BIRT Report Object Model – Lists & Tables

Functional Specification

Draft 1: November 29, 2004

Abstract

Describes the list and table elements of the Java Reporting Platform Report Object Model.

Document Revisions

Version	Date	Description of Changes
Draft 1	11/29/2004	First BIRT release.

Contents

1. Introduction

This specification is part of a set that describes the BIRT Report Object Model. This document explains the two elements for displaying lists of data: the List and Table element. Both elements have a data set and display some or all of the rows from that data set. The primary difference is how the content is presented. A table provides a simple, tabular presentation of the data. A list is more general, and allows the developer to include all types of report content. A table is often used for simple listings, while a list may be used for creating form letters or creating complex report layouts.

2. About Lists and Tables

2.1 Bands

A list defines a number of *bands* that print in response to specific events. Each band can contain one or more sections called *subsections*. List bands include:

The bands of a report include:

- The *header* prints once, at the beginning of the list. It can contain a matrix, a chart, or other controls. Matrices and charts display data for the entire report. Aggregates display totals for the entire report. Controls display values for the first row in the report. Data set fields display the first row from the data set.
- The *detail* prints for every row. It can contain database fields and running totals.
- The *footer* prints once, at the end of the list. It can contain the same items as the report header. Data set fields display the last row from the query.

2.1.1 Groups

Lists with more than a few rows often divide data into groups identified by a group key. Each group may have its own header and footer. Some grouped reports omit the header and simply hide the group-level fields for all except the first row for that group.

Each group has two bands:

- The *group header* prints once for each group, before the first detail row and before the first group header of any nested groups. It can contain matrices[1], charts, aggregates and data set fields. Matrices and charts display all rows within the group. Aggregates compute over the entire group. Database fields show values for the first row in the group.
- The *group footer* prints once for each group, after all detail frames for that group and any group footers for nested groups. It can contain the same items as the group header, except that database fields display values for the last row in the group.

[1] Matrices are not in the first release.

2.2 Data Binding for Lists

Every list is bound to a data set. A list also supports data sorting, data filtering, grouping, and input parameter binding.

2.2.1 Input Parameter Binding

Data sets define named input parameters. A list can bind values to the data set input parameters. This feature is often required when creating a nested report. Suppose an outer list iterates over list in one database. A nested report iterates over orders for that user from a second database. In this case, the inner list can use data set parameter bindings to select orders for the customer identified in the outer list.

2.2.2 Data Sorting

Most lists present data in a defined order called the sort order. The sort order is defined by the group keys, followed by the sorting criteria for detail rows. The list automatically influences the data set sort order to ensure that the sort criteria are met.

Groups can also be sorted. For example, customer groups can be sorted by the total sales to the customer for the quarter.[2]

2.2.3 Data Filters

A list can apply additional filters to its data set. For example, a report may want to display the list of recent transactions broken down in several ways. First, show all sales, then returns, then exchanges. Suppose that each list requires a different format. In this case, we can use three BIRT List elements. Each list has a filter that matches only the rows to appear within that one list.

Groups can also be filtered. For example, a report can show only the top 10 customers as defined by total sales within a quarter.[3]

2.2.4 Aggregates

Header and footer bands often display totals. Totals can be calculated over a set of rows: all the rows for the list, or the rows for a group. For example, sum the sales of all customers within a given sales region. Aggregates can also include a filter over the rows. For example, sum all the sales for customers with questionable credit.

2.3 Conditional Content

Many reports require differing output for an event depending on certain conditions. For example, a list of customers may want to display residential, business and government customers differently.

The developer creates conditional content by creating multiple subsections within a band. Each subsection contains the content needed for a particular condition. The developer defines a condition that hides the subsection except when it is needed.

Other reports require a combination of output. For example, when printing a statement, all customers may need the company logo and statement heading. The customer is then

[2] Group sorting is not in the first release.

[3] Group filters are not in the first release.

displayed differently as discussed above. All customers then need the same column headings.

This example also requires multiple subsections within a band. However, the developer would apply no conditions to the first and last subsections so that they always appear, but the subsections in the middle would have conditions as above.

2.4 Multiple Presentations of Data

Reports sometimes want to display the same data multiple ways. For example, a portfolio report may show holdings as a chart and as a list. A sales report may shows sales charted by time, by product area, and by sales region, each in a separate chart. An employee report may provide a list of employees by name and again by department.

The report developer simply adds elements to the report to show the data. BIRT detects the multiple presentations and automatically makes multiple passes over the result set.

The multiple presentations can occur for the entire result set, or can occur for only the contents of a group. In the portfolio report above, the report may list a series of funds, and each fund displays both a list and a chart. The report would have a list to iterate over all the data. A group would represent each fund. The list and chart would display the data within the fund group.

2.5 Charts or Tables within a List

As explained in a later section, another way to implement the above is to use a grid in a group header, and that grid contains a row with two columns. The left cell displays the chart, the right cell displays a table.

2.6 Listing Element

Abstract base element that captures the behavior common to lists and tables.

Summary

This element is abstract; it does not appear in the UI.

Base element: Report Item

Availability: First release

Properties

`sort`

The sort order for detail rows in the List.

`filter`

Filter criteria to apply to each row in the List.

Methods

`onStart`

Called before the first row is retrieved from the data set for this element.

`onRow`

Called for each row retrieved from the data set for this element, but before creating any content for that row.

`onFinish`

Called after the last row is read from the data set for this element, but before the footer band is created.

Description

The listing element is an abstract base element that captures the commonality between the List and Table element. Both elements support a data set, filtering, sorting, methods, and so on.

See the List and Table items for the specific visual differences between the two items.

See Also

Report Item for information on the data set, input parameter bindings, and so on.

2.6.1 `sort` Property

The sort order for detail rows in the List.

Summary

Display Name: Sort

ROM Type: List of Sort Criteria structures

JavaScript Type: Array of `PropertyStructure` objects

Default value: None

Inherited: Yes

Settable at runtime: No

Availability: First release

Description

Specifies the sorting of the rows within the detail band.

2.6.2 `filter` Property

Filter criteria to apply to each row in the List.

Summary

Display Name: Filter

ROM Type: List of Filter Criteria structures

JavaScript Type: Array of `PropertyStructure` objects

Default value: None

Inherited: Yes

Settable at runtime: No

Availability: First release

Description

2.6.3 `onStart` Method

Called before the first row is retrieved from the data set for this element.

Synopsis

```
element.foo( )
```

Return Type: None

Summary

Availability: First release

Context: Factory

Arguments

None

Returns

None

Description

Called before the first row is retrieved from the data set for this element. Called after the data set is open but before the header band is created. This script is a convenient place to initialize custom running total variables.

The scripts allow the application to calculate custom running totals using code. Simply:

- initialize a global variable in the on-start method (optional),
- accumulate the running total by adding values to the global variable,
- finalize the running total prior to displaying it (optional).

See Also

`onRow` method

`onFinish` method

2.6.4 onRow Method

Called for each row retrieved from the data set for this element, but before creating any content for that row.

Synopsis

```
element.on-row( )
```

Return Type: None

Summary

Availability: First release

Context: Factory

Arguments

None

Returns

Nothing

Description

Called for each row retrieved from the data set for this element, but before creating any content for that row.

See Also

`onStart` method

`onFinish` method

2.6.5 `onFinish` Method

Called after the last row is read from the data set for this element, but before the footer band is created.

Synopsis

```
element.on-finish( )
```

Return Type: None

Summary

Availability: First release

Context: Factory

Arguments

None

Returns

None

Description

Called after the last row is read from the data set for this element, but before the footer band is created.

See Also

`onStart` method

`onRow` method

2.7 Listing Group Element

Abstract element that represents the behavior common to list and table groups.

Properties

`name`

Optional name of the group.

`key`

The expression for the group key.

`interval`

The kind of grouping interval: year, month, interval, etc.

`intervalRange`

When creating intervals, this attribute allows grouping on a range of contiguous values.

`sortDirection`

The sort direction for the groups at this level.

`toc`

The TOC expression for this group.

`filter`

A filter to apply to the each group as a whole.

`sort`

Custom sort criteria to apply to the groups as a whole.

Methods

`onStart`

Called before the first row is retrieved from the data set for this element.

`onFinish`

Called after the last row is read from the data set for this element, but before the footer band is created.

Description

A list or table is most often divided into a number of groups. Groups provide a way of showing common headings for a group of related rows. Groups often have subtotals. For example, a sales report may show totals per sales rep and region.

A group is defined by a *group key.* The key is a column from the query. If the group key is a time field then user often want to group on an *interval* such as month or quarter.

2.7.1 `name` Property

Optional group name that can be used within aggregates.

Summary

Display Name: Name

ROM Type: Name

JavaScript Type: String

Default value: None

Required: No

Settable at runtime: No

Availability: First release

Description

Optional name of the group. The group name must be unique within the list or table. The name is used to identify the group in aggregates as explained in the scripting specification.

2.7.2 key Property

The expression for the group key.

Summary

Display Name: Key

ROM Type: Expression

Expression Type: any

JavaScript Type: any

Required.

Availability: First release

Description

Gives the group key as an expression. The group key says when to cause a group break listing. For example, if the group is by state, then the group key might be "`row.state`". A level break occurs each time the key changes. At each level break, the current group is ended, and a new group started. The expression most often is just a column, but can be a computed value.

2.7.3 `interval` Property

When creating intervals, this property allows grouping on a range of contiguous values.

Summary

Display Name: Interval

ROM Type: Choice

JavaScript Type: String

Default value: None

Settable at runtime: No

Availability: First release

Choices

Display Name	Internal Name	Description
None	`none`	Group on individual values.
Prefix	`prefix`	Group on a string prefix
Year	`year`	Group on a range of dates.
Quarter	`quarter`	
Month	`month`	
Week	`week`	
Day	`day`	
Hour	`hour`	
Minute	`minute`	
Second	`second`	
Interval	`interval`	Group on a numeric interval.

Description

This property allows the group to include a range of contiguous values. For example, a monthly sales report may want to summarize a list of individual transactions, each of

which has a transaction date. Using the Month interval, BIRT will cause a break to occur each time the transaction dates move into a different calendar month.

Use this property with the `intervalRange` property to select a set of contiguous values. For example, a bi-monthly report would use the Month `interval` with an `intervalRange` of 2.

See Also

`intervalRange` Property

2.7.4 `intervalRange` Property

Number of contiguous groups to merge to create a single group.

Summary

Display Name: Interval Range

ROM Type: Number

JavaScript Type: Number

Default value: 1

Settable at runtime: No

Availability: First release

Description

When creating intervals, this attribute allows grouping on a range of contiguous values. For example, to show data over the span of three hours, set the grouping interval to hours and the range to 3. The user enters the interval range here.

The following shows the meaning of the range for each interval type.

Interval	Meaning of Range
None	**Ignored**
Prefix	The number of characters in the prefix.
Date/Time	The number of the units.
Interval	The numeric interval

See Also

`interval` Property

2.7.5 `sortDirection` Property

The sort direction for the groups at this level.

Summary

Display Name: Sort Direction

ROM Type: Choice

JavaScript Type: String

Default value: Ascending

Settable at runtime: No

Availability: First release

Choices

Display Name	Internal Name	Description
Ascending	`asc`	Sort values from lowest to highest.
Descending	`desc`	Sort values from highest to lowest.

Description

The sort direction applies only if the List does not have a custom sort defined using the `sort` property. The default group sort key is the group key.

See Also

`sort` Property

2.7.6 `toc` Property

The table of contents entry for this group.

Summary

Display Name: TOC

ROM Type: Expression

Expression Type: String

JavaScript Type: String

Default value: None

Settable at runtime: No

Availability: After the first release

Description

The table of contents (TOC) expression for the group. The value of the expression appears for the table of contents for the report.

2.7.7 `filter` Property

A filter to apply to the each group as a whole.

Summary

Display Name: Filter

ROM Type: List of Filter Criteria structures

JavaScript Type: Array of `PropertyStructure` objects

Default value: None

Settable at runtime: No

Availability: After the first release

Description

A filter to apply to the each group as a whole. The filter conditions should apply to values that are constant for each group. This means that the filter expressions should be *functionally dependent* on the group key, or should be defined on aggregates computed over the group. For example, display only customers with non-zero sales totals, or display only the top 10 customers as determined by sales.

2.7.8 `sort` Property

Custom sort criteria to apply to the groups as a whole.

Summary

Display Name: Sort

ROM Type: List of Sort Criteria structures

JavaScript Type: Array of `PropertyStructure` objects

Default value: None

Settable at runtime: No

Availability: After the first release

Description

Custom sort criteria to apply to the groups as a whole. For example, the report may group customers by customer ID, but sort the customers by total sales or by customer name.

See Also

`sortDirection` property

2.7.9 `onStart` Method

Called before the first row is retrieved from the data set for this group.

Summary

Availability: First release

Context: Factory

Arguments

None

Returns

None

Called before the first row is retrieved from the data set for this group. Called before the header band is created. This script is a convenient place to initialize custom running total variables.

2.7.10 `onFinish` Method

Called after the last row for this group.

Summary

Availability: First release

Context: Factory

Arguments

None

Returns

None

Description

Called after the last row is read from the data set for this group, but before the footer band is created.

2.8 Sort Criteria Structure

A sort definition consisting of an expression and sort direction.

Summary

Availability: First release.

Properties

`key`

An expression that gives the sort key on which to sort.

`directionChoice`

The sort direction: Ascending (the default) or Descending.

Description

Report sorting and group sorting is defined as a series of column, sort direction pairs. Each pair is a Sort Criteria structure.

2.8.1 `key` Property

Summary

Display Name: Sort Key

ROM Type: Expression

Expression Type: any

Required.

Settable at runtime: No

Availability: First release

Description

An expression that gives the sort key on which to sort. The simplest case is the name of a column. The expression can also be an expression that includes columns. When used for a group, the expression can contain an aggregate computed over the group.

Expressions that include only a column name are candidates for “pushing” into the data set. However, sort keys that are expressions must be evaluated within BIRT, and so the sorting itself must occur within BIRT.

2.8.2 `direction` Property

The sort direction: Ascending (the default) or Descending.

Summary

Display Name: Sort Direction

ROM Type: Choice

JavaScript Type: String

Default value: Ascending

Settable at runtime: No

Availability: First release

Choices

Display Name	Internal Name	Description
Ascending	`asc`	Sort values from lowest to highest.
Descending	`desc`	Sort values from highest to lowest.

Description

The sort direction: Ascending (the default) or Descending.

2.9 Filter Criteria Structure

Properties

`expr`

The name of a data row column or an aggregate expression.

`operator`

The operator to apply to the expression.

`value1`

The value for simple conditions with the operators: <, <=, =, <>, >=, >, between, not between, like. Gives the “N” for the top N, Bottom N conditions. Gives the “percent” for the Top Percent and Bottom Percent conditions.

`value2`

The value for simple conditions with the operators: between, not between

Description

The filter criteria defines a filter to apply to a List, Table, Group or other element. Expressions are of the form “expression op value” or “expression op value 1, value 2”. A special case is “expression is true” in which the expression itself is a Boolean expression.

The simplest expression is comparison with a column in the data row:

```
row.State = “CA”
```

A slightly more complex comparison is to use a report parameter:

```
row.State = params.StateParam
```

Some conditions may require conditional logic:

```
( row.Balance < 0 || row.Status = “Suspended” ) is true
```

Note that the “is true” is present just to say that the entire expression should, itself, be treated as a complete condition.

Ranking expressions let the report choose only the most interesting rows:

```
row.Balance Top 10
```

Which means to display the ten customers with the highest balances.

When used with groups, a filter can refer to total computed over the group:

```
Total.sum( row.InvoiceAmt ) > 1000
```

The filter can also refer to totals computed over the entire data set, or a higher level group. To display only customers whose sales account for at least 5% of total sales:

```
Total.sum( row.InvoiceAmt ) >= Total.sum_overall( row.InvoiceAmt ) *
0.05
```

2.9.1 expr Property

Summary

Display Name: Expression

ROM Type: Expression

Expression Type: any

Required.

Settable at runtime: No

Availability: First release

2.9.2 Description

The filter expression is the name of a data row column or an aggregate expression. A filter only makes sense when computed using a data row column. BIRT accepts a value such as 1 or “hello, world”, but then the same filter criteria will be applied to every row. Such behavior may be useful when testing, but seldom in a production report.

2.9.3 operator Property

Summary

Display Name: Operator

ROM Type: Choice

JavaScript Type: String

Default value: =

Settable at runtime: No

Availability: First release

Choices

Display Name	Internal Name	Description	Value 1	Value 2
<, <=, =, <>, >=, >	<, <=, =, <>, >=, >	Simple relational operators.	✓	
is null, is not null	is-null, is-not-null	Checks if the value is null or not null.		
between, not between	between, not-between	Checks if a value is between or not between two values.	✓ (lower bound)	✓ (upper bound)
is true, is false	is-true, is-false	Checks a Boolean condition. Use these if the expression condition itself already expresses the full filter condition.		
like	like	Check the value of a column against a JavaScript regular expression.	✓ (the regular expression)	
any	any	Acts as a “no op”; the filter condition always matches.		
top N, bottom N	top-n, bottom-n	Accept the value only if it is within the top or bottom *n* values.	✓ (the *n* value)	
top percent, bottom percent	top percent, bottom percent	Accept the value only if it is within the top or bottom percentage of values.	✓ (the % value)	

Description

A filter criteria is of the form:

```
expr op value1 (value2)
```

The operator says how to test the expression. It can be a simple relational operator:

```
expr = 10
```

Or one of the other operations shown above.

The ranking operators (Top N, Bottom N, Top Percent and Bottom Percent) can include ties. For example, suppose we want the top three customers by sales. Suppose we have the following customers and sales:

A 100

B 200

C 300

D 100

E 50

F 0

A top-3 ranking would include customers C, B, A and D because A and D both have the same sales amount.

2.9.4 `value1` Property

The first (or only) operand.

Summary

Display Name: Value 1

ROM Type: Expression

Expression Type: any

Required for operators: <, <=, =, <>, >=, >, between, not between, like.

Settable at runtime: No

Availability: First release

Description

The value for simple conditions with the operators: <, <=, =, <>, >=, >, between, not between, like. Gives the “N” for the top N, Bottom N conditions. Gives the “percent” for the Top Percent and Bottom Percent conditions.

The `value1` property for a ranking operator must be constant with respect to the data set. For example, it can reference a report parameter, but not (normally) a column. The result is undefined when used with a value that varies within the data set.

2.9.5 `value2` Property

The second operator for between & not between operators.

Summary

Display Name: Value 2

ROM Type: Expression

Expression Type: any

Required for the between and not between operators.

Settable at runtime: No

Availability: First release

Description

The value for conditions with the operators between and not between. Gives the upper value of the range.

3. List Item

A list is a mechanism for iterating over the results of a data set and printing sections in response to the first row, last row and every row. Groups provide optional levels for headings and totals.

Some lists may contain no content when run. Perhaps the filter conditions matched no rows, or the data source was empty. Normally, none of the report sections print when the report is empty. However, developers often wish to display a "no data available" message to let the user know that the report worked, but couldn't find any data. The developer does this by setting a property on the header and/or footer section to display it even if (or only if) the list is empty.

3.1 Drop Sections

Users sometimes want to achieve a drop effect. For example, list a chart alongside details, or list header information alongside details.

The bold parts below are drop headings:

	Order #	Value
Customer: 123	456	$100
Matrix Link	678	$200
UK	901	$300
	902	$400
Customer: 124	400	$500
Actuate Corp.	500	$300
USA		
Customer: 125	101	$100
Fred Fin	102	$200
Australia	103	$300
	104	$400

Example with a drop chart:

DSP
Controller
Processor
Dynamic Ram
Static Ram
Driver

Hardware	Price (in $)
DSP	6,207,210
Controller	12,873,548
Processor	12,207,960
Dynamic Ram	8,785,411
Static Ram	24,587,969
Driver	8,483,880

A drop section follows these rules:

- A group or list header contains a "drop" section that is to overlay detail sections. The drop section has the Drop property set to one of the valid choices.

- The drop section is placed into the output.
- BIRT resets the vertical output position to the top of the drop section. Detail sections then are placed next to the drop section.
- Detail sections are printed until the end of the group (or list) that defined the drop section.
- If the sum of the heights of the detail sections is less than that of the drop section, BIRT moves the output position to that following the drop section. (That is, a drop section overlaps only its own detail rows, but not detail rows of later groups or lists.)

Drop sections will not be available in the first release; they require the use of Dynamic HTML and cannot be done with standard HTML that is the target of the first release.

3.2 List Item Element

This section describes the list itself.

Summary

Base element: Listing Item

Availability: First release

XML Element Name: `list`

Predefined Style Name: `list`

Contents

`header`

Sections that appears before the first row of the report.

`footer`

Sections that appears after the last row of the report.

`groups`

Groups for the report. The outermost group appears first in the list, the innermost group appears last.

`detail`

Sections to print for every row in the query.

Description

The list item creates a banded presentation of a set of data rows. The data in each band can be in any of the supported section types: grid, free-form, text, chart and so on.

A list can appear as a top-level section (or as a subsection within another top-level list). If so, then the bands fill the entire width of the page.

A list can also appear within a container (grid, table cell, free-form, etc.). In this case, the list width is fixed. The list height will grow as large as needed to display the list contents.

A list is composed of a series of bands. Each band can contain any number of sections. The sections print sequentially. Or, the user can attach conditions so that only one of the sections prints for any given list or row.

See Also

Table element

3.2.1 `header` Slot

Sections that appears before the first row of the list.

Summary

Display Name: Header

Cardinality: Multiple

Element Name: `header`

Default Style: list-header

Availability: First release

Contents

The header slot can contain any report item.

Description

The header band appears before the first detail row of a list. This band contains any number of sections. Sections appear sequentially one after another, or can be selected conditionally.

The header normally appears whether the report contains any data rows or not, but the user can change this behavior using expressions.

Report items within the header slot have visibility to the entire result set. If the header contains a simple element such as a data item, then it will display values from the first row. If the header contains a chart, then the chart displays rows from the entire data set. If the header displays an aggregate, then the aggregate is computed over the entire data set.

3.2.2 `footer` Slot

A section that appears after the last row of the report.

Summary

Display Name: Footer

Cardinality: Multiple

Element name: `footer`

Default Style: `list-footer`

Availability: First release

Contents

The header slot can contain any report item.

Description

The header band appears after the first detail row of a list. This band contains any number of sections. Sections appear sequentially one after another, or can be selected conditionally.

The footer normally appears whether the report contains any data rows or not, but the user can change this behavior using expressions.

Report items within the footer slot have visibility to the entire result set. If the footer contains a simple element such as a data item, then it will display values from the last row. If the footer contains a chart, then the chart displays rows from the entire data set. If the footer displays an aggregate, then the aggregate is computed over the entire data set.

3.2.3 `detail` Slot

Sections to print for every row in the query.

Summary

Display Name: Detail

Cardinality: Multiple

Element name: `detail`

Default Style: list-detail

Availability: First release

Contents

The header slot can contain any report item.

Description

The detail band contains the sections to print for each row within the list. The list creates a separate instance of the detail band for each row. The detail band normally contains simple elements such as a data item.

The detail band can also contain a nested list or table. In this case, the nested list or table is repeated for each row in the outer list. This is hand to create subreports. For example, outer report displays a list of customers, the inner prints a form letter that contains a list of the customer's sales this month.

3.2.4 `groups` Slot

Groups for the report.

Summary

Display Name: Groups

Cardinality: Multiple

Availability: First release

Contents

List Group element

Description

Groups for the report. The outermost group appears first in the list, the innermost group appears last. Each group has a group key, a header, a footer and more.

See Also

List Group element

3.3 List Group Element

Represents a group break level within a list.

Summary

Base element: Listing Group

Availability: First release

XML Element Name: `group`

Contents

The list is composed of a number of sections and attributes as shown below.

`header`

The header band for this group.

`footer`

The footer band for this group.

Description

A list is most often divided into a number of groups. Groups provide a way of showing common headings for a group of related rows. Groups often have subtotals. For example, a sales report may show totals per sales rep and region.

A group is defined by a *group key.* The key is a column from the query. If the group key is a time field then user often want to group on an *interval* such as month or quarter.

3.3.1 `header` Slot

The header band for this group.

Summary

Cardinality: Multiple

Element name: `header`

Default Style: `list-group-header-`*n*, where *n* is the group nesting level.

Availability: First release

Contents

Any content item except the TOC item.

Description

The header band appears before the first detail row of a group. This band contains any number of sections. Sections appear sequentially one after another, or can be selected conditionally.

Report items within the header slot have visibility to the entire result set. If the header contains a simple element such as a data item, then it will display values from the first row of the group. If the header contains a chart, then the chart displays rows from the entire group. If the header displays an aggregate, then the aggregate is computed over the entire group.

3.3.2 `footer` Slot

The footer band for this group.

Summary

Cardinality: Multiple

Element name: `footer`

Default Style: `list-group-footer-`*n*, where *n* is the group nesting level.

Availability: First release

Contents

Any content item except the TOC item.

Description

The footer band appears after the last detail row of a group. This band contains any number of sections. Sections appear sequentially one after another, or can be selected conditionally.

Report items within the footer slot have visibility to the entire result set. If the footer contains a simple element such as a data item, then it will display values from the first row of the group. If the footer contains a chart, then the chart displays rows from the entire group. If the footer displays an aggregate, then the aggregate is computed over the entire group.

4. Table

A table is a list that is structured into a rows and columns. The columns are defined for the entire table. Rows are created in response to the same events as for a list.

Like a list, a table is defined by a series of bands. A table defines the same bands as a list. Bands within a table divide into a number of rows. Each row is further divided into a set of cells.

Tables differ from Lists in how they handle the header. A List displays the header once. A Table header can optionally appear at the top of each page.

4.1.1 Style Precedence

The following rules govern how BIRT ERD locates a style property for an item in a table cell:

1. Item style
2. Row style
3. Column style
4. Table Band default style (header, footer, etc.)

CSS provides for only a small subset of style properties on columns. Specifically:[4]

> Table cells may belong to two contexts: rows and columns. However, in the source document cells are descendants of rows, never of columns. Nevertheless, some aspects of cells can be influenced by setting properties on columns.
>
> The following properties apply to column and column-group elements:

[4] CSS 2.1, Section 17.3 Columns, available at http://www.w3.org/TR/CSS21/tables.html.

'border'

The various border properties apply to columns only if 'border-collapse' is set to 'collapse' on the table element. In that case, borders set on columns and column groups are input to the conflict resolution algorithm that selects the border styles at every cell edge.

'background'

The background properties set the background for cells in the column, but only if both the cell and row have transparent backgrounds. See "Table layers and transparency."

'width'

The 'width' property gives the minimum width for the column.

4.2 Computing the Number of Columns

BIRT ERD uses rules similar to HTML 4.0 for computing the number of columns. The following is adapted from *HTML 4.01 Specification, W3C Recommendation 24 December 1999* available at http://www.w3.org/TR/html401/.

There are two ways to determine the number of columns in a table (in order of precedence):

1. If the Table element contains any column definition elements, BIRT ERD calculates the number of columns by summing the following:

   For each column definition element, take the value of its repeat attribute (default value 1).

2. Otherwise, if the Table element contains no column definition elements, BIRT ERD bases the number of columns on what is required by the rows. The number of columns is equal to the number of columns required by the row with the most columns, including cells that span multiple columns. For any row that has fewer than this number of columns, the end of that row should be padded with empty cells.

It is an error if a table contains column definition elements and the two calculations do not result in the same number of columns.

4.3 Drop Headings

BIRT tables support drop headings. These are very similar to the drop sections described in the List section above. The examples shown in that section can easily be implemented using a BIRT table.

A drop heading follows these rules:

- A drop heading can be defined only for a group header, but not for a group footer or column heading.
- A drop heading is defined as one or more cells that vertically span the group of table rows that represent the detail data rows for a group. That is, the drop property works much like the vertical span, except that the exact size of the span is computed at run time.
- A row with drop columns appears in a group header band.

- The vertical span starts with the group header. If, however, all other cells within the group header table row are empty, then the vertical span instead starts with the first detail row.
- A group header can contain two or more rows. In such a case, the drop can be defined in any of these rows. The drop will merge group header cells as well as detail cells as long as these header cells are 1) empty, and 2) appears after the row with the drop cell.
- At run time, the detail table rows are added to the table until the end of the group (or list) that defined the drop section.
- Once the total number of detail rows are known, this number is placed into the vertical span property of the drop cell, and this is the number of rows that the drop cell will span in the printed output for this one group.

Note that, by definition, each group will contain at least one detail row, and so a drop header will always appear. If the drop header is taller than the contents of the detail rows, then the CSS UA will insert extra space as needed to display the full header content. If, however, the sum of the detail rows is taller than the header content, then the header content is placed within the drop cell at the top, center, or bottom of the cell depending on the header cell's formatting properties.

Some special conditions require consideration:

- The header row that contains a drop must be the last row within a group header drop for the drop to take effect. If it is not the last row in the band, then the drop is ignored.
- The above rule applies at runtime, not design time. For example, a header may have three different group headers for a group, each with a different condition. Only one is selected for any given report group. In this case, the drop column is in effect.
- The drop setting is honored only if the detail table row has an empty cell in the same position as the group header drop column. If the detail cell is not empty, then the drop setting in the group header is ignored for that column. This decision is made on a column-by-column basis. (Said another way, the Engine computes the span as the lesser of 1) the number of contiguous blank cells under the drop cell, or 2) the total number of detail rows within the group.)
- A blank detail cell is one that contains no report items, or all contained report items are data items that have a null or empty display value (after mapping and formatting.)
- If a drop is defined on the inner-most group header, that drop cell spans the (optional) group header row, and all detail rows for that group (with the above caveats.) The drop property specifies if the drop should also span into the group footer. If it does span into the group footer, the same caveats above apply to the footer cells: they must be blank.
- A drop can apply to any group level. If the group is not the inner-most, then the drop cell will span all the group headers for any nested groups, and will span all detail items for all nested groups.
- A drop cannot be applied to a group footer.

4.4 Table Item Element

This section describes the list itself.

Summary

Base element: Listing element

Availability: First release

XML Element Name: `table`

Predefined Style Name: `table`

Properties

`repeatHeader`

Whether to repeat the headings at the top of each page.

`caption`

Text to appear as a table caption in HTML.

`columns`

A list of Column elements that describe the table columns.

Contents

`header`

Rows to appear at the top of the table, and optionally at the top of each page.

`footer`

Rows to appear at the bottom of the table.

`groups`

Level breaks within the data. Each has its own header & footer.

`detail`

Rows to display for each row in the data set.

Description

A table presents the rows from a data set in a tabular arrangement. The columns property is optional and defines the columns. The header appears at the top of the table, and on each new page. The table can contain groups. The footer displays totals and appears at the bottom of the table. The detail appears for every row in the data set.

A table can omit the data set. In this case, it displays a set of rows defined by its container. See the “Combining a List and a Table with a Single Data Set” section below for details.

4.4.1 `repeatHeader` Property

Whether to repeat the headings at the top of each page.

Summary

Display Name: Repeat Header

ROM Type: Boolean

JavaScript Type: Boolean

Default value: True

Inherited: Yes

Settable at runtime: No

Availability: First release

Description

This property specifies whether to repeat the headings at the top of each page. If true (the default), the headings repeat. If false, the header appears only on the first page.

See Also

`header` slot

4.4.2 `caption` Property

Text to appear as a table caption in HTML.

Summary

Display Name: Caption

ROM Type: Text structure

JavaScript Type: `PropertyStructure` object

Default value: None

Inherited: Yes

Settable at runtime: No

Availability: First release

This property provides text to appear as the table caption in HTML. The text can be localized. From the HTML 4.0 spec:

> When present, the CAPTION element's text should describe the nature of the table. … Visual user agents allow sighted people to quickly grasp the structure of the table from the headings as well as the caption. A consequence of this is that captions will often be inadequate as a summary of the purpose and structure of the table from the perspective of people relying on non-visual user agents.

4.4.3 `columns` Property

Describes the columns that make up the table.

Summary

Display Name: Columns

ROM Type: List of Column Definition structures

JavaScript Type: Array of `PropertyStructure` objects

Default value: None.

Inherited: Yes

Settable at runtime: No

Availability: First release

Description

This property describes the columns within the table. This property is optional. If omitted, BIRT infers the columns from the table structure itself, and BIRT will size the columns

based on their contents. If provided, then the table must contain no more than the number of columns described. Use this property when you want to control the size, color, border, or other properties of each column.

See Also

The Column Definition structure in the *ROM Layout Specification*.

4.4.4 `header` Slot

Rows to appear at the top of the table.

Summary

Cardinality: Multiple

Element name: `header`

Default Style: table-header

Availability: First release

Contents

Row elements.

Description

The table header appears at the top of the table, and optionally at the top of each page. The table header usually contains column headings. Unlike a list, the table header is not designed to display introductory information. Instead, to display such information, use the table combined with a list as described in the “Combining a List and a Table with a Single Data Set” section below.

4.4.5 `footer` Slot

Rows to appear at the bottom of the table.

Summary

Cardinality: Multiple

Element name: `footer`

Default Style: table-footer

Availability: First release

Contents

Row elements.

Description

The footer defines a set of rows to appear at the bottom of a table. The footer often contains totals.

4.4.6 `groups` Slot

A set of grouping levels within the table.

Summary

Cardinality: Multiple

Availability: First release

Contents

A list of Table Group elements. Groups appear from the most general (outermost) to the most specific (innermost).

Description

Groups provide a way of organizing data within a table. For example, a table can display sales by sales region and sales rep. The region and rep represent groups. The region is the outermost group. The rep is the innermost group.

4.4.7 `detail` Slot

A set of rows to display for each data row.

Summary

Cardinality: Multiple

Element name: `detail`

Default Style: table-detail

Availability: First release

Contents

Row elements.

Description

The detail rows appear for each row in the data set. Rows can be conditionally selected. For example, an accounting report can display a different row for debits vs. credits.

4.5 Table Group Element

Represents a group break level within a table.

Summary

Base element: Listing Group

Availability: First release

XML Element Name: `group`

Contents

The list is composed of a number of sections and attributes as shown below.

`header`

The header rows for this group.

`footer`

The footer rows for this group.

Description

A table is often divided into a number of groups. Groups provide a way of showing common headings for a group of related rows. Groups often have subtotals. For example, a sales report may show totals per sales rep and region.

A group is defined by a *group key.* The key is a column from the query. If the group key is a time field then user often want to group on an *interval* such as month or quarter.

4.5.1 `header` Slot

The header band for this group.

Summary

Cardinality: Multiple

Element name: `header`

Default Style: `table-group-header-`*n*, where *n* is the group nesting level.

Availability: First release

Contents

Row elements.

Description

The header band appears before the first detail row of a group. This band contains any number of table rows. Rows appear sequentially; or can be selected conditionally.

Report items within the header slot have visibility to the entire result set. If the header contains a simple element such as a data item, then it will display values from the first row of the group. If the header contains a chart, then the chart displays rows from the entire group. If the header displays an aggregate, then the aggregate is computed over the entire group.

See Also

Row element defined in the *ROM Layout Specification*

4.5.2 `footer` Slot

The footer band for this group.

Summary

Cardinality: Multiple

Element name: `footer`

Default Style: `table-group-footer-`*n*, where *n* is the group nesting level.

Availability: First release

Contents

Row elements.

Description

The footer band appears after the last detail row of a group. This band contains any number of table rows. Rows appear sequentially; or can be selected conditionally.

Report items within the footer slot have visibility to the entire result set. If the footer contains a simple element such as a data item, then it will display values from the first row of the group. If the footer contains a chart, then the chart displays rows from the entire group. If the footer displays an aggregate, then the aggregate is computed over the entire group.

See Also

Row element defined in the *ROM Layout Specification*

5. Combining a List and a Table with a Single Data Set

A report may wish to create a compound layout that includes both a list and a table. Such a structure is easy if both the list and table have their own data set. But, suppose that the report wants to print customer statements. A query returns the list of customers that need statements, along with the orders that are to appear on each statement. The report chooses to display a header portion of the statement that uses a free-form layout along with some text. The body of the statement is a table that shows each order. The structure is something like this:

XYZ Company

Statement for the Month of September, 2004

Bill To: ABC Corp.
123 Somewhere St.
Your Town, ST, 12345

Order No.	Date	Amount
123	8/1/2004	1234.56
145	8/9/2004	2345.67

The report design achieves this effect by using a List to iterate over the customers. The list has a group for each customer. That group has a header that displays the statement header. The header for the customer group contains a table.

```
Data Sets
    Statement Query
Body
    List (Data Set = “Statement Query”)
        Groups
            Customer Group
                Header
                    Order Table (Data Set = “”)
```

The table leaves the data set property unset, which means to “inherit” the data set from the container. The list creates the table at the start of each inner-most group, gives the table the detailed rows, and finishes the table when a group level break occurs in the list.

Specifically, in the example above, the list handles the group levels for the customer, the table displays the details of the list of orders for that customer.

The following general rules apply for this kind of structure:

- A list can contain a table in the header or footer slot of the list. If that table has a blank data set name, the table receives rows from the list.
- A list can contain a table in the header or footer slot of a table within the list. If that table has a blank data set name, the table receives rows from the group.
- Any given header or footer can contain any number of tables.

- Tables can have their own filter and sort criteria that are applied to the rows displayed in the table.
- The appearance of a table in a header or footer does not change any other part of the list. A list can contain both a table in a header, as well as a report item in the detail band.

5.1 Conditional Table Header and Footer

The combination of a List and Table is also useful to create a header or footer for a table that displays data in a format different from the table layout. Define a list that specifies the data set and the header and/or footer. Put the list into the detail band of the list. The header and footer can display overall totals for the list, and will be hidden if the data set returns no rows.

5.2 Charts and Matrices

Charts and matrices follow the same rules. Both can also appear in a header or footer. A given header or footer can also display both a chart and a table, either one after the other as report bands, or side-by-side as items within a grid.