

BIRT Report Object Model – Base Elements

Functional Specification

Draft 1: November 29, 2004

Abstract

Describes the base elements for many of the BIRT components. Report Element is the base element for all elements, while Report Item is the base element for all visual report items.

Document Revisions

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

Contents

1. Introduction	3
2. Design Element	3
2.1 User-Defined Properties.....	3
2.2 Property Element Types.....	3
2.3 Design Element.....	4
2.3.1 <i>userProperties Property</i>	5
2.3.2 <i>propertyMasks Property</i>	6
2.3.3 <i>methods Property</i>	6
2.3.4 <i>customXml Property</i>	7
3. Element Definition Structures	7
3.1 User Property Definition Structure	7
3.1.1 <i>name Property</i>	8
3.1.2 <i>displayName Property</i>	8
3.1.3 <i>UserPropertyDefn.type Property</i>	9
3.1.4 <i>choices Property</i>	10
3.2 Property Choice Structure	11
3.2.1 <i>displayName Property</i>	11
3.2.2 <i>value Property</i>	11
3.3 Property Mask Structure.....	12
3.3.1 <i>name Property</i>	12
3.3.2 <i>mask Property</i>	13
4. Report Element.....	13
4.1 Report Element.....	13
4.1.1 <i>name Property</i>	14
4.1.2 <i>extends Property</i>	14
4.1.3 <i>comments Property</i>	15
4.1.4 <i>displayName Property</i>	16
5. Report Item	16
5.1 Report Item Inheritance.....	16
5.2 Report Items	17
5.3 Report Item Element	17
5.3.1 <i>style Property</i>	19
5.3.2 <i>x Property</i>	19
5.3.3 <i>y Property</i>	20
5.3.4 <i>height Property</i>	20
5.3.5 <i>width Property</i>	21
5.3.6 <i>dataSet Property</i>	21
5.3.7 <i>bookmark Property</i>	23
5.3.8 <i>toc Property</i>	23
5.3.9 <i>visibility Property</i>	24
5.3.10 <i>bindings Property</i>	24
5.3.11 <i>onCreate Method</i>	25
5.3.12 <i>onRender Method</i>	25
6. Report Item Structures	25
6.1 Visibility Rule Structure	26
6.1.1 <i>format Property</i>	26
6.1.2 <i>expr Property</i>	27

1. Introduction

This specification defines the two elements at the root of the report element hierarchy. The Report Element is the base element for anything that major components that make up a report. A report item has a name, properties and so on. The report item element is the base element for all visual report items and includes data and style information.

This specification also defines a number of structures used by these two base items.

2. Design Element

The design element is an internal, abstract element used to implement basic features of ROM elements. It provides the following:

- User-defined property definitions
- Property values as name/value pairs
- Customization of which properties can appear in the UI
- Methods
- Custom XML

The design element provides a framework for customizing properties. The developer can add application-specific properties. These properties appear in the JRP ERD property sheet identically to ROM-defined properties. The developer can identify the property type using a wide variety of choices. JRP ERD automatically displays a property builder UI based on the property type. The developer can also hide JRP ERD or custom-defined properties.

The developer can add custom properties to a component, and can hide ROM-defined properties. Together, these help the developer create highly customized, reusable components tailored to the needs of a given application.

2.1 User-Defined Properties

As noted above, JRP ERD elements can define custom properties. A property has two parts: a *definition* and a *value*. The definition provides the name, type and other descriptive information. The value is something that another developer enters to set the property. For example, suppose the developer defines an “AlertColor” property for a library component. A developer who uses this element might set the value of the property to “Red.”

See the User Property Definition Structure below for details.

2.2 Property Element Types

ROM defines a number of property elements:

property

The value of most simple parameters. ROM determines the type of the value by consulting a set of property meta-data.

```
<property name="propName">value</property>
```

xml-property

The value of a property that contains custom XML. This property is often used by extensions to store its information.

```
<xml-property name="propName">value</xml-property>
```

property-list

For any property that has a list of properties such as custom colors, bindings, and so on.

```
<property-list name="propName">
  [ <structure>...</structure> ] *
</property-list>
```

expression

For any property defined by an expression. (Intended for future use.)

```
<expression name="propName">value</expression>
```

ex-property

An extended property mostly for use with extensions that don't have well-formed property names:

```
<ex-property
  <name>propName</name>
  <value>value</value>
</ex-property>
```

structure

For a property defined by a structure (a collection of two or more properties.)

```
<structure name="propName">
  <property name="member1">value1</property>
  <property name="member2">value2</property>
</structure>
```

2.3 Design Element

Abstract element that represents anything that has properties.

Summary

Display Name: Design Element (however, the GUI does not show abstract elements)

JavaScript Design-time Object: DesignElementDefn

JavaScript Runtime Object: DesignElement

Availability: First release, except as noted below.

XML Summary

```
<derivedElement>
  properties & methods as described above
</derivedElement>
```

Properties

userProperties[]

List of user-defined properties defined on this element. Null if no user-defined properties are available. Each item in the array is a UserProperty object.

`propertyMasks[]`

List of user-defined property masks. Null if no masks are defined. Each item in the array is a `PropertyMask` object.

`methods[]`

An associative array of method names. The method name is the key into the array. Returns a String that holds the method text, or null if no method is defined.

`propName`

Provides the value of a property given a property name. Design Element properties appear directly as properties on the `DesignElementDefn` JavaScript object. Used to access all properties: those defined by ROM, those added through extensions, and user-defined properties. The type of the returned value depends on the property definition. Returns null if the property is not set.

`custom`

Returns the value of the custom XML, if any, set for this element. Returns a String if the XML is provided, null otherwise.

Description

The `DesignElementDefn` object is available to your scripts. It provides the information about the design of an object. Every ROM element has design information, and many have separate run-time information. The design object provides access to the information defined in the XML schema file. It is most useful to get access to user-defined properties and property values so that scripts can customize report behavior.

Not all properties are available for all kinds of elements. See the derived element definitions for details.

Some of the elements derived from this one support inheritance. If so, then the value of the above properties include values inherited up the inheritance chain.

2.3.1 `userProperties` Property

List of user-defined properties for the element.

Synopsis

`DesignElementDefn.userProperties[]`

Summary

Display Name: User-Defined Properties

ROM Type: List of User Property Definition structures

JavaScript Type: Array of `UserPropertyDefn` objects

Default value: None

Inherited: Yes

Settable at runtime: No

Availability: First release

Description

The list of user-defined properties defined for this element.

See Also

User Property Definition structure

2.3.2 propertyMasks Property

List of property masks.

Synopsis

```
DesignElementDefn.propertyMasks[ ]
```

Summary

Display Name: Property Masks

ROM Type: List of Property Mask structures

JavaScript Type: Array of `PropertyMaskDefn` objects

Default value: None

Inherited: Yes

Settable at runtime: No

Availability: After the first release.

Description

Property masks allow the user to customize the set of properties available for a reusable component.

See Also

Property Mask structure

2.3.3 methods Property

List of methods.

Synopsis

```
DesignElementDefn.methods[ ]
```

Runtime Scripting

```
element.methodName
```

Summary

Display Name: Methods

JavaScript Type: Array of String

Default value: None

Inherited: Yes

Settable at runtime: No

Availability: First release

XML Summary

Methods are defined in XML using the method element:

```
<method name="methodName">code</method>
```

The method name must match one of the methods defined in this specification. The code can be any valid JavaScript script.

Description

List of methods defined on the element. The array is indexed by method name. The method code appears inside the array.

At design time, the methods are simply a list of JavaScript scripts. At design time, the methods are actual JavaScript methods on the runtime object.

Methods are designed to be called by the BIRT Report Engine at interesting events. They are usually not meant to be called by user-defined scripts.

The methods available to an element are described on the derived elements.

2.3.4 customXml Property

Custom XML associated with an element.

Summary

Display Name: Custom XML

ROM Type: XML

JavaScript Type: String

Default value: None

Inherited: Yes

Settable at runtime: No

Availability: First release

Description

Applications can associate their own XML with any ROM report element. BIRT ignores this XML other than to make it available to user scripts. User scripts can use this XML to provide custom report behavior.

The XML must be valid. XML elements must be balanced. The XML should use a third-party name space, but the name space is not required.

See Also

User-defined properties for another way to customize an element.

3. Element Definition Structures

The following property structures are used by Design Element.

3.1 User Property Definition Structure

Defines a user-defined property.

Summary

Availability: Supported in ROM in the first release, but not in the ERD UI.

Properties

name

The internal name of the property.

displayName

The name to show for this element in the UI. Display names are most often used for components put into libraries or templates, or that will be used for the future web reporting tool. The name can be externalized.

type

The type of property.

choices

A list of available choices if the property type is Choice.

Description

This structure defines a user-defined property. The user defined property acts just like a system-defined property. It appears in the ERD's property sheet, and it is available at runtime for use in scripts.

A property has an internal name, which is used in the `property` element in the XML file, and is used in scripting. It also has an optional display name that appears in the GUI. The display name can be externalized and translated. If no display name is provided, then the internal name is shown to the user.

A user-defined property also has a type. The available types are shown below. If the type is "Choice", then a list of available choices is also required.

Set an optional default value using the property value element described above.

3.1.1 name Property

The internal name of the property.

Summary

Display Name: Name

ROM Type: Name

Required.

JavaScript Type: String

Availability: First release

Description

Each property has an internal name. This is the name used when saving the property to the XML file, and when accessing the property through scripts. The name must be unique within the properties for this element. The name must be a valid JavaScript identifier; it cannot contain spaces or punctuation.

3.1.2 displayName Property

The display name of the user-defined property. Can be externalized.

Summary

Display Name: Display Name

ROM Type: Text structure

Default value: internal name

Availability: First release

Description

The name to show for this element in the UI. Display names are most often used for components put into libraries or templates, or that will be used for the future web reporting tool. The name can be externalized.

See Also

Static Text ROM data type

3.1.3 UserPropertyDefn.type Property

The data type of a user-defined property.

Summary

Display Name: Type

ROM Type: Choice

Default value: String

JavaScript Type: String

Availability: First release

Choices

Display Name	XML Name	JavaScript Type of Property Value	Description
String	string	String	Simple data types
Number	number	Number	
Integer	integer	Number	
Float	float	Number	
Date-Time	date-time	Date	
Boolean	boolean	Boolean	A ROM expression. Can reference columns, aggregates and so on.
Expression	expression	any	
Choice	choice	String	
Text	text	TextDefn	
Dimension	dimension	String	
Color	color	String	A value that is an RGB color value in CSS, XML or Java format, or one of the recognized CSS or report-defined color names.
HTML	html	TextDefn	Text with embedded HTML formatting.
URI	uri	String	A string that gives a Uniform Resource Indicator (URI).
XML	xml	String	

Description

The type of property. The default is String.

3.1.4 choices Property

List of choices for a user-defined property.

Synopsis

```
UserProperty.choices[ ]
```

Summary

Display Name: Choices

ROM Type: List of Choice Structures

JavaScript Type: Array of ChoiceDefn

Availability: First release

Description

Provides a list of choices for the property as a list of Value Choice elements. The list provides a set of choices from which the user can select, or the user can enter some other value not on the list. Required if the property is a choice, optional otherwise.

Choices give the end user a set of pre-defined value choices. Choices can be open or close-ended. Open-ended choices are suggestions: the user can choose one of the choices, or can enter some other value. Closed-ended choices are the only legal values. The developer indicates a set of closed-ended choices by setting the property type to "choice." For all other property types, the choice list is open-ended.

3.2 Property Choice Structure

Defines a choice for a user-defined property.

Properties

`displayName`

The value to display to the developer in the UI. Can be externalized. If omitted, then the value is displayed.

`value`

The value to use internally to represent this choice.

Description

User-defined properties can provide the user with a list of choices. The choices can be externalized and translated. The value is required; the display name is optional.

3.2.1 `displayName` Property

The value to display to the developer in the UI.

Summary

Display Name: Display Name

ROM Type: Text structure

Default: value

Availability: First release

Description

Provides a display name to appear for the choice in the GUI. The display name can be externalized and translated. The display name is optional. If omitted, the value is displayed instead.

See Also

Static Text ROM data type

3.2.2 `value` Property

The value to use internally to represent this choice.

Summary

Display Name: Value

ROM Type: Any. (Must match the type of the user-defined property.)

JavaScript Type: any

Required.

Availability: First release

Description

Each choice represents a discrete value. For example, if the property represents a list of customer types, the display names may be “Commercial”, “Residential”, and “Government”, while the values may be “C”, “R” and “G”.

3.3 Property Mask Structure

Allows a third-party developer to lock or hide a property.

Summary

Availability: After the first release.

Properties

name

The name of any property.

mask

The action to take for the property.

Description

A property mask says how to hide or lock a property. Applies to any kind of property: ROM-defined, user-defined, or extension-defined. The mask is used when creating a reusable component, it allows the developer to hide properties that the consumer does not need or should not change.

The mask is not meant to be a security rule; it is instead designed to help a component designer customize the property sheet UI. Because of this, a mask on a derived element can hide a mask on a base element: element A can lock a property. Element B can derive from A and unlock the property.

3.3.1 name Property

The name of a property.

Summary

Display Name: Name

ROM Type: Name

JavaScript Type: String

Required

Availability: After the first release

Description

The name of the property to mask. The name must match an valid property.

3.3.2 `mask` Property

The action to take for the property.

Summary

Display Name: Mask

ROM Type: Choice

JavaScript Type: String

Default: Hide

Availability: After the first release

Choices

Display Name	Internal Name	Description
Change	change	Allow the user to change the property.
Lock	lock	Displays the property, but don't allow changes.
Hide	hide	Hides the property. (Default)

Description

The action to take for the property.

4. Report Element

A report element represents anything that can be named and customized. Most of the major components in ROM derive from the Report Element.

4.1 Report Element

Abstract element for user-visible elements.

Summary

Display Name: Report Element (however, the GUI does not show abstract elements)

Base Element: Design Element

JavaScript Design-time Object: `ReportElementDefn` (extends `DesignElementDefn`)

JavaScript Runtime Object: `ReportElement` (extends `DesignElement`)

Availability: First release

Properties

`name`

A name for the element used internally in designer and in user code.

`extends`

Identifies the “parent” element by name.

comments

Information about this element for people who work on the design.

displayName

The localized name to show for this element in the UI.

Description

This abstract element is the base element for report elements that have a name and can be extended. These are the elements visible in the GUI: data sets, styles, master pages, report items and so on.

4.1.1 name Property

Internal element name used to reference the element within the report design and in scripts.

ROM Summary

Display Name: Name

ROM Type: Element Name

JavaScript Type: String

Default value: None

Inherited: No

Settable at runtime: No

Availability: First release

XML Summary

```
<elementName name="value" ...
```

Description

ROM report elements can have a name. The name is required for some elements, but is optional for most others. Items that require a name are those that exist to be referenced elsewhere: parameters, data sources, data sets, styles, and so on. Other elements are generally not referenced. The developer provides a name only if the s/he finds it convenient to do so. Names also make it easier to work with components in custom scripts. If the name is omitted, the component is anonymous but can still be found in code by traversing the design hierarchy. See the description of the derived concrete elements for details.

This property provides the name for the element used internally in designer and in user scripts. The name cannot be localized. As noted above, the name is optional for most elements. If the name is provided, it must be unique with its name space (as defined for each element.) Names can contain any character from any language, including spaces, punctuation and so on. ROM names are case-sensitive.

See Also

ReportElementDefn.displayName property

4.1.2 extends Property

Allows an element to extend another element of the same type.

Summary

Display Name: Extends

ROM Type: Element Reference

JavaScript Type: String

Default value: None

Inherited: No

Settable at runtime: No

Availability: First release

XML Summary

```
<elementName extends="value" ...
```

Description

Most elements can inherit properties, styles and behavior from another element of the same type. For example, a label can inherit from another label. This property identifies the “parent” element by name. The parent must exist. If it does not, the ERD will display an error and the report won’t run.

ROM uses prototype-based inheritance, much like JavaScript. Suppose derived element D extends parent element P. The *effective value* of the property of element D is found by first looking for the property value in the definition of element D. If the property is not set in element D, then BIRT repeats the search in element P. P may in turn extend yet another element, and the search repeats for P’s parent. The search terminates if when there are no more parent elements to search. In this case, the property takes its default value as defined in this specification.

Style properties use an additional level of search as defined in the style specification.

Some properties do not inherit. These include the Extends property itself, and the Name property. The specification for each property identifies whether or not it inherits.

4.1.3 comments Property

Provides design-time comments about the element.

Summary

Name: Comments

Type: String

JavaScript Type: String

Default value: None

Inherited: No

Settable at runtime: No

Availability: After the first release

Description

The comments provide information about this element for people who work on the design. This text can contain embedded HTML formatting tags; but cannot be localized.

4.1.4 displayName Property

Provides a localized, user-visible name for the element.

Summary

Display Name: Display Name

ROM Type: Text structure

Inherited: No

Default value: None. (The GUI will use the element name if no display name is provided.)

Settable at runtime: No

Availability: After the first release

Description

The name to show for this element in the UI. Display names are most often used for components put into libraries, or that will be used for the future web reporting tool. The name can be externalized.

The display name is most often provided for reusable components, especially those to be used by casual users within the web report designer.

See Also

Report Element `name` property

Text structure

5. Report Item

The report item type is the common base type for all visual report elements. Report items include a style. The style provides visual characteristics for anything that prints in a report: a section or report item.

Report items are positioned within the report based on a series of rules described in the ROM Layout Specification. The free-form item provides absolute (x, y) positioning, but most other containers provide some kinds of automated positioning. The report item provides properties for each kind of layout, but the properties are used only when appropriate. For example, every item can have an (x, y) position, but the position is ignored unless the item appears within a free-form container.

5.1 Report Item Inheritance

Report items can extend other report items. However, to keep the system manageable, ROM allows a report item to extend only those items defined in:

- The Components (“private library”) are of the design file, or
- In a library.

A report item cannot extend another report item that appears in the Body area of the report design.

These rules avoid complex designs such as having a container extend the container that contains itself. (A determined user can still create this scenario within a library, however.)

5.2 Report Items

ROM reports contain *report items* that display specific data values, images, labels, and so on. ROM defines the following report items defined in later sections:

- **Label:** displays static text that can be translated into multiple languages.
- **Data item:** displays data from database columns, expressions and so on.
- **Text item:** displays text with embedded formatting directives, and that typically includes placeholders to indicate where to merge in database data. (Similar to an e.RD-Pro rich text control.)
- **Image:** displays a variety of formats.
- **Line and rectangle:** provides graphics within a report. (Not in the first release.)
- **Chart:** display business graphics such as line charts, pie charts, and so on.
- **Matrix:** present a tabular view of data. (Not in the first release.)
- **Table-of-contents item:** lets the user insert a table of contents directly into the report. (Not in the first release.)
- **Browser control:** allows the developer to create custom web-page content. (Not in the first release.)
- **List:** provides a typical banded report.
- **Table:** banded report organized into rows and columns.
- **Free-form:** holds an arbitrary collection of report items at arbitrary (x, y) positions. (Not in the first release.)
- **Grid:** provide a static, table-oriented layout of report items.
- **Include:** imports another design file into the report. (Not in the first release.)
- **Extended item:** allows the developer to specify his own custom item.

5.3 Report Item Element

Abstract base element for all visual report items.

Summary

Display Name: Report Item

Base element: Report Element

Availability: First release

Design object: `ReportItemDesign`

Runtime object: `ReportItem`

Properties

`style`

Name of a shared style applied to this item.

`styleProp`

All properties defined for a style appear as properties on report items also. See the Style element for the list of properties that appear here.

`x`

The x position of the item.

`y`

The y position of the item.

`height`

Height of the item when it appears inside a container.

`width`

Width of the item when it appears inside a container.

`dataSet`

Reference to the data set to use for this component.

`bookmark`

Many report items can be the target of hyperlinks. The bookmark is a string that identifies the item location.

`toc`

A table of contents entry for this item.

`visibility`

A list of visibility rules

`bindings`

Bindings between input parameters in the included design and values in the main design. A list of Binding items.

Methods

`onCreate`

Executed when the element is created in the Factory.

`onRender`

Executed when the element is prepared for rendering in the Presentation engine.

Description

This element provides properties common to all visual report items.

The inherited name is optional for report items, except for those that are designed for reuse.

Each report item also has a private style. The private style is a separate element in the XML definition. However, within the UI and scripting, the style properties appear as though they were properties of this item itself.

5.3.1 *style* Property

Name of the style applied to this item.

ROM Summary

Display Name: Style

ROM Type: Element Reference

JavaScript Type: String

Default value: None

Inherited: Yes

Settable at runtime: Yes

Availability: First release

Description

Report items provide many style properties. The style properties are defined in the *ROM Styles Specification*. An element can inherit its style properties from a parent element, from a named style, or from a style that identifies the context in which the item appears. This property provides the name of the shared style to use.

This item can also set the value of style properties. Properties set on this element “override” those inherited from a parent element or a shared style.

See Also

Style element in the *ROM Styles Specification*.

5.3.2 *x* Property

The x (horizontal) position of the item.

Summary

Display Name: X

ROM Type: Dimension

JavaScript Type: String

Inherited: Yes

Default value: 0

Settable at runtime: Yes

Availability: After the first release (used only for the Free-form container.)

Description

The x position of the item. This is an absolute dimension relative to from the left of the container. Dimensions include both a measure and units: “10in” or “5cm”. The position can also be relative: “10%”.

The position is used only for containers that provide absolute positioning. Such containers include the free-form container and the graphic master page. Otherwise, the position is computed automatically by the container, and this property is ignored.

See Also

Dimension property type

Report Item y property

5.3.3 y Property

The y (vertical) position of the item.

Summary

Display Name: Y

ROM Type: Dimension

JavaScript Type: String

Default value: 0

Inherited: Yes

Settable at runtime: Yes

Availability: After the first release (used only for the Free-form container.)

Description

The x position of the item. This is an absolute dimension relative to from the top of the container. Dimensions include both a measure and units: "10in" or "5cm". The position can also be relative: "10%".

The position is used only for containers that provide absolute positioning. Such containers include the free-form container and the graphic master page. Otherwise, the position is computed automatically by the container, and this property is ignored.

See Also

Dimension property type

Report Item x property

5.3.4 height Property

Height of the item.

Summary

Display Name: Height

ROM Type: Dimension

JavaScript Type: String

Default value: 0

Inherited: Yes

Settable at runtime: Yes

Availability: First release (limited cases, see the specific report items for details.)

Description

Height of the item. Some items have a dynamic size. In this case, this is the minimum height, or may be ignored altogether. The actual size in the printed report usually depends on the content, and may be larger than the height set here. For example, a data, label or text item will grow as tall as required to display its content.

See Also

Dimension property type

Report Item `width` property

5.3.5 `width` Property

Width of the item.

Summary

Display Name: Width

ROM Type: Dimension

JavaScript Type: `String`

Default value: 0

Inherited: Yes

Settable at runtime: Yes

Availability: First release (limited cases, see the specific report items for details.)

Description

Width of the item. When an item appears as a section (a top-level bit of content within the report), the width property is often ignored, and is taken to be the width of the page. See the derived elements for details.

See Also

Dimension property type

Report Item `height` property

5.3.6 `dataset` Property

The name of the data set that provides data for this item.

Summary

Display Name: Data Set

ROM Type: Element Reference

JavaScript Type: `String`

Default value: See description.

Inherited: Yes

Settable at runtime: No

Availability: First release

Description

Reference to the data set to use for this component. If blank, the data set is that defined for the enclosing list, table or other element.

Data Set Usage

The following table shows the use of the data set name for each type of section.

Report Item	Data Set Name Usage
List Table	The data set that provides the data for the groups and detail sections.
Free-form Grid	Provides data to the contents of the container. If the contents work with a set of rows (chart, list, etc.), then each item reads the entire data set. If the item works with a single row (data item), then the item displays the first (or only) row from the data set.
Text	Provides data for any data expressions in the text. The expressions show values from the first (or only) row in the data set.
Matrix	The matrix displays data from the data set.
Chart	The chart displays data from the data set.
TOC	
Include	Ignored.
Label	
Image	Used if the image is obtained from the data set, otherwise ignored.
Browser Control	TBD
Extended Item	Depends on the implementation of the item.

The meaning for list, table, matrix and chart is clear, because each of these works with a set of rows. The meaning for Free-form, Grid and Text requires a bit more explanation.

A text section can display static text and/or data expressions. The data expression can refer to data in the data set. Since the text item appears only once, it can display data from only one row in the data set. BIRT defines that row to be the first row of the data set. A common usage is to display data looked up from a database. For example, suppose that the report is an invoice. The header of the invoice might display the invoice number and customer information. The report may take a parameter that provides the invoice number. The data set associated with the title would look up the invoice record, and join it to the customer table. The result should be a single row, and the data from this row is what appears in the report title.

The free-form and grid items both contain other report items. If those items are simple items such as a data item or text item, then these items follow the same rule discussed above: they display the first row from the data set. However, items that can display multiple rows (such as a chart, list, matrix, etc.) display all rows from the data set. The effect is as though each item makes a separate pass over the data set. This structure is useful for a section that displays the same data multiple ways. For example, a section may show the top 10 sales as a list then show them again as a pie chart.

Default Data Set

The data set name is optional. If none is provided, then the report item “inherits” a data set. A report item used as a section automatically uses the first data set defined in the

report design. This convenience exists to simplify the tasks for customers to create simple reports using the XML design format. A report item nested inside a container uses the data set for the container.

See Also

Data Set element

Report Item `bindings` property

5.3.7 `bookmark` Property

Defines this item as a hyperlink target.

Summary

Display Name: Bookmark

ROM Type: Expression

Expression type: String

JavaScript Type: String

Inherited: Yes

Default value: None

Settable at runtime: Yes

Availability: After the first release

Description

Many report items can be the target of hyperlinks. The bookmark is a string that uniquely identifies the item instance within the report. This property is an expression that evaluates to a string. If the string is blank or null, then the item has no bookmark. Bookmarks should be unique. If there are duplicates, the behavior of the hyperlink is undefined: it may jump to the first item with that name.

See Also

Action property on labels, data items, images and other items.

5.3.8 `toc` Property

A table of contents entry for this item.

ROM Summary

Display Name: TOC

ROM Type: Expression

Expression type: String

Inherited: Yes

Default value: None

Settable at runtime: Yes

Availability: After the first release

Description

A table of contents entry for this item. The TOC property defines an expression that returns a string that is to appear in the Table of Contents for this item or its container.

5.3.9 visibility Property

A list of visibility rules that determines when the item appears.

Summary

Display Name: Visibility

ROM Type: List of Visibility Rule structure

Default value: None

Inherited: Yes

Settable at runtime: Yes

Availability: First release

Description

The visibility rules control when the item appears in the report. An item can be completely hidden, in which case it won't even be created in the Factory. For example, perhaps the Country item should appear for international customers, but not domestic customers. An item can also be hidden conditionally depending on the output format. For example, perhaps a hyperlink should appear when the report is viewed on the web, but not when it is printed.

See Also

Visibility Rule structure for the details of the rules.

5.3.10 bindings Property

Bindings between input parameters in the included design and values in the main design.

Summary

Display Name: Bindings

ROM Type: List of Binding structures

Default value: None

Inherited: Yes

Settable at runtime: Yes

Availability: First release

Description

A report item can bind to a data set as described in the data set property. A report item can pass information to the data set by providing *bindings* to input parameters of the data set. Each binding identifies an input parameter by name, and provides an expression that gives the value of the parameter.

See Also

Report Item dataSet property

Data Set element, especially the Input Parameters property

5.3.11 `onCreate` Method

Script executed when the element is created in the Factory.

Summary

Availability: First release

Context: Factory

Arguments

None

Returns

None

Description

Executed when the element is created in the Factory. Called after the item is created, but before the item is saved to the report document file. See the scripting spec for additional information about this script. Applications should perform visual customization in the on-render script instead.

See Also

Report Item `onRender` method

5.3.12 `onRender` Method

Script executed when the element is prepared for rendering in the Presentation engine.

Summary

Availability: First release

Context: Presentation

Arguments

None

Returns

None

Description

Executed when the element is prepared for rendering in the Presentation engine. Changes made to the element are written to the target output format, but not saved to the report document file. This is the preferred place for visual customizations.

See Also

Report Item `onCreate` method

6. Report Item Structures

The following explains the structures used by the Report Item element.

6.1 Visibility Rule Structure

Defines visibility for a report item for one output format.

Summary

Display Name: Visibility Rule

Availability: First release

Properties

`format`

The target output format.

`expr`

A Boolean expression that determines if the element is visible in the target format.

Description

The visibility rule says when a report item should be hidden. It can be hidden based on the output type, an expression, or both. For example, the browser control is normally hidden in all output formats except HTML. A past-due item might be hidden if the account is not past due.

An item can be completely hidden (hidden in all formats), or hidden only in selected formats. An item hidden in all formats is never actually created in the Factory. An item hidden in selected formats is created, but will be omitted when rendered to the format for which it is hidden.

6.1.1 `format` Property

Identifies the format in which to hide the element.

Summary

Display Name: Format

ROM Type: Choice

JavaScript Type: String

Default value: "All"

Settable at runtime: Yes

Availability: First release

Choices

Display Name	XML Name	Description
All	all	All output formats. (Default if no format is given.)
Viewer	viewer	Hide in the DHTML viewer.
Email	e-mail	Hide when the report is embedded in an e-mail.
Print	print	Hide when printing the report on the printer.
PDF	pdf	Hide in Adobe Acrobat PDF files.
RTF	rtf	Hide in Microsoft Rich Text Format (RTF) files.
Excel	excel	Hide when converting the report to the Microsoft Excel format.
Word	word	Hide when converting the report to Microsoft Word.
PowerPoint	power-point	Hide when converting the report to Microsoft PowerPoint.

The user can specify other application-specific formats as well.

Description

This property identifies the output format in which to hide the element. The element can be hidden in all output formats (using the “All” choice), or in a specific format. If the element is hidden in all formats, then it is not created in the Factory. Otherwise, is created, but is removed before rendering in the Presentation engine when converted to the target output format.

6.1.2 expr Property

Boolean expression that determines whether to hide the value or not.

Summary

Display Name: Expression

Type: Expression

Expression Type: Boolean

Default value: `true`

Settable at runtime: Yes

Availability: First release

Description

The Boolean expression to evaluate. If the expression evaluates to true, then the item is hidden. Defaults to true if no expression is given.