함수를 사용하여 문자열로 변환하여 가져와야 합니다.

● 함수 명 및 설명

메소드 명	설명
addCellAttributeDetailAt()	<p>function addCellAttributeDetailAt(rowNo:int, colNo:int, rowSpan:int, colSpan:int, styleName:String = null, backgroundColor:Number, formatString:String = null, editable:Boolean = true):Boolean</p> <p>셀 속성 추가.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- rowNo:<b>int</b> – 행의 index번호</li> <li>- colNo:<b>int</b> – 열의 index번호(DataGrid의 columns기준)</li> <li>- rowSpan:<b>int</b> – 병합하려는 행의 갯수</li> <li>- colSpan:<b>int</b> – 병합하려는 열의 갯수</li> <li>- styleName:<b>String</b> – 셀에 적용하려는 스타일명</li> <li>- backgroundColor:<b>Number</b> – 배경색</li> <li>- formatString:<b>String</b> – 셀에 적용할 마스크 패턴</li> </ul>
addEventListener()	<p>function addEventListener(type:String, listener:Function):void</p> <p>type에 선언된 이벤트가 발생할 경우 이벤트를 받을 수 있는 listener함수를 등록합니다.</p> <p><b>Parameters</b></p>

	<ul style="list-style-type: none"> <li>- type:<b>String</b> – 이벤트 종류를 표현하는 문자열</li> <li>- listener:<b>Function</b> – 이벤트 발생시 불러질 함수</li> </ul>
addRowAttributeDetailAt()	<p>function addRowAttributeDetailAt(rowNo:int, styleName:String = null, backgroundColor:Number, formatString:String = null, editable:Boolean = true, rowHeight:Number):Boolean</p> <p>행 속성 추가.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- rowNo:<b>int</b> – 행의 index번호</li> <li>- styleName:<b>String</b> – 셀에 적용하려는 스타일명</li> <li>- backgroundColor:<b>Number</b> – 배경색</li> <li>- formatString:<b>String</b> – 셀에 적용할 마스크 패턴</li> <li>- editable:<b>Boolean</b> – 행의 수정가능 여부</li> <li>- rowHeight:<b>Number</b> – 행의 높이</li> </ul>
getCellAttributeAt()	<p>function getCellAttributeAt(rowNo:int, colNo:int):SpanCellAttribute</p> <p>셀 속성 반환</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- rowNo:<b>int</b> – 행의 index번호</li> <li>- colNo:<b>int</b> – 열의 index번호(DataGrid의 columns기준)</li> </ul> <p><b>Returns</b></p> <ul style="list-style-type: none"> <li>- SpanCellAttribute : 셀 속성 객체, 없을시 null</li> </ul>
getRowAttributeAt()	<p>function getRowAttributeAt(rowNo:int):SpanRowAttribute</p> <p>행 속성 반환</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- rowNo:<b>int</b> – 행의 index번호</li> </ul> <p><b>Returns</b></p> <ul style="list-style-type: none"> <li>- SpanRowAttribute : 행 속성 객체, 없을시 null</li> </ul>
refresh()	<p>function refresh():Boolean</p> <p>정렬이나 필터링에 의해 재계산을 수행합니다.</p>
removeAllAttribute()	<p>function removeAllAttribute():Boolean</p>

	모든 셀, 행 속성 삭제
removeCellAttributeAt()	function removeCellAttributeAt(rowNo:int, colNo:int):Boolean  셀 속성 삭제  <b>Parameters</b> - rowNo: <b>int</b> – 행의 index번호 - colNo: <b>int</b> – 열의 index번호(DataGrid의 columns기준)  <b>Returns</b> - Boolean : 성공시 true, 실패시 false
removeRowAttributeAt()	function removeRowAttributeAt(rowNo:int):Boolean  셀 속성 삭제  <b>Parameters</b> - rowNo: <b>int</b> – 행의 index번호  <b>Returns</b> - Boolean : 성공시 true, 실패시 false
removeEventListener()	function removeEventListener(type:String, listener:Function):void  type에 선언된 이벤트에 연결된 함수 listener를 제거합니다. <b>Parameters</b> - type: <b>String</b> – 이벤트 종류를 표현하는 문자열 - listener: <b>Function</b> – 이벤트 발생시 불러질 함수
toXMLString()	function toXMLString():String  데이터의 내용을 XML문자열로 반환합니다.

● 이벤트 및 설명

이벤트	설명
collectionChange	컬렉션의 데이터에 변경이 일어났을때 발생합니다. (그리드의 에디팅 기능을 이용할 경우에는 발생하지 않음)

	이벤트 상세내역 - kind : 변경의 내용. 다음 중 한 개가 됩니다. add : 데이터 행 추가 remove : 데이터 행 삭제 replace : 데이터 행 수정 reset : 컬렉션 자체가 수정되어 reset이 필요한 상태 update : 데이터의 여러행의 수정이 일어난 경우 - location : 데이터가 추가, 삭제, 수정된 경우 해당 행의 번호 - items : 데이터가 추가, 삭제된 경우 해당 행이 들어있는 배열
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 9.20. SpanSummaryCollection

주어진 데이터를 병합하여 합산하는 기능을 제공하는 컬렉션 컴포넌트입니다.

이 컴포넌트로 그리드를 표현하는 경우 데이터가 변형되므로, 그리드를 통한 에디팅을 하실 수 없습니다.

- 속성 및 유효 값 설명

속성 명	유효 값(기본값)	설명
async	Boolean(false)	refresh함수의 비동기 실행여부.
mergingFields	Array	데이터의 병합에 사용되는 필드를 명시하는 SpanMergingField 객체의 배열입니다. 배열에서 SpanMergingField객체의 순서는 병합시 필드의 우선순위를 결정합니다.
summaries	Array	전체 데이터에 대한 합산을 정의한 SpanSummaryRow 인스턴스의 배열입니다. 한개 또는 2개이상의 SpanSummaryRow를 정의할 수 있습니다.

- 함수 명 및 설명

메소드 명	설명
addCellAttributeDetailAt()	function addCellAttributeDetailAt(rowNo:int, colNo:int, rowSpan:int, colSpan:int, styleName:String = null, backgroundColor:Number, formatString:String = null, editable:Boolean = true):Boolean  셀 속성 추가.

	<b>Parameters</b> <ul style="list-style-type: none"> <li>- rowNo:<b>int</b> – 행의 index번호</li> <li>- colNo:<b>int</b> – 열의 index번호(DataGrid의 columns기준)</li> <li>- rowSpan:<b>int</b> – 병합하려는 행의 갯수</li> <li>- colSpan:<b>int</b> – 병합하려는 열의 갯수</li> <li>- styleName:<b>String</b> – 셀에 적용하려는 스타일명</li> <li>- backgroundColor:<b>Number</b> – 배경색</li> <li>- formatString:<b>String</b> – 셀에 적용할 마스크 패턴</li> </ul>
addEventListener()	function addEventListener(type:String, listener:Function):void  type에 선언된 이벤트가 발생할 경우 이벤트를 받을 수 있는 listener함수를 등록합니다.  <b>Parameters</b> <ul style="list-style-type: none"> <li>- type:<b>String</b> – 이벤트 종류를 표현하는 문자열</li> <li>- listener:<b>Function</b> – 이벤트 발생시 불러질 함수</li> </ul>
addRowAttributeDetailAt()	function addRowAttributeDetailAt(rowNo:int, styleName:String = null, backgroundColor:Number, formatString:String = null, editable:Boolean = true, rowHeight:Number):Boolean  행 속성 추가.  <b>Parameters</b> <ul style="list-style-type: none"> <li>- rowNo:<b>int</b> – 행의 index번호</li> <li>- styleName:<b>String</b> – 셀에 적용하려는 스타일명</li> <li>- backgroundColor:<b>Number</b> – 배경색</li> <li>- formatString:<b>String</b> – 셀에 적용할 마스크 패턴</li> <li>- editable:<b>Boolean</b> – 행의 수정가능 여부</li> <li>- rowHeight:<b>Number</b> – 행의 높이</li> </ul>
cancelRefresh()	function cancelRefresh():void  refresh명령 해제.  refresh명령이 async에 의해 비동기로 실행될 경우 refresh명령을 해제합니다.
getCellAttributeAt()	function getCellAttributeAt(rowNo:int, colNo:int):SpanCellAttribute  셀 속성 반환

	<b>Parameters</b> - rowNo: <b>int</b> – 행의 index번호 - colNo: <b>int</b> – 열의 index번호(DataGrid의 columns기준)  <b>Returns</b> - SpanCellAttribute : 셀 속성 객체, 없을시 null
getRowAttributeAt()	function getRowAttributeAt(rowNo:int):SpanRowAttribute  행 속성 반환  <b>Parameters</b> - rowNo: <b>int</b> – 행의 index번호  <b>Returns</b> - SpanRowAttribute : 행 속성 객체, 없을시 null
refresh()	function refresh():Boolean  합산정보의 재계산을 수행합니다.
removeAllAttribute()	function removeAllAttribute():Boolean  모든 셀, 행 속성 삭제
removeCellAttributeAt()	function removeCellAttributeAt(rowNo:int, colNo:int):Boolean  셀 속성 삭제  <b>Parameters</b> - rowNo: <b>int</b> – 행의 index번호 - colNo: <b>int</b> – 열의 index번호(DataGrid의 columns기준)  <b>Returns</b> - Boolean : 성공시 true, 실패시 false
removeRowAttributeAt()	function removeRowAttributeAt(rowNo:int):Boolean  셀 속성 삭제  <b>Parameters</b> - rowNo: <b>int</b> – 행의 index번호

	<b>Returns</b> - Boolean : 성공시 true, 실패시 false
removeEventListener()	function removeEventListener(type:String, listener:Function):void  type에 선언된 이벤트에 연결된 함수 listener를 제거합니다. <b>Parameters</b> - type:String – 이벤트 종류를 표현하는 문자열 - listener:Function – 이벤트 발생시 불러질 함수

● 이벤트 및 설명

이벤트	설명
collectionChange	<p>컬렉션의 데이터에 변경이 일어났을 때 발생합니다. (그리드의 에디팅 기능을 이용할 경우에는 발생하지 않음)</p> <p>이벤트 상세내역</p> <ul style="list-style-type: none"> <li>- kind : 변경의 내용. 다음 중 한 개가 됩니다.             <ul style="list-style-type: none"> <li>add : 데이터 행 추가</li> <li>remove : 데이터 행 삭제</li> <li>replace : 데이터 행 수정</li> <li>reset : 컬렉션 자체가 수정되어 reset이 필요한 상태</li> <li>update : 데이터의 여러행의 수정이 일어난 경우</li> </ul> </li> <li>- location : 데이터가 추가, 삭제, 수정된 경우 해당 행의 번호</li> <li>- items : 데이터가 추가, 삭제된 경우 해당 행이 들어있는 배열</li> </ul>

## 9.21. SpanMergingField

SpanMergingField 클래스는 SpanDataGrid에서 병합할 개별 필드를 정의합니다.

● 속성 및 유효 값 설명.

속성 명	유효 값(기본값)	설명
caseInsensitive	Boolean(false)	병합시 대소문자 구분여부. 대소문자 구분을 하지 않을 경우 true를 지정합니다.
name	String	병합할 필드 명.
rowSpanable	Boolean(true)	병합되는 cell에 대한 rowSpan 적용여부.

summaries	Array	병합 필드의 합계 행을 정의한 SpanSummaryRow 인스턴스의 배열입니다. 한개 또는 2개이상의 SpanSummaryRow를 정의할 수 있습니다.
colNum	int	병합되는 cell에 대한 rowSpan을 적용할 그리드의 컬럼 번호. -1일 경우 mergingFields의 배열 순서에 따라 놓여지며, 지정될 경우 그리드의 해당 컬럼에 병합정보가 저장됩니다.

## 9.22. SpanSummaryRow

SpanSummaryRow 클래스는 SpanDataGrid 병합 행을 계산할 정보를 저장합니다.

SpanSummaryCollection 또는 SpanMergingField의 summaries속성에서 정의하여 병합 데이터를 만들어 낼 수 있습니다.

- 속성 및 유효 값 설명.

속성 명	유효 값(기본값)	설명
cellAttribute	SpanCellAttribute 객체	label이 들어간 dataField에 대한 cell 속성.
fields	Array	합계를 계산하기 위한 data 필드와 계산방식을 나타내는 SpanSummaryField 배열.
label	String	합계의 label. 지정한 labelDataField나 기본 병합 필드에 들어갈 label을 지정합니다.
labelDataField	String	label이 들어갈 dataField. 지정하지 않으면 기본 병합 필드에 label이 저장되고 특별히 다른 필드를 지정해서 넣고자 하는 경우, 필드명을 지정합니다.
labelJsFunction	Function	합계에 표시할 label을 정의할 수 있는 사용자 정의 함수.  합계를 표시할 경우에는 기본적으로 label속성으로 지정한 문자열이 표시되지만,



		<p>사용자 정의 함수를 통해 표시값을 그때그때 변경하여 지정 할 수 있습니다.</p> <p>labelJsFunction으로 지정되는 함수는 다음과 같이 데이터로 쓰이는 한개의 파라미터를 전달 받아서 label로 쓰일 문자열을 반환하게 됩니다.</p> <pre>function myLabelFunction(item:Object, summaryRow:SpanSummaryRow):String     item : 라벨을 표시하는데 필요한     DataGrid의 data item 객체     summaryRow : 함수를 정의한     SpanSummaryRow 객체</pre>
rowAttribute	SpanRowAttribute 객체	합계 행에 대한 row 속성.
summaryPlacement	"first", "last"(last)	<p>DataGrid에서 합계 라인이 표시될 위치를 설정. 가능한 값은 다음과 같습니다</p> <p>first - 병합되는 행의 맨 앞에 합계 행을 만듭니다.</p> <p>last - 병합되는 행의 맨 마지막에 합계 행을 만듭니다.</p>

### 9.23. SpanSummaryField

SpanSummaryField 클래스는 SpanSummaryRow 에서 한개의 합산 정보를 표현합니다.

개별 SpanSummaryRow 에는 한개이상의 SpanSummayField 를 포함하여 합산을 수행합니다.

- 속성 및 유효 값 설명.

속성 명	유효 값(기본값)	설명
cellAttribute	SpanCellAttribute 객체	합계를 넣는 cell에 대한 속성.
dataField	String	합산을 계산할 필드.
summaryOperation	"SUM", "MIN", "MAX", "AVG", "COUNT"(SUM)	<p>수행해야할 합산. 이 속성은 다음 값중 하나를 가질 수 있습니다.</p> <p>SUM : 합계</p> <p>MIN : 최소값</p> <p>MAX : 최대값</p>

		AVG : 평균 COUNT : 건수
targetDataField	String	계산 결과가 저장될 필드명.  SpanSummaryField의 계산 결과는 기본적으로 dataField에 저장되지만 강제로 다른 필드로 표시하고자 할 경우 이 속성을 지정해 주면 지정된 필드로 계산결과가 저장됩니다.

## 9.24. HierarchicalData

계층형(트리형) 자료를 데이터를 저장하는 컬렉션 컴포넌트입니다.

- 속성 및 유효 값 설명.

속성 명	유효 값(기본값)	설명
childrenField	String	하위 필드명 설정 자료가 XML로 되어 있으면 하위 노드를 자동으로 감지하나, XML이 아닐 경우 하위 노드의 정보를 가지고 있는 필드를 지정할 필요가 있습니다. 이때 이 속성이 설정되어 있는 경우 속성값으로 필드를 찾아 하위 노드를 찾아 갑니다. 기본값은 children 입니다.
source	Object	HierarchicalData 의 원천 데이터. 배열이나, XMLList가 될 수 있습니다. 자바스크립트를 통해서 가져올 경우 toXMLString()함수를 사용하여 문자열로 변환하여 가져와야 합니다.

## 9.25. GroupingCollection

데이터를 특정 필드에 의거해 그룹해서 계층형(트리형)으로 저장하는 컬렉션 컴포넌트입니다.

- 속성 및 유효 값 설명.

속성 명	유효 값(기본값)	설명
childrenField	String	하위 필드명 설정 자료가 XML로 되어 있으면 하위 노드를 자동으로 감지하나, XML이 아닐 경우 하위 노드의 정보를 가지고 있는 필드를 지정할 필요가 있습니다. 이때 이 속성이 설정되어 있는 경우 속성값으로 필드를 찾아 하위 노드를 찾아 갑니다. 기본값은 children 입니다.
grouping	Grouping 객체	데이터의 그룹핑에 사용되는 정보를 명시하는 Grouping 객체입니다.
source	Object	GroupingCollection 의 원천 데이터. 배열이나, XMLList가 될 수 있습니다. 자바스크립트를 통해서 가져올 경우 toXMLString()함수를 사용하여 문자열로 변환하여 가져와야 합니다.
summaries	Array	전체 데이터에 대한 합산을 정의한 SummaryRow 인스턴스의 배열입니다. 한개 또는 2개이상의 SummaryRow를 정의할 수 있습니다.

● 함수 명 및 설명.

메소드 명	설명
addEventListener()	function addEventListener(type:String, listener:Function):void  type에 선언된 이벤트가 발생할 경우 이벤트를 받을 수 있는 listener함수를 등록합니다. <b>Parameters</b> - type:String – 이벤트 종류를 표현하는 문자열 - listener:Function – 이벤트 발생시 불려질 함수
cancelRefresh()	function cancelRefresh():void  refresh명령 해제.  refresh명령이 async에 의해 비동기로 실행될 경우 refresh명령을 해제합니다.
refresh()	function refresh():Boolean

	그룹핑의 재계산을 수행합니다. 데이터가 변경된 경우 반드시 실행해야만 재계산이 일어납니다.
removeEventListener()	function removeEventListener(type:String, listener:Function):void  type에 선언된 이벤트에 연결된 함수 listener를 제거합니다. <b>Parameters</b> - type:String – 이벤트 종류를 표현하는 문자열 - listener:Function – 이벤트 발생시 불러질 함수

## 9.26. Grouping

GroupingCollection에서 그룹핑을 할 필드정보를 저장하는 컴포넌트입니다.

- 속성 및 유효 값 설명.

속성 명	유효 값(기본값)	설명
fields	Array	데이터의 그룹핑에 사용되는 필드를 명시하는 GroupingField 객체의 배열입니다. 배열에서 GroupingField객체의 순서는 그룹핑의 우선순위를 결정합니다.
label	String(GroupLabel)	그룹핑시에 그룹핑 정보 데이터를 추가할 필드명

## 9.27. GroupingField

GroupingCollection에서 그룹핑을 할 개별 필드정보를 저장하는 컴포넌트입니다.

- 속성 및 유효 값 설명.

속성 명	유효 값(기본값)	설명
caseInsensitive	Boolean(false)	그룹핑시 대소문자 구분여부. 대소문자 구분을 하지 않을 경우 true를 지정합니다.
descending	Boolean(false)	그룹핑시 역순으로 정렬할지 여부
name	String	그룹핑할 필드 명.
numeric	Boolean(false)	그룹핑할 필드의 데이터가 숫자인지 여부. false이면 숫자도 문자로 처리되어 100이

		99보다 먼저 나타나게 됩니다.
summaries	Array	그룹핑합 필드의 합계 행을 정의한 SummaryRow 인스턴스의 배열입니다. 한개 또는 2개이상의 SummaryRow를 정의할 수 있습니다.

## 9.28. SummaryRow

GroupingCollection에서 그룹핑에 의한 합산 열을 계산할 정보를 저장하는 컴포넌트입니다. GroupingCollection 또는 GroupingField의 summaries속성에서 정의하여 합산 데이터를 만들어 낼 수 있습니다.

- 속성 및 유효 값 설명.

속성 명	유효 값(기본값)	설명
fields	Array	합계를 계산하기 위한 data 필드와 계산방식을 나타내는 SummaryField 배열.
summaryPlacement	"first", "last", "group"(last)	DataGrid에서 합계 라인이 표시될 위치를 설정. 가능한 값은 다음과 같습니다 first - 그룹핑되는 행의 맨 앞에 합계 행을 만듭니다. last - 그룹핑되는 행의 맨 마지막에 합계 행을 만듭니다. group - 그룹핑을 나타내는 행에 합계행을 표시합니다.

## 9.29. SummaryField

SummaryField 클래스는 SpanSummaryRow 에서 한개의 합산 정보를 표현합니다. 개별 SummaryRow 에는 한개이상의 SummayField 를 포함하여 합산을 수행합니다.

- 속성 및 유효 값 설명.

속성 명	유효 값(기본값)	설명
dataField	String	합산을 계산할 필드.
label	String	합산 결과 저장시 사용할 필드명 입력되지 않을 경우, 기본은 dataField를

		사용하나 이 값을 넣어주면 이 값으로 필드명을 삼아 합산값을 저장하게 됩니다.
operation	"SUM","MIN","MAX","AVG", "COUNT"(SUM)	수행해야할 합산. 이 속성은 다음 값중 하나를 가질 수 있습니다. SUM : 합계 MIN : 최소값 MAX : 최대값 AVG : 평균 COUNT : 건수

### 9.30. ArrayCollection

병합정보를 포함하지 않는 배열정보를 저장하는 컬렉션 컴포넌트입니다. Layout에서 dataProvider에 특별히 지정하지 않는 경우 데이터의 유형이 배열이면 이 객체에 저장됩니다.

- 속성 및 유효 값 설명

속성 명	유효 값(기본값)	설명
length	int	(read-only)데이터의 개수
source	Array	ArrayCollection의 원천 배열.

- 함수 명 및 설명

메소드 명	설명
addEventListener()	function addEventListener(type:String, listener:Function):void  type에 선언된 이벤트가 발생할 경우 이벤트를 받을 수 있는 listener함수를 등록합니다. <b>Parameters</b> - type:String – 이벤트 종류를 표현하는 문자열 - listener:Function – 이벤트 발생시 불러질 함수
removeEventListener()	function removeEventListener(type:String, listener:Function):void  type에 선언된 이벤트에 연결된 함수 listener를 제거합니다. <b>Parameters</b> - type:String – 이벤트 종류를 표현하는 문자열

	- listener: <b>Function</b> – 이벤트 발생시 불러질 함수
--	----------------------------------------------

- 이벤트 및 설명

이벤트	설명
collectionChange	<p>컬렉션의 데이터에 변경이 일어났을때 발생합니다.</p> <p>이벤트 상세내역</p> <ul style="list-style-type: none"> <li>- kind : 변경의 내용. 다음 중 한 개가 됩니다.               <ul style="list-style-type: none"> <li>add : 데이터 행 추가</li> <li>remove : 데이터 행 삭제</li> <li>replace : 데이터 행 수정</li> <li>reset : 컬렉션 자체가 수정되어 reset이 필요한 상태</li> <li>update : 데이터의 여러행의 수정이 일어난 경우</li> </ul> </li> <li>- location : 데이터가 추가, 삭제, 수정된 경우 해당 행의 번호</li> <li>- items : 데이터가 추가, 삭제된 경우 해당 행이 들어있는 배열</li> </ul>

### 9.31. XMLListCollection

병합정보를 포함하지 않는 XML정보를 저장하는 컬렉션 컴포넌트입니다. Layout에서 dataProvider에 특별히 지정하지 않는 경우 데이터의 유형이 XML이면 이 객체에 저장됩니다.

- 속성 및 유효 값 설명

속성 명	유효 값(기본값)	설명
length	int	(read-only)데이터의 개수
source	XMLList	XMLListCollection의 원천 XMLList. 자바스크립트를 통해서 가져올 경우 toXMLString()함수를 사용하여 문자열로 변환하여 가져와야 합니다.

- 함수 명 및 설명

메소드 명	설명
addEventListener()	function addEventListener(type:String, listener:Function):void

	<p>type에 선언된 이벤트가 발생할 경우 이벤트를 받을 수 있는 listener함수를 등록합니다.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- type:<b>String</b> – 이벤트 종류를 표현하는 문자열</li> <li>- listener:<b>Function</b> – 이벤트 발생시 불러질 함수</li> </ul>
removeEventListener()	<p>function removeEventListener(type:String, listener:Function):void</p> <p>type에 선언된 이벤트에 연결된 함수 listener를 제거합니다.</p> <p><b>Parameters</b></p> <ul style="list-style-type: none"> <li>- type:<b>String</b> – 이벤트 종류를 표현하는 문자열</li> <li>- listener:<b>Function</b> – 이벤트 발생시 불러질 함수</li> </ul>
toXMLString()	<p>function toXMLString():String</p> <p>데이터의 내용을 XML문자열로 반환합니다.</p>

- 이벤트 및 설명

이벤트	설명
collectionChange	<p>컬렉션의 데이터에 변경이 일어났을때 발생합니다.</p> <p>이벤트 상세내역</p> <ul style="list-style-type: none"> <li>- kind : 변경의 내용. 다음 중 한 개가 됩니다.               <ul style="list-style-type: none"> <li>add : 데이터 행 추가</li> <li>remove : 데이터 행 삭제</li> <li>replace : 데이터 행 수정</li> <li>reset : 컬렉션 자체가 수정되어 reset이 필요한 상태</li> <li>update : 데이터의 여러행의 수정이 일어난 경우</li> </ul> </li> <li>- location : 데이터가 추가, 삭제, 수정된 경우 해당 행의 번호</li> <li>- items : 데이터가 추가, 삭제된 경우 해당 행이 들어있는 배열</li> </ul>

## 9.32. Box

DataGrid위에 메시지를 표시하기 위한 컨테이너 클래스입니다.

- 속명 및 유효 값 설명



속성 명	유효 값(기본값)	설명
alpha	Number(0~1)(기본값:1)	투명도(alpha) 값을 결정합니다.
direction	"vertical", "horizontal"	여러 객체가 포함되는 경우 레이아웃의 방향.
height	Number	세로 크기. 픽셀단위 또는 백분율로 표시할 수 있습니다.
horizontalScrollPolicy	"auto", "on", "off"(off)	가로 스크롤바 표시 여부 및 정책을 결정합니다.
verticalScrollPolicy	"on", "off", "auto"(auto)	스크롤바 생성 정책을 결정합니다. auto 인 경우 내용물이 현재 가로, 세로보다 클 경우 자동 생성 이됩니다.
visible	Boolean(true)	가시성 여부를 나타냅니다.
width	Number	가로 크기. 픽셀단위 또는 백분율로 표시할 수 있습니다.

● 스타일 및 유효 값 설명

스타일 명	유효 값(기본값)	설명
backgroundAlpha	Number(0~1)(기본값:1)	배경의 투명도(alpha) 값을 결정합니다.
backgroundColor	RGB	배경색을 나타냅니다.
borderColor	RGB	테두리 색
borderStyle	"none", "solid", "inset", "outset" (기본값:none)	테두리 스타일
borderThickness	Number(기본값:1)	테두리 두께 지정 (borderStyle이 'solid'일때만 작동).
cornerRadius	Number(기본값:5)	모서리의 둥근 정도
paddingBottom	Number	아래쪽 여백을 나타냅니다.
paddingTop	Number	위쪽 여백을 나타냅니다
paddingLeft	Number	왼쪽 여백을 나타냅니다.
paddingRight	Number	오른쪽 여백을 나타냅니다.

### 9.33. Label

DataGrid위에 메시지를 표시하기 위한 텍스트 클래스입니다.

● 속명 및 유효 값 설명

속성 명	유효 값(기본값)	설명
------	-----------	----

alpha	Number(0~1)(기본값:1)	투명도(alpha) 값을 결정합니다.
height	Number	세로 크기. 픽셀단위 또는 백분율로 표시할 수 있습니다.
htmlText	String	라벨에 표시할 html
text	String	라벨에 표시할 문자열
toolTip	String	툴팁에 표시할 문자열.
truncateToFit	Boolean(false)	true이면 주어진 크기가 텍스트보다 작은 경우, 텍스트를 적당한 크기로 자르고 "..." 를 붙이게 됩니다.
visible	Boolean(true)	가시성 여부를 나타냅니다.
width	Number	가로 크기. 픽셀단위 또는 백분율로 표시할 수 있습니다.

● 스타일 및 유효 값 설명

스타일 명	유효 값(기본값)	설명
color	RGB	텍스트의 색깔을 나타냅니다.
fontFamily	폰트 이름	사용할 폰트를 결정합니다.
fontSize	Number(픽셀단위)	폰트 사이즈를 픽셀단위로 결정합니다.
fontStyle	"normal", "italic"	폰트 이탤릭체를 사용할지를 나타냅니다.
fontWeight	"normal", "bold"	폰트 볼드체를 사용할지를 나타냅니다.
letterSpacing	Number(Default 0)	각각의 글자(character) 사이의 추가적인 공간을 픽셀단위로 지정합니다.
paddingBottom	Number	아래쪽 여백을 나타냅니다.
paddingTop	Number	위쪽 여백을 나타냅니다.
paddingLeft	Number	왼쪽 여백을 나타냅니다.
paddingRight	Number	오른쪽 여백을 나타냅니다.
textAlign	"left", "right", "center"	텍스트 정렬 방식을 나타냅니다.
textDecoration	"none", "underline"	텍스트에 밑줄을 그을지 나타냅니다.
textIndent	Number	텍스트의 시작 들여쓰기를 픽셀단위로 지정합니다.