Oracle® MICROS Reporting and Analytics

Report Designer's Guide

Release 20.1
F30460-08
May 2024

ORACLE

Oracle MICROS Reporting and Analytics Report Designer’s Guide, Release 20.1

F30460-08

Contents

Preface

Purpose

This guide describes the XML structure, elements, and attributes for designing reports in Oracle MICROS Reporting and Analytics with the Report Builder Advanced Editor.

Audience

The guide is intended for report designers who have an understanding of XML and HTML.

Customer Support

To contact Oracle Customer Support, access the Support Portal at the following URL:

https://iccp.custhelp.com/

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received
- Screen shots of each step you take

Documentation

Oracle MICROS product documentation is available on the Oracle Help Center at http://docs.oracle.com/en/industries/food-beverage/.

To learn more about XML, search for online resources.

Revision History

Table 1 Revision History

Date	Description of Change
October 2020	Initial publication.
April 2021	Updated examples.
September 2022	Updated Bar Chart example.
September 2023	Updated Customer Support link.
April 2024	Removed several sections.
May 2024	Updated reportdef/report/layout/component/@visual-style.

1

Getting Started

Reporting and Analytics has two editors that let you design reports that show only the data that is important for understanding and taking action on your business operations.

If your reporting requirements are simple, use the basic editor. The basic editor lets you quickly select a visualization, such as a table or pie chart, and then associate data with the visualization components. The basic editor is easy to use and ideal for non-technical users. It lets you define tables and charts with a simple user interface.

Use the advanced editor if your business requirements are more complex. For example, use the advanced editor to work with data returned through the predefined subject areas and customize how a report displays that data. This guide is a companion to the advanced editor. Use it as a reference to design your report.

You can start designing reports in the basic editor and then switch to the advanced editor if your reporting requirements change. However, you cannot switch back to the basic editor if you started using features in the advanced editor that are not supported in the basic editor. The basic editor cannot validate XML that you add to your report through the advanced editor. The *Oracle MICROS Reporting and Analytics User Guide* provides more information on using the editors.

> **Tip:**
>
> Reporting and Analytics provides a template for designing reports in the advanced editor. When creating a new report, choose the advanced template as the starting point and then modify it to design your own report.

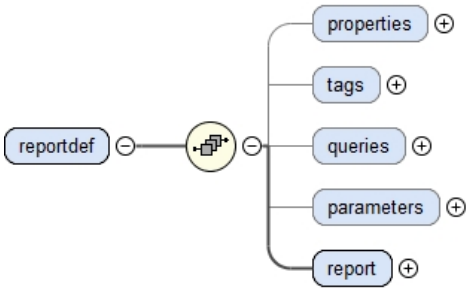
When you design reports, you work with subject areas. You can see a list of subject areas and available data points on the left side of the screen when using the advanced editor.

2
XML Reporting Structure

This section describes the XML elements and attributes that comprise the Reporting and Analytics XML reporting structure.

Elements

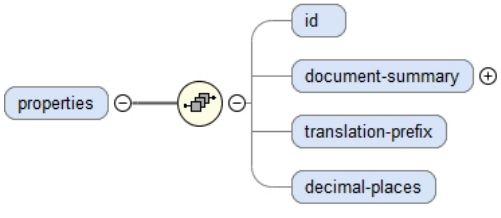
reportdef

Description	The required root element contains all of the other elements. All of the following elements listed (and their own elements) are contained in reportdef start tag and end tag.
Diagram	
Child Elements	parameters, properties, queries, report, tags

Example

```
<reportdef>
        <properties>
            <id>MenuMixDetail</id>
            …
        </properties>
        …
</reportdef>
```

reportdef/properties

Description	An optional element that includes the report name, the printer parameters, and the authoring specifics.
Diagram	

Child Elements	decimal-places, document-summary, id, translation-prefix

Example

```
<reportdef>

  <properties>
    <id>MenuMixMajorDetail</id>

    <document-summary>
      <name>Menu Mix by Major Group</name>
     <description>Menu mix showing maj groups, fam groups and menu items detail </
description>
     <brief-description>Major Group</brief-description>
      <author>John Doe</author>
      <generator>Manual</generator>
      <revision>1.0</revision>
    </document-summary>
    <translation-prefix>stdRpt</translation-prefix>

  </properties>
   …
```

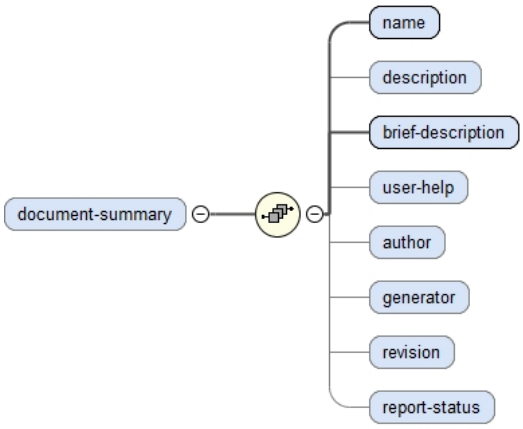
reportdef/properties/id

Description	An optional element that is the name of the report definition. It is a good practice to match this with the name of the report definition file.
Diagram	

Example

```
<reportdef>

  <properties>
    <id>MenuMixMajorDetail</id>

    <document-summary>
      <name>Menu Mix by Major Group</name>
     <description>Menu mix showing maj groups, fam groups and menu items detail </
description>
     <brief-description>Major Group</brief-description>
      <author>John Doe</author>
      <generator>Manual</generator>
      <revision>1.0</revision>
    </document-summary>
    <translation-prefix>stdRpt</translation-prefix>

  </properties>
   …
```

reportdef/properties/document-summary

Description	An optional element that identifies the report name, content, author, method of generation, and revision number. A place to record the creator's intentions, for the benefit of future editors of the file. Including this optional element makes the file easier to maintain over time.
Diagram	
Child Elements	author, brief-description, description, generator, name, report-status, revision, user-help

Example

```
<reportdef>

  <properties>
    <id>MenuMixMajorDetail</id>

    <document-summary>
      <name>Menu Mix by Major Group</name>
     <description>Menu mix showing maj groups, fam groups and menu items detail </
description>
     <brief-description>Major Group</brief-description>
      <author>John Doe</author>
      <generator>Manual</generator>
      <revision>1.0</revision>
    </document-summary>
    <translation-prefix>stdRpt</translation-prefix>

  </properties>
   …
```

reportdef/properties/document-summary/name

Description	A required element that is the name of the actual report.
Diagram	

Example

```
<reportdef>

  <properties>
```

```
    <id>MenuMixMajorDetail</id>

    <document-summary>
      <name>Menu Mix by Major Group</name>
     <description>Menu mix showing maj groups, fam groups and menu items detail </
description>
     <brief-description>Major Group</brief-description>
      <author>John Doe</author>
      <generator>Manual</generator>
      <revision>1.0</revision>
    </document-summary>
    <translation-prefix>stdRpt</translation-prefix>

  </properties>
   …
```

reportdef/properties/document-summary/description

Description	An optional short outline of report contents.
Diagram	description

Example

```
<reportdef>

  <properties>
    <id>MenuMixMajorDetail</id>

    <document-summary>
      <name>Menu Mix by Major Group</name>
     <description>Menu mix showing maj groups, fam groups and menu items detail </
description>
     <brief-description>Major Group</brief-description>
      <author>John Doe</author>
      <generator>Manual</generator>
      <revision>1.0</revision>
    </document-summary>
    <translation-prefix>stdRpt</translation-prefix>

  </properties>
   …
```

reportdef/properties/document-summary/brief-description

Description	A required very short (less than 20 characters) description.
Diagram	brief-description

Example

```
<reportdef>

  <properties>
    <id>MenuMixMajorDetail</id>
```

```
    <document-summary>
      <name>Menu Mix by Major Group</name>
     <description>Menu mix showing maj groups, fam groups and menu items detail </
description>
     <brief-description>Major Group</brief-description>
      <author>John Doe</author>
      <generator>Manual</generator>
      <revision>1.0</revision>
    </document-summary>
    <translation-prefix>stdRpt</translation-prefix>

  </properties>
   …
```

reportdef/properties/document-summary/user-help

Description	Optional text.
Diagram	

Example

```
<reportdef>

  <properties>
    <id>MenuMixMajorDetail</id>

    <document-summary>
      <name>Menu Mix by Major Group</name>
     <description>Menu mix showing maj groups, fam groups and menu items detail </
description>
     <brief-description>Major Group</brief-description>
      <author>John Doe</author>
      <generator>Manual</generator>
      <revision>1.0</revision>
    </document-summary>
    <translation-prefix>stdRpt</translation-prefix>

  </properties>
   …
```

reportdef/properties/document-summary/author

Description	An optional name of the definition file author.
Diagram	

Example

```
<reportdef>

  <properties>
    <id>MenuMixMajorDetail</id>

    <document-summary>
      <name>Menu Mix by Major Group</name>
```

```
     <description>Menu mix showing maj groups, fam groups and menu items detail </
description>
     <brief-description>Major Group</brief-description>
      <author>John Doe</author>
      <generator>Manual</generator>
      <revision>1.0</revision>
    </document-summary>
    <translation-prefix>stdRpt</translation-prefix>

  </properties>
   …
```

reportdef/properties/document-summary/generator

Description	An optional description of the method used to generate the definition file. Primarily this will be `Manual` if a text editor was used, or `Automatic` if a XML IDE such as XMLSpy was utilized.
Diagram	generator

Example

```
<reportdef>

  <properties>
    <id>MenuMixMajorDetail</id>

    <document-summary>
      <name>Menu Mix by Major Group</name>
     <description>Menu mix showing maj groups, fam groups and menu items detail </
description>
     <brief-description>Major Group</brief-description>
      <author>John Doe</author>
      <generator>Manual</generator>
      <revision>1.0</revision>
    </document-summary>
    <translation-prefix>stdRpt</translation-prefix>

  </properties>
   …
```

reportdef/properties/document-summary/revision

Description	An optional element that reflects the revision number of the current file. This is a static value only. It is not related to revision functionality available for published reports.
Diagram	revision

Example

```
<reportdef>

  <properties>
    <id>MenuMixMajorDetail</id>

    <document-summary>
      <name>Menu Mix by Major Group</name>
     <description>Menu mix showing maj groups, fam groups and menu items detail </
```

```
description>
     <brief-description>Major Group</brief-description>
      <author>John Doe</author>
      <generator>Manual</generator>
      <revision>1.0</revision>
    </document-summary>
    <translation-prefix>stdRpt</translation-prefix>

  </properties>
   …
```

reportdef/properties/translation-prefix

Description	An optional corresponding translation file that contains the string variables and their respective translations.
Diagram	translation-prefix

Example

```
<reportdef>

  <properties>
    <id>MenuMixMajorDetail</id>

    <document-summary>
      <name>Menu Mix by Major Group</name>
     <description>Menu mix showing maj groups, fam groups and menu items detail </
description>
     <brief-description>Major Group</brief-description>
      <author>John Doe</author>
      <generator>Manual</generator>
      <revision>1.0</revision>
    </document-summary>
    <translation-prefix>stdRpt</translation-prefix>

  </properties>
   …
```

reportdef/properties/decimal-places

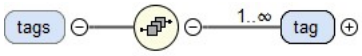
Description	An optional element that provides a single place to override the default of two decimals on cells formatted as `currency`.
Diagram	decimal-places

Example

```
<reportdef>

  <properties>
    <id>MenuMixMajorDetail</id>

    <document-summary>
      <name>Menu Mix by Major Group</name>
     <description>Menu mix showing maj groups, fam groups and menu items detail </
description>
     <brief-description>Major Group</brief-description>
```

```
    <author>John Doe</author>
    <generator>Manual</generator>
    <revision>1.0</revision>
  </document-summary>
  <translation-prefix>stdRpt</translation-prefix>

</properties>
 …
```

reportdef/tags

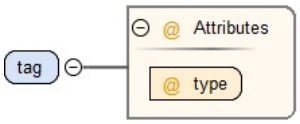
Description	A required element that contains all defined tags assigned to the report.
Diagram	

Child Elements	tag

Example

```
<reportdef>

  <properties>
    …
    …

  </properties>

    <tags>
    <tag type='marketTSR'/>
     <tag type='posSimphony'/>
    <tags>

    …
  …
```

reportdef/tags/tag

Description	A required element that contains all defined tags.
Diagram	
Attributes	type

Example

```
<reportdef>

  <properties>
    …
    …

  </properties>
```

```
    <tags>
    <tag type='marketTSR'/>
     <tag type='posSimphony'/>
    <tags>

    …
  …
```

reportdef/queries

Description	A required element that contains all elements and attributes that define the query commands.
Diagram	
Child Elements	query

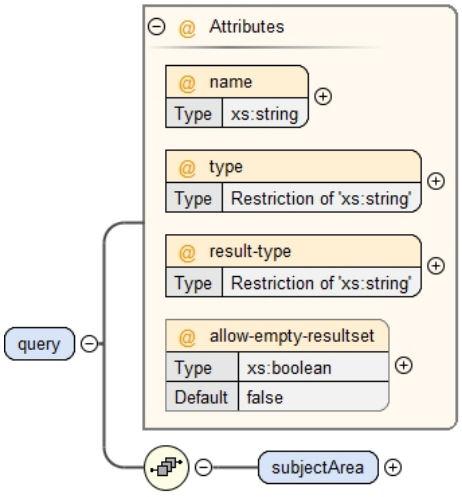
Example

The example shows return records from `Tender Daily Total` between business dates `2019-11-22` and `2019-11-27` with `tndrNum` > `10` and order the returns values by revenue center.

```
...
<query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
    <subjectArea name="Tender Daily Total">
        <columns>
            <column name="rvc" />
            <column name="tndrName" />
            <column name="tndrCnt" aggregate="sum" />
            <column name="tndrTtl" aggregate="sum" />
        </columns>
        <orders>
            <order name="rvc" direction="ASC" />
        </orders>
        <filters>
            <filter name="busDate" type="dateRangeFilter">
                <start>2019-11-22</start>
                <end>2019-11-27</end>
            </filter>
            <filter name="tndrNum" type="relationalFilter" comparison="GREATER_THAN">
                <data>10</data>
            </filter>
        </filters>
    </subjectArea>
</query>
...
```

reportdef/queries/query

Description	This required element contains all the query-related elements and attributes.

Diagram	
Child Elements	subjectArea

Attributes	allow-empty-resultset, name, result-type, type

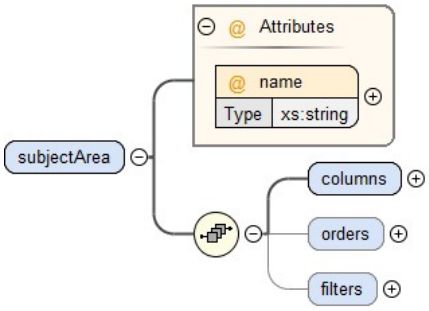
Example

The example shows return records from `Tender Daily Total` between business dates `2019-11-22` and `2019-11-27` with `tndrNum` > `10` and order the returns values by revenue center.

```
...
<query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
    <subjectArea name="Tender Daily Total">
        <columns>
            <column name="rvc" />
            <column name="tndrName" />
            <column name="tndrCnt" aggregate="sum" />
            <column name="tndrTtl" aggregate="sum" />
        </columns>
        <orders>
            <order name="rvc" direction="ASC" />
        </orders>
        <filters>
            <filter name="busDate" type="dateRangeFilter">
                <start>2019-11-22</start>
                <end>2019-11-27</end>
            </filter>
            <filter name="tndrNum" type="relationalFilter" comparison="GREATER_THAN">
                <data>10</data>
            </filter>
        </filters>
    </subjectArea>
</query>
...
```

reportdef/queries/query/subjectArea

Description	A required element that is only applicable if type of the parent <query> element is `iquery`.

Diagram	
Child Elements	columns, filters, orders
Attributes	name

Example

The example shows return records from `Tender Daily Total` between business dates `2019-11-22` and `2019-11-27` with `tndrNum` > `10` and order the returns values by revenue center.

```
...
<query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
    <subjectArea name="Tender Daily Total">
        <columns>
            <column name="rvc" />
            <column name="tndrName" />
            <column name="tndrCnt" aggregate="sum" />
            <column name="tndrTtl" aggregate="sum" />
        </columns>
        <orders>
            <order name="rvc" direction="ASC" />
        </orders>
        <filters>
            <filter name="busDate" type="dateRangeFilter">
                <start>2019-11-22</start>
                <end>2019-11-27</end>
            </filter>
            <filter name="tndrNum" type="relationalFilter" comparison="GREATER_THAN">
                <data>10</data>
            </filter>
        </filters>
    </subjectArea>
</query>
...
```

reportdef/queries/query/subjectArea/columns

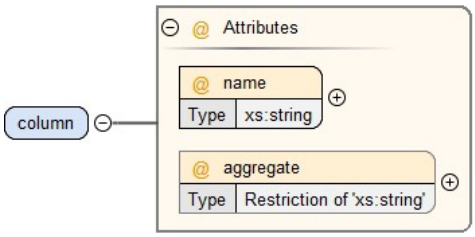
Description	This required element contains the list of columns given in metadata that will be returned in the report. The metadata appears in the **Subject Area (Alias)** section of the advanced editor of the report builder. The alias name, which appears in the parentheses, must be used for the column.
Diagram	
Child Elements	column

Example

The example shows return records from `Tender Daily Total` between business dates `2019-11-22` and `2019-11-27` with `tndrNum` > `10` and order the returns values by revenue center.

```
...
<query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
    <subjectArea name="Tender Daily Total">
        <columns>
            <column name="rvc" />
            <column name="tndrName" />
            <column name="tndrCnt" aggregate="sum" />
            <column name="tndrTtl" aggregate="sum" />
        </columns>
        <orders>
            <order name="rvc" direction="ASC" />
        </orders>
        <filters>
            <filter name="busDate" type="dateRangeFilter">
                <start>2019-11-22</start>
                <end>2019-11-27</end>
            </filter>
            <filter name="tndrNum" type="relationalFilter" comparison="GREATER_THAN">
                <data>10</data>
            </filter>
        </filters>
    </subjectArea>
</query>
...
```

reportdef/queries/query/subjectArea/columns/column

Description	This required element contains the list of columns given in metadata that will be returned in the report. The metadata appears in the **Subject Area (Alias)** section of the advanced editor of the report builder. The alias name, which appears in the parentheses, must be used for the column.
Diagram	
Attributes	aggregate, name

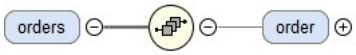
Example

The example shows return records from `Tender Daily Total` between business dates `2019-11-22` and `2019-11-27` with `tndrNum` > `10` and order the returns values by revenue center.

```
...
<query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
    <subjectArea name="Tender Daily Total">
        <columns>
            <column name="rvc" />
            <column name="tndrName" />
            <column name="tndrCnt" aggregate="sum" />
```

```
            <column name="tndrTtl" aggregate="sum" />
        </columns>
        <orders>
            <order name="rvc" direction="ASC" />
        </orders>
        <filters>
            <filter name="busDate" type="dateRangeFilter">
                <start>2019-11-22</start>
                <end>2019-11-27</end>
            </filter>
            <filter name="tndrNum" type="relationalFilter" comparison="GREATER_THAN">
                <data>10</data>
            </filter>
        </filters>
    </subjectArea>
</query>
...
```

reportdef/queries/query/subjectArea/orders

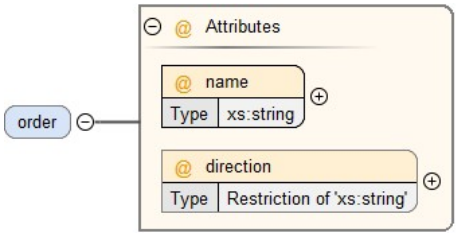
Description	This optional element contains the list of columns given in metadata that is used to order the items. The metadata appears in the **Subject Area (Alias)** section of the advanced editor of the report builder.
Diagram	
Child Elements	order

Example

The example shows return records from `Tender Daily Total` between business dates `2019-11-22` and `2019-11-27` with `tndrNum` > `10` and order the returns values by revenue center.

```
...
<query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
    <subjectArea name="Tender Daily Total">
        <columns>
            <column name="rvc" />
            <column name="tndrName" />
            <column name="tndrCnt" aggregate="sum" />
            <column name="tndrTtl" aggregate="sum" />
        </columns>
        <orders>
            <order name="rvc" direction="ASC" />
        </orders>
        <filters>
            <filter name="busDate" type="dateRangeFilter">
                <start>2019-11-22</start>
                <end>2019-11-27</end>
            </filter>
            <filter name="tndrNum" type="relationalFilter" comparison="GREATER_THAN">
                <data>10</data>
            </filter>
        </filters>
    </subjectArea>
</query>
...
```

reportdef/queries/query/subjectArea/orders/order

Description	This optional element contains the list of columns given in metadata that is used to order the items. The metadata appears in the **Subject Area (Alias)** section of the advanced editor of the report builder.
Diagram	
Attributes	direction, name

Example

The example shows return records from `Tender Daily Total` between business dates `2019-11-22` and `2019-11-27` with `tndrNum` > `10` and order the returns values by revenue center.

```
...
<query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
    <subjectArea name="Tender Daily Total">
        <columns>
            <column name="rvc" />
            <column name="tndrName" />
            <column name="tndrCnt" aggregate="sum" />
            <column name="tndrTtl" aggregate="sum" />
        </columns>
        <orders>
            <order name="rvc" direction="ASC" />
        </orders>
        <filters>
            <filter name="busDate" type="dateRangeFilter">
                <start>2019-11-22</start>
                <end>2019-11-27</end>
            </filter>
            <filter name="tndrNum" type="relationalFilter" comparison="GREATER_THAN">
                <data>10</data>
            </filter>
        </filters>
    </subjectArea>
</query>
...
```

reportdef/queries/query/subjectArea/filters

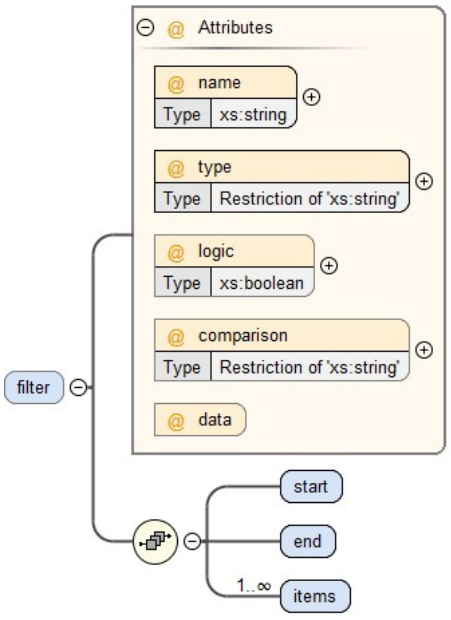
Description	This optional element contains the list of columns given in metadata that is used to filter the records. The metadata appears in the **Subject Area (Alias)** section of the advanced editor of the report builder.
Diagram	
Child Elements	filter

Example

The example shows return records from `Tender Daily Total` between business dates `2019-11-22` and `2019-11-27` with `tndrNum` > `10` and order the returns values by revenue center.

```
...
<query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
    <subjectArea name="Tender Daily Total">
        <columns>
            <column name="rvc" />
            <column name="tndrName" />
            <column name="tndrCnt" aggregate="sum" />
            <column name="tndrTtl" aggregate="sum" />
        </columns>
        <orders>
            <order name="rvc" direction="ASC" />
        </orders>
        <filters>
            <filter name="busDate" type="dateRangeFilter">
                <start>2019-11-22</start>
                <end>2019-11-27</end>
            </filter>
            <filter name="tndrNum" type="relationalFilter" comparison="GREATER_THAN">
                <data>10</data>
            </filter>
        </filters>
    </subjectArea>
</query>
...
```

reportdef/queries/query/subjectArea/filters/filter

Description	This optional element contains the list of columns given in metadata that is used to filter the records. Columns marked as measure (except date, qtrHr, busHour) in metadata can't be used as filters. The metadata appears in the **Subject Area (Alias)** section of the advanced editor of the report builder.
Diagram	

Child Elements	end, items, start
Attributes	comparison, data, logic, name, type

Example

The example shows return records from `Tender Daily Total` between business dates `2019-11-22` and `2019-11-27` with `tndrNum` > `10` and order the returns values by revenue center.

```
...
<query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
    <subjectArea name="Tender Daily Total">
        <columns>
            <column name="rvc" />
            <column name="tndrName" />
            <column name="tndrCnt" aggregate="sum" />
            <column name="tndrTtl" aggregate="sum" />
        </columns>
        <orders>
            <order name="rvc" direction="ASC" />
        </orders>
        <filters>
            <filter name="busDate" type="dateRangeFilter">
                <start>2019-11-22</start>
                <end>2019-11-27</end>
            </filter>
            <filter name="tndrNum" type="relationalFilter" comparison="GREATER_THAN">
                <data>10</data>
            </filter>
        </filters>
    </subjectArea>
</query>
...
```

reportdef/queries/query/subjectArea/filters/filter/data

Description	An optional element. This is the data that will be compared with column in order to filter the records. Appropriate format must be followed for data. For example, for date/datetime column, data must be in ISO8601 format. Date/datetime column: • `yyyy-MM-dd` • `yyyy-MM-dd HH:mm:ss` hour-min-sec formatted column: • `HH:mm:ss` • `HH:mm` • `HH` `IS_NULL`, `IS_NOT_NULL` ignores the data element. It is only applicable if filter type= `relationalFilter`

Example

The example shows return records from `Tender Daily Total` between business dates `2019-11-22` and `2019-11-27` with `tndrNum` > `10` and order the returns values by revenue center.

```
...
<query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
```

```
    <subjectArea name="Tender Daily Total">
        <columns>
            <column name="rvc" />
            <column name="tndrName" />
            <column name="tndrCnt" aggregate="sum" />
            <column name="tndrTtl" aggregate="sum" />
        </columns>
        <orders>
            <order name="rvc" direction="ASC" />
        </orders>
        <filters>
            <filter name="busDate" type="dateRangeFilter">
                <start>2019-11-22</start>
                <end>2019-11-27</end>
            </filter>
            <filter name="tndrNum" type="relationalFilter" comparison="GREATER_THAN">
                <data>10</data>
            </filter>
        </filters>
    </subjectArea>
</query>
...
```

reportdef/queries/query/subjectArea/filters/filter/start

Description	Required when using a range filter.
	Range Filters:
	`dateRangeFilter`, `dateTimeRangeFilter`, `busHrRangeFilter`, `fpRangeFilter`
	Sample for `dateRangeFilter` :
	`<filter name="busDate" type="dateRangeFilter">`
	`  <start>2019-11-22</start>`
	`  <end>2019-11-27</end>`
	`</filter>`
	Sample for `busHrRangeFilter`:
	`<filter name="busHr" type="busHrRangeFilter">`
	`  <start>8</start>`
	`  <end>17</end>`
	`</filter>`

Diagram	

Example

The example shows return records from `Tender Daily Total` between business dates `2019-11-22` and `2019-11-27` with `tndrNum` > `10` and order the returns values by revenue center.

```
...
<query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
    <subjectArea name="Tender Daily Total">
        <columns>
            <column name="rvc" />
            <column name="tndrName" />
            <column name="tndrCnt" aggregate="sum" />
            <column name="tndrTtl" aggregate="sum" />
        </columns>
        <orders>
```

```
            <order name="rvc" direction="ASC" />
        </orders>
        <filters>
            <filter name="busDate" type="dateRangeFilter">
                <start>2019-11-22</start>
                <end>2019-11-27</end>
            </filter>
            <filter name="tndrNum" type="relationalFilter" comparison="GREATER_THAN">
                <data>10</data>
            </filter>
        </filters>
    </subjectArea>
</query>
...
```

reportdef/queries/query/subjectArea/filters/filter/end

Description	Required when using the range filter.
Diagram	

Example

The example shows return records from `Tender Daily Total` between business dates `2019-11-22` and `2019-11-27` with `tndrNum` > `10` and order the returns values by revenue center.

```
...
<query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
    <subjectArea name="Tender Daily Total">
        <columns>
            <column name="rvc" />
            <column name="tndrName" />
            <column name="tndrCnt" aggregate="sum" />
            <column name="tndrTtl" aggregate="sum" />
        </columns>
        <orders>
            <order name="rvc" direction="ASC" />
        </orders>
        <filters>
            <filter name="busDate" type="dateRangeFilter">
                <start>2019-11-22</start>
                <end>2019-11-27</end>
            </filter>
            <filter name="tndrNum" type="relationalFilter" comparison="GREATER_THAN">
                <data>10</data>
            </filter>
        </filters>
    </subjectArea>
</query>
...
```

reportdef/queries/query/subjectArea/filters/filter/items

Description	Required when using the list filter, it is a wrapper element for item element..

Diagram	items
Child Elements	item

Example

The example shows return records from `Tender Daily Total` between business dates `2019-11-22` and `2019-11-27` with `tndrNum` > `10` and order the returns values by revenue center.

```
...
<query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
    <subjectArea name="Tender Daily Total">
        <columns>
            <column name="rvc" />
            <column name="tndrName" />
            <column name="tndrCnt" aggregate="sum" />
            <column name="tndrTtl" aggregate="sum" />
        </columns>
        <orders>
            <order name="rvc" direction="ASC" />
        </orders>
        <filters>
            <filter name="busDate" type="dateRangeFilter">
                <start>2019-11-22</start>
                <end>2019-11-27</end>
            </filter>
            <filter name="tndrNum" type="relationalFilter" comparison="GREATER_THAN">
                <data>10</data>
            </filter>
        </filters>
    </subjectArea>
</query>
...
```

reportdef/queries/query/subjectArea/filters/filter/items/item

Description	Multiple items element is required when using the following list filters: `stringFilter`, `intFilter`, `longFilter`, `doubleFilter`.
	Sample:
	`<filter name="surname" type="stringFilter">`
	`<items>`
	`<item>Smith</item>`
	`<item>Johnson</item>`
	`<item>Brown</item>`
	`</items>`
	`</filter>`

Example

The example shows return records from `Tender Daily Total` between business dates `2019-11-22` and `2019-11-27` with `tndrNum` > `10` and order the returns values by revenue center.

```
...
<query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
    <subjectArea name="Tender Daily Total">
```

```
        <columns>
            <column name="rvc" />
            <column name="tndrName" />
            <column name="tndrCnt" aggregate="sum" />
            <column name="tndrTtl" aggregate="sum" />
        </columns>
        <orders>
            <order name="rvc" direction="ASC" />
        </orders>
        <filters>
            <filter name="busDate" type="dateRangeFilter">
                <start>2019-11-22</start>
                <end>2019-11-27</end>
            </filter>
            <filter name="tndrNum" type="relationalFilter" comparison="GREATER_THAN">
                <data>10</data>
            </filter>
        </filters>
    </subjectArea>
</query>
...
```

reportdef/parameters

Description	An optional element with definitions referenced by the queries. Used to filter the data that is being requested.
Diagram	
Child Elements	parameter

Example

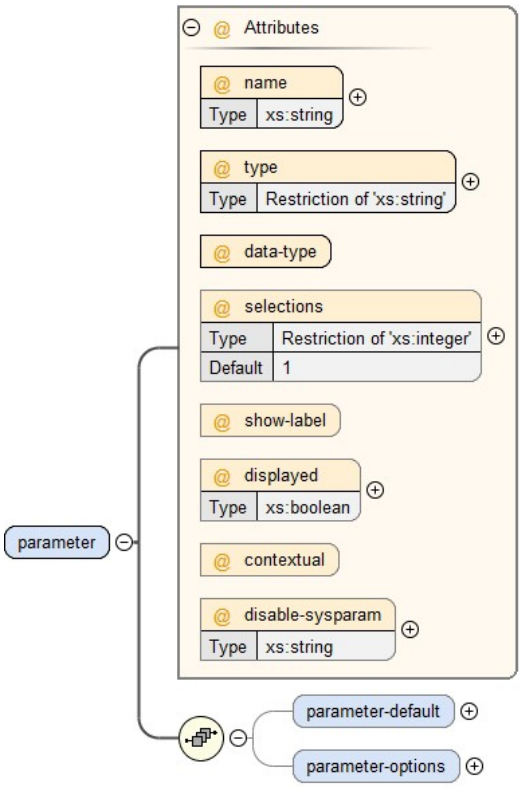
In the example, the data is filtered three ways: by business date (`busDate`), location name (`locName`), and revenue center name (`rvc`). These parameters are options when running a report in Reporting and Analytics, found within **Edit Parameters**.

In the advanced editor, you can view the parameters listed under the subject area selected for this query. Possible parameters and parameter names are shown with a **Parameter** tag in the subject area list on the left. Use the alias name of the column within the `name` attribute.

```
<reportdef>
    <properties />
    <queries>
        <query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
            <subjectArea name="Tender Fixed Period Total">
                <columns>
                    <column name="rvc" />
                    <column name="busHour" />
                    <column name="tndrName" />
                    <column name="tndrCnt" aggregate="sum" />
                    <column name="tndrTtl" aggregate="sum" />
                </columns>
                <orders>
                    <order name="busHour" direction="ASC" />
                    <order name="tndrName" direction="ASC" />
                </orders>
            </subjectArea>
        </query>
```

```
    </queries>
    <parameters>
        <parameter name="busDate" type="list" data-type="date">
            <parameter-default default-type="system" default-
value="MostRecentOperations" />
        </parameter>
        <parameter name="locName" type="list" data-type="text" />
        <parameter name="rvc" type="list" data-type="text" />
    </parameters>
    ...
```

reportdef/parameters/parameter

Description	This required element defines the characteristics of the filter parameters.
Diagram	
Child Elements	parameter-default, parameter-options
Attributes	contextual, data-type, disable-sysparam, displayed, name, selections, show-label, type

Example

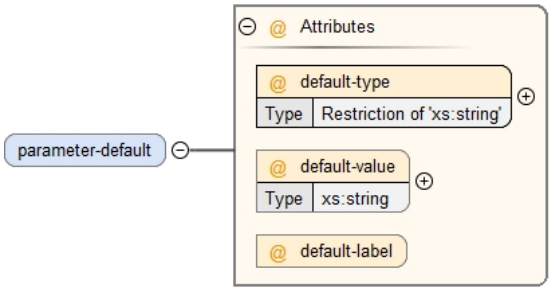
In the example, the data is filtered three ways: by business date (`busDate`), location name (`locName`), and revenue center name (`rvc`). These parameters are options when running a report in Reporting and Analytics, found within **Edit Parameters**.

In the advanced editor, you can view the parameters listed under the subject area selected for this query. Possible parameters and parameter names are shown with a **Parameter** tag in the subject area list on the left. Use the alias name of the column within the `name` attribute.

```
<reportdef>
    <properties />
```

```
    <queries>
        <query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
            <subjectArea name="Tender Fixed Period Total">
                <columns>
                    <column name="rvc" />
                    <column name="busHour" />
                    <column name="tndrName" />
                    <column name="tndrCnt" aggregate="sum" />
                    <column name="tndrTtl" aggregate="sum" />
                </columns>
                <orders>
                    <order name="busHour" direction="ASC" />
                    <order name="tndrName" direction="ASC" />
                </orders>
            </subjectArea>
        </query>
    </queries>
    <parameters>
        <parameter name="busDate" type="list" data-type="date">
            <parameter-default default-type="system" default-
value="MostRecentOperations" />
        </parameter>
        <parameter name="locName" type="list" data-type="text" />
        <parameter name="rvc" type="list" data-type="text" />
    </parameters>
    ...
```

reportdef/parameters/parameter/parameter-default

Description	This optional element modifies **selections** when that attribute is set to a value greater than 1.
Diagram	
Attributes	default-type, default-value,

Example

In the example, the data is filtered three ways: by business date (`busDate`), location name (`locName`), and revenue center name (`rvc`). These parameters are options when running a report in Reporting and Analytics, found within **Edit Parameters**.

In the advanced editor, you can view the parameters listed under the subject area selected for this query. Possible parameters and parameter names are shown with a **Parameter** tag in the subject area list on the left. Use the alias name of the column within the `name` attribute.

```
<reportdef>
    <properties />
    <queries>
        <query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
            <subjectArea name="Tender Fixed Period Total">
                <columns>
```

```
                    <column name="rvc" />
                    <column name="busHour" />
                    <column name="tndrName" />
                    <column name="tndrCnt" aggregate="sum" />
                    <column name="tndrTtl" aggregate="sum" />
                </columns>
                <orders>
                    <order name="busHour" direction="ASC" />
                    <order name="tndrName" direction="ASC" />
                </orders>
            </subjectArea>
        </query>
    </queries>
    <parameters>
        <parameter name="busDate" type="list" data-type="date">
            <parameter-default default-type="system" default-
value="MostRecentOperations" />
        </parameter>
        <parameter name="locName" type="list" data-type="text" />
        <parameter name="rvc" type="list" data-type="text" />
    </parameters>
    ...
```

reportdef/parameters/parameter/parameter-options

Description	Optional element that affects the behavior of parameter selecting by enforcing a selection and/or limiting the number of selections.
Diagram	
Child Elements	parameter-option

Example

In the example, the data is filtered three ways: by business date (`busDate`), location name (`locName`), and revenue center name (`rvc`). These parameters are options when running a report in Reporting and Analytics, found within **Edit Parameters**.

In the advanced editor, you can view the parameters listed under the subject area selected for this query. Possible parameters and parameter names are shown with a **Parameter** tag in the subject area list on the left. Use the alias name of the column within the `name` attribute.

The parameter options used in this specific example limit the total number of business dates that can be selected to 7, limit the total number of locations that can be selected to 1, cause the parameter options to be shown when the report is run, and will force the user to select a location before running the report.

```
<queries>
    <query datasource="" name="Query2" result-type="multiple-rows" type="iquery">
        <subjectArea name="Tender Fixed Period Total">
            <columns>
                <column name="busHour" />
                <column name="tndrName" />
                <column name="tndrCnt" aggregate="sum" />
                <column name="tndrTtl" aggregate="sum" />
            </columns>
            <orders>
                <order name="busHour" direction="ASC" />
                <order name="tndrName" direction="ASC" />
```

```
            </orders>
        </subjectArea>
    </query>
</queries>
<parameters>
    <parameter name="busDate" type="list" data-type="date">
        <parameter-default default-type="system" default-value="MostRecentOperations" />
        <parameter-options>
            <parameter-option name="maximum" value="7" />
        </parameter-options>
    </parameter>
    <parameter name="locName" type="value" data-type="text">
        <parameter-options>
            <parameter-option name="pre-select" value="yes" />
            <parameter-option name="maximum" value="1" />
        </parameter-options>
    </parameter>
    <parameter name="rvc" type="list" data-type="text" />
</parameters>
<report>
...
</report>
...
```

reportdef/parameters/parameter/parameter-options/parameter-option

Description	Optional element that affects the behavior of parameter selecting by enforcing a selection and/or limiting the number of selections.
Attributes	error-message, name, value

Example

In the example, the data is filtered three ways: by business date (`busDate`), location name (`locName`), and revenue center name (`rvc`). These parameters are options when running a report in Reporting and Analytics, found within **Edit Parameters**.

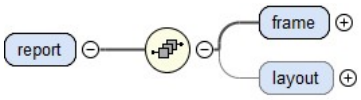
In the advanced editor, you can view the parameters listed under the subject area selected for this query. Possible parameters and parameter names are shown with a **Parameter** tag in the subject area list on the left. Use the alias name of the column within the `name` attribute.

The parameter options used in this specific example limit the total number of business dates that can be selected to 7, limit the total number of locations that can be selected to 1, cause the parameter options to be shown when the report is run, and will force the user to select a location before running the report.

```
<queries>
    <query datasource="" name="Query2" result-type="multiple-rows" type="iquery">
        <subjectArea name="Tender Fixed Period Total">
            <columns>
                <column name="busHour" />
                <column name="tndrName" />
                <column name="tndrCnt" aggregate="sum" />
                <column name="tndrTtl" aggregate="sum" />
            </columns>
            <orders>
                <order name="busHour" direction="ASC" />
                <order name="tndrName" direction="ASC" />
            </orders>
```

```
        </subjectArea>
    </query>
</queries>
<parameters>
    <parameter name="busDate" type="list" data-type="date">
        <parameter-default default-type="system" default-value="MostRecentOperations" />
        <parameter-options>
            <parameter-option name="maximum" value="7" />
        </parameter-options>
    </parameter>
    <parameter name="locName" type="value" data-type="text">
        <parameter-options>
            <parameter-option name="pre-select" value="yes" />
            <parameter-option name="maximum" value="1" />
        </parameter-options>
    </parameter>
    <parameter name="rvc" type="list" data-type="text" />
</parameters>
<report>
...
</report>
...
```

reportdef/report

Description	A required element. Determines the look and feel of the final report output, incorporating all the data and formats laid out in the preceding elements and attributes.
Diagram	
Child Elements	frame, layout

Example

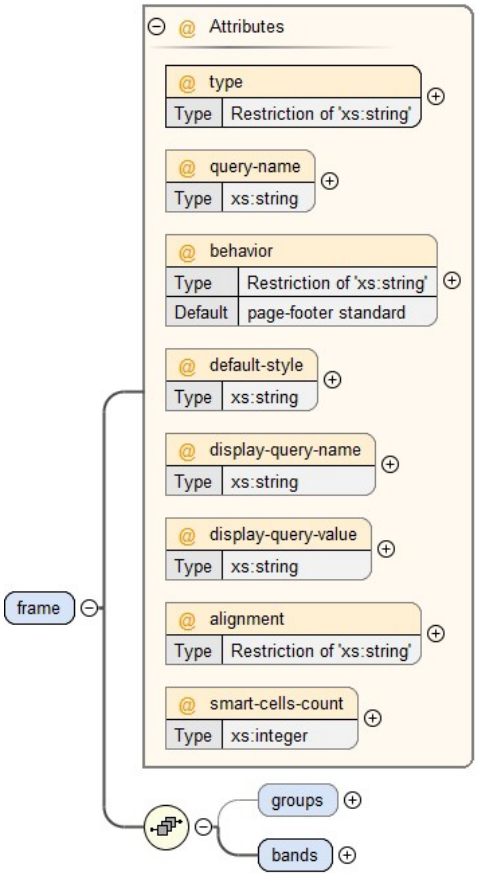
```
<queries>
    <query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
        <subjectArea name="Tender Daily Total">
            <columns>
                <column name="rvc" />
                <column name="tndrName" />
                <column name="tndrCnt" aggregate="sum" />
            </columns>
            <orders>
                <order name="rvc" direction="ASC" />
            </orders>
        </subjectArea>
    </query>
</queries>
<parameters>
...
</parameters>
<report>
    <frame type="table" name="tmbrvc" query-name="Query0">
        <bands>
            <band type="table-column-header">
                <cell type="text" alignment="left" print-width="81" data-
format="text">Revenue Center</cell>
                <cell type="text" alignment="left" print-width="81" data-
```

```
format="text">Tender Name</cell>
                <cell type="text" alignment="left" print-width="81" data-
format="text">Tender Count</cell>
            </band>
            <band type="table-body">
                <cell type="column" alignment="left" print-width="81" data-
format="text">rvc</cell>
                <cell type="column" alignment="left" print-width="81" data-
format="text">tndrName</cell>
                <cell type="column" alignment="right" print-width="81" data-
format="integer">tndrCnt</cell>
            </band>
        </bands>
    </frame>
    ...
</report>
```

reportdef/report/frame

Description	This required element contains the elements and attributes that define the look and feel of a report. The frame element determines the appearance of its part of the report, including layout, source, and format. Within a single report, multiple frames can be used to show the user different views of the same (or related) data. In a table frame, a single table-body band can produce a nearly unlimited number of rows. The data will fill in as many rows as were returned by the source query. A table-column-header band will place a single row of cells above the body of a report, and is used to provide column names. The table-group-header places a single row above each set of body data each time the data is broken into its respective group (as defined by the group-item in the groups element.)

Diagram



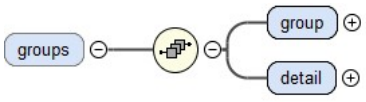
Child Elements	bands, groups
Attributes	alignment, behavior, default-style, display-query-name, display-query-value, query-name, smart-cells-count, type

Example

```
<queries>
    <query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
        <subjectArea name="Tender Daily Total">
            <columns>
                <column name="rvc" />
                <column name="tndrName" />
                <column name="tndrCnt" aggregate="sum" />
            </columns>
            <orders>
                <order name="rvc" direction="ASC" />
            </orders>
        </subjectArea>
    </query>
</queries>
<parameters>
...
</parameters>
<report>
    <frame type="table" name="tmbrvc" query-name="Query0">
        <groups>
            <group name="Total">
                <group-item name="organizationid" data-type="integer" />
```

```
                <group-expr name="Allsum_tndrCnt" type="sum" column="tndrCnt" />
            </group>
        </groups>
        <bands>
            <band type="table-column-header">
                <cell type="text" alignment="left" print-width="81" data-
format="text">Revenue Center</cell>
                <cell type="text" alignment="left" print-width="81" data-
format="text">Tender Name</cell>
                <cell type="text" alignment="left" print-width="81" data-
format="text">Tender Count</cell>
            </band>
            <band group-name="Total" type="table-group-header">
                <cell type="text" print-width="81" alignment="left" data-format="text" />
                <cell type="text" print-width="81" alignment="left" data-format="text" />
                <cell type="column" print-width="81" alignment="right" data-
format="integer">Allsum_tndrCnt</cell>
            </band>
            <band type="table-body">
                <cell type="column" alignment="left" print-width="81" data-
format="text">rvc</cell>
                <cell type="column" alignment="left" print-width="81" data-
format="text">tndrName</cell>
                <cell type="column" alignment="right" print-width="81" data-
format="integer">tndrCnt</cell>
            </band>
        </bands>
    </frame>
    ...
</report>
...
```

reportdef/report/frame/groups

Description	This optional element defines data groups, for sorting and summing of groups. Only one `groups` element can exist in a `frame`. Primarily used to create subtotal and total lines, or summed values to be used in calculations. The optional element `groups` defines how information is categorized, which value fields should be included, and what calculations need to be performed. Groups can act like sub-queries that run dynamically from within the report. While many `group` elements can exist within `groups`, only one `group` element can exist per `frame`.
Diagram	
Child Elements	group

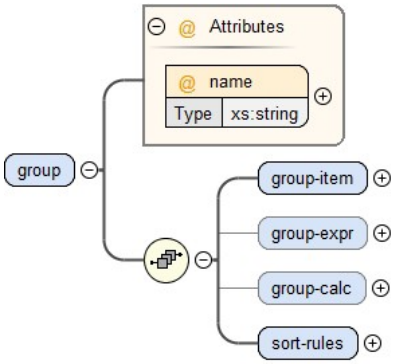
Example

```
<queries>
    <query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
        <subjectArea name="Tender Fixed Period Total">
            <columns>
                <column name="rvc" />
                <column name="busHour" />
                <column name="tndrName" />
                <column name="tndrCnt" aggregate="sum" />
                <column name="tndrTtl" aggregate="sum" />
```

```
            </columns>
            <orders>
                <order name="busHour" direction="ASC" />
                <order name="tndrName" direction="ASC" />
            </orders>
        </subjectArea>
    </query>
</queries>
<parameters>
...
</parameters>

<report>
    <frame type="table" name="Tender Media by Revenue Center" query-name="Query0">
        <groups>
            <group name="Total">
                <group-item name="organizationId" data-type="integer" />
                <group-expr name="AllTndrCnt" type="sum" column="tndrCnt" />
                <group-expr name="AllTndrTtl" type="sum" column="tndrTtl" />
                <group-calc name="AllPrctTndrTtl" data-type="percent">1</group-calc>
            </group>
            <group name="RvcAggregates">
                <group-item name="rvc" data-type="text" />
                <group-expr name="RvcTndrCnt" type="sum" column="tndrCnt" />
                <group-expr name="RvcTndrTtl" type="sum" column="tndrTtl" />
                <group-calc name="RvcPrctTndrTtl" data-type="percent">RvcTndrTtl /
AllTndrTtl</group-calc>
            </group>
        </groups>
        <bands>
            <band type="table-column-header">
                <cell type="text" print-width="68" alignment="left" data-
format="text">Name</cell>
                <cell type="text" print-width="68" alignment="right" data-
format="text">Tender Count</cell>
                <cell type="text" print-width="68" alignment="right" data-
format="text">Tender Amount</cell>
                <cell type="text" print-width="68" alignment="right" data-
format="text">Percent of Total Tender Amount</cell>
            </band>
            <band type="table-group-header" group-name="Total">
                <cell type="text" print-width="68" alignment="left" data-
format="text">TOTAL</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">AllTndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">AllTndrTtl</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="percent">AllPrctTndrTtl</cell>
            </band>
            <band type="table-group-header" group-name="RvcAggregates">
                <cell type="column" print-width="68" alignment="left" data-
format="text">rvc</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">RvcTndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">RvcTndrTtl</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="percent">RvcPrctTndrTtl</cell>
            </band>
            <band type="table-body">
                <cell type="column" print-width="68" alignment="left" data-
```

```
format="text">tndrName</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">tndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">tndrTtl</cell>
                <cell type="expr" print-width="68" alignment="right" data-
format="percent" name="TotalPercentage">tndrTtl / AllTndrTtl</cell>
            </band>
        </bands>
    </frame>
    ...
</report>
```

reportdef/report/frame/groups/group

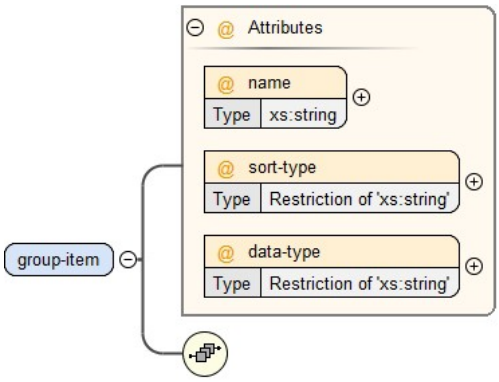
Description	This required element defines how data will be organized and formatted. While many `group` elements can exist within `groups`, only one `group` element can exist per `frame`.
Diagram	
Child Elements	group-calc, group-expr, group-item, sort-rules
Attributes	name

Example

```
<queries>
    <query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
        <subjectArea name="Tender Fixed Period Total">
            <columns>
                <column name="rvc" />
                <column name="busHour" />
                <column name="tndrName" />
                <column name="tndrCnt" aggregate="sum" />
                <column name="tndrTtl" aggregate="sum" />
            </columns>
            <orders>
                <order name="busHour" direction="ASC" />
                <order name="tndrName" direction="ASC" />
            </orders>
        </subjectArea>
    </query>
</queries>
<parameters>
...
</parameters>

<report>
    <frame type="table" name="Tender Media by Revenue Center" query-name="Query0">
```

```
        <groups>
            <group name="Total">
                <group-item name="organizationId" data-type="integer" />
                <group-expr name="AllTndrCnt" type="sum" column="tndrCnt" />
                <group-expr name="AllTndrTtl" type="sum" column="tndrTtl" />
                <group-calc name="AllPrctTndrTtl" data-type="percent">1</group-calc>
            </group>
            <group name="RvcAggregates">
                <group-item name="rvc" data-type="text" />
                <group-expr name="RvcTndrCnt" type="sum" column="tndrCnt" />
                <group-expr name="RvcTndrTtl" type="sum" column="tndrTtl" />
                <group-calc name="RvcPrctTndrTtl" data-type="percent">RvcTndrTtl /
AllTndrTtl</group-calc>
            </group>
        </groups>
        <bands>
            <band type="table-column-header">
                <cell type="text" print-width="68" alignment="left" data-
format="text">Name</cell>
                <cell type="text" print-width="68" alignment="right" data-
format="text">Tender Count</cell>
                <cell type="text" print-width="68" alignment="right" data-
format="text">Tender Amount</cell>
                <cell type="text" print-width="68" alignment="right" data-
format="text">Percent of Total Tender Amount</cell>
            </band>
            <band type="table-group-header" group-name="Total">
                <cell type="text" print-width="68" alignment="left" data-
format="text">TOTAL</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">AllTndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">AllTndrTtl</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="percent">AllPrctTndrTtl</cell>
            </band>
            <band type="table-group-header" group-name="RvcAggregates">
                <cell type="column" print-width="68" alignment="left" data-
format="text">rvc</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">RvcTndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">RvcTndrTtl</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="percent">RvcPrctTndrTtl</cell>
            </band>
            <band type="table-body">
                <cell type="column" print-width="68" alignment="left" data-
format="text">tndrName</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">tndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">tndrTtl</cell>
                <cell type="expr" print-width="68" alignment="right" data-
format="percent" name="TotalPercentage">tndrTtl / AllTndrTtl</cell>
            </band>
        </bands>
    </frame>
    ...
</report>
```

reportdef/report/frame/groups/group/group-item

Description	A required element that defines how the data is grouped. In the example below, data is first grouped by the whole enterprise in the Total row (`<group-item name="organizationId"`) and it is grouped by each Revenue Center in the second group header (`<group-item name="rvc"`).
Diagram	
Attributes	data-type, name, sort-type

Example

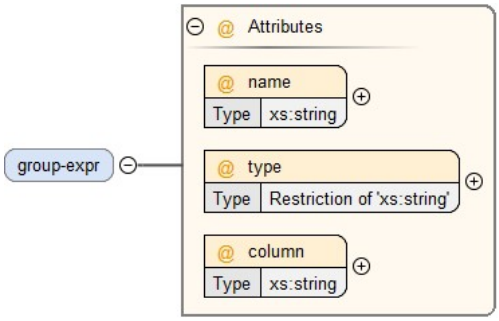
```
<queries>
    <query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
        <subjectArea name="Tender Fixed Period Total">
            <columns>
                <column name="rvc" />
                <column name="busHour" />
                <column name="tndrName" />
                <column name="tndrCnt" aggregate="sum" />
                <column name="tndrTtl" aggregate="sum" />
            </columns>
            <orders>
                <order name="busHour" direction="ASC" />
                <order name="tndrName" direction="ASC" />
            </orders>
        </subjectArea>
    </query>
</queries>
<parameters>
...
</parameters>

<report>
    <frame type="table" name="Tender Media by Revenue Center" query-name="Query0">
        <groups>
            <group name="Total">
                <group-item name="organizationId" data-type="integer" />
                <group-expr name="AllTndrCnt" type="sum" column="tndrCnt" />
                <group-expr name="AllTndrTtl" type="sum" column="tndrTtl" />
                <group-calc name="AllPrctTndrTtl" data-type="percent">1</group-calc>
            </group>
            <group name="RvcAggregates">
                <group-item name="rvc" data-type="text" />
                <group-expr name="RvcTndrCnt" type="sum" column="tndrCnt" />
                <group-expr name="RvcTndrTtl" type="sum" column="tndrTtl" />
                <group-calc name="RvcPrctTndrTtl" data-type="percent">RvcTndrTtl /
```

```
AllTndrTtl</group-calc>
            </group>
        </groups>
        <bands>
            <band type="table-column-header">
                <cell type="text" print-width="68" alignment="left" data-
format="text">Name</cell>
                <cell type="text" print-width="68" alignment="right" data-
format="text">Tender Count</cell>
                <cell type="text" print-width="68" alignment="right" data-
format="text">Tender Amount</cell>
                <cell type="text" print-width="68" alignment="right" data-
format="text">Percent of Total Tender Amount</cell>
            </band>
            <band type="table-group-header" group-name="Total">
                <cell type="text" print-width="68" alignment="left" data-
format="text">TOTAL</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">AllTndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">AllTndrTtl</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="percent">AllPrctTndrTtl</cell>
            </band>
            <band type="table-group-header" group-name="RvcAggregates">
                <cell type="column" print-width="68" alignment="left" data-
format="text">rvc</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">RvcTndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">RvcTndrTtl</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="percent">RvcPrctTndrTtl</cell>
            </band>
            <band type="table-body">
                <cell type="column" print-width="68" alignment="left" data-
format="text">tndrName</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">tndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">tndrTtl</cell>
                <cell type="expr" print-width="68" alignment="right" data-
format="percent" name="TotalPercentage">tndrTtl / AllTndrTtl</cell>
            </band>
        </bands>
    </frame>
    ...
</report>
```

reportdef/report/frame/groups/group/group-expr

Description	This optional element aggregates the values from the data source (the query for this frame) as `sum`, `avg`, `min`, or `max`.

Diagram



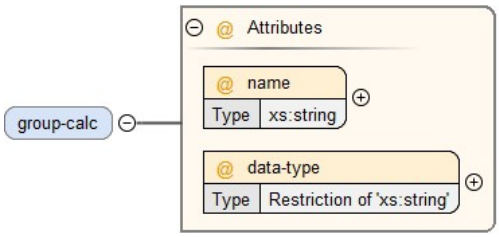
Attributes column, name, type

Example

```
<queries>
    <query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
        <subjectArea name="Tender Fixed Period Total">
            <columns>
                <column name="rvc" />
                <column name="busHour" />
                <column name="tndrName" />
                <column name="tndrCnt" aggregate="sum" />
                <column name="tndrTtl" aggregate="sum" />
            </columns>
            <orders>
                <order name="busHour" direction="ASC" />
                <order name="tndrName" direction="ASC" />
            </orders>
        </subjectArea>
    </query>
</queries>
<parameters>
...
</parameters>

<report>
    <frame type="table" name="Tender Media by Revenue Center" query-name="Query0">
        <groups>
            <group name="Total">
                <group-item name="organizationId" data-type="integer" />
                <group-expr name="AllTndrCnt" type="sum" column="tndrCnt" />
                <group-expr name="AllTndrTtl" type="sum" column="tndrTtl" />
                <group-calc name="AllPrctTndrTtl" data-type="percent">1</group-calc>
            </group>
            <group name="RvcAggregates">
                <group-item name="rvc" data-type="text" />
                <group-expr name="RvcTndrCnt" type="sum" column="tndrCnt" />
                <group-expr name="RvcTndrTtl" type="sum" column="tndrTtl" />
                <group-calc name="RvcPrctTndrTtl" data-type="percent">RvcTndrTtl /
AllTndrTtl</group-calc>
            </group>
        </groups>
        <bands>
            <band type="table-column-header">
                <cell type="text" print-width="68" alignment="left" data-
format="text">Name</cell>
                <cell type="text" print-width="68" alignment="right" data-
format="text">Tender Count</cell>
                <cell type="text" print-width="68" alignment="right" data-
```

```
format="text">Tender Amount</cell>
                <cell type="text" print-width="68" alignment="right" data-
format="text">Percent of Total Tender Amount</cell>
            </band>
            <band type="table-group-header" group-name="Total">
                <cell type="text" print-width="68" alignment="left" data-
format="text">TOTAL</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">AllTndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">AllTndrTtl</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="percent">AllPrctTndrTtl</cell>
            </band>
            <band type="table-group-header" group-name="RvcAggregates">
                <cell type="column" print-width="68" alignment="left" data-
format="text">rvc</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">RvcTndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">RvcTndrTtl</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="percent">RvcPrctTndrTtl</cell>
            </band>
            <band type="table-body">
                <cell type="column" print-width="68" alignment="left" data-
format="text">tndrName</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">tndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">tndrTtl</cell>
                <cell type="expr" print-width="68" alignment="right" data-
format="percent" name="TotalPercentage">tndrTtl / AllTndrTtl</cell>
            </band>
        </bands>
    </frame>
    ...
</report>
```

reportdef/report/frame/groups/group/group-calc

Description	This optional element performs arithmetical operations on one or more `group-expr` elements, or on values from queries that originate outside this `frame`.
Diagram	
Attributes	data-type, name

Example

```
<queries>
    <query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
        <subjectArea name="Tender Fixed Period Total">
            <columns>
```

```
                <column name="rvc" />
                <column name="busHour" />
                <column name="tndrName" />
                <column name="tndrCnt" aggregate="sum" />
                <column name="tndrTtl" aggregate="sum" />
            </columns>
            <orders>
                <order name="busHour" direction="ASC" />
                <order name="tndrName" direction="ASC" />
            </orders>
        </subjectArea>
    </query>
</queries>
<parameters>
...
</parameters>

<report>
    <frame type="table" name="Tender Media by Revenue Center" query-name="Query0">
        <groups>
            <group name="Total">
                <group-item name="organizationId" data-type="integer" />
                <group-expr name="AllTndrCnt" type="sum" column="tndrCnt" />
                <group-expr name="AllTndrTtl" type="sum" column="tndrTtl" />
                <group-calc name="AllPrctTndrTtl" data-type="percent">1</group-calc>
            </group>
            <group name="RvcAggregates">
                <group-item name="rvc" data-type="text" />
                <group-expr name="RvcTndrCnt" type="sum" column="tndrCnt" />
                <group-expr name="RvcTndrTtl" type="sum" column="tndrTtl" />
                <group-calc name="RvcPrctTndrTtl" data-type="percent">RvcTndrTtl /
AllTndrTtl</group-calc>
            </group>
        </groups>
        <bands>
            <band type="table-column-header">
                <cell type="text" print-width="68" alignment="left" data-
format="text">Name</cell>
                <cell type="text" print-width="68" alignment="right" data-
format="text">Tender Count</cell>
                <cell type="text" print-width="68" alignment="right" data-
format="text">Tender Amount</cell>
                <cell type="text" print-width="68" alignment="right" data-
format="text">Percent of Total Tender Amount</cell>
            </band>
            <band type="table-group-header" group-name="Total">
                <cell type="text" print-width="68" alignment="left" data-
format="text">TOTAL</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">AllTndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">AllTndrTtl</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="percent">AllPrctTndrTtl</cell>
            </band>
            <band type="table-group-header" group-name="RvcAggregates">
                <cell type="column" print-width="68" alignment="left" data-
format="text">rvc</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">RvcTndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">RvcTndrTtl</cell>
```

```
                <cell type="column" print-width="68" alignment="right" data-
format="percent">RvcPrctTndrTtl</cell>
            </band>
            <band type="table-body">
                <cell type="column" print-width="68" alignment="left" data-
format="text">tndrName</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">tndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">tndrTtl</cell>
                <cell type="expr" print-width="68" alignment="right" data-
format="percent" name="TotalPercentage">tndrTtl / AllTndrTtl</cell>
            </band>
        </bands>
    </frame>
    ...
</report>
```

reportdef/report/frame/bands

Description	A required element that is the layout of the rows and columns on the report.
Diagram	
Child Elements	band

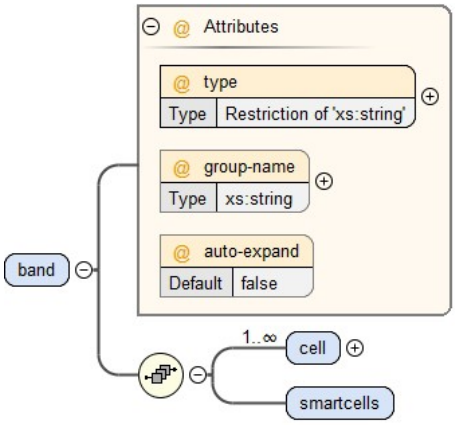
Example

In a `table` frame, a single `table-body` band can produce a nearly unlimited number of rows. The data will fill in as many rows as were returned by the source query. A `table-column-header` band will place a single row of cells above the body of a report, and is used to provide column names. The `table-group-header` places a single row above each set of body data each time the data is broken into its respective group (as defined by the `group-item` in `<groups>`.)

```
<queries>
    <query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
        <subjectArea name="Tender Fixed Period Total">
            <columns>
                <column name="rvc" />
                <column name="busHour" />
                <column name="tndrName" />
                <column name="tndrCnt" aggregate="sum" />
                <column name="tndrTtl" aggregate="sum" />
            </columns>
            <orders>
                <order name="busHour" direction="ASC" />
                <order name="tndrName" direction="ASC" />
            </orders>
        </subjectArea>
    </query>
</queries>
<parameters>
...
</parameters>

<report>
```

```
    <frame type="table" name="Tender Media by Revenue Center" query-name="Query0">
        <groups>
            <group name="Total">
                <group-item name="organizationId" data-type="integer" />
                <group-expr name="AllTndrCnt" type="sum" column="tndrCnt" />
                <group-expr name="AllTndrTtl" type="sum" column="tndrTtl" />
                <group-calc name="AllPrctTndrTtl" data-type="percent">1</group-calc>
            </group>
            <group name="RvcAggregates">
                <group-item name="rvc" data-type="text" />
                <group-expr name="RvcTndrCnt" type="sum" column="tndrCnt" />
                <group-expr name="RvcTndrTtl" type="sum" column="tndrTtl" />
                <group-calc name="RvcPrctTndrTtl" data-type="percent">RvcTndrTtl /
AllTndrTtl</group-calc>
            </group>
        </groups>
        <bands>
            <band type="table-column-header">
                <cell type="text" print-width="68" alignment="left" data-
format="text">Name</cell>
                <cell type="text" print-width="68" alignment="right" data-
format="text">Tender Count</cell>
                <cell type="text" print-width="68" alignment="right" data-
format="text">Tender Amount</cell>
                <cell type="text" print-width="68" alignment="right" data-
format="text">Percent of Total Tender Amount</cell>
            </band>
            <band type="table-group-header" group-name="Total">
                <cell type="text" print-width="68" alignment="left" data-
format="text">TOTAL</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">AllTndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">AllTndrTtl</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="percent">AllPrctTndrTtl</cell>
            </band>
            <band type="table-group-header" group-name="RvcAggregates">
                <cell type="column" print-width="68" alignment="left" data-
format="text">rvc</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">RvcTndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">RvcTndrTtl</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="percent">RvcPrctTndrTtl</cell>
            </band>
            <band type="table-body">
                <cell type="column" print-width="68" alignment="left" data-
format="text">tndrName</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">tndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">tndrTtl</cell>
                <cell type="expr" print-width="68" alignment="right" data-
format="percent" name="TotalPercentage">tndrTtl / AllTndrTtl</cell>
            </band>
        </bands>
    </frame>
    ...
</report>
```

reportdef/report/frame/bands/band

Description	A required element that is the layout and contents of each frame. Many `band` elements can exist within the `bands` element.
Diagram	
Child Elements	cell, smartcells
Attributes	auto-expand, group-name, type

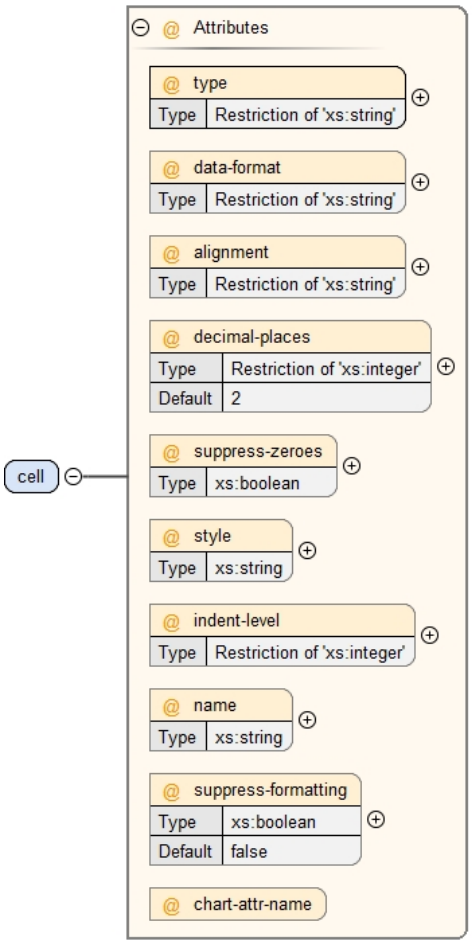
Example

In a `table` frame, a single `table-body` band can produce a nearly unlimited number of rows. The data will fill in as many rows as were returned by the source query. A `table-column-header` band will place a single row of cells above the body of a report, and is used to provide column names. The `table-group-header` places a single row above each set of body data each time the data is broken into its respective group (as defined by the `group-item` in `<groups>`.)

```
<queries>
    <query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
        <subjectArea name="Tender Fixed Period Total">
            <columns>
                <column name="rvc" />
                <column name="busHour" />
                <column name="tndrName" />
                <column name="tndrCnt" aggregate="sum" />
                <column name="tndrTtl" aggregate="sum" />
            </columns>
            <orders>
                <order name="busHour" direction="ASC" />
                <order name="tndrName" direction="ASC" />
            </orders>
        </subjectArea>
    </query>
</queries>
<parameters>
...
</parameters>

<report>
    <frame type="table" name="Tender Media by Revenue Center" query-name="Query0">
        <groups>
            <group name="Total">
```

```
                <group-item name="organizationId" data-type="integer" />
                <group-expr name="AllTndrCnt" type="sum" column="tndrCnt" />
                <group-expr name="AllTndrTtl" type="sum" column="tndrTtl" />
                <group-calc name="AllPrctTndrTtl" data-type="percent">1</group-calc>
            </group>
            <group name="RvcAggregates">
                <group-item name="rvc" data-type="text" />
                <group-expr name="RvcTndrCnt" type="sum" column="tndrCnt" />
                <group-expr name="RvcTndrTtl" type="sum" column="tndrTtl" />
                <group-calc name="RvcPrctTndrTtl" data-type="percent">RvcTndrTtl /
AllTndrTtl</group-calc>
            </group>
        </groups>
        <bands>
            <band type="table-column-header">
                <cell type="text" print-width="68" alignment="left" data-
format="text">Name</cell>
                <cell type="text" print-width="68" alignment="right" data-
format="text">Tender Count</cell>
                <cell type="text" print-width="68" alignment="right" data-
format="text">Tender Amount</cell>
                <cell type="text" print-width="68" alignment="right" data-
format="text">Percent of Total Tender Amount</cell>
            </band>
            <band type="table-group-header" group-name="Total">
                <cell type="text" print-width="68" alignment="left" data-
format="text">TOTAL</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">AllTndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">AllTndrTtl</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="percent">AllPrctTndrTtl</cell>
            </band>
            <band type="table-group-header" group-name="RvcAggregates">
                <cell type="column" print-width="68" alignment="left" data-
format="text">rvc</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">RvcTndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">RvcTndrTtl</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="percent">RvcPrctTndrTtl</cell>
            </band>
            <band type="table-body">
                <cell type="column" print-width="68" alignment="left" data-
format="text">tndrName</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">tndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">tndrTtl</cell>
                <cell type="expr" print-width="68" alignment="right" data-
format="percent" name="TotalPercentage">tndrTtl / AllTndrTtl</cell>
            </band>
        </bands>
    </frame>
    ...
</report>
```

reportdef/report/frame/bands/band/cell

Description	A required element that is layout and contents of a single cell within a band.
Diagram	
Attributes	alignment, chart-attr-name, data-format, decimal-places, indent-level, name, style, suppress-formatting, suppress-zeroes, type

Example

In a `table` frame, a single `table-body` band can produce a nearly unlimited number of rows. The data will fill in as many rows as were returned by the source query. A `table-column-header` band will place a single row of cells above the body of a report, and is used to provide column names. The `table-group-header` places a single row above each set of body data each time the data is broken into its respective group (as defined by the `group-item` in `<groups>`.)

```
<queries>
    <query datasource="" name="Query0" result-type="multiple-rows" type="iquery">
        <subjectArea name="Tender Fixed Period Total">
            <columns>
                <column name="rvc" />
                <column name="busHour" />
                <column name="tndrName" />
                <column name="tndrCnt" aggregate="sum" />
                <column name="tndrTtl" aggregate="sum" />
```

```
            </columns>
            <orders>
                <order name="busHour" direction="ASC" />
                <order name="tndrName" direction="ASC" />
            </orders>
        </subjectArea>
    </query>
</queries>
<parameters>
...
</parameters>

<report>
    <frame type="table" name="Tender Media by Revenue Center" query-name="Query0">
        <groups>
            <group name="Total">
                <group-item name="organizationId" data-type="integer" />
                <group-expr name="AllTndrCnt" type="sum" column="tndrCnt" />
                <group-expr name="AllTndrTtl" type="sum" column="tndrTtl" />
                <group-calc name="AllPrctTndrTtl" data-type="percent">1</group-calc>
            </group>
            <group name="RvcAggregates">
                <group-item name="rvc" data-type="text" />
                <group-expr name="RvcTndrCnt" type="sum" column="tndrCnt" />
                <group-expr name="RvcTndrTtl" type="sum" column="tndrTtl" />
                <group-calc name="RvcPrctTndrTtl" data-type="percent">RvcTndrTtl /
AllTndrTtl</group-calc>
            </group>
        </groups>
        <bands>
            <band type="table-column-header">
                <cell type="text" print-width="68" alignment="left" data-
format="text">Name</cell>
                <cell type="text" print-width="68" alignment="right" data-
format="text">Tender Count</cell>
                <cell type="text" print-width="68" alignment="right" data-
format="text">Tender Amount</cell>
                <cell type="text" print-width="68" alignment="right" data-
format="text">Percent of Total Tender Amount</cell>
            </band>
            <band type="table-group-header" group-name="Total">
                <cell type="text" print-width="68" alignment="left" data-
format="text">TOTAL</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">AllTndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">AllTndrTtl</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="percent">AllPrctTndrTtl</cell>
            </band>
            <band type="table-group-header" group-name="RvcAggregates">
                <cell type="column" print-width="68" alignment="left" data-
format="text">rvc</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">RvcTndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">RvcTndrTtl</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="percent">RvcPrctTndrTtl</cell>
            </band>
            <band type="table-body">
                <cell type="column" print-width="68" alignment="left" data-
```

```
format="text">tndrName</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="integer">tndrCnt</cell>
                <cell type="column" print-width="68" alignment="right" data-
format="currency">tndrTtl</cell>
                <cell type="expr" print-width="68" alignment="right" data-
format="percent" name="TotalPercentage">tndrTtl / AllTndrTtl</cell>
            </band>
        </bands>
    </frame>
    ...
</report>
```

reportdef/report/frame/bands/band/smartcells

Description	This optional element defines selectable column headings in the UI. The same properties and attributes assigned to normal cells are assigned to smart cells individually within this element. All cells within the smart cell element must use the name attribute. Each `band` element of a `bands` element must define the same number of smart cells.
Diagram	

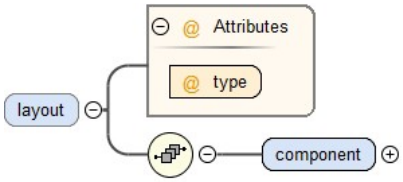
Example

In a `table` frame, a single `table-body` band can produce a nearly unlimited number of rows. The data will fill in as many rows as were returned by the source query. The `table-group-header` or `-footer` places a single row above or beneath each set of body data each time the data is broken into its respective group (as defined by the `group-item` in `<groups>`.)

```
<report>
    <frame type="table" name="Tender Media Totals" query-name="Query2" smart-cells-
count="2">
        <groups>
            <group name="Total">
                <group-item name="organizationId" data-type="integer" />
                <group-expr name="AllTndrCnt" type="sum" column="tndrCnt" />
                <group-expr name="AllTndrTtl" type="sum" column="tndrTtl" />
                <group-calc name="AllPrctTndrTtl" data-type="percent">1</group-calc>
            </group>
        </groups>
        <bands>
            <band type="table-column-header">
                <cell type="text" print-width="68" alignment="left" data-
format="text">Name</cell>
                <smart-cells>
                    <cell type="text" print-width="68" alignment="right" data-
format="text" name="Tender Count">Tender Count</cell>
                    <cell type="text" print-width="68" alignment="right" data-
format="text" name="Tender Amount">Tender Amount</cell>
                    <cell type="text" print-width="68" alignment="right" data-
format="text" name="PercentofTotalAmount">Percent of Total Tender Amount</cell>
                </smart-cells>
            </band>
            <band type="table-group-header" group-name="Total">
                <cell type="text" print-width="68" alignment="left" data-
format="text">TOTAL</cell>
                <smart-cells>
                    <cell type="column" print-width="68" alignment="right" data-
format="integer" name="Tender Count">AllTndrCnt</cell>
                    <cell type="column" print-width="68" alignment="right" data-
```

```
format="currency" name="Tender Amount">AllTndrTtl</cell>
                    <cell type="column" print-width="68" alignment="right" data-
format="percent" name="PercentofTotalAmount">AllPrctTndrTtl</cell>
                </smart-cells>
            </band>
            <band type="table-body">
                <cell type="column" print-width="68" alignment="left" data-
format="text">tndrName</cell>
                <smart-cells>
                    <cell type="column" print-width="68" alignment="right" data-
format="integer" name="Tender Count">tndrCnt</cell>
                    <cell type="column" print-width="68" alignment="right" data-
format="currency" name="Tender Amount">tndrTtl</cell>
                    <cell type="expr" print-width="68" alignment="right" data-
format="percent" name="PercentofTotalAmount">tndrTtl / AllTndrTtl</cell>
                </smart-cells>
            </band>
        </bands>
    </frame>
...
```

reportdef/report/layout

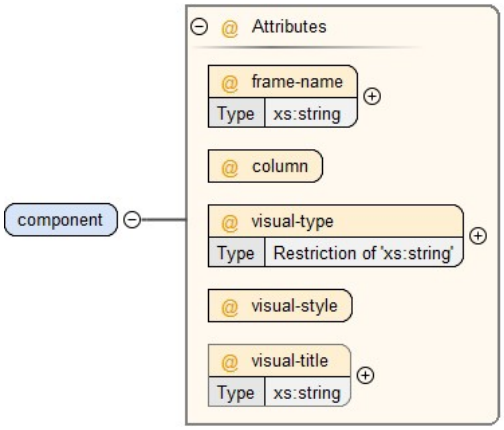
Description	This optional element defines the positioning of the `frame` component in the UI based on a column-grid layout.
Diagram	
Child Elements	component
Attributes	type

Example

```
<layout type="columner">

      <component frame-name="frame3Name" column="1" visual-type="ojPie" visual-style="oj-
sm-6" visual-title="TotalRevenue"/>
      <component frame-name="frame4Name" column="2" visual-type="ojBar" visual-style="oj-
sm-6" visual-title="TotalOperatingCosts"/>
      <component frame-name="frame5Name" column="1" visual-type="ojGrid" visual-
style="oj-sm-12" visual-title="TotalOperatingMargin"/>
      <component frame-name="frame6Name" column="1" visual-type="ojTable" visual-
style="oj-sm-12" visual-title="RevenueCenter"/>

</layout>
```

reportdef/report/layout/component

Description	This required element defines the component layout.

Diagram	
Attributes	column, frame-name, visual-style, visual-title, visual-type

Example

```
<layout type="columner">

      <component frame-name="frame3Name" column="1" visual-type="ojPie" visual-style="oj-sm-6" visual-title="TotalRevenue"/>
      <component frame-name="frame4Name" column="2" visual-type="ojBar" visual-style="oj-sm-6" visual-title="TotalOperatingCosts"/>
      <component frame-name="frame5Name" column="1" visual-type="ojGrid" visual-style="oj-sm-12" visual-title="TotalOperatingMargin"/>
      <component frame-name="frame6Name" column="1" visual-type="ojTable" visual-style="oj-sm-12" visual-title="RevenueCenter"/>

</layout>
```

Attributes

reportdef/tags/tag/@type

Description	A required attribute that is the type of tag.

Value Restrictions	`marketTSR`
	`marketQSR`
	`addOnTax`
	`inclusiveTax`
	`posSuperlite`
	`posE7`
	`posRES3700`
	`reportBuilder`

Used by	reportdef/tags/tag

reportdef/queries/query/subjectArea/columns/column/@name

Description	A required attribute that is the name of the column specified within a subject area. The value is a unique name assigned to column within subject area, given in metadata. The metadata appears in the **Subject Area (Alias)** section of the advanced editor of the report builder.
Used by	reportdef/queries/query/subjectArea/columns/column

reportdef/queries/query/subjectArea/columns/column/@aggregate

Description	An optional attribute. It is an aggregate function that should be applied to the column. Not all columns can be aggregated, only columns marked as supporting aggregate in metadata can use this attribute. The metadata appears in the **Subject Area (Alias)** section of the advanced editor of the report builder.

Value Restrictions	`COUNT`
	`AVG`
	`SUM`
	`MAX`
	`MIN`

Used by	reportdef/queries/query/subjectArea/columns/column

reportdef/queries/query/subjectArea/orders/order/@name

Description	A required attribute that is the name of the column specified within a subject area. The value is a unique name assigned to column within subject area, given in metadata. The metadata appears in the **Subject Area (Alias)** section of the advanced editor of the report builder.
Used by	reportdef/queries/query/subjectArea/orders/order

reportdef/queries/query/subjectArea/orders/order/@direction

Description	An optional attribute that specifies whether to use ascending or descending order for the given column.
Value Restrictions	`ASC` `DESC`
Used by	reportdef/queries/query/subjectArea/orders/order

reportdef/queries/query/subjectArea/filters/filter/@name

Description	A required attribute that is the name of the column specified within a subject area which will be used as filter. The value is a unique name assigned to column within subject area, given in metadata. The metadata appears in the **Subject Area (Alias)** section of the advanced editor of the report builder.
Used by	reportdef/queries/query/subjectArea/filters/filter

reportdef/queries/query/subjectArea/filters/filter/@type

Description	Required attribute that specifies the type of the filter.
Value Restrictions	List filters: • `stringFilter` • `intFilter` • `longFilter` • `doubleFilter` Range filters: • `dateRangeFilter` • `dateTimeRangeFilter` • `busHrRangeFilter` • `fpRangeFilter` Other filters: • `relationalFilter`
Used by	reportdef/queries/query/subjectArea/filters/filter

reportdef/queries/query/subjectArea/filters/filter/@logic

Description	An optional attribute. Logical operator `AND`/`OR` used with succeeding filter element, defaults to `AND` if not provided.
Value Restrictions	`AND` `OR`
Used by	reportdef/queries/query/subjectArea/filters/filter

reportdef/queries/query/subjectArea/filters/filter/@comparison

Description	An optional attribute. Specifies how the data attribute is to be compared in order to filter the records. `IS_NULL`, `IS_NOT_NULL` ignores the data attribute. `LIKE` will be treated as SQL `column LIKE 'DATA%'` It is only applicable if filter type= `relationalFilter`

Value Restrictions	`EQUAL`
	`NOT_EQUAL`
	`GREATER_THAN`
	`GREATER_THAN_EQ`
	`LESS_THAN`
	`LESS_THAN_EQ`
	`LIKE`
	`IS_NULL`
	`IS_NOT_NULL`

Used by	reportdef/queries/query/subjectArea/filters/filter

reportdef/queries/query/subjectArea/@name

Description	A required attribute that is the name of the subject area provided in metadata. The metadata appears in the **Subject Area (Alias)** section of the advanced editor of the report builder. The value is a unique name assigned to subject area.
Used by	reportdef/queries/query/subjectArea

reportdef/queries/query/@name

Description	A required attribute that is the name of the query. The value is any unique name assigned to a specific query.
Used by	reportdef/queries/query

reportdef/queries/query/@type

Description	A required attribute that is the type of query.
Value Restrictions	`iquery`
Used by	reportdef/queries/query

reportdef/queries/query/@result-type

Description	A required attribute that is the number of rows of data that will be returned by the query.
Value Restrictions	`multiple-rows` `single-row`
Used by	reportdef/queries/query

reportdef/queries/query/@allow-empty-resultset

Description	An optional attribute. When true, this permits query result sets to be empty, preventing the termination of report execution.
Value Restrictions	`true` `false` (default)
Used by	reportdef/queries/query

reportdef/parameters/parameter/parameter-default/@default-type

Description	A required attribute that is the parameter type. If set to `system`, then a default value must be assigned. When set to `all` in conjunction with the reserved system parameter `organizationid`, the list of locations will be filtered to include only those associated with the logged-in user. Do not use a parameter default type of none. This parameter default-type evaluates to null/ empty selections.

Value Restrictions	`all`
	`system`
	`none`

Used by	reportdef/parameters/parameter/parameter-default

reportdef/parameters/parameter/parameter-default/@default-value

Description	An optional attribute that is the value of the businessdate parameter.
Value Restrictions	**default-value** options for **businessdate**: • `MostRecentOperations` – This will return the last completed business date. • `Today` • `Yesterday` • `SameDayLastWeek` • `Past7Days` **Standard Calendar** options: • `SameDayLastYear` • `CurrentWeek` • `LastWeek` • `LastWeekWTD` • `SameWTDLastYear` • `CurrentMonth` • `LastMonthMTD` • `SameMTDLastYear` • `CurrentQuarter` • `LastQuarterQTD` • `SameQTDLastYear` • `CurrentYear` • `LastYearYTD`

	Financial Calendar options: • `SameDayLastYearFinancial` • `CurrentFinancialWeek` • `LastWeekFinancial` • `LastWeekWTDFinancial` • `SameWTDLastYearFinancial` • `CurrentFinancialPeriod` • `LastPeriodPTDFinancial` • `SamePTDLastYearFinancial` • `CurrentFinancialQuarter` • `LastQuarterQTDFinancial` • `SameQTDLastYearFinancial` • `CurrentFinancialYear` • `LastYearYTDFinancial` Several default-value options such as `Today` or `Past7Days` will select the same Date(s) regardless of the type of calendar used. However, most options use different logic to derive Dates for Standard than is used for Financial. If a report's default business date is set to use a Financial Calendar selection, and the organization does not have a Financial Calendar set up, then the reporting engine will fall back on the Standard Calendar equivalent as the default for the report. These options use the same logic as the QuickSelects seen in the Business Dates selection applet (notice the differences in the QuickSelect options listed when in the Standard Calendar versus Financial).
Used by	reportdef/parameters/parameter/parameter-default

reportdef/parameters/parameter/parameter-options/parameter-option/@name

Description	A required attribute. If set to `maximum` , then only a certain number of dates or locations may be selected. The number allowed is given in the value attribute. When set to `pre-select`, as soon as this report is run by a user the applet for which this attribute is set will appear.
Value Restrictions	`maximum` `pre-select`
Used by	reportdef/parameters/parameter/parameter-options/parameter-option

reportdef/parameters/parameter/parameter-options/parameter-option/@value

Description	A required attribute. For the `maximum` option, this will be any integer. For the `pre-select` option, a value of `yes` is required.
Value Restrictions	any integer `yes`
Used by	reportdef/parameters/parameter/parameter-options/parameter-option

reportdef/parameters/parameter/parameter-options/parameter-option/@error-message

Description	An optional attribute. This is the text that will appear if the selection rules are not met by the user.
Used by	reportdef/parameters/parameter/parameter-options/parameter-option

reportdef/parameters/parameter/@name

Description	A required attribute that is the name of the parameter. You can search the name of the parameter in the subject area list within the Advanced Editor. All options in this list that have a **Parameter** tag can be used as parameters and the alias should be used as the name.

Value Restrictions	`busDate`
	`locName`
	`rvc`
	`orderName`

Used by	reportdef/parameters/parameter

reportdef/parameters/parameter/@type

Description	A required attribute that is the type of the parameter. The `List` value lets you select multiple items and `Value` limits the selection to one item.
Value Restrictions	`value` `list`
Used by	reportdef/parameters/parameter

reportdef/parameters/parameter/@data-type

Description	A required attribute that is the data type of the parameter.

Value Restrictions	`text`
	`integer`
	`int64(bigint)`
	`daypart`
	`date`
	`date-time`

Used by	reportdef/parameters/parameter

reportdef/parameters/parameter/@selections

Description	An optional attribute. Number of parameter selections that can be executed in the portal user interface.

Value Restrictions	`1`
Used by	reportdef/parameters/parameter

reportdef/parameters/parameter/@displayed

Description	An optional attribute that displays an applet (for example, Location, Order Type, or Revenue Center) when this report is shown. This is used on a parent report.
Value Restrictions	`true` `false` (default)
Used by	reportdef/parameters/parameter

reportdef/parameters/parameter/@contextual

Description	An optional attribute that identifies whether the given parameter is displayed only if a certain condition is met.
Used by	reportdef/parameters/parameter

reportdef/parameters/parameter/@disable-sysparam

Description	An optional attribute. If the contextual conditions are met, this element identifies the system parameter to disable as a result of the contextual parameter taking precedence. The value is the name of the system parameter to disable.
Used by	reportdef/parameters/parameter

reportdef/report/frame/groups/group/group-item/@name

Description	A required attribute that is the name of the column for which there should be a break on each distinct value. The value is one of the columns from the query that this frame is based on.
Used by	reportdef/report/frame/groups/group/group-item

reportdef/report/frame/groups/group/group-item/@sort-type

Description	A required element that is the sort order of grouped data.
Value Restrictions	`ascending` `descending`
Used by	reportdef/report/frame/groups/group/group-item

reportdef/report/frame/groups/group/group-item/@data-type

Description	A required element that is the grouped value's data type.

Value Restrictions	`text`
	`currency`
	`date`
	`time`
	`integer`

Used by	reportdef/report/frame/groups/group/group-item

reportdef/report/frame/groups/group/group-expr/@name

Description	A required attribute that is the name of column (other than `group-item`) to be included in this group's data set. The value is defined by the author of the report. It must be a name of one of the `frame`'s source query fields (or a field alias).
Used by	reportdef/report/frame/groups/group/group-expr

reportdef/report/frame/groups/group/group-expr/@type

Description	A required attribute that is the arithmetical operation to be performed on the data field. Count can be used on columns with text or numbers.

Value Restrictions	`sum`
	`min`
	`count`
	`max`
	`avg`

Used by	reportdef/report/frame/groups/group/group-expr

reportdef/report/frame/groups/group/group-expr/@column

Description	A required attribute that is the column in a grouped data set on which arithmetic is performed. The value is a valid column name from the `frame`'s source query (or an alias).
Used by	reportdef/report/frame/groups/group/group-expr

reportdef/report/frame/groups/group/group-calc/@name

Description	A required unique name of the derived value. The value is defined by the author of the report. It must be a name of one of the `frame`'s source query fields (or a field alias).
Used by	reportdef/report/frame/groups/group/group-calc

reportdef/report/frame/groups/group/group-calc/@data-type

Description	A required attribute that is the data type of the new derived value.

Value Restrictions	`text`
	`currency`
	`date`
	`time`
	`integer`
	`numeric`
	`percent`

Used by	reportdef/report/frame/groups/group/group-calc

reportdef/report/frame/groups/group/@name

Description	A required attribute. This attribute gives the group its name. The value is defined by the author of the definition file.
Used by	reportdef/report/frame/groups/group

reportdef/report/frame/bands/band/cell/@type

Description	A required attribute and is the type of cell.

Value Restrictions	`column`
	`expr`
	`text`
	`macro`
	`char_text`

Used by	reportdef/report/frame/bands/band/cell

reportdef/report/frame/bands/band/cell/@data-format

Description	An optional attribute that is the data format of the cell contents.

Value Restrictions	`text`
	`currency`
	`date`
	`time`
	`integer`
	`numeric`
	`date-time`
	`gmt-date-time`
	`hour-min-sec`
	`percent`
	`period-start`

	`period-stop`
	`period-hour`
	`period-range`
	`hour-minute`
	`minute-second`
	`preText`

Used by	reportdef/report/frame/bands/band/cell

reportdef/report/frame/bands/band/cell/@alignment

Description	An optional attribute that is the justification of the cell contents.

Value Restrictions	`left`
	`center`
	`right`

Used by	reportdef/report/frame/bands/band/cell

reportdef/report/frame/bands/band/cell/@decimal-places

Description	An optional attribute that can define the number of decimal places for numeric or currency data types. • The data type **integer** does not have any decimal places. If used for currency or numeric values, the numbers will be rounded. • For the data type **numeric**, the report author can set `1` or more decimal places but cannot set `0`. If `0` decimal places are set then it will default to `2` decimal places. • For the data type **currency** the developer can set `1` or more decimal places but cannot set `0`. If `0` decimal places are set, then it will use the user's locale to set the number of decimal places.
Value Restrictions	`1` to any positive integer `2` (default)
Used by	reportdef/report/frame/bands/band/cell

reportdef/report/frame/bands/band/cell/@suppress-zeroes

Description	An optional attribute that eliminates the rendering of zero values in a cell. It applies only if data-type is `numeric`.
Value Restrictions	`true` `false`
Used by	reportdef/report/frame/bands/band/cell

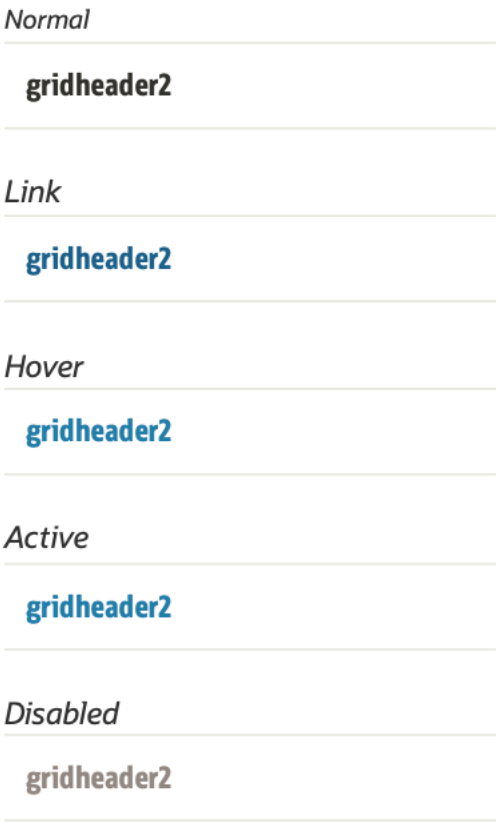
reportdef/report/frame/bands/band/cell/@style

Description	An optional attribute that can reference any of the styles listed below. If omitted, the default is the system style `normal`.

Value Restrictions	`normal`
	`col_header_1`
	`col_header_2`
	`header_0`
	`header_1`
	`header_2`
	`body_8`
	`body_7`
	`body_8_bold`
	`imgLayer`
	`clonedHeader`
	`gridheader1`
	Example:
	Normal
	gridheader1
	Link
	gridheader1
	Hover
	gridheader1
	Active
	gridheader1
	Disabled
	gridheader1

`gridheader2` and `gridheader3`

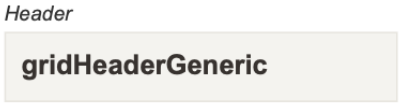
Example:



`grid_header_generic`

Use this value for all cells in a row that are not the first cell and are not a total.

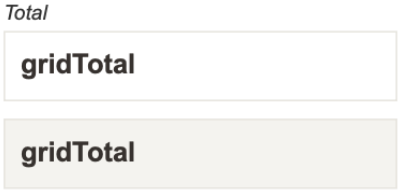
Example:



`grid_total`

Use this value for rows that have a total.

Example:



`empty_cell`

`empty_col`

`empty_cell_col`

`header_bare`

Use this value when a cell is at the top of a grid and only a bottom border is needed.

	`hidden_cell` Use this value when a cell is in the JSON but you do not want to show the data.
Used by	reportdef/report/frame/bands/band/cell

reportdef/report/frame/bands/band/cell/@indent-level

Description	An optional attribute that is used to add indentation to the beginning of text in a cell.
Value Restrictions	numeric range `0-5`
Used by	reportdef/report/frame/bands/band/cell

reportdef/report/frame/bands/band/cell/@name

Description	An optional attribute that is a unique name for the cell; must be used if value of `type` is `expr` and if the cell is part of a group of smart cells.
Used by	reportdef/report/frame/bands/band/cell

reportdef/report/frame/bands/band/cell/@suppress-formatting

Description	An optional attribute used to suppress even the default formats, such as commas as a thousands separator in numbers.
Value Restrictions	`true` `false` (default)
Used by	reportdef/report/frame/bands/band/cell

reportdef/report/frame/bands/band/cell/@chart- attr-name

Description	An optional attribute used to defined visualization-related attributes as described in the Visualizations chapter.
Used by	reportdef/report/frame/bands/band/cell

reportdef/report/frame/bands/band/@type

Description	A required attribute that is the band's function.
Value Restrictions	`stripe` (only if `frame`'s type is grid or flow) `table-column-header` (only if `frame`'s type is table) `table-group-header` (only if `frame`'s type is table) `table-body` (only if `frame`'s type is table)
Used by	reportdef/report/frame/bands/band

reportdef/report/frame/bands/band/@group-name

Description	An optional attribute that is the previously defined group being displayed by this band. This is only used when the `type` value is `table-group-header` or `table-group`.
Used by	reportdef/report/frame/bands/band

reportdef/report/frame/bands/band/@auto-expand

Description	An optional attribute used to set table row as expanded.
Value Restrictions	`true` `false` (default)
Used by	reportdef/report/frame

reportdef/report/frame/@type

Description	A required attribute that is the form of the report.
Value Restrictions	`grid` (Microsoft Excel-like style) `table` (columns) `flow` (usually used for accompanying text)
Used by	reportdef/report/frame

reportdef/report/frame/@query-name

Description	An optional attribute. It is the query that provides data to the report frame. The value is one of the names previously defined in `query`.
Used by	reportdef/report/frame

reportdef/report/frame/@behavior

Description	An optional attribute that sets this frame as a header, footer or body text. It can only be used when `type` value is flow.
Value Restrictions	`report-header` (for parameter display) `page-header` `page-footer standard` (default)
Used by	reportdef/report/frame

reportdef/report/frame/@default-style

Description	An optional attribute that is the style of text used in this frame instead of the default. The value is one of the globally defined styles. See reportdef/report/frame/bands/band/cell/@style.
Used by	reportdef/report/frame

reportdef/report/frame/@display-query-name

Description	An optional attribute used to determine whether a conditional frame is shown or hidden; must supply a `query` and a `query-display-value` for comparison; the query must be a single-row and the name of the field returned by the condition query must be called `condition`. The value is the name given to the `query` that returns the conditional value.
Used by	reportdef/report/frame

reportdef/report/frame/@display-query-value

Description	An optional attribute. If this value equals the returned `condition` field, then the frame will be displayed. Value to be compared to the value returned by the condition query identified in `query-display-name`.
Used by	reportdef/report/frame

reportdef/report/frame/@alignment

Description	An optional attribute that is the justification of the frame contents.

Value Restrictions	`left`
	`center`
	`right`

Used by	reportdef/report/frame

reportdef/report/frame/@smart-cells-count

Description	An optional attribute that is the number of displayed smart cells. Each band in the frame must have at least as many smart-cells definitions as smart cells count. The value is any number that equals the count of smart cells.
Used by	reportdef/report/frame

reportdef/report/layout/component/@frame-name

Description	A required attribute that is the frame component that will be included in the layout definition. The value is a valid frame name.
Used by	reportdef/report/layout/component

reportdef/report/layout/component/@column

Description	This is a required attribute that is the position index in a column-grid layout format.
Used by	reportdef/report/layout/component

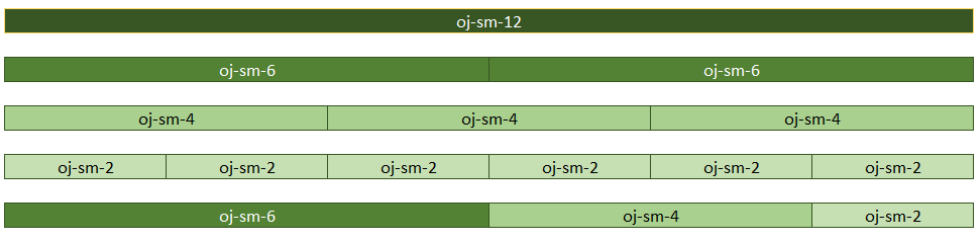
reportdef/report/layout/component/@visual-type

Description	A required attribute that is the visualization output type of the frame data. The Visualizations chapter provides more information.

Value Restrictions	`ojGrid`
	`ojTable`
	`ojBar`
	`ojPie`
	`ojPrimaryTile`
	`ojSecondaryTile`
	`ojDynamicTitle`
	`ojBarRefObj`
	`ojDualY`
	`ojScatter`
	`ojTitleLink`
	`ojCombo`

Used by	reportdef/report/layout/component

reportdef/report/layout/component/@visual-style

Description	A required attribute that defines the layout style of the components. The screen layout is divided into 12 equally sized sections. That lets you define how many components can fit into a row on the screen, plus how large each component is in relation to the other components and in relation to the available screen space. By default, a report component will stretch on the screen from left to right. The attribute for that component is `oj-sm-12`. To position two components with equal sizes into the same row on a report, use `oj-sm-6` for both. Other possible values are `oj-sm-4` and `oj-sm-2`. The report designer can also mix multiple, differently sized components into one row, however the total number of sections cannot exceed 12. The following illustration shows examples of how to use and combine different layout styles. 

Used by	reportdef/report/layout/component

reportdef/report/layout/component/@visual-title

Description	An optional attribute that is the header title of the visualization component. The value is the title displayed above the report component.
Used by	reportdef/report/layout/component

reportdef/report/layout/@type

Description	The type of layout.
Value Restrictions	`columner`
Used by	reportdef/report/layout

3

Expressions, Operators, and Functions

This section describes supported expressions, operators, and functions.

Supported Operators

All common arithmetic operators are supported. Boolean operators are also fully supported. Boolean expressions are evaluated to be either 1 or 0 (true or false respectively). The constants TRUE and FALSE are also available for use.

Table 2 - Supported Operators

Order of Precedence	Operator	Symbol
1st (highest)	Reference	:
2nd	Percent	%
3rd	Power (aka Exponent)	^
4th	Modulus	%%
5th	Division	/
6th	Multiplication	*
7th	Addition, Subtraction	+, -
8th (tie)	Less Than, Greater Than*	<, >
	Less Than or Equal, Greater Than or Equal*	<=, >=
9th	Not Equal, Equal*	<>, =
10th	Boolean Not	!
11th (lowest)	Boolean Or, Boolean And	||, &&
	String Concatenation	&
	Unary Plus, Unary Minus	+x, -x

Gray shading indicates operators not supported by Excel

CDATA Exceptions

If < or > operators are used in an XML report definition, they must be contained within a 'CDATA' section. This is done to avoid confusing the XML parsing engine, which identifies < and > as markup symbols.

For Example: Let's look at CDATA used in a conditional-style definition.

Below the condition NetSales > 10000 that would be used as the element value must be enclosed within the CDATA syntax as follows:

<conditional-style>

<condition style="Hi"><![CDATA[NetSales>10000]]></condition>

</conditional-style>

Cell and Range Syntax for Grid Frames

Excel-like absolute cell references are supported (i.e., A1, B3, C4), but only within the context of an expression within a report definition grid frame.

Absolute cell range references are supported (i.e., A1:A15, B3:D3). Likewise, these are only valid within expressions in report grid frames.

There are several other Excel cell and range reference syntaxes supported:

The cell in column A and row 10 of a report grid frame is A10

The range of cells in column A and rows 10 through 20 A10:A20

The range of cells in row 15 and columns B through EB15:E15

The following cell range syntax is NOT supported:

All cells in row 5 5:5

All cells in rows 5 through 105:10

All cells in column H H:H

All cells in columns H through JH:J

Also NOT supported are relative cell references (i.e., R1C1).

Excel-Like Functions

The framework allows for custom constants and functions to be registered with the parser. The following Excel-like functions have been implemented.

Table 3 - Exel-Like Functions

FUNCTION	COMMENTS
ABS(number)	Returns the absolute value of number
AND(logical1,logical2, …)	Returns true or false
AVERAGE(number1,number2, …)	Returns the average of all number arguments
CONCATENATE(string1,string2, …)	Returns concatenation of all string arguments
EQUALS (expression, expression)	evaluates two expression to see if they are equal and returns true or false
IF(logical_test,value_if_true, value_if_false)	Returns value of arg2 if arg1 evaluates to true, else returns value of arg3
ISERROR(expression)	Returns true if expression evaluates to either (null, NaN, or positive/negative infinity)
ISNULL (expression)	evaluates expression and if NULL returns true otherwise returns false
MOD (number,divisor)	where number is the number for which you want to find the remainder. divisor is the number by which you want to divide ‘number’. The number and divisor can be expressions
NOT(logical)	Returns true or false
OR(logical1,logical2, …)	Returns true or false

FUNCTION	COMMENTS
SUM(expression,expression, …)	Returns sum of expression list

Excel-Like Constants

The following Excel-like constants have been identified and defined to the parser.

Table 5 Excel-Like Constants

CONSTANT	COMMENTS
TRUE	Returns the integer value 1.
FALSE	Returns the integer value 0.
TODAY	Returns the current date.
NOW	Return the current date and time.

As with Excel functions, these functions can be nested to any level. Also, all expressions can include parentheses to force expression evaluation order.

The following is an example of nested functions, although granted it is not a very realistic example.

IF(SUM(salestotal1,salestotal2)<2000,TRUE,IF(salestotal>2000,SUM(salescount,salestotal),salestotal*.5))

It should also be noted that the following two feature extends the capabilities of what Excel supports.

Functions that take a variable number of arguments (ex. OR), are not limited on the number of arguments they can take. Excel has a limitation.

4
Visualizations

This section describes supported report visualizations. The visualizations are based on the Oracle Javascript Extension Toolkit (OJET) library. The existing XML report definition has been enhanced to generate the expected JSON data format.

The OJET Developer Cookbook provides more information and examples of visualizations.

Bar Chart

Attribute Name	XML Declaration	Description / Notes
name	`<cell type="text" data-format="text" alignment="left" style="col_header_1" print-width="130" chart-attr-name="seriesName">TotalOperatingCosts</cell>`	"seriesName" is a constant attribute name. The value is derived based on the cell type definition.
items	`<cell type="expr" data-format="integer" alignment="right" style="header_1" suppress-formatting="true">DBGET("DailyOpsQuery","prepCostTotal")</cell>`	This attribute must directly follow the "seriesName" declaration. This is the chart data points/items values. The "suppress-formatting" property must be set to true. The value is derived based on the cell type definition.
groupData	`<cell type="column" data-format="text" alignment="left" chart-attr-name="groupData">Group A</cell>`	"groupData" is a constant attribute name. The value is derived based on the cell type definition.
yAxisTitle	`<cell type="text" data-format="text" alignment="left" chart-attr-name="yAxisTitle">Y Axis Title</cell>`	"yAxisTitle" is a constant attribute name. The value is derived based on the cell type definition.
xAxisTitle	`<cell type="text" data-format="text" alignment="left" chart-attr-name="xAxisTitle">X Axis Title</cell>`	"xAxisTitle" is a constant attribute name. The value is derived based on the cell type definition.
hoverBehavior	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="hoverBehavior">dim</cell>`	"hoverBehavior" is a constant attribute name. The cell type must be set to "chart_text". The valid value is dim.

Attribute Name	XML Declaration	Description / Notes
stackValue	<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="stackValue">on</cell>	"stackValue" is a constant attribute name. The cell type must be set to "chart_text". Valid values areon or off.
hoverSeriesLabel	<cell type="text" data-format="text" alignment="left" chart-attr-name="hoverSeriesLabel">Test Series</cell>	"hoverSeriesLabel" is a constant attribute name. The value is derived based on the cell type definition.
hoverValueLabel	<cell type="text" data-format="text" alignment="left" chart-attr-name="hoverValueLabel">Test Value</cell>	"hoverValueLabel" is a constant attribute name. The value is derived based on the cell type definition.
hoverGroupLabel	<cell type="text" data-format="text" alignment="left" chart-attr-name="hoverGroupLabel">Test Group</cell>	"hoverGroupLabel" is a constant attribute name. The value is derived based on the cell type definition.
dataTypeSeriesFormat	<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="dataTypeSeriesFormat">percent</cell>	"dataTypeSeriesFormat" is a constant attribute name. This applies data formatting to "seriesName" values. The cell type must be set to "chart_text". The valid value is the same as those that can be defined in cell data-format.
dataTypeGroupFormat	<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="dataTypeGroupFormat">currency</cell>	"dataTypeGroupFormat" is a constant attribute name. This applies data formatting to "groupData" values. The cell type must be set to "chart_text". The valid value is the same as those that can be defined in cell data-format.

Attribute Name	XML Declaration	Description / Notes
dataType	<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="dataType">integer</cell>	"dataType" is a constant attribute name. This applies data formatting to "items" values. The cell type must be set to "chart_text". The valid value is the same as those that can be defined in cell data-format.
decimalPlaces	<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="decimalPlaces">2</cell>	"decimalPlaces" is a constant attribute name. This number of decimal places is applied to "items" values. The cell type must be set to "chart_text". The Value is an integer.

Example of XML Report Definition to Generate Bar Chart JSON Data

```
<report>
        <frame type="table" query-name="MajorGroupsQuery" name="barFrame">

            <bands>

                <band type="table-body">
                    <cell type="text" data-format="text" alignment="left" print-
width="140" chart-attr-name="seriesName">Series1</cell>
                    <cell type="column" data-format="currency" alignment="right"
suppress-formatting="true" print-width="70">salesTotal</cell>

        <cell type="text" data-format="text" alignment="left" print-width="140" chart-
attr-name="seriesName">Series2</cell>
                    <cell type="column" data-format="currency" alignment="right"
suppress-formatting="true" print-width="70">salesTotal</cell>

        <cell type="column" data-format="text" alignment="left" chart-attr-
name="groupData">majorGroupNameMaster</cell>

        <!-- extra cell to match count in table-column -->
        <cell type="chart_text" data-format="text" alignment="left"></cell>
        <cell type="chart_text" data-format="text" alignment="left"></cell>
        <cell type="chart_text" data-format="text" alignment="left"></cell>
        <cell type="chart_text" data-format="text" alignment="left"></cell>
        <cell type="chart_text" data-format="text" alignment="left"></cell>
        <cell type="chart_text" data-format="text" alignment="left"></
cell>
                </band>

       <band type="table-column-header">
         <!--  yAxisTitle chart attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="yAxisTitle">Y Axis Title</cell>

        <!--  xAxisTitle chart attribute  -->
```

```
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="xAxisTitle">X Axis Title</cell>

        <!-- hover series label attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="hoverSeriesLabel">Test Series</cell>

        <!-- hover value label attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="hoverValueLabel">Test Value</cell>

        <!-- hover group label attribute  -->
        <cell type="text" data-format="text" alignment="left" chart-attr-
name="hoverGroupLabel">Test Group</cell>

        <!--  hoverBehavior chart attribute  -->
                    <cell type="chart_text" data-format="text" alignment="left" chart-
attr-name="hoverBehavior">dim</cell>

        <!-- stackValue chart attribute  -->
                    <cell type="chart_text" data-format="text" alignment="left" chart-
attr-name="stackValue">off</cell>

        <!-- dataType chart attribute  -->
                    <cell type="chart_text" data-format="text" alignment="left" chart-
attr-name="dataType">integer</cell>

        <!-- decimalPlaces chart attribute  -->
                    <cell type="chart_text" data-format="text" alignment="left" chart-
attr-name="decimalPlaces">2</cell>

        <!-- dataTypeSeriesFormat chart attribute  -->
                    <cell type="chart_text" data-format="text" alignment="left" chart-
attr-name="dataTypeSeriesFormat">percent</cell>

        <!--  dataTypeGroupFormat chart attribute  -->
                    <cell type="chart_text" data-format="text" alignment="left" chart-
attr-name="dataTypeGroupFormat">currency</cell>
       </band>

           </bands>
        </frame>

        <layout type="columner">
            <component frame-name="barFrame" column="1" visual-type="ojBar" visual-
style="oj-sm-12" visual-title="TotalOperatingCosts"/>
        </layout>

</report>
```

Pie Chart

This section provides more information on pie chart attributes and XML.

Tip:

The Reporting and Analytics report builder cannot show negative values in a pie chart. Use the advanced editor to convert negative values to positive values to show the values in the pie chart. Convert negative values by adding `* -1` to the column name. The following example shows how to convert a negative discount amount to a positive amount:

Before: `sum_discAmt`

After: `sum_discAmt * -1`

Attribute Name	XML declaration	Description / Notes
name	`<cell type="text" data-format="text" alignment="left" style="col_header_1" print-width="130" chart-attr-name="seriesName">TotalOperatingCosts</cell>`	"seriesName" is a constant attribute name. The value is derived based on the cell type definition.
items	`<cell type="expr" data-format="integer" alignment="right" style="header_1" suppress-formatting="true">DBGET("DailyOpsQuery","prepCostTotal")</cell>`	This attribute must directly follow the "seriesName" declaration. This is the chart data points/items values. The "suppress-formatting" property must be set to true. The value is derived based on the cell type definition.
hoverSeriesLabel	`<cell type="text" data-format="text" alignment="left" chart-attr-name="hoverSeriesLabel">Test Series</cell>`	"hoverSeriesLabel" is a constant attribute name. The value is derived based on the cell type definition.
hoverValueLabel	`<cell type="text" data-format="text" alignment="left" chart-attr-name="hoverValueLabel">Test Value</cell>`	"hoverValueLabel" is a constant attribute name. The value is derived based on the cell type definition.
extraItem	`<cell type="column" data-format="integer" alignment="left" print-width="140" chart-attr-name="extraItem" >salesTotal</cell>`	These are extra chart data points/items values. The value is derived based on the cell type definition.
hoverExtraItemLabel	`<cell type="text" data-format="text" alignment="left" chart-attr-name="hoverExtraItemLabel">Test ExtraItem</cell>`	"hoverExtraItemLabel" is a constant attribute name. The value is derived based on the cell type definition.

Attribute Name	XML declaration	Description / Notes
dataTypeExtraItemFormat	<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="dataTypeExtraItemFormat">percent</cell>	"dataTypeExtraItemFormat" is a constant attribute name. This applies data formatting to "extraItem" values. The cell type must be set to "chart_text". The valid value is the same as those that can be defined in cell data-format.

Example of XML Report Definition to Generate Pie Chart JSON Data

```
<report>

        <frame type="table" query-name="MajorGroupsQuery" name="barFrame">
               <bands>

                <band type="table-body">
                    <cell type="column" data-format="text" alignment="left" print-
width="140" chart-attr-name="seriesName">majorGroupNameMaster</cell>
                    <cell type="column" data-format="currency" alignment="right"
suppress-formatting="true" print-width="70">salesTotal</cell>

        <cell type="column" data-format="integer" alignment="left" print-width="140"
chart-attr-name="extraItem">salesTotal</cell>

        <!-- extra cell to match count in table-column -->
        <cell type="chart_text" data-format="text" alignment="left"></cell>
        </band>

                <band type="table-column-header">
        <!-- hover series label attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="hoverSeriesLabel">Test Series</cell>

                    <!-- hover value label attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="hoverValueLabel">Test Value</cell>

        <!-- hover extraItem label attribute  -->
        <cell type="text" data-format="text" alignment="left" chart-attr-
name="hoverExtraItemLabel">Test ExtraItem</cell>

        <!-- dataTypeExtraItemFormat attribute  -->
        <cell type="chart_text" data-format="text" alignment="left" chart-attr-
name="dataTypeExtraItemFormat">percent</cell>
       </band>

           </bands>
        </frame>

        <layout type="columner">
            <component frame-name="barFrame" column="1" visual-type="ojPie" visual-
style="oj-sm-12" visual-title="TotalOperatingCosts"/>
        </layout>

 </report>
```

Dual-Y Chart

JSON Data Attribute	XML declaration	Description / Notes
name	`<cell type="text" data-format="text" alignment="left" style="col_header_1" print-width="130" chart-attr-name="seriesName">TotalOperatingCosts</cell>`	"seriesName" is a constant attribute name. The value is derived based on the cell type definition.
items	`<cell type="expr" data-format="integer" alignment="right" style="header_1" suppress-formatting="true">DBGET("DailyOpsQuery","prepCostTotal")</cell>`	This attribute must directly follow the "seriesName" declaration. This is the chart data points/items values. The "suppress-formatting" property must be set to true. The value is derived based on the cell type definition.
assignedToY2	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="assignedToY2">on</cell>`	This attribute must directly follow the chart data points/items value declaration to assign Y-axis plotting. "assignedToY2" is a constant attribute name. The cell type must be set to "chart_text". Valid values are on or off.
groupData	`<cell type="column" data-format="text" alignment="left" chart-attr-name="groupData">Group A</cell>`	"groupData" is a constant attribute name. The value is derived based on the cell type definition.
yAxisTitle	`<cell type="text" data-format="text" alignment="left" chart-attr-name="yAxisTitle">Y Axis Title</cell>`	"yAxisTitle" is a constant attribute name. The value is derived based on the cell type definition.
y2AxisTitle	`<cell type="text" data-format="text" alignment="left" chart-attr-name="y2AxisTitle">Y2 Axis Title</cell>`	"y2AxisTitle" is a constant attribute name. The value is derived based on the cell type definition.
xAxisTitle	`<cell type="text" data-format="text" alignment="left" chart-attr-name="xAxisTitle">Y Axis Title</cell>`	"xAxisTitle" is a constant attribute name. The value is derived based on the cell type definition.

JSON Data Attribute	XML declaration	Description / Notes
dualY	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="dualY">on</cell>`	"dualY" is a constant attribute name. The cell type must be set to "chart_text". Valid values are on or off.
stackValue	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="stackValue">on</cell>`	"stackValue" is a constant attribute name. The cell type must be set to "chart_text". Valid values are on or off.
hoverSeriesLabel	`<cell type="text" data-format="text" alignment="left" chart-attr-name="hoverSeriesLabel">Test Series</cell>`	"hoverSeriesLabel" is a constant attribute name. The value is derived based on the cell type definition.
hoverValueLabel	`<cell type="text" data-format="text" alignment="left" chart-attr-name="hoverValueLabel">Test Value</cell>`	"hoverValueLabel" is a constant attribute name. The value is derived based on the cell type definition.
hoverGroupLabel	`<cell type="text" data-format="text" alignment="left" chart-attr-name="hoverGroupLabel">Test Group</cell>`	"hoverGroupLabel" is a constant attribute name. The value is derived based on the cell type definition.
hoverValueY2Label	`<cell type="text" data-format="text" alignment="left" chart-attr-name="hoverValueY2Label">Test Value Y2</cell>`	"hoverValueY2Label" is a constant attribute name. The value is derived based on the cell type definition.
dataType	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="dataType">integer</cell>`	"dataType" is a constant attribute name. This applies data formatting to "items" values. The cell type must be set to "chart_text". The valid value is the same as those that can be defined in cell data-format.
decimalPlaces	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="decimalPlaces">2</cell>`	"decimalPlaces" is a constant attribute name. This set number of decimal places to "items" values. The cell type must be set to "chart_text". The Value is an integer.

JSON Data Attribute	XML declaration	Description / Notes
dataTypeY2	<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="dataTypeY2">integer</cell>	"dataTypeY2" is a constant attribute name. This applies data formatting to "items" values in the Y-2 axis. The cell type must be set to "chart_text". The valid value is the same as those that can be defined in cell data-format.
decimalPlacesY2	<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="decimalPlacesY2">2</cell>	"decimalPlacesY2" is a constant attribute name. This set number of decimal places to "items" values in the Y-2 axis. The cell type must be set to "chart_text". The value is an integer.
dataTypeSeriesFormat	<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="dataTypeSeriesFormat">percent</cell>	"dataTypeSeriesFormat" is a constant attribute name. This applies data formatting to "seriesName" values. The cell type must be set to "chart_text". The valid value is the same as those that can be defined in cell data-format.
dataTypeGroupFormat	<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="dataTypeGroupFormat">currency</cell>	"dataTypeGroupFormat" is a constant attribute name. This applies data formatting to "groupData" values. The cell type must be set to "chart_text". The valid value is the same as those that can be defined in cell data-format.

Example of XML Report Definition to Generate Dual-Y Chart JSON Data

```
<report>

        <frame type="table" query-name="MajorGroupsQuery" name="dualYFrame">
           <bands>

               <band type="table-body">
                   <cell type="text" data-format="text" alignment="left" print-
width="140" indent-level="3" chart-attr-name="seriesName">Series1</cell>
```

```
                    <cell type="column" data-format="currency" alignment="right"
suppress-formatting="true" print-width="70">salesTotal</cell>
          <cell type="chart_text" data-format="text" alignment="left" chart-attr-
name="assignedToY2">on</cell>

          <cell type="text" data-format="text" alignment="left" print-width="140" indent-
level="3" chart-attr-name="seriesName">Series2</cell>
                     <cell type="column" data-format="currency" alignment="right"
suppress-formatting="true" print-width="70">salesTotal</cell>
          <cell type="chart_text" data-format="text" alignment="left" chart-attr-
name="assignedToY2">off</cell>

          <cell type="column" data-format="text" alignment="left" chart-attr-
name="groupData">majorGroupNameMaster</cell>

          <!-- extra cell to match count in table-column -->
         <cell type="chart_text" data-format="text" alignment="left"></cell>
         <cell type="chart_text" data-format="text" alignment="left"></cell>
         <cell type="chart_text" data-format="text" alignment="left"></cell>
         <cell type="chart_text" data-format="text" alignment="left"></
cell>
         <cell type="chart_text" data-format="text" alignment="left"></
cell>
        <cell type="chart_text" data-format="text" alignment="left"></
cell>
                </band>

     <band type="table-column-header">

                     <!--  yAxisTitle chart attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="yAxisTitle">Y Axis Title</cell>

         <!--  y2AxisTitle chart attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="y2AxisTitle">Y2 Axis Title</cell>

         <!--  xAxisTitle chart attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="xAxisTitle">X Axis Title</cell>

          <!-- hover series label attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="hoverSeriesLabel">Test Series</cell>

          <!-- hover value label attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="hoverValueLabel">Test Value</cell>

          <!-- hover group label attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="hoverGroupLabel">Test Group</cell>

           <!-- hover value Y2 label attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="hoverValueY2Label">Test Value Y2</cell>

          <!--  dualY chart attribute  -->
                    <cell type="chart_text" data-format="text" alignment="left" chart-
attr-name="dualY">on</cell>

          <!-- stackValue chart attribute  -->
```

```
                    <cell type="chart_text" data-format="text" alignment="left" chart-
attr-name="stackValue">off</cell>

          <!-- dataType chart attribute  -->
                    <cell type="chart_text" data-format="text" alignment="left" chart-
attr-name="dataType">currency</cell>

          <!-- decimalPlaces chart attribute  -->
                    <cell type="chart_text" data-format="text" alignment="left" chart-
attr-name="decimalPlaces">2</cell>

          <!-- dataTypeY2 chart attribute  -->
                    <cell type="chart_text" data-format="text" alignment="left" chart-
attr-name="dataTypeY2">currency</cell>

           <!-- decimalPlacesY2 chart attribute  -->
                    <cell type="chart_text" data-format="text" alignment="left" chart-
attr-name="decimalPlacesY2">2</cell>
     </band>

          </bands>
        </frame>

        <layout type="columner">
            <component frame-name="dualYFrame" column="1" visual-type="ojDualY" visual-
style="oj-sm-12" visual-title=""/>
        </layout>

    </report>
```

Bar Reference Object Chart

JSON Data Attribute	XML declaration	Description / Notes
name	`<cell type="text" data-format="text" alignment="left" style="col_header_1" print-width="130" chart-attr-name="seriesName">TotalOperatingCosts</cell>`	`"seriesName"` is a constant attribute name. The value is derived based on the cell type definition.
items	`<cell type="expr" data-format="integer" alignment="right" style="header_1" suppress-formatting="true">DBGET("DailyOpsQuery","prepCostTotal")</cell>`	This attribute must directly follow the `"seriesName"` declaration. This is the chart data points/items values. The `"suppress-formatting"` property must be set to `true`. The value is derived based on the cell type definition.
groupData	`<cell type="column" data-format="text" alignment="left" chart-attr-name="groupData">Group A</cell>`	`"groupData"` is a constant attribute name. The value is derived based on the cell type definition.

JSON Data Attribute	XML declaration	Description / Notes
stackValue	<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="stackValue">on</cell>	"stackValue" is a constant attribute name. The cell type must be set to "chart_text". Valid values areon or off.
xAxisTitle	<cell type="text" data-format="text" alignment="left" chart-attr-name="xAxisTitle">Y Axis Title</cell>	"xAxisTitle" is a constant attribute name. The value is derived based on the cell type definition.
xAxisRefObjValue	<cell type="text" data-format="text" alignment="left" chart-attr-name="xAxisRefObjValue">Group A</cell>	"xAxisRefObjValue" is a constant attribute name. The value is derived based on the cell type definition.
xAxisRefObjLegendText	<cell type="text" data-format="text" alignment="left" chart-attr-name="xAxisRefObjLegendText">X Axis Object Reference Text</cell>	"xAxisRefObjLegendText" is a constant attribute name. The value is derived based on the cell type definition.
xAxisRefObjHoverDescValue	<cell type="text" data-format="text" alignment="left" chart-attr-name="xAxisRefObjHoverDescValue">X Axis Sample Reference Line</cell>	"xAxisRefObjHoverDescValue" is a constant attribute name. The value is derived based on the cell type definition.
yAxisTitle	<cell type="text" data-format="text" alignment="left" chart-attr-name="yAxisTitle">Y Axis Title</cell>	"yAxisTitle" is a constant attribute name. The value is derived based on the cell type definition.
yAxisRefObjValue	<cell type="text" data-format="text" alignment="left" chart-attr-name="yAxisRefObjValue">25000</cell>	"yAxisRefObjValue" is a constant attribute name. The value is derived based on the cell type definition.
yAxisRefObjLegendText	<cell type="text" data-format="text" alignment="left" chart-attr-name="yAxisRefObjLegendText">Y Axis Object Reference Text</cell>	"yAxisRefObjLegendText" is a constant attribute name. The value is derived based on the cell type definition.
yAxisRefObjHoverDescValue	<cell type="text" data-format="text" alignment="left" chart-attr-name="yAxisRefObjHoverDescValue">Y Axis Sample Reference Line</cell>	"yAxisRefObjHoverDescValue" is a constant attribute name. The value is derived based on the cell type definition.

JSON Data Attribute	XML declaration	Description / Notes
hoverSeriesLabel	`<cell type="text" data-format="text" alignment="left" chart-attr-name="hoverSeriesLabel">Test Series</cell>`	"hoverSeriesLabel" is a constant attribute name. The value is derived based on the cell type definition.
hoverValueLabel	`<cell type="text" data-format="text" alignment="left" chart-attr-name="hoverValueLabel">Test Value</cell>`	"hoverValueLabel" is a constant attribute name. The value is derived based on the cell type definition.
hoverGroupLabel	`<cell type="text" data-format="text" alignment="left" chart-attr-name="hoverGroupLabel">Test Group</cell>`	"hoverGroupLabel" is a constant attribute name. The value is derived based on the cell type definition.
dataType	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="dataType">integer</cell>`	"dataType" is a constant attribute name. This applies data formatting to "items" values. The cell type must be set to "chart_text". The valid value is the same as those that can be defined in cell data-format.
decimalPlaces	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="decimalPlaces">2</cell>`	"decimalPlaces" is a constant attribute name. This set number of decimal places to "items" values. The cell type must be set to "chart_text". The Value is an integer.
dataTypeSeriesFormat	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="dataTypeSeriesFormat">percent</cell>`	"dataTypeSeriesFormat" is a constant attribute name. This applies data formatting to "seriesName" values. The cell type must be set to "chart_text". The valid value is the same as those that can be defined in cell data-format.

JSON Data Attribute	XML declaration	Description / Notes
dataTypeGroupFormat	<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="dataTypeGroupFormat">currency</cell>	"dataTypeGroupFormat" is a constant attribute name. This applies data formatting to "groupData" values. The cell type must be set to "chart_text". The valid value is the same as those that can be defined in cell data-format.

Example of XML Report Definition to Generate Bar Reference Object JSON Data

```
<report>

       <frame type="table" query-name="MajorGroupsQuery" name="barRefObjFrame">
            <bands>

                <band type="table-body">
                    <cell type="column" data-format="text" alignment="left" print-
width="140" indent-level="3" chart-attr-name="seriesName">majorGroupNameMaster</cell>
                    <cell type="column" data-format="currency" alignment="right"
suppress-formatting="true" print-width="70">salesTotal</cell>

          <!-- extra cell to match count in table-column -->
          <cell type="chart_text" data-format="text" alignment="left"></cell>
          <cell type="chart_text" data-format="text" alignment="left"></cell>
          <cell type="chart_text" data-format="text" alignment="left"></cell>
          <cell type="chart_text" data-format="text" alignment="left"></cell>
          <cell type="chart_text" data-format="text" alignment="left"></cell>
          <cell type="chart_text" data-format="text" alignment="left"></cell>
          <cell type="chart_text" data-format="text" alignment="left"></cell>
          <cell type="chart_text" data-format="text" alignment="left"></cell>
          <cell type="chart_text" data-format="text" alignment="left"></cell>
          <cell type="chart_text" data-format="text" alignment="left"></cell>
          <cell type="chart_text" data-format="text" alignment="left"></cell>
      </band>

    <band type="table-column-header">
                    <!--  groupData chart attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="groupData">Group A</cell>

          <!--  xAxisTitle chart attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="xAxisTitle">X Axis Title</cell>

          <!--  xAxisRefObjValue chart attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="xAxisRefObjValue">Group A</cell>

          <!--  xAxisRefObjLegendText chart attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="xAxisRefObjLegendText">X Axis Object Reference Text</cell>

          <!--  xAxisRefObjHoverDescValue chart attribute  -->
```

```
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="xAxisRefObjHoverDescValue">X Axis Sample Reference Line</cell>

          <!--  yAxisTitle chart attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="yAxisTitle">Y Axis Title</cell>

           <!--  yAxisRefObjValue chart attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="yAxisRefObjValue">25000</cell>

          <!--  yAxisRefObjLegendText chart attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="yAxisRefObjLegendText">Y Axis Object Reference Text</cell>

           <!--  yAxisRefObjHoverDescValue chart attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="yAxisRefObjHoverDescValue">Y Axis Sample Reference Line</cell>

          <!-- hover series label attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="hoverSeriesLabel">Test Series</cell>

          <!-- hover value label attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="hoverValueLabel">Test Value</cell>

           <!-- hover group label attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="hoverGroupLabel">Test Group</cell>

           <!-- stackValue chart attribute  -->
                    <cell type="chart_text" data-format="text" alignment="left" chart-
attr-name="stackValue">on</cell>
                </band>

            </bands>
        </frame>

        <layout type="columner">
            <component frame-name="barRefObjFrame" column="1" visual-type="ojBarRefObj"
visual-style="oj-sm-12" visual-title=""/>
        </layout>

    </report>
```

Scatter Chart

JSON Data Attribute	XML declaration	Description / Notes
name	`<cell type="text" data-format="text" alignment="left" style="col_header_1" print-width="130" chart-attr-name="seriesName">TotalOperatingCosts</cell>`	`"seriesName"` is a constant attribute name. The value is derived based on the cell type definition.

JSON Data Attribute	XML declaration	Description / Notes
items – x, y	<cell type="column" data-format="currency" alignment="right" suppress-formatting="true" print-width="70" chart-attr-name="xAxisItem">salesTotal</cell> <cell type="column" data-format="currency" alignment="right" suppress-formatting="true" print-width="70" chart-attr-name="yAxisItem">salesTotal</cell>	This attribute must directly follow the "seriesName" declaration. This is the chart data points/items values. The suppress-formatting property must be set to true. Value is derived based on the cell type definition. "xAxisItem" is a constant attribute name. The value is derived based on the cell type definition. "yAxisItem" is a constant attribute name. The value is derived based on the cell type definition.
groupData	<cell type="column" data-format="text" alignment="left" chart-attr-name="groupData">Group A</cell>	"groupData" is a constant attribute name. The value is derived based on the cell type definition.
xAxisTitle	<cell type="text" data-format="text" alignment="left" chart-attr-name="xAxisTitle">Y Axis Title</cell>	"xAxisTitle" is a constant attribute name. The value is derived based on the cell type definition.
yAxisTitle	<cell type="text" data-format="text" alignment="left" chart-attr-name="yAxisTitle">Y Axis Title</cell>	"yAxisTitle" is a constant attribute name. The value is derived based on the cell type definition.
hoverBehavior	<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="hoverBehavior">dim</cell>	"hoverBehavior" is a constant attribute name. The cell type must be set to "chart_text". The valid value is dim.
hoverSeriesLabel	<cell type="text" data-format="text" alignment="left" chart-attr-name="hoverSeriesLabel">Test Series</cell>	"hoverSeriesLabel" is a constant attribute name. The value is derived based on the cell type definition.
hoverValueLabel	<cell type="text" data-format="text" alignment="left" chart-attr-name="hoverValueLabel">Test Value</cell>	"hoverValueLabel" is a constant attribute name. The value is derived based on the cell type definition.

JSON Data Attribute	XML declaration	Description / Notes
hoverGroupLabel	`<cell type="text" data-format="text" alignment="left" chart-attr-name="hoverGroupLabel">Test Group</cell>`	"hoverGroupLabel" is a constant attribute name. The value is derived based on the cell type definition.
hoverYLabel	`<cell type="text" data-format="text" alignment="left" chart-attr-name="hoverYLabel">Test Y</cell>`	"hoverYLabel" is a constant attribute name. The value is derived based on the cell type definition.
hoverXLabel	`<cell type="text" data-format="text" alignment="left" chart-attr-name="hoverXLabel">Test X</cell>`	"hoverXLabel" is a constant attribute name. Value is derived based on the cell type definition.
dataType	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="dataType">integer</cell>`	"dataType" is a constant attribute name. This applies data formatting to "items" values. The cell type must be set to "chart_text". The valid value is the same as those that can be defined in cell data-format.
decimalPlaces	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="decimalPlaces">2</cell>`	"decimalPlaces" is a constant attribute name. This set number of decimal places to "items" values. The cell type must be set to "chart_text". The Value is an integer.
dataTypeSeriesFormat	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="dataTypeSeriesFormat">percent</cell>`	"dataTypeSeriesFormat" is a constant attribute name. This applies data formatting to "seriesName" values. The cell type must be set to "chart_text". The valid value is the same as those that can be defined in cell data-format.

JSON Data Attribute	XML declaration	Description / Notes
dataTypeGroupFormat	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="dataTypeGroupFormat">currency</cell>`	"dataTypeGroupFormat" is a constant attribute name. This applies data formatting to "groupData" values. The cell type must be set to "chart_text". The valid value is the same as those that can be defined in cell data-format.
dataTypeXAxisFormat	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="dataTypeXAxisFormat">integer</cell>`	"dataTypeXAxisFormat" is a constant attribute name. The cell type must be set to "chart_text". The valid value is the same as those that can be defined in cell data-format.
dataTypeYAxisFormat	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="dataTypeYAxisFormat">percent</cell>`	"dataTypeYAxisFormat" is a constant attribute name. The cell type must be set to "chart_text". The valid value is the same as those that can be defined in cell data-format.

Example of XML Report Definition to Generate Scatter Chart JSON Data

```
<report>
    <frame type="table" query-name="MajorGroupsQuery" name="scatterFrame">
        <bands>
            <band type="table-body">
                <cell type="text" data-format="text" alignment="left" print-width="140"
indent-level="3" chart-attr-name="seriesName">Series1</cell>
                <cell type="column" data-format="currency" alignment="right" suppress-
formatting="true" print-width="70" chart-attr-name="xAxisItem">salesTotal</cell>
                <cell type="column" data-format="currency" alignment="right" suppress-
formatting="true" print-width="70" chart-attr-name="yAxisItem">salesTotal</cell>

                <cell type="text" data-format="text" alignment="left" print-width="140"
indent-level="3" chart-attr-name="seriesName">Series2</cell>
                <cell type="column" data-format="currency" alignment="right" suppress-
formatting="true" print-width="70" chart-attr-name="xAxisItem">salesTotal</cell>
                <cell type="column" data-format="currency" alignment="right" suppress-
formatting="true" print-width="70" chart-attr-name="yAxisItem">salesTotal</cell>

                <cell type="column" data-format="text" alignment="left" chart-attr-
name="groupData">majorGroupNameMaster</cell>

                <!-- extra cell to match count in table-body -->
                <cell type="chart_text" data-format="text" alignment="left"></cell>
                <cell type="chart_text" data-format="text" alignment="left"></cell>
```

```
                <cell type="chart_text" data-format="text" alignment="left"></cell>
                <cell type="chart_text" data-format="text" alignment="left"></
cell>
            </band>
            <band type="table-column-header">
                <!--  xAxisTitle chart attribute  -->
                <cell type="text" data-format="text" alignment="left" chart-attr-
name="xAxisTitle">X Axis Title</cell>

                <!--  yAxisTitle chart attribute  -->
                <cell type="text" data-format="text" alignment="left" chart-attr-
name="yAxisTitle">Y Axis Title</cell>

                <!-- hover series label attribute  -->
                <cell type="text" data-format="text" alignment="left" chart-attr-
name="hoverSeriesLabel">Test Series</cell>

                <!-- hover value label attribute  -->
                <cell type="text" data-format="text" alignment="left" chart-attr-
name="hoverValueLabel">Test Value</cell>

                <!-- hover group label attribute  -->
                <cell type="text" data-format="text" alignment="left" chart-attr-
name="hoverGroupLabel">Test Group</cell>

                <!-- hover Y label attribute  -->
                <cell type="text" data-format="text" alignment="left" chart-attr-
name="hoverYLabel">Test Y</cell>

                <!-- hover X label attribute  -->
                <cell type="text" data-format="text" alignment="left" chart-attr-
name="hoverXLabel">Test X</cell>

                <!--  hoverBehavior chart attribute  -->
                <cell type="chart_text" data-format="text" alignment="left" chart-attr-
name="hoverBehavior">dim</cell>

                <!--  hideShowBehavior chart attribute  -->
                <cell type="chart_text" data-format="text" alignment="left" chart-attr-
name="hideShowBehavior">withoutRescale</cell>

                <!--  dataTypeXAxisFormat chart attribute  -->
                <cell type="chart_text" data-format="text" alignment="left" chart-attr-
name="dataTypeXAxisFormat">integer</cell>

                <!--  dataTypeYAxisFormat chart attribute  -->
                <cell type="chart_text" data-format="text" alignment="left" chart-attr-
name="dataTypeYAxisFormat">percent</cell>
            </band>
        </bands>
    </frame>
<layout type="columner">
    <component frame-name="scatterFrame" column="1" visual-type="ojScatter" visual-
style="oj-sm-12" visual-title=""/>
</layout>
</report>
```

Combo Chart

JSON Data Attribute	XML declaration	Description / Notes
name	`<cell type="text" data-format="text" alignment="left" style="col_header_1" print-width="130" chart-attr-name="seriesName">TotalOperatingCosts</cell>`	"seriesName" is a constant attribute name. The value is derived based on the cell type definition.
items	`<cell type="expr" data-format="integer" alignment="right" style="header_1" suppress-formatting="true">DBGET("DailyOpsQuery","prepCostTotal")</cell>`	This attribute must directly follow the "seriesName" declaration. This is the chart data points/items values. The "suppress-formatting" property must be set to true. The value is derived based on the cell type definition.
type	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="chartType">line</cell>`	"chartType" is a constant attribute name. The cell type must be set to "chart_text". Valid values are line, bar, or area.
color	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="chartColor">dvtCategory5Color</cell>`	"chartColor" is a constant attribute name. The cell type must be set to "chart_text". Valid values are listed in this table.
markerDisplayed	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="markerDisplayed">on</cell>`	"markerDisplayed" is a constant attribute name. The cell type must be set to "chart_text". Valid values are on or off.
groupData	`<cell type="column" data-format="text" alignment="left" chart-attr-name="groupData">Group A</cell>`	"groupData" is a constant attribute name. The value is derived based on the cell type definition.
stackValue	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="stackValue">on</cell>`	"stackValue" is a constant attribute name. The cell type must be set to "chart_text". Valid values areon or off.

JSON Data Attribute	XML declaration	Description / Notes
dualY	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="dualY">on</cell>`	"dualY" is a constant attribute name. The cell type must be set to "chart_text". Valid values are on or off.
xAxisTitle	`<cell type="text" data-format="text" alignment="left" chart-attr-name="xAxisTitle">Y Axis Title</cell>`	"xAxisTitle" is a constant attribute name. The value is derived based on the cell type definition.
xAxisRefObjValue	`<cell type="text" data-format="text" alignment="left" chart-attr-name="xAxisRefObjValue">Group A</cell>`	"xAxisRefObjValue" is a constant attribute name. The value is derived based on the cell type definition.
xAxisRefObjLegendText	`<cell type="text" data-format="text" alignment="left" chart-attr-name="xAxisRefObjLegendText">X Axis Object Reference Text</cell>`	"xAxisRefObjLegendText" is a constant attribute name. The value is derived based on the cell type definition.
xAxisRefObjHoverDescValue	`<cell type="text" data-format="text" alignment="left" chart-attr-name="xAxisRefObjHoverDescValue">X Axis Sample Reference Line</cell>`	"xAxisRefObjHoverDescValue" is a constant attribute name. The value is derived based on the cell type definition.
yAxisTitle	`<cell type="text" data-format="text" alignment="left" chart-attr-name="yAxisTitle">Y Axis Title</cell>`	"yAxisTitle" is a constant attribute name. The value is derived based on the cell type definition.
yAxisRefObjValue	`<cell type="text" data-format="text" alignment="left" chart-attr-name="yAxisRefObjValue">25000</cell>`	"yAxisRefObjValue" is a constant attribute name. The value is derived based on the cell type definition.
yAxisRefObjLegendText	`<cell type="text" data-format="text" alignment="left" chart-attr-name="yAxisRefObjLegendText">Y Axis Object Reference Text</cell>`	"yAxisRefObjLegendText" is a constant attribute name. The value is derived based on the cell type definition.
yAxisRefObjHoverDescValue	`<cell type="text" data-format="text" alignment="left" chart-attr-name="yAxisRefObjHoverDescValue">Y Axis Sample Reference Line</cell>`	"yAxisRefObjHoverDescValue" is a constant attribute name. Value is derived based on the cell type definition.

JSON Data Attribute	XML declaration	Description / Notes
y2AxisTitle	`<cell type="text" data-format="text" alignment="left" chart-attr-name="y2AxisTitle">Y2 Axis Title</cell>`	"y2AxisTitle" is a constant attribute name. The value is derived based on the cell type definition.
hoverSeriesLabel	`<cell type="text" data-format="text" alignment="left" chart-attr-name="hoverSeriesLabel">Test Series</cell>`	"hoverSeriesLabel" is a constant attribute name. The value is derived based on the cell type definition.
hoverValueLabel	`<cell type="text" data-format="text" alignment="left" chart-attr-name="hoverValueLabel">Test Value</cell>`	"hoverValueLabel" is a constant attribute name. The value is derived based on the cell type definition.
hoverGroupLabel	`<cell type="text" data-format="text" alignment="left" chart-attr-name="hoverGroupLabel">Test Group</cell>`	"hoverGroupLabel" is a constant attribute name. The value is derived based on the cell type definition.
hoverValueY2Label	`<cell type="text" data-format="text" alignment="left" chart-attr-name="hoverValueY2Label">Test Value Y2</cell>`	"hoverValueY2Label" is a constant attribute name. The value is derived based on the cell type definition.
dataType	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="dataType">integer</cell>`	"dataType" is a constant attribute name. This applies data formatting to "items" values. The cell type must be set to "chart_text". The valid value is the same as those that can be defined in cell data-format.
decimalPlaces	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="decimalPlaces">2</cell>`	"decimalPlaces" is a constant attribute name. This set number of decimal places to "items" values. The cell type must be set to "chart_text". The Value is an integer.

JSON Data Attribute	XML declaration	Description / Notes
dataTypeY2	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="dataTypeY2">integer</cell>`	“dataType” is a constant attribute name. This applies data formatting to `"items"` values in the Y-2 axis. The cell type must be set to `"chart_text"`. The valid value is the same as those that can be defined in cell data-format.
decimalPlacesY2	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="decimalPlacesY2">2</cell>`	`"decimalPlacesY2"` is a constant attribute name. This set number of decimal places to `"items"` values in the Y-2 axis. The cell type must be set to `"chart_text"`. The value is an integer.
dataTypeSeriesFormat	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="dataTypeSeriesFormat">percent</cell>`	`"dataTypeSeriesFormat"` is a constant attribute name. This applies data formatting to `"seriesName"` values. The cell type must be set to `"chart_text"`. The valid value is the same as those that can be defined in cell data-format.
dataTypeGroupFormat	`<cell type="chart_text" data-format="text" alignment="left" chart-attr-name="dataTypeGroupFormat">currency</cell>`	`"dataTypeGroupFormat"` is a constant attribute name. This applies data formatting to `"groupData"` values. The cell type must be set to `"chart_text"`. The valid value is the same as those that can be defined in cell data-format.

Table 4-1 Valid Color Values

Color Value	Sample
`dvtCategory1Color`	#238090

Table 4-1 (Cont.) Valid Color Values

Color Value	Sample
dvtCategory1ColorAlt	#27707D
dvtCategory2Color	#7BBF73
dvtCategory2ColorAlt	#6DA365
dvtCategory3Color	#F9D571
dvtCategory3ColorAlt	#D2B464
dvtCategory4Color	#DF7326
dvtCategory4ColorAlt	#BD6628
dvtCategory5Color	#8C698F
dvtCategory5ColorAlt	#7B5E7C
dvtCategory6Color	#91D7D9
dvtCategory6ColorAlt	#7FB6B7
dvtCategory7Color	#F4B962
dvtCategory7ColorAlt	#CE9E58
dvtCategory8Color	#DF7C7C
dvtCategory8ColorAlt	#BD6D6D
dvtCategory9Color	#85B5E5
dvtCategory9ColorAlt	#759BC1
dvtCategory10Color	#75C572
dvtCategory10ColorAlt	#69A865
dvtCategory11Color	#AF6476
dvtCategory11ColorAlt	#975A68

Table 4-1 (Cont.) Valid Color Values

Color Value	Sample
dvtCategory12Color	#F9F195
dvtCategory12ColorAlt	#D2CB81

Example of XML Report Definition to Generate Combo Chart JSON Data

```
<report>
        <frame type="table" query-name="MajorGroupsQuery" name="comboFrame">
            <bands>

                <band type="table-body">
                    <cell type="text" data-format="text" alignment="left" print-
width="140" indent-level="3" chart-attr-name="seriesName">Series1</cell>
                    <cell type="column" data-format="currency" alignment="right"
suppress-formatting="true" print-width="70">salesTotal</cell>
        <cell type="chart_text" data-format="text" alignment="left" chart-attr-
name="chartType">bar</cell>

        <cell type="text" data-format="text" alignment="left" print-width="140" indent-
level="3" chart-attr-name="seriesName">Series2</cell>
                    <cell type="column" data-format="currency" alignment="right"
suppress-formatting="true" print-width="70">salesTotal</cell>
        <cell type="chart_text" data-format="text" alignment="left" chart-attr-
name="chartType">line</cell>
        <cell type="chart_text" data-format="text" alignment="left" chart-attr-
name="chartColor">dvtCategory5Color</cell>
        <cell type="chart_text" data-format="text" alignment="left" chart-attr-
name="markerDisplayed">on</cell>

        <cell type="text" data-format="text" alignment="left" print-width="140" indent-
level="3" chart-attr-name="seriesName">Series3</cell>
        <cell type="column" data-format="currency" alignment="right" suppress-
formatting="true" print-width="70">salesTotal</cell>
        <cell type="chart_text" data-format="text" alignment="left" chart-attr-
name="chartType">area</cell> <!-- combo chart type attribute -->

        <cell type="text" data-format="text" alignment="left" print-width="140" indent-
level="3" chart-attr-name="seriesName">Series4</cell>
                    <cell type="column" data-format="currency" alignment="right"
suppress-formatting="true" print-width="70">salesTotal</cell>
        <cell type="chart_text" data-format="text" alignment="left" chart-attr-
name="chartType">bar</cell>
        <cell type="chart_text" data-format="text" alignment="left" chart-attr-
name="chartColor">dvtCategory6Color</cell>
        <cell type="chart_text" data-format="text" alignment="left" chart-attr-
name="assignedToY2">on</cell>

        <cell type="column" data-format="text" alignment="left" chart-attr-
name="groupData">majorGroupNameMaster</cell>
                </band>

        <band type="table-column-header">
        <!--  xAxisTitle chart attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="xAxisTitle">X Axis Title</cell>
```

```
        <!--  xAxisRefObjValue chart attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="xAxisRefObjValue">Group A</cell>
        <!--  xAxisRefObjLegendText chart attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="xAxisRefObjLegendText">X Axis Object Reference Text</cell>
        <!--  xAxisRefObjHoverDescValue chart attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="xAxisRefObjHoverDescValue">X Axis Sample Reference Line</cell>
        <!--  yAxisTitle chart attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="yAxisTitle">Y Axis Title</cell>
        <!--  yAxisRefObjValue chart attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="yAxisRefObjValue">Group B</cell>
        <!--  yAxisRefObjLegendText chart attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="yAxisRefObjLegendText">Y Axis Object Reference Text</cell>
        <!--  yAxisRefObjHoverDescValue chart attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="yAxisRefObjHoverDescValue">Y Axis Sample Reference Line</cell>
        <!--  y2AxisTitle chart attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="y2AxisTitle">Y2 Axis Title</cell>
        <!-- hover series label attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="hoverSeriesLabel">Test Series</cell>
        <!-- hover value label attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="hoverValueLabel">Test Value</cell>
        <!-- hover group label attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="hoverGroupLabel">Test Group</cell>
        <!-- hover value Y2 label attribute  -->
                    <cell type="text" data-format="text" alignment="left" chart-attr-
name="hoverValueY2Label">Test Value Y2</cell>
        <!--  dualY chart attribute  -->
                    <cell type="chart_text" data-format="text" alignment="left" chart-
attr-name="dualY">off</cell>
        <!-- stackValue chart attribute  -->
                    <cell type="chart_text" data-format="text" alignment="left" chart-
attr-name="stackValue">off</cell>
        <!-- extra cell to match count in table-body -->
        <cell type="chart_text" data-format="text" alignment="left"></cell>
        <cell type="chart_text" data-format="text" alignment="left"></
cell>
              </band>

          </bands>
     </frame>

     <layout type="columner">
          <component frame-name="comboFrame" column="1" visual-type="ojCombo" visual-
style="oj-sm-12" visual-title=""/>
     </layout>

</report>
```

Table

To add a table, set the `frame` element's `type` attribute to `table` as shown in the following example.

Example of XML Report Definition to Generate Table Type JSON Data

```
    <report>
         ...
        <frame type="table" query-name="MainQuery" name="MainFrame">
            <groups>
                <group name="grpAllDiscounts">
                    <group-item name="organizationID" data-type="integer"></group-item>
                </group>
                <detail>
                    <detail-item name="discountName" data-type="text"></detail-item>
                    <detail-item name="discountTotal" data-type="number"></detail-item>
                    <detail-item name="discountCount" data-type="number"></detail-item>
                    <detail-item name="AvgDiscount" data-type="number"></detail-item>
                    <sort-rules>
                        <sort-rule name="s1">
                            <sort-field name="discountName" sort-type="ascending"></sort-
field>
                        </sort-rule>
                        <sort-rule name="s3">
                            <sort-field name="discountTotal" sort-type="ascending"></
sort-field>
                            <sort-field name="discountName" sort-type="ascending"></sort-
field>
                        </sort-rule>
                        <sort-rule name="s4">
                            <sort-field name="discountCount" sort-type="descending"></
sort-field>
                            <sort-field name="discountName" sort-type="ascending"></sort-
field>
                        </sort-rule>
                        <sort-rule name="s5">
                            <sort-field name="AvgDiscount" sort-type="descending"></sort-
field>
                            <sort-field name="discountName" sort-type="ascending"></sort-
field>
                        </sort-rule>
                    </sort-rules>
                </detail>
            </groups>
            <sort-actions default="a3">
                <sort-action name="a1">
                    <sort-action-rule group-name="detail" sort-rule="s1"></sort-action-
rule>
                </sort-action>
                <sort-action name="a3">
                    <sort-action-rule group-name="detail" sort-rule="s3"></sort-action-
rule>
                </sort-action>
                <sort-action name="a4">
                    <sort-action-rule group-name="detail" sort-rule="s4"></sort-action-
rule>
                </sort-action>
                <sort-action name="a5">
                    <sort-action-rule group-name="detail" sort-rule="s5"></sort-action-
```

```
rule>
                </sort-action>
            </sort-actions>
            <bands>
                <band type="table-column-header">
                   <cell type="text" data-format="text" alignment="left"
style="header_1" sort-action="a1">DiscountName</cell>
                    <cell type="text" data-format="text" alignment="left"
style="header_1">DiscountPercentage</cell>
                     <cell type="text" data-format="text" alignment="right"
style="header_1" sort-action="a4">DiscountCount</cell>
                     <cell type="text" data-format="text" alignment="right"
style="header_1" sort-action="a5">AverageAmount</cell>
                </band>
                <band type="table-body">
                  <cell type="column" data-format="text" alignment="left" print-
width="130" indent-level="6" drilldown="DrillAllChecks">discountName</cell>
                   <cell type="column" data-format="percent" alignment="right" suppress-
zeroes="true" print-width="30" decimal-places="0" name="DiscPercent">posPercent</cell>
                   <cell type="column" data-format="integer" alignment="right" print-
width="80">discountCount</cell>
                   <cell type="expr" data-format="currency" alignment="right" print-
width="80" name="AvgDiscount">discountTotal/discountCount</cell>
                </band>
            </bands>
        </frame>

        <layout type="columner">
            <component frame-name="MainFrame" column="1" visual-type="ojTable" visual-
style="oj-sm-12" visual-title=""/>
        </layout>

    </report>
```