Report Definition Language Specification

THIRD VERSION
JULY 2008

Contents

About this Document

This document specifies the structure and semantics of Report Definition Language (RDL), an XML schema for representing reports.

Audience

This specification assumes:

- Working knowledge of XML.
- General knowledge of database concepts and query languages.

Introduction

In today's database reporting market, most vendor applications use a proprietary format for representing the definition of a report. In addition, vendors that provide a report execution environment usually only support their own design tools. For customers, this means that reports cannot be easily moved between different reporting implementations and that there are few options for choosing new tools that work with their existing execution environments.

Goals of Report Definition Language

The goal of Report Definition Language (RDL) is to promote the interoperability of commercial reporting products by defining a common schema that allows interchange of report definitions. An important aspect to understand is that RDL is a schema definition, not a programmatic interface or protocol like HTTP or ODBC. RDL does not specify how report definitions are passed between applications or how reports are processed. Also, RDL is meant to be fully encapsulated; meaning that successfully interpreting an RDL document should not require any understanding of the source application.
RDL is designed to be output format neutral. This means that reports defined using RDL should be able to be output to a variety of formats including Web and print-ready formats or data-focused formats like XML. It is expected that the in process of generating different output formats, products may represent RDL constructs slightly differently or ignore certain constructs completely. For example, a product generating a textual format may choose to ignore images in the report.

What is a Report?

A report is a combination of three kinds of information:

- **Data** or information about how to obtain the data (queries) as well as the structure of the data.
- **Layout** or formatting information that describes how the data is presented.
- **Properties** of the report, such as author, parameters, images within the report, and so on

Report Definition Overview Diagrams

This section contains diagrams that illustrate the schema of the Report Definition Language.

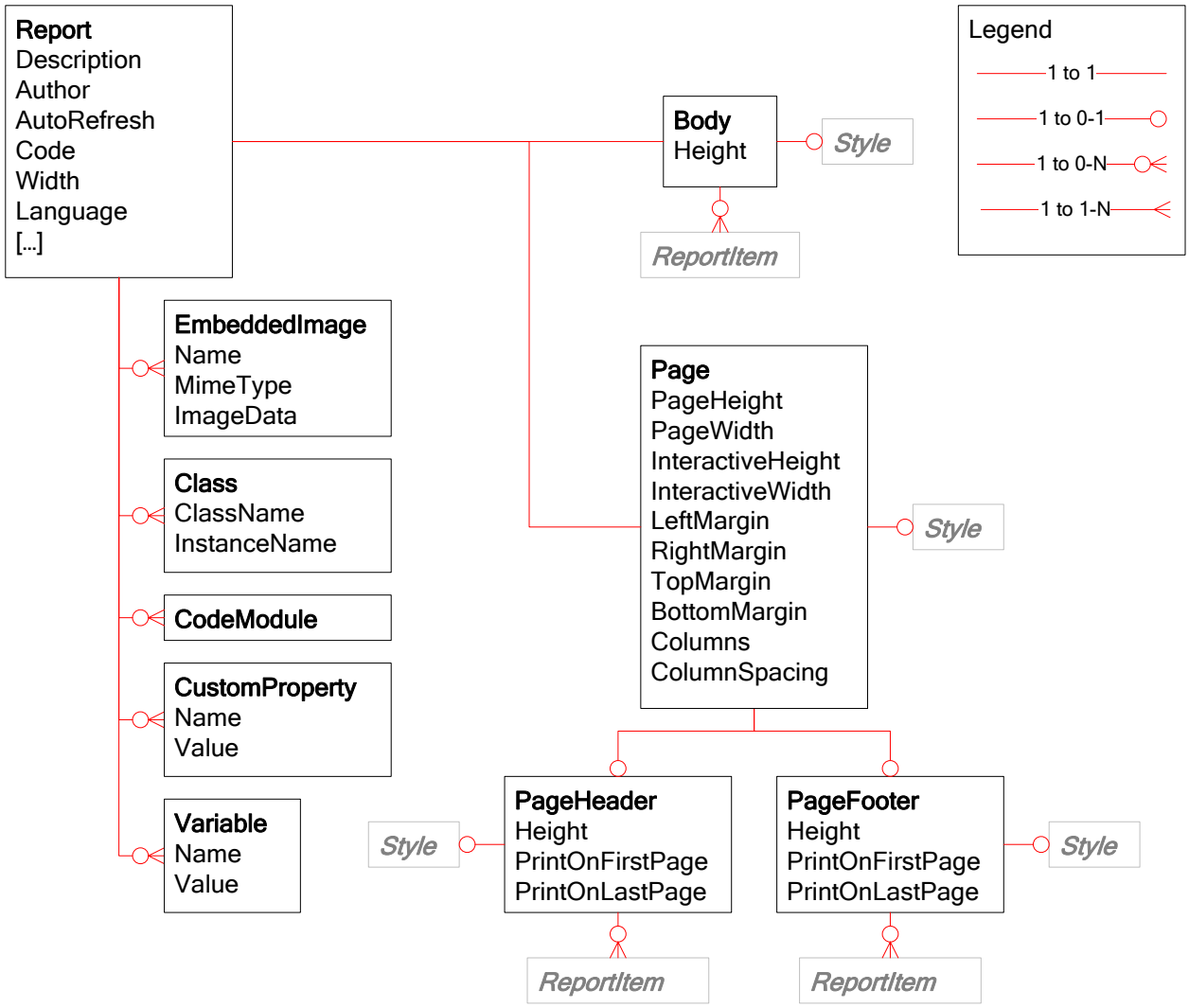

Figure 1 – Report Layout

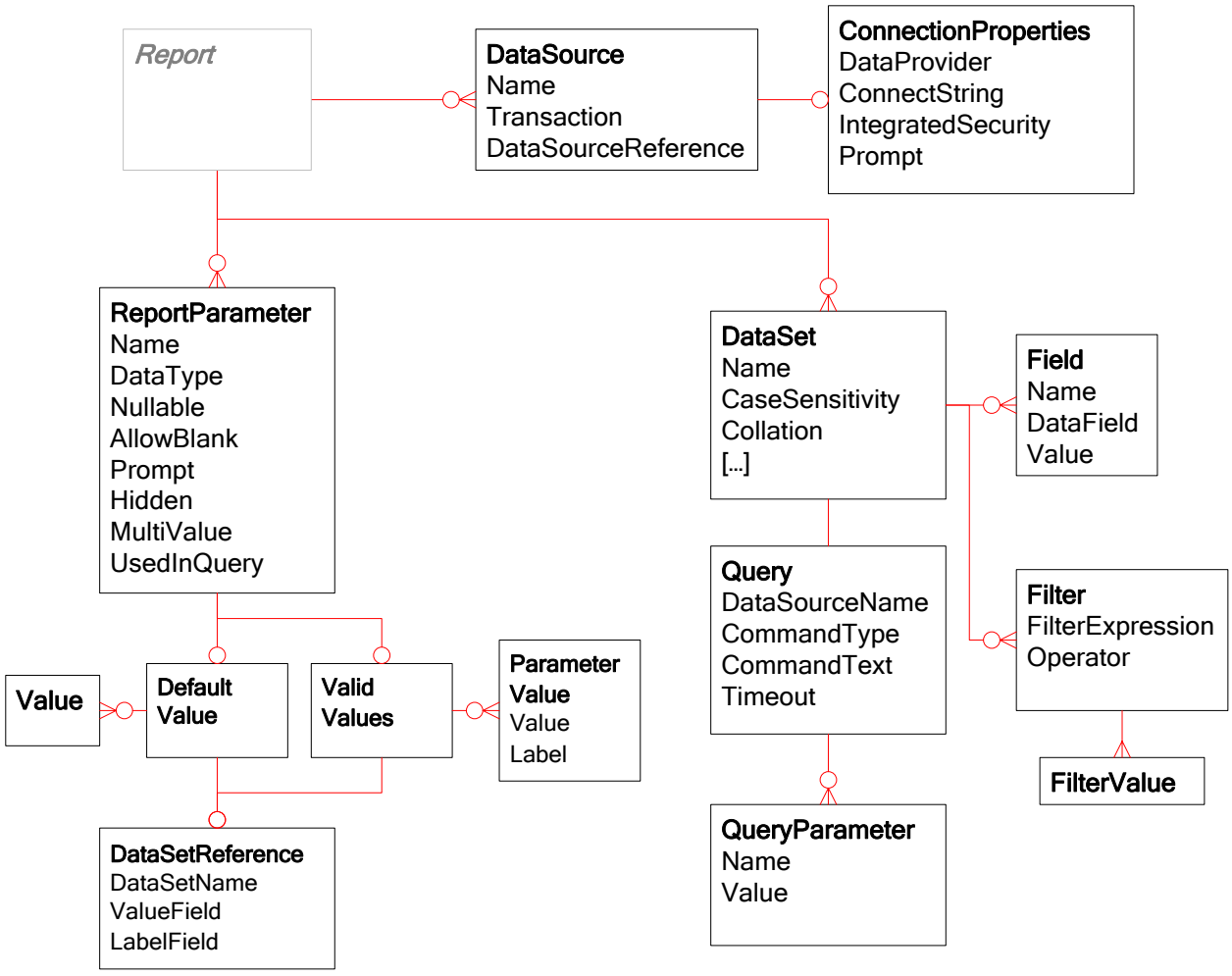

Figure 2 – Report Data

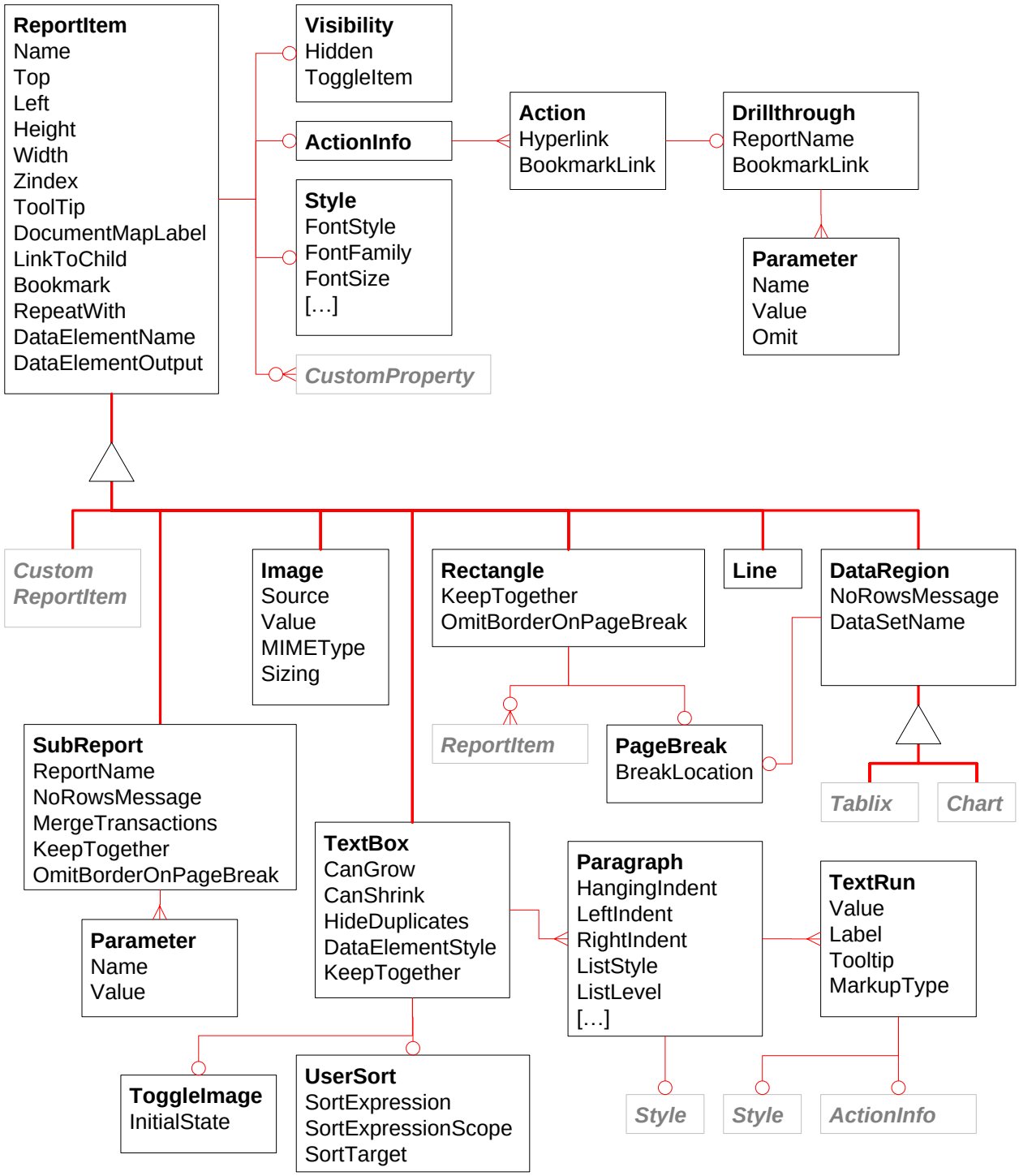

Figure 3 – Report Items

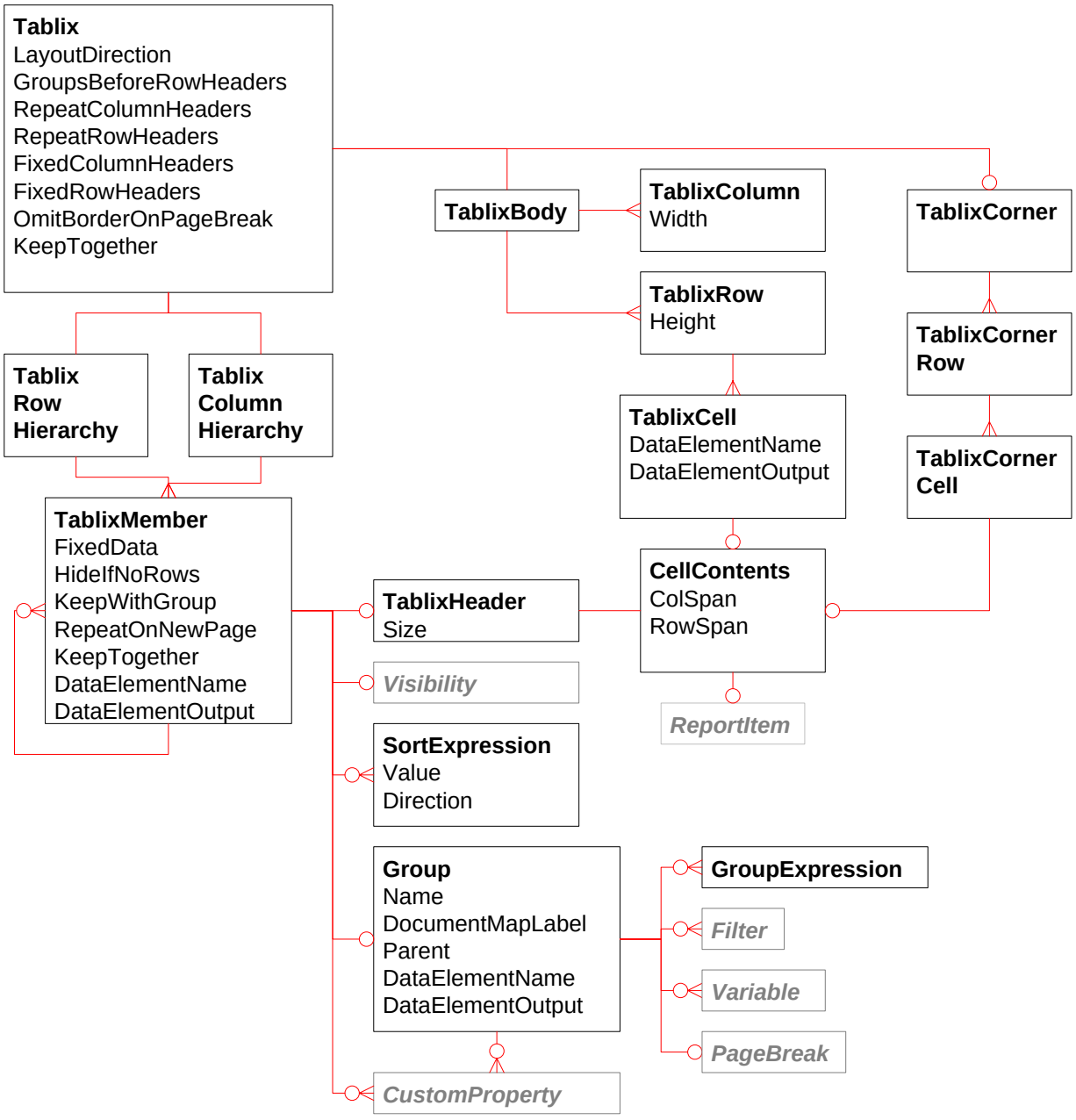

Figure 4 – Tablix

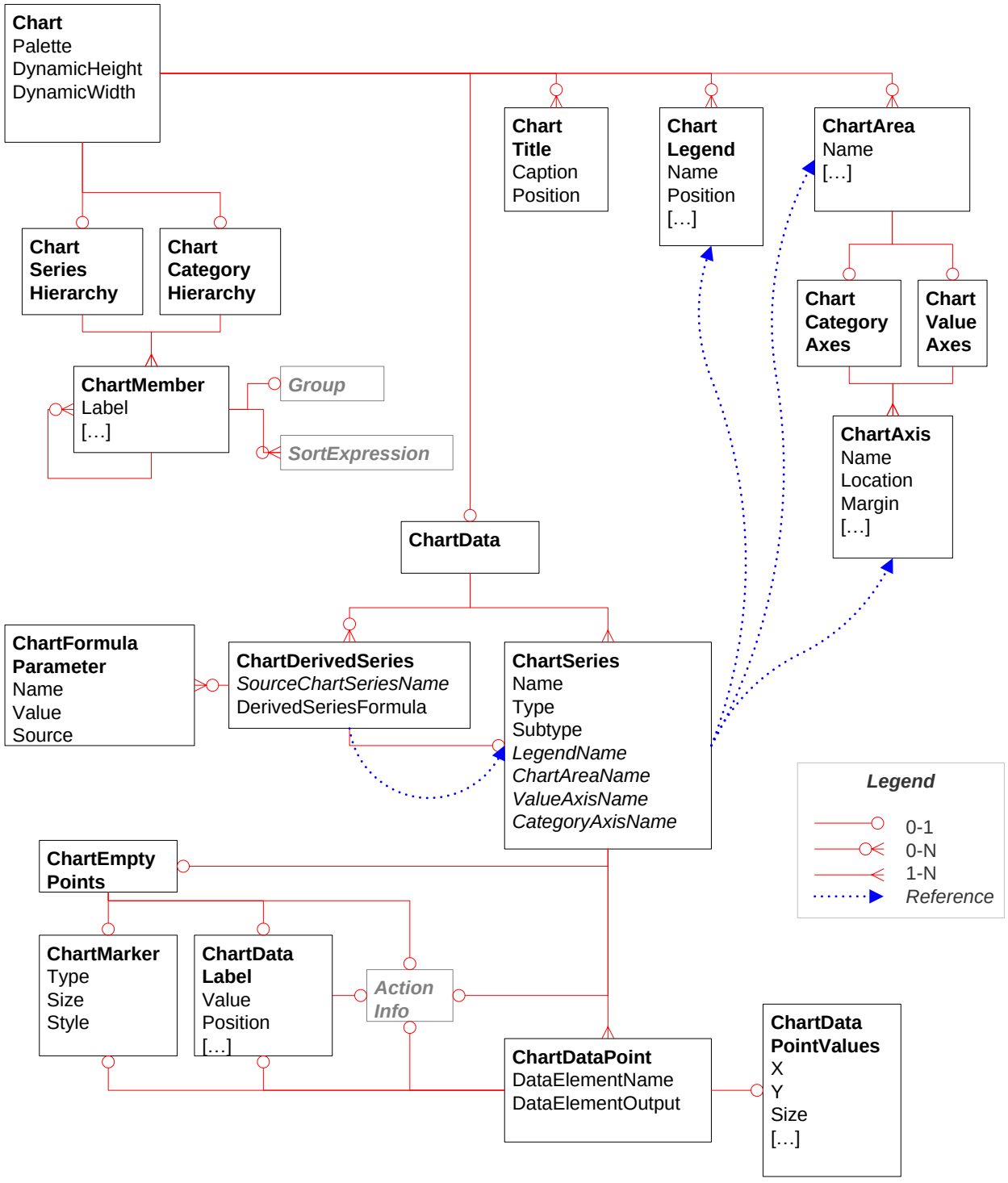

Figure 5 – Chart

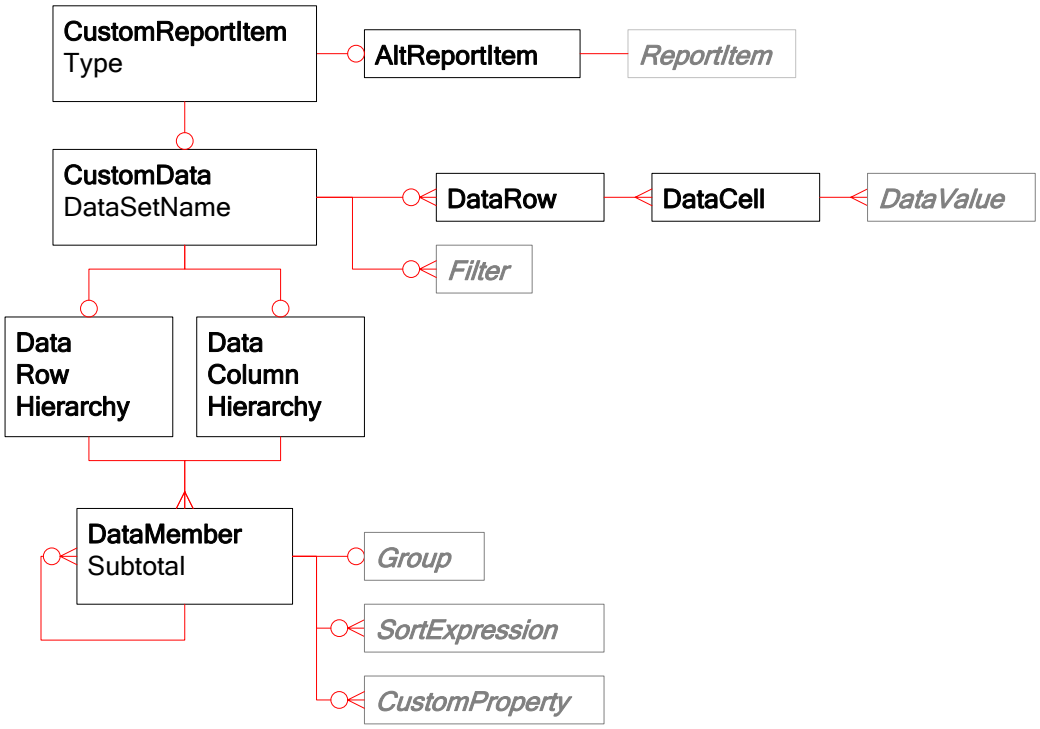

Figure 6 – Custom Report Item

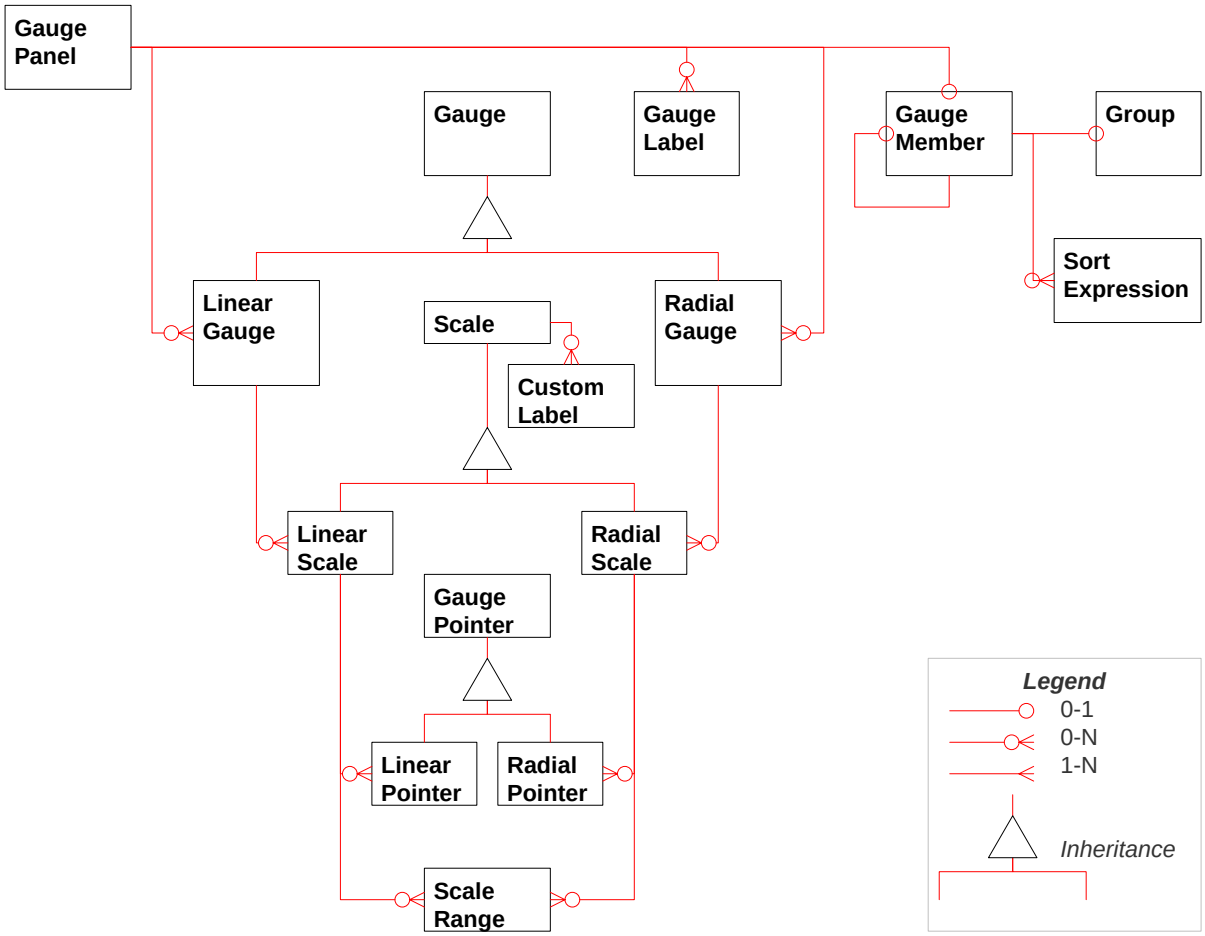

Figure 7 – Gauge Panel

Report Definition XML

This section defines the XML elements contained in RDL.

XML Namespace and Versioning

The namespace URI for RDL is

`http://schemas.microsoft.com/sqlserver/reporting/yyyy/mm/reportdefinition`

The date component (yyyy/mm) indicates the date of the release of that version of RDL.

The standard file name extension for RDL files is RDL. The MIMEType to use for RDL files is text/xml.

Extending RDL

RDL is an open schema. Application authors may extend/annotate RDL with their own attributes and subelements (in their own namespace). Note, however, that tools using RDL are not required to preserve unrecognized elements when loading and persisting. For properties that must be preserved independent of tool, the CustomProperties element should be used instead.

Element Definition Conventions

The following convention is used for types of attributes/elements in the element definitions in this document:

- Name – An attribute or subelement with a string text value that uniquely identifies the object within its element type.[1] This is an attribute if the name of the property is name and a subelement otherwise. Must be a case-sensitive CLS-compliant identifier[2]
- Element – A subelement (structure to be defined elsewhere in the document)
- String – A subelement or attribute with a string text value.
- Integer – A subelement with an integer (int32) value (without a thousands separators).
- Boolean – A subelement with true/false as the value of the element
  Unless otherwise specified, the value of an omitted optional Boolean element is taken to be false.
- Float – A subelement with a float value (without a thousands separators and "." used as the optional decimal separator)
- Size – A subelement with a size value. A size value is a floating-point number (without a thousands separators and "." used as the optional decimal separator) followed by an optional space and an absolute physical units designator (cm, mm, in, pt, pc). For more information about the supported length units, see CSS Length Units. Unless otherwise specified on the property, the maximum size is 160 in and the minimum size is 0[3]. Negative (absolute) sizes are only allowed where explicitly stated in the spec.
- Date – A subelement with a fully specified date or datetime value in ISO 8601 date format[4]: YYYY-MM-DD[THH:MM[:SS[.S]]]
- Color – A subelement with a color value[5]. A color value is a either a color name[6] or a hex HTML RGB color string of the form #HHHHHH or ARGB color string of the form #HHHHHHHH. Note: Color properties ignore the Alpha channel unless explicitly documented to support transparency.
- Expression – See the Expressions section later in this article. Expected return type is listed below each expression. Note: A return value of null on an optional element should behave as if the element were omitted.

---

[1] All ReportItems elements are considered to be in the same namespace

[2] See http://www.unicode.org/unicode/reports/tr15/tr15-18.html (Annex 7). Note: Non-normalized comparison is used for equality checks. Identifiers are limited to 256 characters.

[3] Objects in the report with height or width of 0 should be present in the target rendering. This means, for example, that an empty textbox of height 0 and width 1in with a top or bottom border will render as a horizontal line of length 1 in. Note that initial size of 0 is not a special case for growth behavior (such as CanGrow and containers expanding to accommodate contents) or clipping behavior (such as textboxes without CanGrow).

[4] See http://www.w3.org/TR/NOTE-datetime

[5] See http://msdn.microsoft.com/workshop/author/dhtml/reference/colors/colors.asp

[6] User Defined System Colors are not supported

- URL – A subelement with a string value that is a valid non-relative URL.
- Enum – A subelement with a string text value that must be among the designated list of values.
- Language – A Subelement with a text value of XML type language that contains a language code[7] such as "en-us" for US English. The value must be either a specific language in the .NET Framework or a neutral language for which a default specific language is defined in the .NET Framework.
- Nullable elements – If an element is specified as nullable, it may optionally have the attribute xsi:nil="true" instead of a value. This indicates the value of the element is explicitly null (instead of merely unspecified).

Whitespace is not trimmed from values in RDL. Subelements are order-independent. This includes items in collection elements unless the collection is explicitly defined as an ordered list.

[7] An ISO standard language abbreviation. See http://www.w3.org/TR/REC-html40/struct/dirlang.html#langcodes and http://msdn2.microsoft.com/en-us/library//system.globalization.cultureinfo.aspx

Element Definitions

Report

The Report element contains property, data and layout information about the report.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Description	0-1	String	Description of the report
Author	0-1	String	Author of the report
AutoRefresh	0-1	Integer	Rate, in seconds, at which the report page (when rendered as HTML) automatically refreshes. Must be nonnegative. If omitted or zero, the report page should not automatically refresh. Max: 2147483647
DataSources	0-1	Element	Describes the data sources from which data sets are taken for this report.
DataSets	0-1	Element	Describes the data that is displayed as part of the report.
Body	1	Element	Describes how the body of the report is structured and rendered.
ReportParameters	0-1	Element	Parameters for the report.
CustomProperties	0-1	Element	Custom information to be handed to the report rendering component
Code	0-1	String	Definitions for custom functions to be used in expressions in the report. Custom functions must be instance methods. If a function OnInit() is defined within Code, it is called during parameter, report and page header/footer initialization. The function must be defined as Protected and Overrides.
Width	1	Size	Width of the report.
Page	1	Element	Contains page layout information about the report.
EmbeddedImages	0-1	Element	Images embedded in the report

Language	0-1	Expression (Language)	The primary language of the text. Default is server language[8]. Used as the default for all language-dependent expressions in the report.
CodeModules	0-1	Element	Code modules to make available to the report for use in expressions.
Classes	0-1	Element	Classes to instantiate during report initialization.
Variables	0-1	Element	Variables defined for the report as a whole.
DeferVariableEvaluation	0-1	Boolean	Indicates that Variables throughout the report are not required to be pre-evaluated at the start of report processing and may be evaluated on-demand based on usage. Deferred variable evaluation can improve performance but should not be used if any variables are time-dependent.
ConsumeContainerWhitespace	0-1	Boolean	Indicates that all whitespace in containers (such as Body and Rectangle) should be consumed when contents grow rather than preserving the minimum whitespace between the contents and the container.
DataTransform	0-1	String	The location to a transformation to apply to a report data rendering. This can be a full folder path (for example, "/xsl/xfrm.xsl") or relative path (for example "xfrm.xsl"). Relative paths start in the same folder as the report.
DataSchema	0-1	String	The schema or namespace to use for a report data rendering.
DataElementName	0-1	String	Name of a top level element that represents the report data. Default: Report. Must be a CLS-compliant identifier.
DataElementStyle	0-1	Enum	Indicates whether leaf-level values (for example, text box values and chart data values) should render as elements or attributes.

Value	**Description**
Attribute	Default Render values as attributes
Element	Render values as elements

[8] Operating system language of the server on which the report is running

Page

The Page element contains page layout information for the report.

Attributes/Elements

Name	*Card*	*Type*	*Description*
PageHeader	0-1	Element	The header that is rendered at the top of each page of the report.
PageFooter	0-1	Element	The footer that is rendered at the bottom of each page of the report.
PageHeight	0-1	Size	Default height for rendering the report in a physical-page oriented renderer. Default: 11 in. Must be greater than 0 in.
PageWidth	0-1	Size	Default width for rendering the report in a physical-page oriented renderer. Default: 8.5 in. Must be greater than 0 in.
InteractiveHeight	0-1	Size	Default height for rendering the report when in an interactive renderer. There is no maximum size. A value of 0 (with any unit) indicates height should be unlimited. Defaults to PageHeight
InteractiveWidth	0-1	Size	Default height for rendering the report when in an interactive renderer. There is no maximum size. A value of 0 (with any unit) indicates width should be unlimited. Defaults to PageWidth.
LeftMargin	0-1	Size	Width of the left margin. Default: 0 in.
RightMargin	0-1	Size	Width of the right margin. Default: 0 in.
TopMargin	0-1	Size	Width of the top margin. Default: 0 in.
BottomMargin	0-1	Size	Width of the bottom margin. Default: 0 in.
Columns	0-1	Integer	Default number of columns for rendering the report Default: 1. Min: 1. Max: 1000
ColumnSpacing	0-1	Size	Spacing between each column in multi-column renderings. Default: 0.5 in.
Style	0-1	Element	Style information for the page.

ReportParameters

The ReportParameters element contains an ordered list of parameters for the report.

Attributes/Elements

Name	*Card*	*Type*	*Description*
ReportParameter	1-N	Element	Definition of a parameter for the report.

ReportParameter

The ReportParameter element contains information about a parameter to the report.
Within a ReportParameter, User.Language (instead of Report.Language) is used for all language-dependent expressions and operations[9].

Attributes/Elements

Name	*Card*	*Type*	*Description*
Name	1	Name	Name of the parameter. (This is the name used when expressions refer to the parameter.) Note: Parameter names need only be unique within the containing Parameters collection.
DataType	1	Enum	The data type of the parameter **Value** / **Description** Boolean – Parameter values are Boolean DateTime – Parameter values are DateTime Integer – Parameter values are Integer Float – Parameter values are Float String – Parameter values are String
Nullable	0-1	Boolean	Indicates the value for this parameter can be Null. Cannot be true if this is a multivalue parameter. (Not currently supported by any data extensions that support multivalue parameters.)
DefaultValue	0-1	Element	Default value to use for the parameter (if not provided by the user). If no value is provided as a part of the definition or by the user, the value is null. Required if there is no Prompt and either Nullable is False or a ValidValues list is provided that does not contain Null (an omitted Value).
AllowBlank	0-1	Boolean	Indicates the value for this parameter can be the empty string. Ignored if DataType is not String.
Prompt	0-1	Expression (String)	The user prompt to display when asking for parameter values

[9] If the report is used as a subreport, instead the language of the containing report is used.

<table>
<tr><td></td><td></td><td></td><td>If omitted, the user should not be prompted for or allowed to otherwise provide a value for this parameter.</td></tr>
<tr><td>Hidden</td><td>0-1</td><td>Boolean</td><td>Indicates the parameter should not be displayed to the user (however, it will still be available for programmatic use with subreports, drillthrough reports etc.)</td></tr>
<tr><td>ValidValues</td><td>0-1</td><td>Element</td><td>Possible values for the parameter (for the end-user UI).</td></tr>
<tr><td>MultiValue</td><td>0-1</td><td>Boolean</td><td>Indicates this is a multivalue parameter (a parameter that can take a set of values).
Multivalue parameters are accessed in expressions as zero-based arrays in the Value and Label properties (for example, Parameters!Cities.Value(0) and Parameters!Cities.Label(0)).
Ignored for Boolean parameters.</td></tr>
<tr><td>UsedInQuery</td><td>0-1</td><td>Enum</td><td>Indicates whether the parameter is used in a query in the report. This is necessary to determine if the queries must be rerun if the parameter changes.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Auto</td><td>Default
True if any query parameter value expression is a simple reference to this parameter[10] or there are any subreports in the report or there exists any query parameter value expression that is anything other than a constant or a simple parameter reference.</td></tr>
<tr><td>True</td><td>The parameter is used in a query in the report.</td></tr>
<tr><td>False</td><td>The parameter is not used in any query in the report.</td></tr>
</table>
</td></tr>
</table>

[10] A simple parameter reference is of the following form: =Parameters!*ParameterName*.Value

ValidValues

The possible values for this parameter, for populating UI selection lists for users to select a parameter value.

Attributes/Elements

Name	*Card*	*Type*	*Description*
DataSetReference	0-1	Element	The query to execute to obtain a list of possible values for the parameter.
ParameterValues	0-1	Element	Hardcoded values for the parameter.

Values must have one and only one of the following: DataSetReference or ParameterValues

DataSetReference

The query to execute to obtain a list of values or default values for a parameter.

Attributes/Elements

Name	*Card*	*Type*	*Description*
DataSetName	1	String	Name of the data set to use.
ValueField	1	String	Name of the field to use for the values/defaults for the parameter.
LabelField	0-1	String	Name of the field to use for the value to display to the user for the selection. If not supplied or the returned value is null, the value in the ValueField is used. Not used for DefaultValue.

ParameterValues

The ordered list of possible values for a parameter, used for populating UI selection lists for users to select a parameter value.

Attributes/Elements

Name	*Card*	*Type*	*Description*
ParameterValue	1-N	Element	Possible value for the parameter.

ParameterValue

A possible value for a parameter.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Value	0-1	Expression (Variant)	Possible value for the parameter. For Boolean parameters, use "true" and "false" For DateTime parameters, use ISO 8601 For Float parameters, use "." As the optional decimal separator. If the Value expression returns an array, each item in the array is treated as a single value. The items in the array must not be arrays.
Label	0-1	Expression (String[11])	Label for the value to display in the UI. If not supplied, the Value is used as the label (if Value is not supplied, Label is the empty string). If the Value expression returns an array, the Label expression must return an array with the same number of items. If the Value expression does not return an array, the Label expression must not return an array.

DefaultValue

The default value for this parameter.

Attributes/Elements

Name	*Card*	*Type*	*Description*
DataSetReference	0-1	Element	The query to execute to obtain the default value(s) for the parameter. For single-value parameters, the default is the first value of the ValueField. For multivalue parameters, the default is all values of the ValueField.
Values	0-1	Element	The default values for the parameter.

DefaultValue must have one and only one of the following: Values or DataSetReference
If one of the default values is not valid, the entire set of default values is treated as not valid.

[11] A Variant is allowed for this expression, but will be autocast to String before use.

Values

A set of values (used as defaults for a parameter).
For single-value parameters, only a single Value is allowed.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Value	1-N	Expression (Variant)	A value used as a default for a parameter. Cannot refer to Fields or ReportItems or any parameters that occur after the current parameter. If the Value expression returns an array, each item in the array is treated as a single value. Items in the array must not be arrays. For single-value parameters, only the first item in the array is used. This element is nullable.

Note: Only one default value is allowed for Boolean parameters.

DataSets

The DataSets element contains information about the sets of data to display as a part of the report.

Attributes/Elements

Name	*Card*	*Type*	*Description*
DataSet	1-N	Element	The sets of data for the report.

DataSet

The DataSet element contains information about a set of data to display as a part of the report.

Attributes/Elements

<table>
<tr><th>Name</th><th>Card</th><th>Type</th><th>Description</th></tr>
<tr><td>Name</td><td>1</td><td>Name</td><td>Name of the data set.
Cannot be the same name as any data region or group.</td></tr>
<tr><td>Fields</td><td>0-1</td><td>Element</td><td>The fields in the data set.</td></tr>
<tr><td>Query</td><td>1</td><td>Element</td><td>Information about the data source, including connection information, query, and so on, required to get the data from the data source.</td></tr>
<tr><td>CaseSensitivity</td><td>0-1</td><td>Enum</td><td>Indicates if the data is case sensitive.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Auto</td><td>Default
The case sensitivity setting should be autoderived by querying the data provider. Defaults to False if the data provider does not support that method.</td></tr>
<tr><td>True</td><td>Data in this data set is case sensitive.</td></tr>
<tr><td>False</td><td>Data in this data set is case insensitive.</td></tr>
</table></td></tr>
<tr><td>Collation</td><td>0-1</td><td>String</td><td>The locale to use for the collation sequence for sorting data.
Uses the standard Microsoft SQL Server collation names[12].
If no Collation is specified, the collation setting should be autoderived by querying the data provider.
Defaults to the collation corresponding to the report’s Language property if the data provider does not support that method or returns an unsupported or invalid value.</td></tr>
<tr><td>AccentSensitivity</td><td>0-1</td><td>Enum</td><td>Indicates if the data is accent sensitive.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Auto</td><td>Default
The accent sensitivity setting should be autoderived by querying the data provider. Defaults to False if the data provider does not support that method.</td></tr>
<tr><td>True</td><td>Data in this data set is accent sensitive.</td></tr>
<tr><td>False</td><td>Data in this data set is accent insensitive.</td></tr>
</table></td></tr>
</table>

[12] See http://msdn2.microsoft.com/en-us/library/ms184391(SQL.100).aspx

<table>
<tr><td>KanatypeSensitivity</td><td>0-1</td><td>Enum</td><td>Indicates if the data is kanatype sensitive.<table><tr><th>Value</th><th>Description</th></tr><tr><td>Auto</td><td>Default
The kanatype sensitivity setting should be autoderived by querying the data provider. Defaults to False if the data provider does not support that method.</td></tr><tr><td>True</td><td>Data in this data set is kanatype sensitive</td></tr><tr><td>False</td><td>Data in this data set is kanatype insensitive.</td></tr></table></td></tr>
<tr><td>WidthSensitivity</td><td>0-1</td><td>Enum</td><td>Indicates if the data is width sensitive.<table><tr><th>Value</th><th>Description</th></tr><tr><td>Auto</td><td>Default
The width sensitivity setting should be autoderived by querying the data provider. Defaults to False if the data provider does not support that method.</td></tr><tr><td>True</td><td>Data in this data set is width sensitive.</td></tr><tr><td>False</td><td>Data in this data set is width insensitive.</td></tr></table></td></tr>
<tr><td>Filters</td><td>0-1</td><td>Element</td><td>Filters to apply to each row of data in the data set.</td></tr>
<tr><td>InterpretSubtotalsAs Details</td><td>0-1</td><td>Enum</td><td>Indicates whether subtotal rows returned from a data provider that supports server subtotals should be interpreted as detail rows instead.<table><tr><th>Value</th><th>Description</th></tr><tr><td>Auto</td><td>Default
Subtotal rows will be treated as details if the report does not use the Aggregate() function to access any fields in this data set.</td></tr><tr><td>True</td><td>Subtotal rows should be interpreted as detail rows.</td></tr><tr><td>False</td><td>Subtotals rows are retrieved only via the Aggregate function.</td></tr></table></td></tr>
</table>

Fields

The Fields element defines the fields in the data model.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Field	1-N	Element	Field in the data model.

The data model maps to the fields in SQL and OLE-DB queries based on name. Each field in the data model corresponds to the field in the OLE-DB rowset of the same name.

Multi-dimensional data rowsets (OLE-DB for OLAP) also map to the data model based on name. Each level and measure in the multi-dimensional cube corresponds to a field in the data model.

Example

Consider the following MDX query:

```
SELECT CROSSJOIN([Time].[Quarter].members, Measures.[Store Sales]) ON COLUMNS,
CROSSJOIN([Store].[Store State].members, [Product].[Product Category].members) ON
ROWS
FROM [Sales]
```

To map this to a data set:

```
<Fields>
       <Field Name=”State”> <DataField>Store State</DataField></Field>
       <Field Name=”Category”> <DataField>Product Category</DataField></Field>
       <Field Name=”Quarter”> <DataField>Quarter</DataField></Field>
       <Field Name=”Sales”> <DataField>Store Sales</DataField></Field>
</Fields>
```

Field

The Field element contains information about a field in the data model of the report.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Name	1	Name	Name to use for the field in the report. Note: Field names need only be unique within the containing Fields collection.
DataField	0-1	String	Name of the field in the query. Note: Data field names do not have to be unique. Multiple fields can refer to the same data field name (although a warning will be generated during publishing).
Value	0-1	Expression (Variant)	An expression that evaluates to the value of this field. For example: =Fields!Price.Value+Fields!Tax.Value The expression cannot contain aggregates or references to report items. The Value element has an optional DataType attribute which specifies the data type of the value in the event it is a constant. It may be set to any RDL data type (see ReportParameter.DataType). If omitted, constant values are assumed to be strings.

Field must have exactly one of the following: DataField or Value.

Query

The Query element contains the description of the query to execute to retrieve the data for the report.

Attributes/Elements

Name	***Card***	***Type***	***Description***
DataSourceName	1	String	Name of the data source to execute the query against.
CommandType	0-1	Enum	Indicates what type of query is contained in the CommandText. **Value** / **Description** Text / Default. The CommandText contains a query command to execute. StoredProcedure / The CommandText contains the name of a stored procedure to execute. TableDirect / The CommandText contains the name of a table from which to retrieve rows.
CommandText	1	Expression (String)	The query to execute to obtain the data for the report.
QueryParameters	0-1	Element	A list of parameters that are passed to the data source as part of the query.
Timeout	0-1	Integer	Number of seconds to allow for the query to run before timing out. Must be nonnegative. If omitted or zero, the query should not time out. Max: 2147483647

DataSources

The DataSources element contains information about how to connect to the sources of data for the various DataSets.

Attributes/Elements

Name	***Card***	***Type***	***Description***
DataSource	1-N	Element	A source of data for the report.

DataSource

The DataSource element contains information about a data source.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Name	1	Name	The name of the data source. Must be unique in the report.
Transaction	0-1	Boolean	Indicates the data sets that use this data source should be executed in a single transaction[13].
ConnectionProperties	0-1	Element	Information about how to connect to the data source.
DataSourceReference	0-1	String	The full folder path (for example, "/salesreports/salesdatabase") or relative path (for example, "salesdatabase") to a data source on the same server. Relative paths start in the same folder as the report. The data source uses the connection properties from the DataSourceReference.

DataSource must have one and only one of the following: DataSourceReference or ConnectionProperties.

ConnectionProperties

The ConnectionProperties element contains information about how to connect to a data source.

Attributes/Elements

Name	*Card*	*Type*	*Description*
DataProvider	1	String	The type of the data source. (for example "SQL", "OLEDB", "OLEDB-MD") This is the name of a registered data provider.
ConnectString	1	Expression (String)	The connection string for the data source.
IntegratedSecurity	0-1	Boolean	Indicates that this data source should be connected to using integrated security.
Prompt	0-1	String	The prompt displayed to the user when prompting for database credentials for this data source.

[13] When a data set is used to populate a report parameter default value or valid values list, it is executed outside of the transaction.

QueryParameters

The QueryParameters element contains parameters that are passed to the data source as part of the query.

Attributes/Elements

Name	*Card*	*Type*	*Description*
QueryParameter	1-N	Element	A parameter to pass to the data source with the query.

QueryParameter

The QueryParameter element contains information about a parameter that is passed to the data source as part of the query.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Name	1	String	Name of the parameter
Value	1	Expression (Variant or Variant Array)	An expression that evaluates to the value to hand to the data source. The expression can refer to report parameters but cannot contain references to report elements, fields in the data model or aggregate functions. In the case of a parameter to a Values or DefaultValue query, the expression can only refer to report parameters that occur earlier in the parameters list. The value for this query parameter is then taken from the user selection for that earlier report parameter. The Value element has an optional DataType attribute which specifies the data type of the value in the event it is a constant. It may be set to any RDL data type (see ReportParameter.DataType). If omitted, constant values are assumed to be strings.

CodeModules

The CodeModules element contains the names of code modules to load for use in Code and expressions.

Attributes/Elements

Name	*Card*	*Type*	*Description*
CodeModule	1-N	String	Name of the code module to load.

Classes

The Classes element contains information about classes to instantiate during report initialization. These class instances can be used in expressions throughout the report.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Class	1-N	Element	The classes to instantiate.

Class

The Class element contains information about a class to instantiate during report initialization. This class instance can be used in expressions throughout the report.

Attributes/Elements

Name	*Card*	*Type*	*Description*
ClassName	1	String	The name of the class.
InstanceName	1	Name	The name of the member variable of Class to assign the class to. This member variable can be used in expressions throughout the report.

ReportElement

The virtual ReportElement element defines an element of a report. The ReportElement element itself is not used. Only the subtypes of ReportElement are used: Body, PageSection, ReportItem.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Style	0-1	Element	Style information for the report element.

Body

The Body element defines the visual elements of the body of the report, how the data is structured/grouped and binds the visual elements to the data for the report. It has the following properties in addition to what it inherits from ReportElement:

Attributes/Elements

Name	*Card*	*Type*	*Description*
ReportItems	0-1	Element	The region that contains the elements of the report body.
Height	1	Size	Height of the body.

PageSection

The virtual PageSection element defines the layout of report items to appear at the top or bottom of every page of the report. The PageSection element itself is not used. Only subtypes of PageSection are used: PageHeader, PageFooter. It has the following properties in addition to what it inherits from ReportElement:

Attributes/Elements

Name	*Card*	*Type*	*Description*
Height	1	Size	Height of the page section.
PrintOnFirstPage	0-1	Boolean	Indicates if the page section should be shown on the first page of the report. Not used in single-page reports if this is a PageFooter.
PrintOnLastPage	0-1	Boolean	Indicates if the page section should be shown on the last page of the report. Not used in single-page reports if this is a PageHeader.
ReportItems	0-1	Element	The region that contains the elements of the page section layout No data regions or subreports are allowed in the page section. All page breaks are ignored in the page section.

PageHeader

The PageHeader element defines the layout of report items to appear at the top of every page of the report. It has no properties beyond those it inherits from PageSection.

PageFooter

The PageFooter element defines the layout of report items to appear at the bottom of every page of the report. It has no properties beyond those it inherits from PageSection.

CustomProperties

The CustomProperties element allows report design tools to pass information to custom report renderers and custom report items.

Attributes/Elements

Name	*Card*	*Type*	*Description*
CustomProperty	1-N	Element	A custom property for this report, report item or member.

CustomProperty

The contents of CustomProperty are passed through to rendering and custom report item components.
Client applications using the CustomProperty element should add an application-specific namespace prefix their custom property names to reduce the possibility of name collisions when multiple applications are used for editing the same report definition; for example, "msd:FormattedValue" rather than "FormattedValue".

Attributes/Elements

Name	*Card*	*Type*	*Description*
Name	1	Expression (String)	Name of the property. Properties with null or duplicate names are not allowed.
Value	1	Expression (Variant)	Value of the property.

EmbeddedImages

The EmbeddedImages element is a collection of images embedded in the report.

Attributes/Elements

Name	*Card*	*Type*	*Description*
EmbeddedImage	1-N	Element	An image embedded in the report.

EmbeddedImage

The EmbeddedImage element is an image embedded in the report.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Name	1	Name	Name of the image.
MIMEType	1	String	The MIMEType for the image. Valid values are as follows: image/bmp, image/jpeg, image/gif, image/png, image/x-png
ImageData	1	String	Base-64 encoded image data.

Filters

The Filters element is a collection of filters to apply to a data set, data region or group.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Filter	1-N	Element	An ordered list of filters used to restrict the rows in a data set or data region or to restrict the group instances in a group. Filters are applied in sequence (this allows Top/Bottom filters to be applied to a data set that has already had some filters applied).

Filter

The Filter element describes a filter to apply to rows of data in a data set or data region or to apply to group instances.

Attributes/Elements

<table>
<tr><th>Name</th><th>Card</th><th>Type</th><th>Description</th></tr>
<tr><td>FilterExpression</td><td>1</td><td>Expression (Variant)</td><td>An expression that is evaluated for each instance within the group or each row of the data set or data region and compared (via the Operator) to the FilterValues. Failed comparisons result in the row/instance being filtered out of the data set, data region or group. See Filter Expression Restrictions later in this document.</td></tr>
<tr><td>Operator</td><td>1</td><td>Enum</td><td>The operator used to compare the FilterExpression and FilterValues.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Equal</td><td>Equality comparison.</td></tr>
<tr><td>Like</td><td>Like comparison.
Uses the same special characters as the Visual Basic LIKE operator (for example “?” to represent a single character and “*” to represent any series of characters)[14].</td></tr>
<tr><td>NotEqual</td><td>Inequality comparison.</td></tr>
<tr><td>GreaterThan</td><td>Inequality comparison.</td></tr>
<tr><td>GreaterThanOrEqual</td><td>Inequality comparison.</td></tr>
<tr><td>LessThan</td><td>Inequality comparison.</td></tr>
<tr><td>LessThanOrEqual</td><td>Inequality comparison.</td></tr>
<tr><td>TopN</td><td>Check if FilterExpression is in top N (as defined by the FilterValue) values.</td></tr>
<tr><td>BottomN</td><td>Check if FilterExpression is in top N (as defined by the FilterValue) values.</td></tr>
<tr><td>TopPercent</td><td>Check if FilterExpression is in top N percent (as defined by the FilterValue) values.</td></tr>
<tr><td>BottomPercent</td><td>Check if FilterExpression is in bottom N percent (as defined by the FilterValue) values.</td></tr>
<tr><td>In</td><td>Check if FilterExpression is</td></tr>
</table></td></tr>
</table>

[14] See http://msdn.microsoft.com/library/en-us/vblr7/html/vaoprlike.asp

			<table><tr><td></td><td>equal to any FilterValue.</td></tr><tr><td>Between</td><td>Check if FilterExpression is between the two FilterValues.</td></tr></table> Notes: Top and Bottom operators include ties in the resulting data. String comparisons are locale-dependent. Null equals Null. TopPercent and BottomPercent round up and down respectively, if the percentage would result in a partial item being included (for example Top 25% of 13 items is 4 items whereas Bottom 75% is 9 items).
FilterValues	1	Element	The values to compare to the FilterExpression[15]. For Equal, Like, NotEqual, GreaterThan, GreaterThanOrEqual, LessThan, LessThanOrEqual, TopN, BottomN, TopPercent and BottomPercent, there must be exactly one FilterValue. For TopN and BottomN, the FilterValue expression must evaluate to an integer. For TopPercent and BottomPercent, the FilterValue expression must evaluate to an integer or float. For Between, there must be exactly two FilterValue elements. For In, the FilterValues are treated as a set (if the FilterExpression value appears anywhere in the set of FilterValues, the instance is not filtered out.)

[15] For TopN, BottomN, TopPercent and BottomPercent, the FilterValue expression is evaluated only once: For the first row (after all earlier filters have been applied) in the case of data set or data region a filter, for the first group instance (after all earlier filters have been applied) in the case of a group filter.

FilterValues

The FilterValues element is a collection of values to compare to in a filter.

Attributes/Elements

Name	*Card*	*Type*	*Description*
FilterValue	1-N	Expression (Variant or Variant Array)	A value to use for comparison (via the Operation) to the value of the FilterExpression. See Filter Expression Restrictions later in this document. The FilterValue element has an optional DataType attribute which specifies the data type of the value in the event it is a constant. It may be set to any RDL data type (see ReportParameter.DataType). If omitted, constant values are assumed to be strings.

Note: Multivalue parameters are supported if the operator is In. They are treated as multiple FilterValues.
For example:

```
<FilterValues>
        <FilterValue>=Parameters!Cities</FilterValue>
</FilterValues>
```

is equivalent to:

```
<FilterValues>
        <FilterValue>=Parameters!Cities.Value[0]</FilterValue>
        <FilterValue>=Parameters!Cities.Value[1]</FilterValue>
        […]
</FilterValues>
```

Filter Expression Restrictions

Filter expressions/values cannot contain references to report items.
Data Set and Data Region filter expressions/values cannot contain aggregate functions.
Group filter expressions/values cannot contain RunningValue or RowNumber.
Group filter expressions/values cannot use the First or Last aggregate with anything other than the default (current) scope.
Failure when evaluating any filter expression or filter value causes the report to immediately return an error.

ReportItems

The ReportItems element is a collection of report items (used to define the contents of a region of a report).

Attributes/Elements

Name	*Card*	*Type*	*Description*
ReportItem	1-N	Element	An element of the report layout (for example Tablix, Textbox, Line).

ReportItem

A report item is one of the following types of objects: Line, Rectangle, Textbox, Image, Subreport, CustomReportItem or DataRegion. DataRegions are: Tablix and Chart.

The ReportItem element itself is not used. Instead, specific report item element is used wherever ReportItem is allowed.

Common ReportItem Attributes/Elements

The following attributes and elements are shared among all report item element types (in addition to what is inherited from ReportElement). Note, however, that not all of these attributes/elements are necessarily meaningful for all types of report items. Attributes/elements that do not apply are ignored.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Name	1	Name	Name of the report item.
ActionInfo	0-1	Element	Actions (for example, a hyperlink) associated with the ReportItem.
Top	0-1	Size	The distance of the item from the top of the containing object. Defaults to 0 if omitted.
Left	0-1	Size	The distance of the item from the left of the containing object. Defaults to 0 if omitted.
Height	0-1	Size	Height of the item. Negative sizes allowed only for lines (The height/width gives the offset of the endpoint of the line from the start point) Defaults to the height of the containing object minus Top if omitted[16].
Width	0-1	Size	Width of the item. Negative sizes allowed only for lines. Defaults to the width of the containing object minus Left if omitted.
ZIndex	0-1	Integer	Drawing order of the report item within the containing object. Items with lower indices are drawn first (appearing behind items with higher indices). Items with equal indices have an unspecified rendering order. Default: 0 Min: 0 Max: 2147483647
Visibility	0-1	Element	Indicates if the item should be hidden[17]

[16] For Tablix, the default Height and Width are instead derived from the sizes of the component parts (columns, rows, cells).

<table>
<tr><td>ToolTip</td><td>0-1</td><td>Expression (String)</td><td>A textual label for the report item. Used for such things as rendering TITLE and ALT attributes in HTML reports.</td></tr>
<tr><td>DocumentMap Label</td><td>0-1</td><td>Expression (String)</td><td>A label to identify an instance of a report item.within the client UI (to provide a user-friendly label for searching).
Hierarchical listing of report item and group labels within the UI (the Document Map) should reflect the object containment hierarchy in the report definition. Peer items should be listed in left-to-right top-to-bottom order.
If the expression returns null, no item is added to the Document Map.
Not used for report items in the page header or footer.</td></tr>
<tr><td>Bookmark</td><td>0-1</td><td>Expression (String)</td><td>A bookmark that can be linked to via a Bookmark action.</td></tr>
<tr><td>RepeatWith</td><td>0-1</td><td>String</td><td>The name of a data region that this report item should be repeated with if that data region spans multiple pages.
The data region must be in the same ReportItems collection as this ReportItem (Since data regions are not allowed in page headers/footers, this means RepeatWith will be unusable in page headers/footers).
Not allowed if this report item is a data region, subreport or rectangle that contains a data region or subreport.</td></tr>
<tr><td>CustomProperties</td><td>0-1</td><td>Element</td><td>Custom information to be handed to the report rendering component</td></tr>
<tr><td>DataElementName</td><td>0-1</td><td>String</td><td>The name to use for the data element/attribute for this report item.
Default: Name of the report item.
Must be a CLS-compliant identifier.</td></tr>
<tr><td>DataElementOutput</td><td>0-1</td><td>Enum</td><td>Indicates whether the item should appear in a data rendering.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Auto</td><td>Default
Will behave as NoOutput for any report item with Hidden set to True (not an expression) that does not have a ToggleItem, and for any report item in a static tablix member that cannot be toggled with Hidden set to non-expression True. Otherwise, acts</td></tr>
</table></td></tr>
</table>

				as NoOutput for Textboxes with constant TextRun values, as ContentsOnly for Rectangles and as Output for all other items.
			Output	Indicates the item should appear in the output.
			NoOutput	Indicates the item should not appear in the output.
			ContentsOnly	Indicates the item should not appear in the XML, but its contents should be rendered as if they were in this item's container. Only applies to Rectangles.

Attributes and elements specific to each ReportItem element type are described later in this document.

Which properties apply to which ReportItem types?

	Line	Rectangle	Textbox	Image	Subreport	Custom Report Item	Tablix	Chart
Name	X	X	X	X	X	X	X	X
Action			X	X				
Top	X	X	X	X	X	X	X	X
Left	X	X	X	X	X	X	X	X
Height	X	X	X	X	X	X	X	X
Width	X	X	X	X	X	X		X
ZIndex	X	X	X	X	X	X	X	X
Visibility	X	X	X	X	X	X	X	X
Tooltip		X	X	X	X		X	X
DocumentMapLabel	X	X	X	X	X		X	X
Bookmark	X	X	X	X	X		X	X
RepeatWith	X	X	X	X		X		
CustomProperties	X	X	X	X	X		X	X
DataElementName		X	X		X		X	X
DataElementOutput		X	X		X		X	X

ActionInfo

The ActionInfo element defines a list of actions and action style associated with a ReportItem.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Actions	1	Element	The actions for the report item.

Actions

The Actions element defines a list of actions associated with a ReportItem.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Action	1-N	Element	An action for the report item. [Restricted to one Action in SQL Server 2008]

Action

The Action element defines a hyperlink, bookmark link or drillthrough action associated with a ReportItem.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Hyperlink	0-1	Expression (URL)	An expression that evaluates to the URL of the hyperlink[18]
Drillthrough	0-1	Element	The drillthrough report that should be executed by clicking on the hyperlink
BookmarkLink	0-1	Expression (String)	An expression that evaluates to the ID of a bookmark in the report to go to when this report item is clicked on. (If no bookmark with this ID is found, the link will not be included in the report. If the bookmark is hidden, the link will go to the start of the page the bookmark is on. If multiple bookmarks with this ID are found, the link will go to the first bookmark.)

Action must have one and only one of the following: Hyperlink, BookmarkLink or Drillthrough.

Drillthrough

The Drillthrough element has the following attributes/elements:

Attributes/Elements

Name	*Card*	*Type*	*Description*
ReportName	1	Expression (String)	The full folder path (for example, "/salesreports/orderdetails"), relative path (for example, "../orderdetails") or URL (for example, "http://reportserver/reports/sales/orderdetails") of the drillthrough report. Relative paths start in the same folder as the report. Note: If the current report is being used as a subreport, the top-level report location is used as the base of the relative path.
Parameters	0-1	Element	Parameters to the drillthrough report[19].

[18] The Access IsHyperlink property of Textboxes will be supported via this more general mechanism. The Hyperlink property of the Textbox can be set to the same expression as the Value property of the Textbox.

[19] None of the report server system parameters (rc: and rs: parameters) are supported in Drillthrough

Visibility

The Visibility element indicates if the ReportItem should be shown in the rendered report. If no Visibility element is present, the item is unconditionally shown.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Hidden	0-1	Expression (Boolean)	Indicates if the item should be hidden at first.
ToggleItem	0-1	String	The name of the text box used to hide/unhide this report item. Clicking on an instance of the ToggleItem will toggle the hidden state of every corresponding instance of this item[20]. If the ToggleItem becomes hidden (because either the item or an ancestor is toggled or conditionally hidden), this item should become hidden.[21] Must be a text box in the same group scope as this item or in any containing (ancestor) group scope. If omitted, no item will toggle the hidden state of this item. Not allowed on and cannot refer to report items contained in a page header or footer. Cannot refer to a report item contained in the current report item unless current group scope has a Parent.

Note: A hidden report item (where the Hidden property is the constant True) that cannot be toggled should be treated as if it is not present, when rendering a report. This means the report layout does not change because the item is hidden (unlike hidden items that can toggle or are conditionally hidden, thereby shifting layout to make room/remove empty space).

[20] A hyperlink attached to the textbox will take precedence over that textbox being a toggle item. In this case, the only way to trigger the toggle is to click on the ToggleImage for the textbox.

[21] This cascading does not apply to tablix members if the toggle item becomes hidden as a result of a containing tablix member on the opposite tablix hierarchy becoming hidden

Using ToggleItem with a Recursive Hierarchy

If the ToggleItem refers to a text box contained by and in the same group scope as the item whose visibility is being toggled and that item is a group (or is directly contained in a group) which has a Parent element, the show/hide toggling behavior will reflect the recursive hierarchy. Specifically: Clicking on the text box in one instance of the group will toggle the visibility of items in child instances of the group (see Group.Parent).

Example:
For a recursive hierarchy table that contains an EmployeeID, EmployeeName and ManagerID, a report can be created with a tablix that contains only a static header and detail row:

Employee
Bill
  Jason
    Brian
    Albert
  Amir

To allow the rows to be shown/hidden by clicking on the manager's name, the Hidden element for the tablix's detail row would resemble this:

```
<Visibility>
        <Hidden>=iif(Fields!ManagerID is Nothing, false, true)</Hidden>
        <ToggleItem>NameTextBox</ToggleItem>
</Visibility>
```

The group for the tablix details would resemble this:

```
<Group Name="Table1_DetailsGroup">
        <GroupExpressions>
                <GroupExpression>=Fields!EmployeeID.Value</GroupExpression>
        </GroupExpressions>
        <Parent>=Fields!ManagerID.Value</Parent>
</Group>
```

Line

The Line element has no additional attributes/elements beyond what it inherits from ReportItem Negative heights/widths allow for lines that are drawn up and/or left from their origin. Although negative Height and Width are allowed, both Top+Height and Left+Width must be nonnegative valid sizes.

Rectangle

The Rectangle element has the following attributes/elements in addition to what it inherits from ReportItem:

Attributes/Elements

Name	*Card*	*Type*	*Description*
ReportItems	0-1	Element	Report items contained within the bounds of the rectangle
PageBreak	0-1	Element	Defines page break behavior for the rectangle.
KeepTogether	0-1	Boolean	Indicates all of the contents of the rectangle should be kept together on one page if possible[22].
OmitBorderOnPageBreak	0-1	Boolean	Indicates the borders should not appear at locations where the rectangle spans multiple pages. Also causes repeated background images to continue rather than restart after a page break.
LinkToChild	0-1	String	The name of a report item contained directly within this rectangle that is the target location for the Document Map label (if any). Ignored if DocumentMapLabel is not present.

Textbox

The Textbox element has the following attributes/elements in addition to what it inherits from ReportItem:

Attributes/Elements

Name	*Card*	*Type*	*Description*
CanGrow	0-1	Boolean	Indicates the Textbox height can increase to accommodate the contents.
CanShrink	0-1	Boolean	Indicates the Textbox height can decrease to match the contents.
HideDuplicates	0-1	String	Indicates the text should not be displayed when the value of the expression associated with the report item is the same as the preceding visible instance. The value of HideDuplicates is the name of a containing group (other than the current group) or data set over which to apply the hiding. Each time a new instance of that group is encountered, the first visible instance of this report item will not be hidden. Rows on a previous page are ignored for the purposes of hiding duplicates. If the text box is in a

[22] KeepTogether functions independently in both horizontal and vertical pagination.

			tablix cell, only the text will be omitted. The text box will remain to provide background and border for the cell. Outside of a tablix cell, the background and borders are omitted as well. Ignored unless the text box contains only one TextRun.
ToggleImage	0-1	Element	Indicates the initial state of a toggling image should one be displayed as a part of the text box.
UserSort	0-1	Element	Indicates an end-user sort control should be displayed as a part of this text box in the UI.
DataElementStyle	0-1	Enum	Indicates whether all TextRun values for this text box value should render as an element or attribute. **Value** / **Description** Auto / Default Use the setting on the Report element. Attribute / Render values as attributes. Element / Render values as elements.
KeepTogether	0-1	Boolean	Indicates all of the contents of the text box should be kept together on one page if possible[23].
Paragraphs	1	Element	Collection of Paragraph elements.

Paragraphs

The Paragraphs element is a collection of Paragraph elements.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Paragraph	1-N	Element	Represents a paragraph.

[23] In the event of a textbox spanning multiple pages (due to KeepTogether=False or the textbox being too large for a page) the textbox is split between text lines into multiple textboxes. Each individual line of text is always kept together.

Paragraph

The Paragraph element represents a paragraph of text in a Textbox and contains a collection of TextRun elements.

Attributes/Elements

<table>
<tr><th>Name</th><th>Card</th><th>Type</th><th>Description</th></tr>
<tr><td>TextRuns</td><td>1</td><td>Element</td><td>Collection of TextRun elements.</td></tr>
<tr><td>LeftIndent</td><td>0-1</td><td>Expression (Size)</td><td>Indentation from the left edge of the Textbox, less left padding.
Default: 0.</td></tr>
<tr><td>RightIndent</td><td>0-1</td><td>Expression (Size)</td><td>Indentation from the right edge of the Textbox, less right padding.
Default: 0.</td></tr>
<tr><td>HangingIndent</td><td>0-1</td><td>Expression (Size)</td><td>Indicates the first line indent or hanging line indent for the paragraph. Relative to left indent; can be negative. If positive, indents just the first line (first line indent). If negative, indents all lines but the first line (hanging indent).
Default: 0</td></tr>
<tr><td>Style</td><td>0-1</td><td>Element</td><td>Style properties for the paragraph.</td></tr>
<tr><td>ListStyle</td><td>0-1</td><td>Enum</td><td>Indicates whether this paragraph is part of a list, and identifies the numbering type.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>None</td><td>Default
Indicates that this is not a list paragraph and that there is no bullet/number for this paragraph.</td></tr>
<tr><td>Numbered</td><td>Indicates that this is a list paragraph with numbering</td></tr>
<tr><td>Bulleted</td><td>Indicates that this is a list paragraph with bullets.</td></tr>
</table></td></tr>
<tr><td>ListLevel</td><td>0-1</td><td>Integer</td><td>Indicates the numbering style and/or indentation level. Must be >= 0 and <= 9. For paragraphs with ListStyle=None, this property serves to indent the paragraph. When ListStyle is Bulleted or Numbered, it serves as indentation level and bullet/number style.
Default: 0.</td></tr>
<tr><td>SpaceBefore</td><td>0-1</td><td>Expression (Size)</td><td>Spacing before the paragraph. Cannot be negative.
Default: 0.</td></tr>
<tr><td>SpaceAfter</td><td>0-1</td><td>Expression (Size)</td><td>Spacing after the paragraph. Cannot be negative.
Default: 0</td></tr>
</table>

TextRuns

The TextRuns element is a collection of TextRun elements.

Attributes/Elements

Name	*Card*	*Type*	*Description*
TextRun	1-N	Element	Represents a TextRun.

TextRun

The TextRun element defines the value and formatting of a contiguous span of text.

Attributes/Elements

<table>
<tr><th>Name</th><th>Card</th><th>Type</th><th>Description</th></tr>
<tr><td>Label</td><td>0-1</td><td>String</td><td>Label for the placeholder for this TextRun. This name appears as the display placeholder in designer tools UI.</td></tr>
<tr><td>Value</td><td>1</td><td>Expression (Variant)</td><td>An expression, the value of which is displayed at runtime for the TextRun.

Optional attributes:
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>DataType</td><td>Specifies the data type of the value in the event it is a constant. It may be set to any RDL data type (see ReportParameter.DataType). If omitted, constant values are assumed to be strings.</td></tr>
<tr><td>EvaluationMode</td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Auto</td><td>Default
Evaluates as an expression if Value starts with =. Otherwise, treats the value as a constant.</td></tr>
<tr><td>Expression</td><td>Value is evaluated as an expression.</td></tr>
<tr><td>Constant</td><td>Value is treated as a constant.</td></tr>
</table>
</td></tr>
<tr><td>xml:space[24]</td><td>Indicates whether to preserve white space in the Value.</td></tr>
</table>
</td></tr>
<tr><td>Style</td><td>0-1</td><td>Element</td><td>Style properties for the TextRun.</td></tr>
<tr><td>ActionInfo</td><td>0-1</td><td>Element</td><td>Defines the actions for this TextRun. Actions on TextRuns are ignored if an action is defined on the parent Textbox (even if the Textbox action resolves to NULL).</td></tr>
</table>

[24] This is an attribute from the xml namespace: http://www.w3.org/TR/REC-xml/#sec-white-space

<table>
<tr><td>ToolTip</td><td>0-1</td><td>Expression (String)</td><td>A textual tooltip label for the TextRun.</td></tr>
<tr><td>MarkupType</td><td>0-1</td><td>Expression (Enum)</td><td>Indicates whether markup appearing in the Value should be processed.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>None</td><td>Default
No markup is processed. Any markup is assumed to be literal (part of the value).</td></tr>
<tr><td>HTML</td><td>HTML markup appearing in the Value is processed and displayed in supporting rendering extensions.</td></tr>
</table></td></tr>
</table>

ToggleImage

Indicates the initial state of a toggle image should such an image be displayed as a part of the text box. The image is always displayed if the text box is a toggle item for another report item. Whenever the text box/image is clicked on, the toggle image state flips and the image associated with the new state is displayed instead.

Attributes/Elements

Name	***Card***	***Type***	***Description***
InitialState	1	Expression (Boolean)	A Boolean expression, the value of which determines the initial state of the toggle image. True = “expanded” (that is, a minus sign). False = “collapsed” (that is, a plus sign).

UserSort

Indicates an end-user sort control should be displayed as a part of this text box in the UI. The control allows the user to select a sort direction (ascending, descending, none).

Attributes/Elements

Name	***Card***	***Type***	***Description***
SortExpression	1	Expression (Variant)	The expression on which to sort. Has the same restrictions as a Group Filter expression. Aggregates used in the SortExpression may only use scopes which equal or contain the SortExpressionScope. Aggregates without an explicit scope are not allowed in the SortExpression if no SortExpressionScope is specified.
SortExpression Scope	0-1	String	Name of the scope (data region or group) in which to evaluate the SortExpression. If omitted, the expression will be evaluated and the sort will be performed independently in each detail scope within the SortTarget. Must be a scope that is equal to or contained within the current scope. If the text box has no current scope (in other words, it is not contained in any data region), SortExpressionScope must be equal to or contained within the SortTarget. Cannot be a detail scope (that is, a group with no group expressions). The data set for the SortExpressionScope must be the same as the data set for the SortTarget. Sorting takes place within the group containing the SortExpressionScope. For example: In a tablix with a country group and a city group with UserSort on each header and SortExpressionScope of the corresponding group, the country sort will sort the country groups within the tablix and the city sort will sort the city groups within each country group (without rearranging the country groups).
SortTarget	0-1	String	Name of the data region, group or data set to apply the sort to. If omitted, the sort will apply to the instance of the current scope. Must be the current scope, an ancestor scope[25], or a peer scope which is a data region.

[25] Tablix groupings are only valid SortTargets from within tablix grouping scopes along the same tablix axis

Image

The Image element has the following attributes/elements in addition to what it inherits from ReportItem:

Attributes/Elements

<table>
<tr><th>Name</th><th>Card</th><th>Type</th><th>Description</th></tr>
<tr><td>Source</td><td>1</td><td>Enum</td><td>Identifies the source of the image.

<table><tr><th>Value</th><th>Description</th></tr><tr><td>External</td><td>The Value contains a constant or expression that evaluates to the location of the image. This can be a full folder path (for example, “/images/logo.gif”), relative path (for example, “logo.gif”) or URL (for example, “http://reportserver/images/logo.gif”). Relative paths start in the same folder as the report.</td></tr><tr><td>Embedded</td><td>The Value contains a constant or expression that evaluates to the name of an EmbeddedImage within the report.</td></tr><tr><td>Database</td><td>The Value contains an expression (typically a field in the database) that evaluates to the binary data for the image.</td></tr></table></td></tr>
<tr><td>Value</td><td>1</td><td>Expression</td><td>See Source. Expected data type is string or binary, depending on Source. If the Value is null, no image is displayed.</td></tr>
<tr><td>MIMEType</td><td>0-1</td><td>Expression (String)</td><td>An expression, the value of which is the MIMEType for the image.
Valid values are: image/bmp, image/jpeg, image/gif, image/png, image/x-png
Required if Source = Database. Ignored otherwise.</td></tr>
<tr><td>Sizing</td><td>0-1</td><td>Enum</td><td>Defines the behavior if the image does not fit in the specified size.

<table><tr><th>Value</th><th>Description</th></tr><tr><td>AutoSize</td><td>Default
The borders should grow/shrink to accommodate the image.</td></tr><tr><td>Fit</td><td>The image is resized to exactly match the height and width of the image element[26].</td></tr><tr><td>FitProportional</td><td>The image should be resized to fit, preserving aspect ratio.</td></tr></table></td></tr>
</table>

[26] Renderers unable to support FitProportional or Clip should render as Fit instead.

			Clip	The image should be clipped to fit.

Subreport

The Subreport element has the following attributes/elements in addition to what it inherits from ReportItem:

Attributes/Elements

Name	***Card***	***Type***	***Description***
ReportName	1	String	The full folder path (for example, "/salesreports/orderdetails") or relative path (for example, "orderdetails") to a subreport on the same server. Relative paths start in the same folder as the current report. Cannot be an empty string (ignoring whitespace).
Parameters	0-1	Element	Parameters to the subreport.
NoRowsMessage	0-1	Expression (String)	Message to display in the Subreport (instead of the region layout[27]) when no rows of data are available in any data set which is used in the body of the subreport. Note: Style information on the subreport applies to this text.
MergeTransactions	0-1	Boolean	Indicates that transactions in the subreport should be merged with transactions in the parent report (into a single transaction for the entire report) if the data sources use the same connection.
KeepTogether	0-1	Boolean	Indicates the entire subreport should be kept together on one page if possible.
OmitBorderOnPageBreak	0-1	Boolean	Indicates the borders should not appear at locations where the subreport spans multiple pages. Also causes repeated background images to continue rather than restart after a page break.

Failure to execute a subreport results in a text box containing the string "Error: Subreport could not be shown" replacing the subreport (Style information on the subreport applies to the text box).

[27] If the subreport is in a tablix cell and does not have a NoRowsMessage property, the contents of the subreport will be omitted but the subreport's border properties will still apply to the cell.

Subreports that are hidden (but cannot be made visible via a toggle item) are not executed. The following Report properties do not apply when a report is used as a subreport: Description, Author, AutoRefresh, Width, Page, DataTranslation, DataSchema.

Parameters

The Parameters element contains a list of parameters and their values for a subreport or drillthrough.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Parameter	1-N	Element	Definition of a parameter for the report.

Parameter

The Parameter element contains information about a parameter to a subreport or drillthrough.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Name	1	String	Name of the parameter
Value	1	Expression (Variant)	An expression that evaluates to the value to hand in for the parameter to the subreport or control. For Drillthrough in Chart, this is the name of a DataField from which to obtain the value rather than an expression.
Omit	0-1	Expression (Boolean)	Indicates the parameter should be skipped. Valid only for Drillthrough parameters.

DataRegion

A DataRegion element is one of the following element types:

Tablix, Chart

The following attributes and elements are shared among all types of DataRegion elements:

Attributes/Elements

Name	***Card***	***Type***	***Description***
NoRowsMessage	0-1	Expression (String)	Message to display in the DataRegion (instead of the region layout[28]) when no rows of data are available. Note: Style information on the data region applies to this text.
DataSetName	0-1	String	Indicates which data set to use for this data region. Mandatory for top level DataRegions (not contained within another DataRegion) if there is not exactly one data set in the report. If there is exactly one data set in the report, the data region uses that data set. (Note: If there are zero data sets in the report, data regions can not be used, as there is no valid DataSetName to use) Ignored for DataRegions that are not top level.
PageBreak	0-1	Element	Defines the page break behavior for the data region.
Filters	0-1	Element	Filters to apply to each row of data in the data region.
SortExpressions	0-1	Element	The expressions by which to sort the rows of data in the data region.

Attributes and elements specific to each DataRegion element type are described later in this document.

[28] If the data region is in a tablix cell and does not have a NoRowsMessage property, the contents of the data region will be omitted but the data region's background and border properties will still apply to the cell.

Group

The Group element defines the expressions to group the data by.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Name	1	Name	Name of the Group. No two group elements may have the same name. No group element may have the same name as a data set or a data region.
DocumentMapLabel	0-1	Expression (String)	A label to identify an instance of the group in the client UI (to provide a user-friendly label for searching). See ReportItem.Label.
GroupExpressions	0-1	Element	The expressions by which to group the data. If omitted, this is a detail group (that is, there is one instance of the group per detail row of data).
PageBreak	0-1	Element	Defines PageBreak behavior for this group.
Filters	0-1	Element	Filters to apply to each instance of the group.
Parent	0-1	Expression (Variant)	An expression that identifies the parent group in a recursive hierarchy. Only allowed if the group has exactly one group expression. Indicates the following: 1. Groups should be sorted according to the recursive hierarchy (Sort is still used to sort peer groups). 2. Labels (in the document map) should be placed/indented according to the recursive hierarchy. 3. Intra-group show/hide should toggle items according to the recursive hierarchy (see ToggleItem). If filters on the group eliminate a group instance's parent, it is instead treated as a child of the parent's parent. In the event of a loop, one of the parent-child relationships will be ignored.
DataElementName	0-1	String	The name to use for the data element for instances of this group. Default: Name of the group Must be a CLS-compliant identifier.
DataElementOutput	0-1	Enum	Indicates whether the instances of the group should appear in a data rendering. **Value** / **Description** Output — Default. Indicates the instances of the group should appear in the output. NoOutput — Indicates the instances of the group

			should not appear in the output.
Variables	0-1	Element	A set of variables to evaluate at the group level.

GroupExpressions

The GroupExpressions element defines an ordered list of expressions to group the data by.

Attributes/Elements

Name	***Card***	***Type***	***Description***
GroupExpression	1-N	Expression (Variant)	An ordered list of expressions to group the data by. The only aggregate function allowed in group expressions is RowNumber (RowNumber must use the immediately containing scope and cannot be used in a GroupExpression anywhere within a Tablix Cell). References to report items are not allowed.

PageBreak

The PageBreak element defines page break behavior for a group or report item.

Attributes/Elements

Name	***Card***	***Type***	***Description***
BreakLocation	1	Enum	Indicates where the page break should occur.

Value	**Description**
Start	There should be a page break before the report item or each instance of the group.
End	There should be a page break after the report item or each instance of the group.
StartAndEnd	There should be a page break both before and after the report item or each instance of the group.
Between	There should be a page break between each instance of the group (does not apply to report items).

Variables

The Variables element defines a set of named expressions to be evaluated within the group or report.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Variable	1-N	Element	A named expression to be evaluated for the group or report and made available in the Variables global collection.

Variable

The Variable element defines a named expression to be evaluated within the group or report.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Name	1	Name	Name of the variable to be used in expressions in the report.
Value	1	Expression (Variant)	Expression to evaluate globally for the report or for each group instance. Unlike expressions evaluated in visual elements of the report, each instance of this expression is calculated only once when the report is executed and never recalculated during subsequent renderings. This is necessary for time-dependent calculations.

SortExpressions

The SortExpressions element defines the expressions to sort the groups by.

Attributes/Elements

Name	*Card*	*Type*	*Description*
SortExpression	1-N	Element	The expressions to sort the groups by.[29] This is an ordered list.

[29] Sorting preserves the order of rows from the original data for SortExpressions with identical values.

SortExpression

The SortExpression element defines an expression to sort the groups by.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Value	1	Expression (Variant)	The value to sort the groups by. The functions RunningValue and RowNumber are not allowed in SortExpression. References to report items are not allowed.
Direction	0-1	Enum	Indicates the direction of the sort **Value** / **Description** Ascending / Default Sort ascending. Descending / Sort descending.

Chart

The Chart element defines a set of chart areas to be drawn as a single data visualization data region. The Chart is defined much like a Tablix, but instead of Columns, Rows, and Cells, the Chart has Categories, Series, and DataPoints. It has the following attributes and elements in addition to what it inherits from DataRegion:

Attributes/Elements

Name	*Card*	*Type*	*Description*
ChartSeriesHierarchy	1	Element	The hierarchy of series members for the chart.
ChartCategoryHierarchy	1	Element	The hierarchy of category members for the chart.
ChartData	0-1	Element	Defines the data values for the chart.
ChartAreas	0-1	Element	Defines the set of chart areas for the chart.
ChartLegends	0-1	Element	Defines the set of legends for the chart.
ChartTitles	0-1	Element	Defines the set of titles for the chart.
Palette	0-1	Expression (Enum)	Determines the color palette for the chart items. Values are: **Value** / **Description** Default / Default Use Default palette EarthTones / Use EarthTones palette Excel / Use Excel palette GrayScale / Use GrayScale palette Light / Use Light palette Pastel / Use Pastel palette

<table>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>SemiTransparent</td><td>Use SemiTransparent palette</td></tr>
<tr><td>Berry</td><td>Use Berry palette</td></tr>
<tr><td>Chocolate</td><td>Use Chocolate palette</td></tr>
<tr><td>Fire</td><td>Use Fire palette</td></tr>
<tr><td>SeaGreen</td><td>Use SeaGreen palette</td></tr>
<tr><td>BrightPastel</td><td>Use BrightPastel palette</td></tr>
<tr><td>Custom</td><td>Use Custom palette</td></tr>
</table>
</td></tr>
<tr><td>PaletteHatchBehavior</td><td>0-1</td><td>Expression (Enum)</td><td>Indicates whether hatching should be automatically applied to data points in the chart.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Default</td><td>Default
Treated as None.</td></tr>
<tr><td>None</td><td>No hatching will be added to the data points.</td></tr>
<tr><td>Always</td><td>Automatic hatching will be applied to all data points (unless BackgroundHatchType is specified as non-Default).</td></tr>
</table>
</td></tr>
<tr><td>DynamicHeight</td><td>0-1</td><td>Expression (Size)</td><td>The height to which the chart should grow/shrink. Height is used as the initial height for relative layout changes due to resizing.</td></tr>
<tr><td>DynamicWidth</td><td>0-1</td><td>Expression (Size)</td><td>The width to which the chart should grow/shrink. Width is used as the initial width for relative layout changes due to resizing.</td></tr>
<tr><td>ChartBorderSkin</td><td>0-1</td><td>Element</td><td>Defines a border skin for the chart.</td></tr>
<tr><td>ChartNoDataMessage</td><td>0-1</td><td>Element</td><td>Title to display if the chart contains no data.</td></tr>
</table>

ChartAreas

The ChartAreas element defines a set of chart areas for the chart.

Attributes/Elements

Name	***Card***	***Type***	***Description***
ChartArea	1-N	Element	An area for the chart.

ChartArea

The ChartArea element defines a chart to be drawn within a Chart data region.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Name	1	Name	Name of the chart area.

Hidden	0-1	Expression (Boolean)	Indicates the chart area should be hidden.
ChartCategoryAxes	0-1	Element	Defines the category axes.
ChartValueAxes	0-1	Element	Defines the value axes.
ChartThreeDProperties	0-1	Element	Properties for a 3D chart layout.
Style	0-1	Element	Defines style properties for the chart area. Each of the properties of type Color support transparency.
AlignOrientation	0-1	Expression (Enum)	Indicates in which directions the chart area should be aligned with the target chart area. **Value** / **Description** None — Default No alignment. Vertical — Vertical alignment. Horizontal — Horizontal alignment. All — Both vertical and horizontal alignment. Ignored if AlignWithChartArea is not set.
ChartAlignType	0-1	Element	Indicates which aspects of the chart area should be aligned with the target chart area. Ignored if AlignWithChartArea is not set.
AlignWithChartArea	0-1	String	Name of a chart area with which to align this chart area.
ChartElementPosition	0-1	Element	Defines a custom position for the chart area. If omitted, automatic positioning will be used.
ChartInnerPlotPosition	0-1	Element	Defines a custom position for the inner plot area. If omitted, automatic positioning will be used.
EquallySizedAxesFont	0-1	Expression (Boolean)	Indicates the same font size should be used for all axes (if the font size is automatic).

ChartElementPosition

The ChartElementPosition element defines the position in which to draw a chart element.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Top	0-1	Expression (Float)	The distance of the item from the top of the containing object, as a percentage of the container. Defaults to 0 if omitted.
Left	0-1	Expression (Float)	The distance of the item from the left of the containing object, as a percentage of the container. Defaults to 0 if omitted.
Height	0-1	Expression (Float)	Height of the item as a percentage of its containing object. Defaults to 100 minus Top if omitted.
Width	0-1	Expression	Width of the item as a percentage of its containing object.

		(Float)	Defaults to 100 minus Left if omitted.

ChartInnerPlotPosition

The ChartInnerPlotPosition element defines the position in which to draw the inner plot of a chart area. It has no attributes/elements beyond what it inherits from ChartElementPosition.

ChartAlignType

The ChartAlignType element defines which aspects of the chart area should be aligned with the target chart area.

Attributes/Elements

Name	***Card***	***Type***	***Description***
AxesView	0-1	Expression (Boolean)	Indicates the chart areas should align on axes views.
Cursor	0-1	Expression (Boolean)	Indicates the chart areas should align on cursors.
Position	0-1	Expression (Boolean)	Indicates the chart areas should align on chart area positions.
InnerPlotPosition	0-1	Expression (Boolean)	Indicates the chart areas should align on inner plot positions.

ChartBorderSkin

The ChartBorderSkin element defines the appearance of the border skin around the plot area.

Attributes/Elements

Name	***Card***	***Type***	***Description***
ChartBorderSkinType	0-1	Expression (Enum)	Border skin type for the chart None (Default) |

Value	**Description**
None	Default No border skin
Emboss	Use Emboss border skin
Raised	Use Raised border skin
Sunken	Use Sunken border skin
FrameThin1	Use FrameThin1 border skin
FrameThin2	Use FrameThin2 border skin
FrameThin3	Use FrameThin3 border skin
FrameThin4	Use FrameThin4 border skin
FrameThin5	Use FrameThin5 border skin
FrameThin6	Use FrameThin6 border skin
FrameTitle1	Use FrameTitle1 border skin
FrameTitle2	Use FrameTitle2 border skin
FrameTitle3	Use FrameTitle3 border skin

			FrameTitle4 — Use FrameTitle4 border skin FrameTitle5 — Use FrameTitle5 border skin FrameTitle6 — Use FrameTitle6 border skin FrameTitle7 — Use FrameTitle7 border skin FrameTitle8 — Use FrameTitle8 border skin
Style	0-1	Element	Style properties for the border skin. Each of the properties of type Color support transparency.

ChartHierarchy

The virtual ChartHierarchy element defines a hierarchy of members for a Chart.

Attributes/Elements

Name	*Card*	*Type*	*Description*
ChartMembers	1	Element	The list of members at the base of the hierarchy.

ChartSeriesHierarchy

The ChartSeriesHierarchy element has no additional attributes/elements beyond what it inherits from ChartHierarchy.

ChartCategoryHierarchy

The ChartCategoryHierarchy element has no additional attributes/elements beyond what it inherits from ChartHierarchy.

ChartMembers

The ChartMembers element defines a list of members for a Chart.

Attributes/Elements

Name	*Card*	*Type*	*Description*
ChartMember	1-N	Element	An ordered list of members for a Chart hierarchy.

ChartMember

The ChartMember element defines a category or series member for a Chart.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Group	0-1	Element	The expressions by which to group the data. If omitted, this is a static member (otherwise, this is a dynamic member). Not allowed if any ancestor member is a detail group. Page breaks in the group are not allowed..
SortExpressions	0-1	Element	The expressions by which to sort the member instances. Not allowed if Group is omitted.
ChartMembers	0-1	Element	Submembers contained within this member.
Label	1	Expression (Variant)	The label displayed on the legend (for series members and category members when ChartSeries.Type = Shape) or category axis (for category members).
CustomProperties	0-1	Element	Custom properties for the member.
DataElementName	0-1	String	The name to use for the data element for this member. Must be a CLS-compliant identifier. Default for dynamic members: [Group.Name] Collection Default for static members: [Label][30]
DataElementOutput	0-1	Enum	Indicates whether the member should appear in a data rendering.

Value	Description
Auto	Default Behaves as Output for dynamic members. Behaves as ContentsOnly for static members.
Output	Indicates the member should appear in the output.
NoOutput	Indicates the member should not appear in the output.

[30] Since Label is an expression, this is the one case where the DataElementName technically may vary per instance. In the event the Label property evaluates to a string which is not a CLS-compliant identifier, the value provided to the renderer will be Null.

ChartTitles

The Titles element defines a set of title areas for the chart.

Attributes/Elements

Name	*Card*	*Type*	*Description*
ChartTitle	1-N	Element	A title for the chart.

ChartTitle

The ChartTitle element defines a title for the chart.

Attributes/Elements

<table>
<tr><th>Name</th><th>Card</th><th>Type</th><th>Description</th></tr>
<tr><td>Name</td><td>1</td><td>Name</td><td>Name of the title.</td></tr>
<tr><td>Caption</td><td>1</td><td>Expression (String)</td><td>Caption of the title.</td></tr>
<tr><td>Hidden</td><td>0-1</td><td>Expression (Boolean)</td><td>Indicates the title should be hidden.</td></tr>
<tr><td>Style</td><td>0-1</td><td>Element</td><td>Defines style properties for the title.
Color, BackgroundColor and BackgroundGradientEndColor all support transparency.</td></tr>
<tr><td>Position</td><td>0-1</td><td>Expression (Enum)</td><td>The position of the title.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>TopCenter</td><td>Default
Position title at TopCenter</td></tr>
<tr><td>TopLeft</td><td>Position title at TopLeft</td></tr>
<tr><td>TopRight</td><td>Position title at TopRight</td></tr>
<tr><td>LeftTop</td><td>Position title at LeftTop</td></tr>
<tr><td>LeftCenter</td><td>Position title at LeftCenter</td></tr>
<tr><td>LeftBottom</td><td>Position title at LeftBottom</td></tr>
<tr><td>RightTop</td><td>Position title at RightTop</td></tr>
<tr><td>RightCenter</td><td>Position title at RightCenter</td></tr>
<tr><td>RightBottom</td><td>Position title at RightBottom</td></tr>
<tr><td>BottomRight</td><td>Position title at BottomRight</td></tr>
<tr><td>BottomCenter</td><td>Position title at BottomCenter</td></tr>
<tr><td>BottomLeft</td><td>Position title at BottomLeft</td></tr>
</table></td></tr>
<tr><td>DockToChartArea</td><td>0-1</td><td>String</td><td>Name of the chart area on which to draw the title. If omitted (or does not match any chart area name), the title is drawn relative to the chart rather than a specific chart area.</td></tr>
<tr><td>DockOutsideChartArea</td><td>0-1</td><td>Expression (Boolean)</td><td>Indicates the title should be docked outside the chart area rather than inside the chart area.
Ignored if DockToChartArea is not set.</td></tr>
<tr><td>DockOffset</td><td>0-1</td><td>Expression (Integer)</td><td>Offset from the dock location, as a percentage of the chart size.
Default: 0</td></tr>
<tr><td>ChartElementPosition</td><td>0-1</td><td>Element</td><td>Defines a custom position for the title.
If omitted, automatic positioning will be used.</td></tr>
</table>

<table>
<tr><td>ToolTip</td><td>0-1</td><td>Expression (String)</td><td>Tool tip to display for the title.</td></tr>
<tr><td>ActionInfo</td><td>0-1</td><td>Element</td><td>Actions for the title.</td></tr>
<tr><td>TextOrientation</td><td>0-1</td><td>Expression (Enum)</td><td>Indicates the orientation of the text.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Auto</td><td>Default
Indicates the orientation will be selected automatically based on context (for example, Rotated270 for titles docked on the left).</td></tr>
<tr><td>Horizontal</td><td>Horizontal text.</td></tr>
<tr><td>Rotated90</td><td>Vertical text – Rotated 90 degrees.</td></tr>
<tr><td>Rotated270</td><td>Vertical text – Rotated 270 degrees.</td></tr>
<tr><td>Stacked</td><td>Vertical text – No character rotation.</td></tr>
</table>
</td></tr>
</table>

ChartNoDataMessage

The ChartNoDataMessage element defines a title to display if the chart contains no data.
It has no attributes/elements in addition to what it inherits from ChartTitle.

ChartLegends

The ChartLegends element defines a set of legend areas for the chart.

Attributes/Elements

Name	***Card***	***Type***	***Description***
ChartLegend	1-N	Element	A legend for the chart.

ChartLegend

The ChartLegend element defines the properties that can be used to display instances of the series groups in a chart legend.

Attributes/Elements

<table>
<tr><th>Name</th><th>Card</th><th>Type</th><th>Description</th></tr>
<tr><td>Name</td><td>1</td><td>Name</td><td>Name of the legend.</td></tr>
<tr><td>Hidden</td><td>0-1</td><td>Expression (Boolean)</td><td>Indicates the legend is hidden.</td></tr>
<tr><td>Style</td><td>0-1</td><td>Element</td><td>Defines style properties for the legend.</td></tr>
<tr><td>Position</td><td>0-1</td><td>Expression (Enum)</td><td>The position of the legend.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>RightTop</td><td>Default
Position legend at RightTop</td></tr>
<tr><td>TopLeft</td><td>Position legend at TopLeft</td></tr>
<tr><td>TopCenter</td><td>Position legend at TopCenter</td></tr>
<tr><td>TopRight</td><td>Position legend at TopRight</td></tr>
<tr><td>LeftTop</td><td>Position legend at LeftTop</td></tr>
<tr><td>LeftCenter</td><td>Position legend at LeftCenter</td></tr>
<tr><td>LeftBottom</td><td>Position legend at LeftBottom</td></tr>
<tr><td>RightCenter</td><td>Position legend at RightCenter</td></tr>
<tr><td>RightBottom</td><td>Position legend at RightBottom</td></tr>
<tr><td>BottomRight</td><td>Position legend at BottomRight</td></tr>
<tr><td>BottomCenter</td><td>Position legend at BottomCenter</td></tr>
<tr><td>BottomLeft</td><td>Position legend at BottomLeft</td></tr>
</table></td></tr>
<tr><td>Layout</td><td>0-1</td><td>Expression (Enum)</td><td>The arrangement of labels within the legend.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>AutoTable</td><td>Default
Automatically arrange labels to fit</td></tr>
<tr><td>Column</td><td>Arrange labels in a column</td></tr>
<tr><td>Row</td><td>Arrange labels in a row</td></tr>
<tr><td>WideTable</td><td>Arrange labels in a wide table</td></tr>
<tr><td>TallTable</td><td>Arrange labels in a tall table</td></tr>
</table></td></tr>
<tr><td>DockToChartArea</td><td>0-1</td><td>String</td><td>Name of the chart area on which to draw the legend. If omitted, (or does not match any</td></tr>
</table>

			chart area name), the legend is drawn relative to the chart rather than a specific chart area.
DockOutsideChartArea	0-1	Expression (Boolean)	Indicates the title should be docked outside the chart area rather than inside the chart area. Ignored if DockToChartArea is not set.
ChartElementPosition	0-1	Element	Defines a custom position for the legend. If omitted, automatic positioning will be used.

<table>
<tr><td>ChartLegendTitle</td><td>0-1</td><td>Element</td><td>Title display in the legend.</td></tr>
<tr><td>AutoFitTextDisabled</td><td>0-1</td><td>Expression (Boolean)</td><td>Indicates text will not be autosized to fit in the legend area.</td></tr>
<tr><td>MinFontSize</td><td>0-1</td><td>Expression (Size)</td><td>Minimum size for autosized legend text
Default: 7pt.</td></tr>
<tr><td>HeaderSeparator</td><td>0-1</td><td>Expression (Enum)</td><td>Indicates what type of separator to use for the legend header.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>None</td><td>Default
No separator</td></tr>
<tr><td>Line</td><td>Separate with Line</td></tr>
<tr><td>ThickLine</td><td>Separate with ThickLine</td></tr>
<tr><td>DoubleLine</td><td>Separate with DoubleLine</td></tr>
<tr><td>DashLine</td><td>Separate with DashLine</td></tr>
<tr><td>DotLine</td><td>Separate with DotLine</td></tr>
<tr><td>GradientLine</td><td>Separate with GradientLine</td></tr>
<tr><td>ThickGradientLine</td><td>Separate with ThickGradientLine</td></tr>
</table></td></tr>
<tr><td>HeaderSeparatorColor</td><td>0-1</td><td>Expression (Color)</td><td>Indicates what color to use for the legend header separator.</td></tr>
<tr><td>ColumnSeparator</td><td>0-1</td><td>Expression (Enum)</td><td>Indicates what type of separator to use for the columns.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>None</td><td>Default
No separator</td></tr>
<tr><td>Line</td><td>Separate with Line</td></tr>
<tr><td>ThickLine</td><td>Separate with ThickLine</td></tr>
<tr><td>DoubleLine</td><td>Separate with DoubleLine</td></tr>
<tr><td>DashLine</td><td>Separate with DashLine</td></tr>
<tr><td>DotLine</td><td>Separate with DotLine</td></tr>
<tr><td>GradientLine</td><td>Separate with GradientLine</td></tr>
<tr><td>ThickGradientLine</td><td>Separate with ThickGradientLine</td></tr>
</table></td></tr>
<tr><td>ColumnSeparatorColor</td><td>0-1</td><td>Expression (Color)</td><td>Indicates what color to use for the column separator.</td></tr>
<tr><td>ColumnSpacing</td><td>0-1</td><td>Expression (Integer)</td><td>Spacing between columns as a percent of the font size.</td></tr>
</table>

			Default: 50
InterlacedRows	0-1	Expression (Boolean)	Indicates legend rows should use interlaced colors.
InterlacedRowsColor	0-1	Expression (Color)	The background color to use for interlaced legend rows. If omitted, the chart area background color will be used.
EquallySpacedItems	0-1	Expression (Boolean)	Indicated legend items should be equally spaced
Reversed	0-1	Expression (Enum)	Indicates the direction of the legend should be reversed. Auto (Default) | True | False

Value	Description
Auto	Default Indicates the direction should be autodetected based on the series types.
True	Reverse the order of items in the legend.
False	Standard legend item ordering.

MaxAutoSize	0-1	Expression (Integer)	Maximum size for the legend, as a percent of the chart size. Default: 50
TextWrapThreshold	0-1	Expression (Integer)	Number of characters after which to wrap the legend text Default: 25

ChartLegendTitle

The ChartLegendTitle element defines a title for a legend.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Caption	1	Expression (String)	Caption of the title.
TitleSeparator	0-1	Expression (Enum)	Indicates what type of separator to use for the legend title.

Value	Description
None	Default No separator
Line	Separate with Line
ThickLine	Separate with ThickLine
DoubleLine	Separate with DoubleLine
DashLine	Separate with DashLine
DotLine	Separate with DotLine
GradientLine	Separate with GradientLine

			ThickGradientLine: Separate with ThickGradientLine
Style	0-1	Element	Defines style properties for the title. BackgroundColor and Border.Color support transparency.

ChartCategoryAxes

The ChartCategoryAxes element defines the list of category (X) axes.

Attributes/Elements

Name	*Card*	*Type*	*Description*
ChartAxis	1-N	Element	The category axes for the chart area. Can contain at most one ChartAxis with Location=Default and at most one ChartAxis with Location=Opposite.

ChartValueAxes

The ChartValueAxes element defines the list of value (Y) axes.

Attributes/Elements

Name	*Card*	*Type*	*Description*
ChartAxis	1-N	Element	The value axes for the chart area. Can contain at most one ChartAxis with Location=Default and at most one ChartAxis with Location=Opposite.

ChartAxis

The Axis element defines properties for labels, titles and gridlines along an axis.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Name	1	Name	Name of the axis (used when there is more than one axis along a dimension).
Visible	0-1	Expression (Enum)	Whether the axis is displayed. **Value** / **Description** Auto: Default. Indicates the axis should be displayed if it is in use (for example, a series is plotted against it or it has a title). True: Display the axis. False: Hide the axis.
Style	0-1	Element	Defines text style properties for the axis labels and

<table>
<tr><td></td><td></td><td></td><td>line style properties for the axis line.</td></tr>
<tr><td>ChartAxisTitle</td><td>0-1</td><td>Element</td><td>Defines a title for the axis.</td></tr>
<tr><td>Margin</td><td>0-1</td><td>Expression (Enum)</td><td>Indicates whether an axis margin will be created. The size of the margin is automatically generated based on the Scale and the number of data points.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Auto</td><td>Default
Indicates the margins are included based on the series type/subtype.</td></tr>
<tr><td>True</td><td>The axis has a margin.</td></tr>
<tr><td>False</td><td>The axis has no margin.</td></tr>
</table>
.</td></tr>
</table>

<table>
<tr><td>Interval</td><td>0-1</td><td>Expression (Float)</td><td>Default interval between gridlines, tick marks and labels.
Default (0), means the axis is autodivided.</td></tr>
<tr><td>IntervalType</td><td>0-1</td><td>Expression (Enum)</td><td>Default units for the Interval
<table><tr><th>Value</th><th>Description</th></tr><tr><td>Auto</td><td>Default
Interval unit is autoderived based on the data plotted against the axis.</td></tr><tr><td>Number</td><td>Interval is numeric</td></tr><tr><td>Years</td><td>Interval is Years</td></tr><tr><td>Months</td><td>Interval is Months</td></tr><tr><td>Weeks</td><td>Interval is Weeks</td></tr><tr><td>Days</td><td>Interval is Days</td></tr><tr><td>Hours</td><td>Interval is Hours</td></tr><tr><td>Minutes</td><td>Interval is Minutes</td></tr><tr><td>Seconds</td><td>Interval is Seconds</td></tr><tr><td>Milliseconds</td><td>Interval is Milliseconds</td></tr></table></td></tr>
<tr><td>IntervalOffset</td><td>0-1</td><td>Expression (Float)</td><td>Default offset for the first tick mark from the axis min.
Default: 0</td></tr>
<tr><td>IntervalOffsetType</td><td>0-1</td><td>Expression (Enum)</td><td>Default units for the IntervalOffset
<table><tr><th>Value</th><th>Description</th></tr><tr><td>Auto</td><td>Default
IntervalOffset unit is autoderived based on the data plotted against the axis.</td></tr><tr><td>Number</td><td>IntervalOffset is numeric</td></tr><tr><td>Years</td><td>IntervalOffset is Years</td></tr><tr><td>Months</td><td>IntervalOffset is Months</td></tr><tr><td>Weeks</td><td>IntervalOffset is Weeks</td></tr><tr><td>Days</td><td>IntervalOffset is Days</td></tr><tr><td>Hours</td><td>IntervalOffset is Hours</td></tr><tr><td>Minutes</td><td>IntervalOffset is Minutes</td></tr><tr><td>Seconds</td><td>IntervalOffset is Seconds</td></tr><tr><td>Milliseconds</td><td>IntervalOffset is Milliseconds</td></tr></table></td></tr>
<tr><td>VariableAutoInterval</td><td>0-1</td><td>Expression (Boolean)</td><td>Indicates if an automatic interval is calculated, it should be based on available size. Otherwise, the interval will be calculated based only on the data range.</td></tr>
<tr><td>LabelInterval</td><td>0-1</td><td>Expression (Float)</td><td>Interval between labels.
Default (0) uses ChartAxis.Interval</td></tr>
<tr><td>LabelIntervalType</td><td>0-1</td><td>Expression (Enum)</td><td>Units for the LabelInterval.
Default (Default) | Auto | Number | Years | Months</td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td>| Weeks | Days | Hours | Minutes | Seconds | Milliseconds
Default uses ChartAxis.IntervalType.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Default</td><td>Default
Uses ChartAxis.IntervalType.</td></tr>
<tr><td>Auto</td><td>LabelInterval unit is autoderived based on the data plotted against the axis.</td></tr>
<tr><td>Number</td><td>LabelInterval is numeric</td></tr>
<tr><td>Years</td><td>LabelInterval is Years</td></tr>
<tr><td>Months</td><td>LabelInterval is Months</td></tr>
<tr><td>Weeks</td><td>LabelInterval is Weeks</td></tr>
<tr><td>Days</td><td>LabelInterval is Days</td></tr>
<tr><td>Hours</td><td>LabelInterval is Hours</td></tr>
<tr><td>Minutes</td><td>LabelInterval is Minutes</td></tr>
<tr><td>Seconds</td><td>LabelInterval is Seconds</td></tr>
<tr><td>Milliseconds</td><td>LabelInterval is Milliseconds</td></tr>
</table></td></tr>
<tr><td>LabelIntervalOffset</td><td>0-1</td><td>Expression (Float)</td><td>Offset for the first label from the axis min.
Default (0) uses ChartAxis.IntervalOffset</td></tr>
<tr><td>LabelIntervalOffsetType</td><td>0-1</td><td>Expression (Enum)</td><td>Units for the LabelIntervalOffset
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Default</td><td>Default
Uses ChartAxis.IntervalOffsetType.</td></tr>
<tr><td>Auto</td><td>LabelIntervalOffset unit is autoderived based on the data plotted against the axis.</td></tr>
<tr><td>Number</td><td>LabelIntervalOffset is numeric</td></tr>
<tr><td>Years</td><td>LabelIntervalOffset is Years</td></tr>
<tr><td>Months</td><td>LabelIntervalOffset is Months</td></tr>
<tr><td>Weeks</td><td>LabelIntervalOffset is Weeks</td></tr>
<tr><td>Days</td><td>LabelIntervalOffset is Days</td></tr>
<tr><td>Hours</td><td>LabelIntervalOffset is Hours</td></tr>
<tr><td>Minutes</td><td>LabelIntervalOffset is Minutes</td></tr>
<tr><td>Seconds</td><td>LabelIntervalOffset is Seconds</td></tr>
<tr><td>Milliseconds</td><td>LabelIntervalOffset is Milliseconds.</td></tr>
</table></td></tr>
<tr><td>ChartMajorGridLines</td><td>0-1</td><td>Element</td><td>Indicates major gridlines should be displayed for this axis.</td></tr>
<tr><td>ChartMinorGridLines</td><td>0-1</td><td>Element</td><td>Indicates minor gridlines should be displayed for this axis.</td></tr>
<tr><td>ChartMajorTickMarks</td><td>0-1</td><td>Element</td><td>Defines major tick marks for the axis.</td></tr>
<tr><td>ChartMinorTickM</td><td>0-1</td><td>Element</td><td>Defines minor tick marks for the axis.</td></tr>
</table>

<table>
<tr><td>arks</td><td></td><td></td><td></td></tr>
<tr><td>MarksAlwaysAtPlotEdge</td><td>0-1</td><td>Expression (Boolean)</td><td>Indicates the marks should stay with the edge of the plot area rather than moving with the axis.</td></tr>
<tr><td>Reverse</td><td>0-1</td><td>Expression (Boolean)</td><td>Indicates the axis should be plotted in the reverse direction.</td></tr>
<tr><td>CrossAt</td><td>0-1</td><td>Expression (Variant[31])</td><td>Value at which to cross the other axis.
If omitted (or error in expression), uses the default behavior for the chart type. Overrides Location.</td></tr>
<tr><td>Location</td><td>0-1</td><td>Expression (Enum)</td><td>Indicates whether the axis is drawn on the default side (for example, left for the value axis on a line chart) or on the opposite side.
Default (Default) | Opposite
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Default</td><td>Default
Draw the axis on the default side.</td></tr>
<tr><td>Opposite</td><td>Draw the axis on the opposite side.</td></tr>
</table></td></tr>
<tr><td>Interlaced</td><td>0-1</td><td>Expression (Boolean)</td><td>If this property is true then strip lines are drawn every other grid line interval for the axis. If grid lines are not used for the axis then the axis’ tick marks or labels are used to determine the interlaced strip lines interval.</td></tr>
<tr><td>InterlacedColor</td><td>0-1</td><td>Expression (Color)</td><td>Color of the interlaced strips.</td></tr>
<tr><td>ChartStripLines</td><td>0-1</td><td>Element</td><td>Custom strip lines for the axis.</td></tr>
<tr><td>Arrows</td><td>0-1</td><td>Expression (Enum)</td><td>Type of arrows for axis labels.
None (Default) | Triangle | SharpTriangle | Lines
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>None</td><td>Default
No arrows</td></tr>
<tr><td>Triangle</td><td>Triangle arrows</td></tr>
<tr><td>SharpTriangle</td><td>SharpTriangle arrows</td></tr>
<tr><td>Lines</td><td>Lines only</td></tr>
</table></td></tr>
<tr><td>Scalar</td><td>0-1</td><td>Boolean</td><td>Indicates the values along this axis are scalar values (that is, numeric or date) which should be displayed on the chart in a continuous axis. Scalar cannot be true if the axis has more than one group, if it has a static group or a group with more than one group expression. The type of scalar (date, integer, float) is derived from the first non-null value found. All values are converted to that type. If any non-scalar value is present, the axis will</td></tr>
</table>

[31] Only Integer, Float and DateTime values are allowed.

<table>
<tr><td></td><td></td><td></td><td>revert to non-scalar.
Treated as True if this is a ChartCategoryAxis and any ChartSeries plotted against this axis contains a ChartDataPoint with ChartDataPointValues.X defined.</td></tr>
<tr><td>Minimum</td><td>0-1</td><td>Expression (Variant)</td><td>Minimum value for the axis.
If omitted (or error in expression), the axis autoscales.</td></tr>
<tr><td>Maximum</td><td>0-1</td><td>Expression (Variant)</td><td>Maximum value for the axis.
If omitted (or error in expression), the axis autoscales.</td></tr>
<tr><td>LogScale</td><td>0-1</td><td>Expression (Boolean)</td><td>Indicates the axis is logarithmic.</td></tr>
<tr><td>LogBase</td><td>0-1</td><td>Expression (Float)</td><td>Base to use for logarithmic scale.
Default: 10</td></tr>
<tr><td>HideLabels</td><td>0-1</td><td>Expression (Boolean)</td><td>Indicates the axis labels are hidden.</td></tr>
<tr><td>Angle</td><td>0-1</td><td>Expression (Float)</td><td>The angle at which to display axis labels.
Must be an integer between –90 and 90Default: 0</td></tr>
<tr><td>PreventFontShrink</td><td>0-1</td><td>Expression (Boolean)</td><td>Indicates the axis label font size will not be reduced to fit within the chart.</td></tr>
<tr><td>PreventFontGrow</td><td>0-1</td><td>Expression (Boolean)</td><td>Indicates the axis label font size will not be increased to fit within the chart.</td></tr>
<tr><td>PreventLabelOffset</td><td>0-1</td><td>Boolean</td><td>Indicates the axis labels will not be staggered to fit within the chart.</td></tr>
<tr><td>PreventWordWrap</td><td>0-1</td><td>Boolean</td><td>Indicates the axis labels will not be word-wrapped to fit within the chart.</td></tr>
<tr><td>AllowLabelRotation</td><td>0-1</td><td>Expression (Enum)</td><td>Indicates the “step” by which axis labels can be incrementally rotated to fit within the chart.
Available values:

<table><tr><th>Value</th><th>Description</th></tr><tr><td>Rotate90</td><td>Default
Rotate in 90 degree increments</td></tr><tr><td>Rotate30</td><td>Rotate in 30 degree increments</td></tr><tr><td>Rotate45</td><td>Rotate in 45 degree increments</td></tr><tr><td>None</td><td>Rotation is not allowed</td></tr></table></td></tr>
<tr><td>IncludeZero</td><td>0-1</td><td>Expression (Boolean)</td><td>Indicates the axis should always include zero.
Ignored if Minimum is set.</td></tr>
<tr><td>LabelsAutoFitDisabled</td><td>0-1</td><td>Expression (Boolean)</td><td>Indicates axis labels should not be automatically adjusted to fit.</td></tr>
<tr><td>MinFontSize</td><td>0-1</td><td>Expression (Size)</td><td>Minimum font size when autofitting labels.</td></tr>
<tr><td>MaxFontSize</td><td>0-1</td><td>Expression (Size)</td><td>Maximum font size when autofitting labels.</td></tr>
<tr><td>OffsetLabels</td><td>0-1</td><td>Expression</td><td>Indicates the labels should be offset.</td></tr>
</table>

		(Boolean)	
HideEndLabels	0-1	Expression (Boolean)	Indicates labels should be hidden at axis ends.
ChartAxisScaleBreak	0-1	Element	Defines scale break behavior for the axis.
CustomProperties	0-1	Element	Custom properties for the axis.

ChartAxisTitle

The ChartAxisTitle element defines a title for an axis.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Caption	1	Expression (String)	Caption of the title
Position	0-1	Expression (Enum)	The position of the title along the axis. **Value** / **Description** Center: Default. Position the title on the center of the axis. Near: Position the title on the near side of the axis. Far: Position the title on the near side of the axis.
Style	0-1	Element	Defines style properties for the title.
TextOrientation	0-1	Expression (Enum)	Indicates the orientation of the text. **Value** / **Description** Auto: Default. Indicates the orientation will be selected automatically based on context (for example, Rotated270 for titles docked on the left). Horizontal: Horizontal text. Rotated90: Vertical text – Rotated 90 degrees. Rotated270: Vertical text – Rotated 270 degrees. Stacked: Vertical text – No character rotation.

ChartAxisScaleBreak

The ChartAxisScaleBreak element defines scale break behavior and style for an axis.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Enabled	0-1	Expression (Boolean)	Indicates scale breaks can be automatically applied.
BreakLineType	0-1	Expression (Enum)	Type of line used to show the scale break. **Value** / **Description** Ragged: Default. Display as a ragged line Straight: Display as a straight line

			Wave: Display as a wavy line None: Do not display a line for the scale break
CollapsibleSpaceThreshold	0-1	Expression (Integer)	Percent of empty space allowed on the axis before a scale break is triggered. Must be greater than 0. Default: 25
MaxNumberOfBreaks	0-1	Expression (Integer)	Maximum number of scale breaks to apply. Default: 2
Spacing	0-1	Expression (Float)	Amount of space to leave for a scale break, as a percent of the chart size. Default: 1.5
IncludeZero	0-1	Expression (Enum)	Indicates whether to prevent a scale break from spanning zero. Auto (Default) | True | False **Value** / **Description** Auto: Default. Determine whether to allow scale breaks to span zero based on the data plotted against the axis True: Do not allow a scale break to span zero. False: Allow a scale break to span zero.
Style	0-1	Element	Defines style properties for the scale break.

ChartStripLines

The ChartStripLines element defines a list of custom strip lines for an axis.

Attributes/Elements

Name	***Card***	***Type***	***Description***
ChartStripLine	1-N	Element	Custom strip line for an axis.

ChartStripLine

The ChartStripLine element defines a custom strip line for an axis.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Style	0-1	Element	Style properties for the strip line.
Title	0-1	Expression (String)	Title for the strip line.
TextOrientation	0-1	Expression (Enum)	Indicates the orientation of the title text. **Value** / **Description** Auto: Default. Indicates the orientation will be selected automatically based on context (for example,

				Rotated270 for titles docked on the left).
			Horizontal	Horizontal text.
			Rotated90	Vertical text – Rotated 90 degrees.
			Rotated270	Vertical text – Rotated 270 degrees.
			Stacked	Vertical text – No character rotation.

<table>
<tr><td>ActionInfo</td><td>0-1</td><td>Element</td><td>Actions for the strip line.</td></tr>
<tr><td>ToolTip</td><td>0-1</td><td>Expression (String)</td><td>Tool tip to display for the strip line.</td></tr>
<tr><td>Interval</td><td>0-1</td><td>Expression (Float)</td><td>Size of the strip line.
Default: 0</td></tr>
<tr><td>IntervalType</td><td>0-1</td><td>Expression (Enum)</td><td>Units for the Interval.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Auto</td><td>Default
Interval unit is autoderived based on the data plotted against the axis.</td></tr>
<tr><td>Number</td><td>Interval is numeric</td></tr>
<tr><td>Years</td><td>Interval is Years</td></tr>
<tr><td>Months</td><td>Interval is Months</td></tr>
<tr><td>Weeks</td><td>Interval is Weeks</td></tr>
<tr><td>Days</td><td>Interval is Days</td></tr>
<tr><td>Hours</td><td>Interval is Hours</td></tr>
<tr><td>Minutes</td><td>Interval is Minutes</td></tr>
<tr><td>Seconds</td><td>Interval is Seconds</td></tr>
<tr><td>Milliseconds</td><td>Interval is Milliseconds</td></tr>
</table></td></tr>
<tr><td>IntervalOffset</td><td>0-1</td><td>Expression (Float)</td><td>Offset from the previous strip line or axis min (for the first strip line).
Default: 0</td></tr>
<tr><td>IntervalOffsetType</td><td>0-1</td><td>Expression (Enum)</td><td>Units for the IntervalOffset
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Auto</td><td>Default
IntervalOffset unit is autoderived based on the data plotted against the axis.</td></tr>
<tr><td>Number</td><td>IntervalOffset is numeric</td></tr>
<tr><td>Years</td><td>IntervalOffset is Years</td></tr>
<tr><td>Months</td><td>IntervalOffset is Months</td></tr>
<tr><td>Weeks</td><td>IntervalOffset is Weeks</td></tr>
<tr><td>Days</td><td>IntervalOffset is Days</td></tr>
<tr><td>Hours</td><td>IntervalOffset is Hours</td></tr>
<tr><td>Minutes</td><td>IntervalOffset is Minutes</td></tr>
<tr><td>Seconds</td><td>IntervalOffset is Seconds</td></tr>
<tr><td>Milliseconds</td><td>IntervalOffset is Milliseconds</td></tr>
</table></td></tr>
<tr><td>StripWidth</td><td>0-1</td><td>Expression (Float)</td><td>Width of the strip line</td></tr>
<tr><td>StripWidthType</td><td>0-1</td><td>Expression (Enum)</td><td>Units for the StripWidth
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Auto</td><td>Default
StripWidth unit is autoderived based on the data plotted</td></tr>
</table></td></tr>
</table>

				against the axis.
			Number	StripWidth is numeric
			Years	StripWidth is Years
			Months	StripWidth is Months
			Weeks	StripWidth is Weeks
			Days	StripWidth is Days
			Hours	StripWidth is Hours
			Minutes	StripWidth is Minutes
			Seconds	StripWidth is Seconds
			Milliseconds	StripWidth is Milliseconds

ChartData

The ChartData element defines the segmentation of the data into multiple series.

Attributes/Elements

Name	***Card***	***Type***	***Description***
ChartSeriesCollection	1	Element	Data points for each series in the chart.
ChartDerivedSeriesCollection	0-1	Element	Derived series which are calculated from formulas applied to other series.

ChartSeriesCollection

The ChartSeriesCollection element defines a list of ChartSeries.

Attributes/Elements

Name	***Card***	***Type***	***Description***
ChartSeries	1-N	Element	Data points for each series in the chart. There must be as many ChartSeries elements are there are leaf-node (that is, has no sub-groups) ChartMembers in ChartSeriesHierarchy.

ChartSeries

The ChartSeries element defines the list of data points for one series.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Name	1	Name	Name of the series.
Hidden	0-1	Expression (Boolean)	Indicates the series should be hidden.
ChartDataPoints	0-1	Element	Data points within the series. Mandatory for ChartSeries within ChartSeriesCollection. Must be omitted for ChartSeries within DerivedChartSeriesCollection.
Type	0-1	Expression	Visualization type for the series.

<table>
<tr><td></td><td></td><td>(Enum)</td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Column</td><td>Default Column chart</td></tr>
<tr><td>Bar</td><td>Bar chart</td></tr>
<tr><td>Line</td><td>Line chart</td></tr>
<tr><td>Shape</td><td>Shape chart</td></tr>
<tr><td>Scatter</td><td>Scatter chart</td></tr>
<tr><td>Area</td><td>Area chart</td></tr>
<tr><td>Range</td><td>Range chart</td></tr>
<tr><td>Polar</td><td>Polar chart</td></tr>
</table>
</td></tr>
<tr><td>Subtype</td><td>0-1</td><td>Expression (Enum)</td><td>Visualization subtype for the series. Available subtypes (and default subtype) depends on Type.

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Plain</td><td>Default for all Types except Shape</td></tr>
<tr><td>Stacked</td><td>For Column, Bar and Area</td></tr>
<tr><td>PercentStacked</td><td>For Column, Bar and Area</td></tr>
<tr><td>Smooth</td><td>For Line, Area and Range</td></tr>
<tr><td>Stepped</td><td>For Line only</td></tr>
<tr><td>Pie</td><td>Default for Shape</td></tr>
<tr><td>ExplodedPie</td><td>For Shape only</td></tr>
<tr><td>Doughnut</td><td>For Shape only</td></tr>
<tr><td>Exploded Doughnut</td><td>For Shape only</td></tr>
<tr><td>Funnel</td><td>For Shape only</td></tr>
<tr><td>Pyramid</td><td>For Shape only</td></tr>
<tr><td>Bubble</td><td>For Scatter only</td></tr>
<tr><td>Stacked</td><td>For Area only</td></tr>
<tr><td>PercentStacked</td><td>For Area only</td></tr>
<tr><td>Candlestick</td><td>For Range only</td></tr>
<tr><td>Stock</td><td>For Range only</td></tr>
<tr><td>Bar</td><td>For Range only</td></tr>
<tr><td>Column</td><td>For Range only</td></tr>
<tr><td>BoxPlot</td><td>For Range only</td></tr>
<tr><td>ErrorBar</td><td>For Range only</td></tr>
<tr><td>Radar</td><td>For Polar only</td></tr>
</table>
If an invalid Subtype is specified, the default Subtype for the specified Type is used.</td></tr>
<tr><td>ChartEmptyPoints</td><td>0-1</td><td>Element</td><td>Defines behavior of empty points in the series.</td></tr>
<tr><td>Style</td><td>0-1</td><td>Element</td><td>Defines style properties for the series.</td></tr>
</table>

ChartDataLabel	0-1	Element	Indicates the values should be marked with data labels. Applies only within DerivedSeries.
ChartMarker	0-1	Element	Defines appearance of the data point marker. Applies only within DerivedSeries.
CustomProperties	0-1	Element	Custom properties for the series. This includes all custom chart attributes for series.
LegendName	0-1	String	Name of the legend in which this series should appear.
ChartItemInLegend	0-1	Element	Defines how the series appears when displayed in a legend.
ChartAreaName	0-1	String	Name of the chart area in which to plot the series. Defaults to the first chart area in the chart.
ValueAxisName	0-1	String	Name of the value axis against which to plot this series. If omitted, the series should be plotted against the first value axis.
CategoryAxisName	0-1	String	Name of the category axis against which to plot this series. If omitted, the series should be plotted against the first category axis.
ChartSmartLabel	0-1	Element	Smart label properties.

In the event the multiple ChartSeries in a ChartArea have Types and/or Subtypes which cannot be displayed together in the same area, the Type and Subtype of the first ChartSeries overrides that of subsequent ChartSeries with incompatible Type/Subtype.

Custom Chart Attributes

ChartSeries and ChartDataPoint support a set of custom attributes which modify the visualization behavior of certain series types and subtypes.

See http://support.dundas.com/OnlineDocumentation/WinChart2003/CustomAttributes_All.html

Upgrade note: This includes PointWidth and DrawingStyle, which were previously RDL elements.

ChartDataPoints

Collection of data points for a chart series. There must be a corresponding ChartDataPoint for each StaticMember within a StaticCategory.

Attributes/Elements

Name	***Card***	***Type***	***Description***
ChartDataPoint	1-N	Element	Collection of data points. There must be as many ChartDataPoint elements are there are leaf-node (that is, has no sub-groups) ChartGroups in CategoryGroups.

ChartDataPoint

The ChartDataPoint element defines a data point for the chart. A ChartDataPoint may consist of a single value expression (for example in bar or line charts) or multiple value expressions (stock and bubble charts).

Attributes/Elements

<table>
<tr><th>Name</th><th>Card</th><th>Type</th><th>Description</th></tr>
<tr><td>ChartDataPointValues</td><td>1</td><td>Element</td><td>Data values for the point.</td></tr>
<tr><td>ChartDataLabel</td><td>0-1</td><td>Element</td><td>Indicates the values should be marked with data labels.</td></tr>
<tr><td>AxisLabel</td><td>0-1</td><td>Expression (Variant)</td><td>Label to use on the axis for the data point.</td></tr>
<tr><td>ToolTip</td><td>0-1</td><td>Expression (String)</td><td>Tool tip to display for the data point.</td></tr>
<tr><td>ActionInfo</td><td>0-1</td><td>Element</td><td>Actions associated with this data point.</td></tr>
<tr><td>Style</td><td>0-1</td><td>Element</td><td>Defines style properties for the data point.</td></tr>
<tr><td>ChartMarker</td><td>0-1</td><td>Element</td><td>Defines appearance of the data point marker.</td></tr>
<tr><td>DataElementName</td><td>0-1</td><td>String</td><td>The name to use for the data element for this data point.
Default: Name of corresponding static series or category. If there is no static series or categories, “Value”
Must be a CLS-compliant identifier.</td></tr>
<tr><td>DataElementOutput</td><td>0-1</td><td>Enum</td><td>Indicates whether the data point should appear in a data rendering.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>ContentsOnly</td><td>Default
Indicates the data point should not appear in the output, but its values should be rendered as if they were in the cell’s container.</td></tr>
<tr><td>Output</td><td>Indicates the data point should appear in the output.</td></tr>
<tr><td>NoOutput</td><td>Indicates the data point should not appear in the output.</td></tr>
</table></td></tr>
<tr><td>ChartItemInLegend</td><td>0-1</td><td>Element</td><td>Defines how the data point appears when displayed in a legend (when Series.Type = Shape).</td></tr>
<tr><td>CustomProperties</td><td>0-1</td><td>Element</td><td>Custom properties for the data point.
This includes all custom chart attributes.</td></tr>
</table>

ChartDataPointValues

The ChartDataPointValues element defines a set of data values for a data point in the chart. Each series type has a different set of mandatory and optional data values. Data values not used for the series type are ignored.

Attributes/Elements

Name	*Card*	*Type*	*Description*
X	0-1	Expression (Scalar)	Indicates the X value for the data point. Mandatory in series with Type = Scatter.
Y	0-1	Expression (Numeric)	Indicates the Y value for the data point. Mandatory in series with Type <> Range and Type = Range with Subtype = ErrorBar.
Size	0-1	Expression (Numeric)	Indicates the size value for the data point. Optional in series with Type = Scatter with Subtype = Bubble.
High	0-1	Expression (Numeric)	Indicates the high value for the data point. Mandatory in series with Type = Range. May be omitted if Y is specified. If so, it defaults to Y.
Low	0-1	Expression (Numeric)	Indicates the high value for the data point. Mandatory in series with Type = Range. May be omitted if Y is specified. If so, it defaults to Y.
Start	0-1	Expression (Numeric)	Indicates the start/open value for the data point. Optional in series with Type = Range with SubType = Stock, Candlestick or BoxPlot.
End	0-1	Expression (Numeric)	Indicates the end/close value for the data point. Optional in series with Type = Range with SubType = Stock, Candlestick or BoxPlot.
Mean	0-1	Expression (Numeric)	Indicates the mean value for the data point. Optional in series with Type = Range with SubType = BoxPlot.
Median	0-1	Expression (Numeric)	Indicates the median value for the data point. Optional in series with Type = Range with SubType = BoxPlot.

ChartEmptyPoints

The ChartEmptyPoints element defines the behavior for empty points in a series.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Style	0-1	Element	Defines style properties for the data point.
ChartMarker	0-1	Element	Defines appearance of the data point marker.
ChartDataLabel	0-1	Element	Indicates the values should be marked with data labels.

AxisLabel	0-1	Expression (Variant)	Label to use on the axis for empty data points.
ToolTip	0-1	Expression (String)	Tool tip to display for the data point.
ActionInfo	0-1	Element	Actions associated with the data point.
CustomProperties	0-1	Element	Custom properties for the data point. This includes all custom series type attributes.

ChartItemInLegend

The ChartItemInLegend element defines the behavior for a series or set of data points displayed in a legend.

Attributes/Elements

Name	***Card***	***Type***	***Description***
LegendText	0-1	Expression (String)	Label to use in the legend for the item For ChartDataPoint, if LegendText is omitted, the Label properties from the ChartCategoriesHierarchy are used as the legend text (concatenated with " – " between each pair).
ToolTip	0-1	Expression (String)	Tool tip to display for the item in the legend.
ActionInfo	0-1	Element	Actions associated with the item in the legend.
Hidden	0-1	Expression (Boolean)	Indicates the item should not be shown in the legend.

ChartDerivedSeriesCollection

The ChartDerivedSeriesCollection element defines a list of ChartDerivedSeries.

Attributes/Elements

Name	***Card***	***Type***	***Description***
ChartDerivedSeries	1-N	Element	A derived series which is calculated from a formula applied to another series.

ChartDerivedSeries

The ChartDerivedSeries element defines a derived series which is calculated from a formula applied to another series.

Attributes/Elements

Name	*Card*	*Type*	*Description*
ChartSeries	1	Element	Series properties for the derived series.
SourceChartSeriesName	1	String	Name of the series from which to derive.
DerivedSeriesFormula	1	Enum	Formula to apply to the data values from the source series. See http://support.dundas.com/OnlineDocumentation/WinChart2003/FormulasOverview.html
ChartFormulaParameters	0-1	Element	Parameters to the formula.

ChartFormulaParameters

The ChartFormulaParameters element defines a list of parameters to a formula for a derived series.

Attributes/Elements

Name	*Card*	*Type*	*Description*
ChartFormulaParameter	1-N	Element	A parameter for the formula for a derived series.

ChartFormulaParameter

The ChartFormulaParameter element defines a parameter to a formula for a derived series.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Name	1	Name	Name of the parameter.
Value	0-1	Expression (Variant)	Value of the parameter if the value does not depend on the actual data points.
Source	0-1	String	Name of the ChartDataPointValue property to use as the value of this parameter.

Each DerivedSeriesFormula has its own set of ChartFormulaParameters. Each is defined to use either the Value or the Source. The other property, if specified, is ignored. The default value is also dependent on the DerivedSeriesFormula.

ChartDataLabel

The DataLabel element defines the data labels to display on data values.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Style	0-1	Element	Defines style properties for the labels. Supplied

<table>
<tr><td></td><td></td><td></td><td>styles override Series styles.</td></tr>
<tr><td>UseValueAsLabel</td><td>0-1</td><td>Expression (Boolean)</td><td>Indicates the Y value of the data point should be used as the label.</td></tr>
<tr><td>Label</td><td>0-1</td><td>Expression (Variant)</td><td>Label for the data point.
Not used if UseValueAsLabel = True</td></tr>
<tr><td>Visible</td><td>0-1</td><td>Expression (Boolean)</td><td>Whether the data label is displayed on the chart.
Defaults to False.</td></tr>
<tr><td>Position</td><td>0-1</td><td>Expression (Enum)</td><td>Position of the label.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Auto</td><td>Default</td></tr>
<tr><td>Top</td><td>Position label at Top of data point</td></tr>
<tr><td>TopLeft</td><td>Position label at TopLeft of data point</td></tr>
<tr><td>TopRight</td><td>Position label at TopRight of data point</td></tr>
<tr><td>Left</td><td>Position label at Left of data point</td></tr>
<tr><td>Center</td><td>Position label at Center of data point</td></tr>
<tr><td>Right</td><td>Position label at Right of data point</td></tr>
<tr><td>BottomRight</td><td>Position label at BottomRight of data point</td></tr>
<tr><td>Bottom</td><td>Position label at Bottom of data point</td></tr>
<tr><td>BottomLeft</td><td>Position label at BottomLeft of data point</td></tr>
<tr><td>Outside</td><td>Position label Outside of data point
For non-Pie charts, Outside is treated as Top.</td></tr>
</table></td></tr>
<tr><td>Rotation</td><td>0-1</td><td>Expression (Integer)</td><td>Angle of rotation of the label text.</td></tr>
<tr><td>ToolTip</td><td>0-1</td><td>Expression (String)</td><td>Tool tip to display for the data label.</td></tr>
<tr><td>ActionInfo</td><td>0-1</td><td>Element</td><td>Actions associated with this data label.</td></tr>
</table>

ChartSmartLabel

The ChartSmartLabel element defines behavior of smart labels.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Disabled	0-1	Expression	Indicates smart labels should be turned off.

<table>
<tr><td></td><td></td><td>(Boolean)</td><td></td></tr>
<tr><td>AllowOutSidePlotArea</td><td>0-1</td><td>Expression (Enum)</td><td>Indicates whether datapoint labels can be drawn outside of the plot area.
True | False | Partial (Default)<table><tr><th>Value</th><th>Description</th></tr><tr><td>Partial</td><td>Default
Labels can be partially outside the plot area.</td></tr><tr><td>True</td><td>Labels can be entirely outside the plot area.</td></tr><tr><td>False</td><td>Labels must be entirely inside the plot area.</td></tr></table></td></tr>
<tr><td>CalloutBackColor</td><td>0-1</td><td>Expression (Color)</td><td>Fill color of the box around the point label text when the CalloutStyle = Box</td></tr>
<tr><td>CalloutLineAnchor</td><td>0-1</td><td>Expression (Enum)</td><td>Shape that should be drawn on the point end of the callout line.
None | Arrow (Default) | Diamond | Square | Round</td></tr>
<tr><td>CalloutLineColor</td><td>0-1</td><td>Expression (Color)</td><td>Color of the callout line.
Default: Black</td></tr>
<tr><td>CalloutLineStyle</td><td>0-1</td><td>Expression (Enum)</td><td>Style of the callout line.<table><tr><th>Value</th><th>Description</th></tr><tr><td>Solid</td><td>Default
Solid line</td></tr><tr><td>None</td><td>No line</td></tr><tr><td>Dotted</td><td>Dotted line</td></tr><tr><td>Dashed</td><td>Dashed line</td></tr><tr><td>Double</td><td>Double solid line</td></tr><tr><td>DashDot</td><td>Dash-dot line</td></tr><tr><td>DashDotDot</td><td>Dash-dot-dot line</td></tr></table></td></tr>
<tr><td>CalloutLineWidth</td><td>0-1</td><td>Expression (Size)</td><td>Width of the callout line.
Default: 0.75pt</td></tr>
<tr><td>CalloutStyle</td><td>0-1</td><td>Expression (Enum)</td><td>Style to use when drawing the callout lines.
None | Underline (Default) | Box<table><tr><th>Value</th><th>Description</th></tr><tr><td>Underline</td><td>Default
Attach the callout line to an underline on the label</td></tr><tr><td>Box</td><td>Attach the callout line to a box around the label</td></tr><tr><td>None</td><td>No additional label style for the callout line</td></tr></table></td></tr>
<tr><td>ShowOverlapped</td><td>0-1</td><td>Expression (Boolean)</td><td>Indicates labels should be displayed even when overlapping issues cannot be resolved.</td></tr>
</table>

MarkerOverlapping	0-1	Expression (Boolean)	Indicates point labels are allowed to overlap point markers.
MaxMovingDistance	0-1	Expression (Size)	The maximum distance from the data point that data point labels can be moved to prevent overlapping. Default: 23 pt.
MinMovingDistance	0-1	Expression (Size)	The minimum distance from the data point that data point labels can be moved to prevent overlapping.
ChartNoMoveDirections	0-1	Element	Indicates which directions the label is not allowed to move

ChartNoMoveDirections

The ChartNoMoveDirections element defines which directions a smart label is not allowed to move.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Up	0-1	Expression (Boolean)	Indicates the smart label will not move directly up.
Left	0-1	Expression (Boolean)	Indicates the smart label will not move directly left.
Right	0-1	Expression (Boolean)	Indicates the smart label will not move directly right.
Down	0-1	Expression (Boolean)	Indicates the smart label will not move directly down.
UpLeft	0-1	Expression (Boolean)	Indicates the smart label will not move up-left.
UpRight	0-1	Expression (Boolean)	Indicates the smart label will not move up-right.
DownLeft	0-1	Expression (Boolean)	Indicates the smart label will not move down-left.
DownRight	0-1	Expression (Boolean)	Indicates the smart label will not move down-right.

ChartMarker

The ChartMarker element defines a marker for displayed chart values.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Type	0-1	Expression (Enum)	Defines the marker type for values.

<table>
<tr><td></td><td></td><td></td><td><table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>None</td><td>Default
No marker</td></tr>
<tr><td>Square</td><td>Square marker</td></tr>
<tr><td>Circle</td><td>Circle marker</td></tr>
<tr><td>Diamond</td><td>Diamond marker</td></tr>
<tr><td>Triangle</td><td>Triangle marker</td></tr>
<tr><td>Cross</td><td>Cross marker</td></tr>
<tr><td>Star4</td><td>Star (4 points) marker</td></tr>
<tr><td>Star5</td><td>Star (5 points) marker</td></tr>
<tr><td>Star6</td><td>Star (6 points) marker</td></tr>
<tr><td>Star10</td><td>Star (10 points) marker</td></tr>
<tr><td>Auto</td><td>Automatically cycle through marker types for each series</td></tr>
</table></td></tr>
<tr><td>Size</td><td>0-1</td><td>Expression (Size)</td><td>Represents the height and width of the plotting area of marker(s).
Default: 3.75pt.</td></tr>
<tr><td>Style</td><td>0-1</td><td>Element</td><td>Defines the style properties for the marker.</td></tr>
</table>

ChartThreeDProperties

The ChartThreeDProperties element defines properties for 3D layout.

Attributes/Elements

<table>
<tr><th>Name</th><th>Card</th><th>Type</th><th>Description</th></tr>
<tr><td>Enabled</td><td>0-1</td><td>Expression (Boolean)</td><td>Whether or not a chart is displayed in 3D. Default is False (2D).</td></tr>
<tr><td>ProjectionMode</td><td>0-1</td><td>Expression (Enum)</td><td>The projection mode used for the 3D rendering.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Oblique</td><td>Default
Use an oblique projection</td></tr>
<tr><td>Perspective</td><td>Use a perspective projection</td></tr>
</table></td></tr>
<tr><td>Perspective</td><td>0-1</td><td>Expression (Integer)</td><td>Represents the percent of perspective.
Applies only for Perspective projection.
Default: 0</td></tr>
<tr><td>Rotation</td><td>0-1</td><td>Expression (Integer)</td><td>Rotation angle
Default: 30</td></tr>
<tr><td>Inclination</td><td>0-1</td><td>Expression (Integer)</td><td>Inclination angle
Default: 30</td></tr>
<tr><td>DepthRatio</td><td>0-1</td><td>Expression (Integer)</td><td>Ratio (in percent)between depth and width.
Default: 100</td></tr>
<tr><td>Shading</td><td>0-1</td><td>Expression (Enum)</td><td>Type of 3D shading.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Real</td><td>Default
Realistic shading</td></tr>
<tr><td>Simple</td><td>Simplified shading</td></tr>
<tr><td>None</td><td>No shading</td></tr>
</table></td></tr>
<tr><td>GapDepth</td><td>0-1</td><td>Expression (Integer)</td><td>Percent depth gap between 3D bars and columns.
Default: 100</td></tr>
<tr><td>WallThickness</td><td>0-1</td><td>Expression (Integer)</td><td>Percent thickness of outer walls.
Default: 7</td></tr>
<tr><td>Clustered</td><td>0-1</td><td>Expression (Boolean)</td><td>Determines if data series are clustered (displayed along distinct rows). Only applies to bar and column chart types. Defaults to false.</td></tr>
</table>

ChartGridLines

The virtual ChartGridLines element defines gridlines along an axis.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Enabled	0-1	Expression (Enum)	Indicates the gridlines should be shown. **Value** / **Description** Auto: Default. True for major grid lines and false for minor grid lines. True: Show the grid lines. False: Hide the grid lines.
Style	0-1	Element	Line style properties for the grid lines.
Interval	0-1	Expression (Float)	Interval between gridlines. Default (0) uses ChartAxis.Interval.
IntervalType	0-1	Expression (Enum)	Units for the Interval. **Value** / **Description** Default: Default. Uses ChartAxis.IntervalType Auto: Interval unit is autoderived based on the data plotted against the axis. Number: Interval is numeric Years: Interval is Years Months: Interval is Months Weeks: Interval is Weeks Days: Interval is Days Hours: Interval is Hours Minutes: Interval is Minutes Seconds: Interval is Seconds Milliseconds: Interval is Milliseconds
IntervalOffset	0-1	Expression (Float)	Offset for the first gridline from the axis min. Default (0) uses ChartAxis.IntervalOffset.
IntervalOffsetType	0-1	Expression (Enum)	Units for the IntervalOffset. **Value** / **Description** Default: Default. Uses ChartAxis.IntervalOffsetType. Auto: IntervalOffset unit is autoderived based on the data plotted against the axis. Number: IntervalOffset is numeric Years: IntervalOffset is Years Months: IntervalOffset is Months Weeks: IntervalOffset is Weeks

			Days	IntervalOffset is Days
			Hours	IntervalOffset is Hours
			Minutes	IntervalOffset is Minutes
			Seconds	IntervalOffset is Seconds
			Milliseconds	IntervalOffset is Milliseconds

ChartMajorGridLines

The ChartMajorGridLines element defines style properties for major gridlines along an axis. It has no attributes/elements other than what it inherits from ChartGridLines.

ChartMinorGridLines

The ChartMinorGridLines element defines style properties for minor gridlines along an axis. It has no attributes/ elements other than what it inherits from ChartGridLines.

ChartTickMarks

The virtual ChartTickMarks element defines tick marks along an axis.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Enabled	0-1	Expression (Enum)	Indicates the tick marks should be shown. Auto (Default) | True | False **Value** / **Description** Auto: Default. True for major tick marks and false for minor tick marks. True: Show the tick marks. False: Hide the tick marks.
Type	0-1	Expression (Enum)	Type of the tick mark None | Inside | Outside (Default) | Cross **Value** / **Description** Outside: Default. Tick mark outside the axis. Inside: Tick mark inside the axis. Cross: Tick mark across the axis. None: No tick mark.
Style	0-1	Element	Line style properties for the tick marks.
Length	0-1	Expression (Float)	Length of the tick mark, as a percentage of the chart size. Default: 1
Interval	0-1	Expression (Float)	Interval between tick marks. Default (0) uses ChartAxis.Interval.
IntervalType	0-1	Expression (Enum)	Units for the Interval. **Value** / **Description** Default: Default. Uses ChartAxis.IntervalType. Auto: Interval unit is autoderived based on the data plotted against the axis. Number: Interval is numeric Years: Interval is Years Months: Interval is Months Weeks: Interval is Weeks Days: Interval is Days Hours: Interval is Hours Minutes: Interval is Minutes Seconds: Interval is Seconds Milliseconds: Interval is Milliseconds
IntervalOffset	0-1	Expression (Float)	Offset for the first tick mark from the axis min. Default (0) uses ChartAxis.IntervalOffset.

<table>
<tr><td>IntervalOffsetType</td><td>0-1</td><td>Expression (Enum)</td><td>Units for the IntervalOffset.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Default</td><td>Default
Uses ChartAxis.IntervalOffsetType.</td></tr>
<tr><td>Auto</td><td>IntervalOffset unit is autoderived based on the data plotted against the axis.</td></tr>
<tr><td>Number</td><td>IntervalOffset is numeric</td></tr>
<tr><td>Years</td><td>IntervalOffset is Years</td></tr>
<tr><td>Months</td><td>IntervalOffset is Months</td></tr>
<tr><td>Weeks</td><td>IntervalOffset is Weeks</td></tr>
<tr><td>Days</td><td>IntervalOffset is Days</td></tr>
<tr><td>Hours</td><td>IntervalOffset is Hours</td></tr>
<tr><td>Minutes</td><td>IntervalOffset is Minutes</td></tr>
<tr><td>Seconds</td><td>IntervalOffset is Seconds</td></tr>
<tr><td>Milliseconds</td><td>IntervalOffset is Milliseconds</td></tr>
</table></td></tr>
</table>

ChartMajorTickMarks

The MajorTickMarks element defines style properties for major tick marks along an axis.
It has no attributes/elements other than what it inherits from ChartTickMarks.

ChartMinorTickMarks

The MinorTickMarks element defines style properties for minor tick marks along an axis.
It has no attributes/ elements other than what it inherits from ChartTickMarks.

ChartCustomPaletteColors

The ChartCustomPaletteColors element defines the colors to use for the Custom palette.

Attributes/Elements

Name	***Card***	***Type***	***Description***
ChartCustomPaletteColor	1-N	Expression (Color)	A color to use in the Custom palette.

ChartCodeParameters

The CodeParameters element defines a set of parameters for the code in the chart.

Attributes/Elements

Name	*Card*	*Type*	*Description*
ChartCodeParameter	1-N	Element	A parameter for the code in the chart.

ChartCodeParameter

The ChartCodeParameter element defines a parameter for the code in the chart.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Name	1	Name	Name of the parameter.
Value	1	Expression (Variant)	Value of the parameter.

Chart Keyword Substitution
Various String properties throughout Chart can perform keyword substitutions when the chart is generated.
The following keywords are available if the object is in the context of a series:

Keyword	Description
#TOTAL	Total of all Y values in the series
#AVG	Average of all Y values in the series
#MIN	Minimum of all Y values in the series
#MAX	Maximum of all Y values in the series
#FIRST	First of all Y values in the series
#LAST	Last of all Y values in the series
#SERIESNAME	Series name

The following keywords are available if the object is in the context of a data point:

Keyword	Description
#VALX	X value of the data point
#VAL, #VALY #VALY2, #VALY3, and so on	Y values of the data point
#SERIESNAME	Series name
#LABEL	Data point label
#AXISLABEL	Axis data point label
#INDEX	Data point index
#PERCENT	Percentage of the data point Y value

Each keyword may be modified with standard .NET Framework format strings enclosed in braces. For example: #VALY{C2} would be replaced with the Y value of the data point, formatted as currency with two decimal places.

GaugePanel

The GaugePanel element defines gauge visualization for data point or set of data points.
It has the following attributes and elements in addition to what it inherits from DataRegion:

Attributes/Elements

Name	***Card***	***Type***	***Description***
LinearGauges	0-1	Element	Defines the set of linear gauges for the gauge panel.
RadialGauges	0-1	Element	Defines the set of radial gauges for the gauge panel.
GaugeLabels	0-1	Element	Defines the set of labels for the gauge panel.
GaugeMember	0-1	Element	Defines group, sort and filter behavior for the data.
AntiAliasing	0-1	Expression (Enum)	Antialiasing type for the gauge panel. **Value** | **Description**

			All: Default None: Text: Graphics:
AutoLayout	0-1	Expression (Boolean)	Indicates automatic layout should be used for elements in the gauge panel.
BackFrame	0-1	Element	The background/frame for the gauge panel.
ShadowIntensity	0-1	Expression (Float)	Intensity of shadows throughout the gauge panel. Must be between 0 and 100. Default: 25
TextAntiAliasingQuality	0-1	Expression (Enum)	Antialiasing quality for text. **Value** / **Description** High: Default Normal: SystemDefault:
TopImage	0-1	Element	Image to display over the top of the gauge panel.

GaugeMember

The GaugeMember element defines group, sort and filter behavior for the data.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Group	1	Element	Grouping to apply to the data.
SortExpressions	0-1	Element	Sorting to apply to the groups.
GaugeMember	0-1	Element	Nested grouping/sorting filtering.

GaugeInputValue

The GaugeInputValue element defines an expression and optional formula used in a gauge.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Value	1	Expression (Numeric)	Expression for the value.

<table>
<tr><td>Formula</td><td>0-1</td><td>Expression (Enum)</td><td>Type of calculation to perform on the values, if more than one is present.<table><tr><th>Value</th><th>Description</th></tr><tr><td>None</td><td>Default
Indicates the last value is used.</td></tr><tr><td>Average</td><td></td></tr><tr><td>Linear</td><td></td></tr><tr><td>Max</td><td></td></tr><tr><td>Min</td><td></td></tr><tr><td>Median</td><td></td></tr><tr><td>OpenClose</td><td></td></tr><tr><td>Percentile</td><td></td></tr><tr><td>Variance</td><td></td></tr><tr><td>RateOfChange</td><td></td></tr><tr><td>Integral</td><td></td></tr></table></td></tr>
<tr><td>MinPercent</td><td>0-1</td><td>Expression (Float)</td><td>Minimum percent.
Used only for Percentile formulas.</td></tr>
<tr><td>MaxPercent</td><td>0-1</td><td>Expression (Float)</td><td>Maximum percent.
Used only for Percentile formulas.</td></tr>
<tr><td>Multiplier</td><td>0-1</td><td>Expression (Float)</td><td>Constant by which to multiply the value.
Used only for Linear formulas.</td></tr>
<tr><td>AddConstant</td><td>0-1</td><td>Expression (Float)</td><td>Constant to add to the value (after multiplying).
Used only for Linear formulas.</td></tr>
<tr><td>DataElementName</td><td>0-1</td><td>String</td><td>The name to use for the data element/attribute for this value.
Must be a CLS-compliant identifier.</td></tr>
<tr><td>DataElementOutput</td><td>0-1</td><td>Enum</td><td>Indicates whether the item should appear in a data rendering.<table><tr><th>Value</th><th>Description</th></tr><tr><td>Output</td><td>Default
Indicates the item should appear in the output.</td></tr><tr><td>NoOutput</td><td>Indicates the item should not appear in the output.</td></tr></table></td></tr>
</table>

MaximumValue, MinimumValue, StartValue, EndValue

The MaximumValue, MinimumValue, StartValue and EndValue elements define an expression and optional formula used in a gauge.
They have no attributes/elements in addition to what they inherit from GaugeInputValue.

GaugePanelItem

The GaugePanelItem virtual element defines an item (gauge, image, label) to be drawn within a gauge panel.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Name	1	Name	Name of the item.
Top	0-1	Expression (Float)	Distance from the top as a percentage of the parent element. If the ParentItem property is not specified, the distance is relative to the GaugePanel .
Left	0-1	Expression (Float)	Distance from the left as a percentage of the parent element. If the ParentItem property is not specified, the distance is relative to the GaugePanel .
Height	0-1	Expression (Float)	Height of the item as a percentage of the parent element. If the ParentItem property is not specified, the height is relative to the GaugePanel.
Width	0-1	Expression (Float)	Width of the item as a percentage of the parent element. If the ParentItem property is not specified, the width is relative to the GaugePanel.
ZIndex	0-1	Expression (Integer)	Drawing order of the item within the panel.
Hidden	0-1	Expression (Boolean)	Indicates this item should be hidden.
ToolTip	0-1	Expression (String)	Tooltip text for the item.
ActionInfo	0-1	Element	Actions for the item.
ParentItem	0-1	String	Name of the parent GaugePanelItem.

Gauge

The Gauge virtual element defines a gauge to be drawn within a gauge panel.
It has the following attributes and elements in addition to what it inherits from GaugePanelItem:

Attributes/Elements

Name	*Card*	*Type*	*Description*
GaugeScales	0-1	Element	Scales to display on the gauge.
BackFrame	0-1	Element	The background/frame for the gauge.
ClipContent	0-1	Expression (Boolean)	Indicates the content of the gauge should be clipped by the bounds/frame of the gauge.
TopImage	0-1	Element	Image to display over the top of the gauge.
AspectRatio	0-1	Expression (Float)	Indicates the aspect ratio (width/height) to be used for drawing the gauge. Must be greater than or equal to zero. If zero or not specified, the aspect ratio used will be automatically determined based on the content of the gauge.

LinearGauges

The LinearGauges element defines a set of linear gauges for the gauge panel.

Attributes/Elements

Name	*Card*	*Type*	*Description*
LinearGauge	1-N	Element	A linear gauge for the gauge panel.

LinearGauge

The LinearGauge element defines a linear gauge to be drawn within a gauge panel.
It has the following attributes and elements in addition to what it inherits from Gauge:

Attributes/Elements

Name	*Card*	*Type*	*Description*
Orientation	0-1	Expression (Enum)	Orientation of the gauge. **Value** / **Description** Auto / Default Horizontal Vertical

RadialGauges

The RadialGauges element defines a set of radial gauges for the gauge panel.

Attributes/Elements

Name	*Card*	*Type*	*Description*
RadialGauge	1-N	Element	A radial gauge for the gauge panel.

RadialGauge

The RadialGauge element defines a radial gauge to be drawn within a gauge panel.
It has the following attributes and elements in addition to what it inherits from Gauge:

Attributes/Elements

Name	*Card*	*Type*	*Description*
PivotX	0-1	Expression (Float)	X position of the pivot point, as a percent of the gauge width. Default 50
PivotY	0-1	Expression (Float)	Y position of the pivot point, as a percent of the gauge height. Default 50

GaugeScales

The GaugeScales element defines a set of scales for a gauge.

Attributes/Elements

Name	*Card*	*Type*	*Description*
GaugeScale	1-N	Element	A scale to display within the gauge. Must contain only LinearScale within LinearGauge and

			RadialScale within RadialGauge.

GaugeScale

The GaugeScale virtual element defines a scale to be drawn within a gauge.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Name	1	Name	Name of the scale.
GaugePointers	0-1	Element	Pointers to display on the scale.
ScaleRanges	0-1	Element	Ranges to display on the scale.
Style	0-1	Element	Style properties for the scale.
CustomLabels	0-1	Element	Custom labels for the scale.
Interval	0-1	Expression (Float)	Default interval between tick marks and labels.
IntervalOffset	0-1	Expression (Float)	Default offset for the first tick mark and label.
Logarithmic	0-1	Expression (Boolean)	Indicates the scale is logarithmic.
LogarithmicBase	0-1	Expression (Float)	Base to use for logarithmic scale. Default: 10
MaximumValue	0-1	Element	Maximum value for the scale. Default: 100
MinimumValue	0-1	Element	Minimum value for the scale. Default: 0
Multiplier	0-1	Expression (Float)	Amount by which the gauge value is multiplied before being displayed.
Reversed	0-1	Expression (Boolean)	Indicates the direction of the scale is reversed.
GaugeMajorTickMarks	0-1	Element	Major tick marks to display on the scale.
GaugeMinorTickMarks	0-1	Element	Minor tick marks to display on the scale.
MaximumPin	0-1	Element	Maximum value at which a pointer on the scale will stop.
MinimumPin	0-1	Element	Minimum value at which a pointer on the scale will stop.
ScaleLabels	0-1	Element	Labels to display on the scale.
TickMarksOnTop	0-1	Expression (Boolean)	Indicates tick marks should be drawn atop pointers.
ToolTip	0-1	Expression (String)	Tool tip text for the scale
ActionInfo	0-1	Element	Actions for the scale.
Hidden	0-1	Expression (Boolean)	Indicates the scale is hidden.
Width	0-1	Expression (Float)	Width of the scale bar, as a percent of the size of the gauge.

LinearScale

The LinearScale element defines a linear scale to be drawn within a linear gauge.
It has the following attributes and elements in addition to what it inherits from GaugeScale:

Attributes/Elements

Name	*Card*	*Type*	*Description*
StartMargin	0-1	Expression (Float)	Distance between the start of the gauge and the start of the scale, as a percentage of the size of the gauge.
EndMargin	0-1	Expression (Float)	Distance between the end of the gauge and the end of the scale, as a percentage of the size of the gauge.
Position	0-1	Expression (Float)	Position of the scale, as a percentage of the size of the gauge (height for horizontal gauges, width for vertical gauges).

RadialScale

The RadialScale element defines a radial scale to be drawn within a radial gauge.
It has the following attributes/elements in addition to what it inherits from GaugeScale.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Radius	0-1	Expression (Float)	Radius of the Scale as a percentage of the gauge. Default: 37
StartAngle	0-1	Expression (Float)	The start angle of the scale in degrees (0-360). Default: 20
SweepAngle	0-1	Expression (Float)	The sweep angle of the scale in degrees (0-360). Default: 320

GaugePointers

The GaugePointers element defines a set of pointers for a scale.

Attributes/Elements

Name	*Card*	*Type*	*Description*
GaugePointer	1-N	Element	A pointer to display on the scale. Must contain only LinearPointer within LinearScale and RadialPointer within RadialScale.

GaugePointer

The GaugePointer virtual element defines a pointer to be drawn against a scale.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Name	1	Name	Name of the pointer.
Style	0-1	Element	Style properties for the pointer.
GaugeInputValue	0-1	Element	Value to use for the pointer.

<table>
<tr><td>BarStart</td><td>0-1</td><td>Expression (Enum)</td><td>Indicates where the pointer will start if it is of type Bar.<table><tr><th>Value</th><th>Description</th></tr><tr><td>ScaleStart</td><td>Default</td></tr><tr><td>Zero</td><td></td></tr></table></td></tr>
<tr><td>DistanceFromScale</td><td>0-1</td><td>Expression (Float)</td><td>Distance from the tip of the pointer to the scale, as a percentage of the scale size (radius for radial scales, length for linear scales).</td></tr>
<tr><td>PointerImage</td><td>0-1</td><td>Element</td><td>Image to use for the pointer.</td></tr>
<tr><td>MarkerLength</td><td>0-1</td><td>Expression (Float)</td><td>Length of the marker as a percentage of the parent scale radius.</td></tr>
<tr><td>MarkerStyle</td><td>0-1</td><td>Expression (Enum)</td><td>Type of the marker.<table><tr><th>Value</th><th>Description</th></tr><tr><td>Triangle</td><td>Default</td></tr><tr><td>Rectangle</td><td></td></tr><tr><td>Circle</td><td></td></tr><tr><td>Diamond</td><td></td></tr><tr><td>Trapezoid</td><td></td></tr><tr><td>Star</td><td></td></tr><tr><td>Wedge</td><td></td></tr><tr><td>Pentagon</td><td></td></tr><tr><td>None</td><td></td></tr></table></td></tr>
<tr><td>Placement</td><td>0-1</td><td>Expression (Enum)</td><td>Determines where the pointer should be placed relative to the scale.<table><tr><th>Value</th><th>Description</th></tr><tr><td>Inside</td><td></td></tr><tr><td>Outside</td><td>Default for Linear Gauge</td></tr><tr><td>Cross</td><td>Default for Radial Gauge</td></tr></table></td></tr>
<tr><td>SnappingEnabled</td><td>0-1</td><td>Expression (Boolean)</td><td>Indicates values should round to the snapping interval.</td></tr>
<tr><td>SnappingInterval</td><td>0-1</td><td>Expression (Float)</td><td>Interval to which the values should round.</td></tr>
<tr><td>ToolTip</td><td>0-1</td><td>Expression (String)</td><td>Tool tip text for the pointer.</td></tr>
<tr><td>ActionInfo</td><td>0-1</td><td>Element</td><td>Actions for the pointer.</td></tr>
<tr><td>Hidden</td><td>0-1</td><td>Expression (Boolean)</td><td>Indicates the pointer is hidden.</td></tr>
<tr><td>Width</td><td>0-1</td><td>Expression (Float)</td><td>Width of the pointer, as a percentage of the scale size (radius for radial scales, width for linear scales).</td></tr>
</table>

LinearPointer

The LinearPointer element defines a linear pointer to be drawn against a linear scale.
It has the following attributes and elements in addition to what it inherits from GaugePointer:

Attributes/Elements

Name	*Card*	*Type*	*Description*
Type	0-1	Expression (Enum)	Type of pointer. **Value** / **Description** Marker / Default Bar Thermometer
Thermometer	0-1	Element	Thermometer style properties.

RadialPointer

The RadialPointer element defines a radial pointer to be drawn against a radial scale.
It has the following attributes and elements in addition to what it inherits from GaugePointer:

Attributes/Elements

Name	*Card*	*Type*	*Description*
Type	0-1	Expression (Enum)	Type of pointer. **Value** / **Description** Needle / Default Marker Bar
PointerCap	0-1	Element	Style properties for the pointer cap.
NeedleStyle	0-1	Expression (Enum)	Style of the needle. **Value** / **Description** Triangular / Default Rectangular TaperedWithTail Tapered ArrowWithTail Arrow StealthArrowWithTail StealthArrow TaperedWithStealthArrow StealthArrowWithWideTail TaperedWithRoundedPoint

ScaleRanges

The ScaleRanges element defines a set of ranges for a scale.

Attributes/Elements

Name	*Card*	*Type*	*Description*
ScaleRange	1-N	Element	A range to display on the scale.

ScaleRange

The ScaleRange element defines a range to be drawn against a scale.

Attributes/Elements

<table>
<tr><th>Name</th><th>Card</th><th>Type</th><th>Description</th></tr>
<tr><td>Name</td><td>1</td><td>Name</td><td>Name of the range.</td></tr>
<tr><td>Style</td><td>0-1</td><td>Element</td><td>Style properties for the range.</td></tr>
<tr><td>BackgroundGradientType</td><td>0-1</td><td>Expression (Enum)</td><td>The type of background gradient.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>StartToEnd</td><td>Default</td></tr>
<tr><td>LeftRight</td><td></td></tr>
<tr><td>TopBottom, Center</td><td></td></tr>
<tr><td>DiagonalLeft</td><td></td></tr>
<tr><td>DiagonalRight</td><td></td></tr>
<tr><td>HorizontalCenter</td><td></td></tr>
<tr><td>VerticalCenter</td><td></td></tr>
<tr><td>None</td><td></td></tr>
</table>
This property is used instead of Style.BackgroundGradientType.</td></tr>
<tr><td>DistanceFromScale</td><td>0-1</td><td>Expression (Float)</td><td>Distance from the range to the scale, as a percentage of the size of the scale.
Default: 10</td></tr>
<tr><td>StartValue</td><td>0-1</td><td>Element</td><td>Starting value for the range.</td></tr>
<tr><td>EndValue</td><td>0-1</td><td>Element</td><td>Ending value for the range.</td></tr>
<tr><td>StartWidth</td><td>0-1</td><td>Expression (Float)</td><td>Width of the range at the start, as a percentage of the size of the scale.</td></tr>
<tr><td>EndWidth</td><td>0-1</td><td>Expression (Float)</td><td>Width of the range at the start, as a percentage of the size of the scale.</td></tr>
<tr><td>InRangeBarPointerColor</td><td>0-1</td><td>Expression (Color)</td><td>Color of the bar pointer if it falls within this range.</td></tr>
<tr><td>InRangeLabelColor</td><td>0-1</td><td>Expression (Color)</td><td>Color of scale labels that fall within this range.</td></tr>
<tr><td>InRangeTickMarksColor</td><td>0-1</td><td>Expression (Color)</td><td>Color of tick marks that fall within this range.</td></tr>
<tr><td>Placement</td><td>0-1</td><td>Expression (Enum)</td><td>Determines where the range should be placed relative to the scale.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Inside</td><td>Default for Radial Range.</td></tr>
<tr><td>Outside</td><td>Default for Linear Range.</td></tr>
<tr><td>Cross</td><td></td></tr>
</table></td></tr>
<tr><td>ToolTip</td><td>0-1</td><td>Expression (String)</td><td>Tool tip text for the range.</td></tr>
<tr><td>ActionInfo</td><td>0-1</td><td>Element</td><td>Actions for the range.</td></tr>
<tr><td>Hidden</td><td>0-1</td><td>Expression (Boolean)</td><td>Indicates the range is hidden.</td></tr>
</table>

GaugeLabels

The GaugeLabels element defines a set of labels to display within a gauge panel.

Attributes/Elements

Name	*Card*	*Type*	*Description*
GaugeLabel	1-N	Element	A label to display within a gauge panel.

GaugeLabel

The GaugeLabel element defines a label to display within a gauge panel.
It has the following attributes and elements in addition to what it inherits from GaugePanelItem:

Attributes/Elements

Name	*Card*	*Type*	*Description*
Text	0-1	Expression (String)	Text of the label.
Style	0-1	Element	Defines style properties for the item.
Angle	0-1	Expression (Float)	Angle of rotation for the label.
ResizeMode	0-1	Expression (Enum)	Indicates whether the content will resize to fit in the available space. **Value** / **Description** AutoFit / Default None /
TextShadowOffset	0-1	Expression (Size)	Size of the text shadow. Default: 0
UseFontPercent	0-1	Expression (Boolean)	Determines if the font size is measured as a percentage of the parent or in units specified.

BaseGaugeImage

The BaseGaugeImage virtual element defines an image to be displayed as a part of a gauge.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Source	1	Expression (Enum)	Identifies the source of the image. **Value** / **Description** External / The Value contains a string constant or expression that evaluates to the location of the image. Embedded / The Value contains a string constant or expression that evaluates to the name of an EmbeddedImage within the report. Database / The Value contains an expression (for example, a field in the database) that evaluates to the binary data for the image.

Value	1	Expression (Variant)	See Source. Expected datatype is string or binary, depending on Source. If the Value is null, no image is displayed.
MIMEType	0-1	Expression (String)	The MIMEType of the image. Valid values are: image/bmp, image/jpeg, image/gif, image/png, image/x-png Required if Source = Database. Ignored otherwise.
TransparentColor	0-1	Expression (Color)	Color to treat as transparent in the image.

TopImage

The TopImage element defines an image to be displayed atop part of a gauge.
It has the following attributes and elements in addition to what it inherits from BaseGaugeImage:

Attributes/Elements

Name	***Card***	***Type***	***Description***
HueColor	0-1	Expression (Color)	Color with which to tint the image.

PointerImage

The PointerImage element defines an image to be used for a pointer in a gauge.
It has the following attributes and elements in addition to what it inherits from BaseGaugeImage:

Attributes/Elements

Name	***Card***	***Type***	***Description***
HueColor	0-1	Expression (Color)	Color with which to tint the image.
Transparency	0-1	Expression (Float)	Percent transparency for the image. Ignored within IndicatorState
OffsetX	0-1	Expression (Size)	X offset in the image for the pointer origin.
OffsetY	0-1	Expression (Size)	Y offset in the image for the pointer origin.

CapImage

The CapImage element defines an image to be used for a pointer cap in a radial gauge.
It has the following attributes and elements in addition to what it inherits from BaseGaugeImage:

Attributes/Elements

Name	***Card***	***Type***	***Description***
HueColor	0-1	Expression (Color)	Color with which to tint the image.
OffsetX	0-1	Expression (Size)	X offset in the image for the cap origin.

OffsetY	0-1	Expression (Size)	Y offset in the image for the cap origin.

TickMarkImage

The TickMarkImage element defines an image to be used for a tick mark on a scale. It has the following attributes and elements in addition to what it inherits from BaseGaugeImage:

Attributes/Elements

Name	*Card*	*Type*	*Description*
HueColor	0-1	Expression (Color)	Color with which to tint the image.

Thermometer

The Thermometer element defines display properties for a linear gauge thermometer pointer.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Style	0-1	Element	Style properties for the thermometer.
BulbOffset	0-1	Expression (Float)	Offset of the bulb from the zero position, as a percent of the scale length. Default: 5
BulbSize	0-1	Expression (Float)	Size of the bulb, as a percent of the scale width. Default: 50
ThermometerStyle	0-1	Expression (Enum)	Type of the thermometer. **Value** – **Description** Standard – Default Flask –

PointerCap

The PointerCap element defines display properties for a radial gauge pointer cap.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Style	0-1	Element	Style properties for the pointer cap.
CapImage	0-1	Element	Image to use for the pointer cap.
OnTop	0-1	Expression (Boolean)	Indicates the cap is displayed on top of the pointer.
Reflection	0-1	Expression (Boolean)	Indicates the cap has a reflection effect.
CapStyle	0-1	Expression (Enum)	Type of the cap **Value** – **Description** RoundedDark – Default Rounded – RoundedLight –

			RoundedWithAdditionalTop RoundedWithWideIndentation FlattenedWithIndentation FlattenedWithWideIndentation RoundedGlossyWithIndentation RoundedWithIndentation
Hidden	0-1	Expression (Boolean)	Indicates the cap is hidden.
Width	0-1	Expression (Float)	Width of the cap, as a percent of the scale radius.

GaugeTickMarks

The GaugeTickMarks element defines major tick marks along a scale.
It has the following attributes/elements in addition to what it inherits from TickMarkStyle.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Interval	0-1	Expression (Float)	Interval between tick marks. Defaults to GaugeScale.Interval
IntervalOffset	0-1	Expression (Float)	Offset for the first tick mark. Defaults to GaugeScale.IntervalOffset

GaugeMajorTickMarks

The GaugeMajorTickMarks element defines major tick marks along a scale.
It has no attributes/elements in addition to what it inherits from GaugeTickMarks.

GaugeMinorTickMarks

The GaugeMinorTickMarks element defines minor tick marks along a scale.
It has no attributes/elements in addition to what it inherits from GaugeTickMarks.

ScaleLabels

The ScaleLabels element defines the appearance of labels on a scale.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Style	0-1	Element	Style properties for the labels.
Interval	0-1	Expression (Float)	Interval between labels. Defaults to GaugeScale.Interval
IntervalOffset	0-1	Expression (Float)	Offset for the first label. Defaults to GaugeScale.IntervalOffset
AllowUpsideDown	0-1	Expression (Boolean)	Indicates the labels can be rotated by more than 90 degrees.
DistanceFromScale	0-1	Expression (Float)	Distance from the labels to the scale, as a percentage of the scale size (radius for radial

			scales, length for linear scales).
FontAngle	0-1	Expression (Float)	Angle of rotation for the text.
Placement	0-1	Expression (Enum)	Determines where the labels should be placed relative to the scale. **Value** / **Description** Inside / Default Outside Cross
RotateLabels	0-1	Expression (Boolean)	Indicates the text rotates together with the scale.
ShowEndLabels	0-1	Expression (Boolean)	Indicates the labels at the ends of the scale should be shown.
Hidden	0-1	Expression (Boolean)	Indicates the labels are hidden.
UseFontPercent	0-1	Expression (Boolean)	Determines if the font size is measured as a percentage of the parent or in units specified.

CustomLabels

The CustomLabels element defines a set of custom labels for a scale.

Attributes/Elements

Name	***Card***	***Type***	***Description***
CustomLabel	1-N	Element	A custom label to display on a scale.

CustomLabel

The CustomLabel element defines a custom label for a scale.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Name	1	Name	Name of the custom label.
Style	0-1	Element	Style properties for the custom label.
Text	0-1	Expression (String)	Text of the custom label.
AllowUpsideDown	0-1	Expression (Boolean)	Indicates the label can be rotated by more than 90 degrees. (Only used for radial scale).
DistanceFromScale	0-1	Expression (Float)	Distance from the label to the scale, as a percentage of the scale size (radius for radial scales, length for linear scales).
FontAngle	0-1	Expression (Float)	Angle of rotation for the text.
Placement	0-1	Expression (Enum)	Determines where the label should be placed relative to the scale. **Value** / **Description** Inside / Default

			Outside Cross
RotateLabel	0-1	Expression (Boolean)	Indicates the text rotates together with the scale. (Only used for radial scale)
TickMarkStyle	0-1	Element	Style properties for the tick mark.
Value	0-1	Expression (Numeric)	Position on the scale where the label will be placed.
Hidden	0-1	Expression (Boolean)	Indicates the custom label is hidden.
UseFontPercent	0-1	Expression (Boolean)	Determines if the font size is measured as a percentage of the parent or in units specified.

TickMarkStyle

The TickMarkStyle element defines a tick mark associated with a custom label.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Style	0-1	Element	Style properties for the tick mark
DistanceFromScale	0-1	Expression (Float)	Distance from the label to the tick mark, as a percentage of the scale size (radius for radial scales, length for linear scales).
Placement	0-1	Expression (Enum)	Determines where the tick mark should be placed relative to the scale **Value** – **Description** Inside – Default Outside Cross
EnableGradient	0-1	Expression (Boolean)	Indicates whether a gradient effect is used for the tick mark.
GradientDensity	0-1	Expression (Float)	Intensity of the gradient effect (from 0 to 100).
TickMarkImage	0-1	Element	Image to use for the tick mark.
Length	0-1	Expression (Float)	Length of the tick mark as a percentage of the parent scale radius for radial gauge and as a percentage of the Width/Height for linear gauge whichever is smaller.
Width	0-1	Expression (Float)	Width of the tick mark as a percentage of the parent scale radius for radial gauge and as a percentage of the Width/Height for linear gauge whichever is smaller.
Shape	0-1	Expression (Enum)	Shape of the tick mark. **Value** – **Description** Rectangle – Default Triangle Circle

			Diamond Trapezoid Star Wedge Pentagon None
Hidden	0-1	Expression (Boolean)	Indicates the custom label is hidden.

ScalePin

The ScalePin virtual element defines a pin value at one end of a scale.
It has the following attributes and elements in addition to what it inherits from TickMarkStyle:

Attributes/Elements

Name	***Card***	***Type***	***Description***
Location	0-1	Expression (Float)	Location of the pin, relative to the start or end of the scale. Degrees for radial scales, percentage for linear scales.
Enable	0-1	Expression (Boolean)	Indicates the pin is enabled.
PinLabel	0-1	Element	Defines a label for the pin.

MaximumPin

The MaximumPin element defines a pin value at the top of a scale.
It has no attributes/elements in addition to what it inherits from ScalePin.

MinimumPin

The MinimumPin element defines a pin value at the bottom of a scale.
It has no attributes/elements in addition to what it inherits from ScalePin.

PinLabel

The PinLabel virtual element defines a label for a pin value at one end of a scale.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Style	0-1	Element	Style properties for the pin label.
Text	0-1	Expression (String)	Text of the label.
AllowUpsideDown	0-1	Expression (Boolean)	Indicates the label can be rotated by more than 90 degrees. Applies only to radial scales.
DistanceFromScale	0-1	Expression (Float)	Distance from the label to the scale, as a percentage of the scale size (radius for radial scales, length for linear scales).
FontAngle	0-1	Expression (Float)	Angle of rotation for the text.

Placement	0-1	Expression (Enum)	Determines where the label should be placed relative to the scale. **Value** / **Description** Inside / Default Outside Cross
RotateLabel	0-1	Expression (Boolean)	Indicates the text rotates together with the scale Applies only to radial scales.
UseFontPercent	0-1	Expression (Boolean)	Determines if the font size is measured as a percentage of the parent or in units specified.

BackFrame

The BackFrame element defines the frame and frame background for a gauge or gauge panel.

Attributes/Elements

Name	***Card***	***Type***	***Description***
Style	0-1	Element	Style properties for the frame.
FrameStyle	0-1	Expression (Enum)	Style of the frame. **Value** / **Description** None / Default Simple Edged
FrameShape	0-1	Expression (Enum)	Shape of the frame **Value** / **Description** Default / Default Treated as Circular for radial gauges, Rectangular for linear gauges and gauge panels. Circular Rectangular RoundedRectangular AutoShape CustomCircular1 CustomCircular2 CustomCircular3 CustomCircular4 CustomCircular5 CustomCircular6 CustomCircular7 CustomCircular8 CustomCircular9 CustomCircular10 CustomCircular11

			CustomCircular12
			CustomCircular13
			CustomCircular14
			CustomCircular15 |
			CustomSemiCircularN1
			CustomSemiCircularN2
			CustomSemiCircularN3
			CustomSemiCircularN4
			CustomSemiCircularS1
			CustomSemiCircularS2
			CustomSemiCircularS3
			CustomSemiCircularS4
			CustomSemiCircularE1
			CustomSemiCircularE2
			CustomSemiCircularE3
			CustomSemiCircularE4 |
			CustomSemiCircularW1
			CustomSemiCircularW2
			CustomSemiCircularW3
			CustomSemiCircularW4 |
			CustomQuarterCircularNE1
			CustomQuarterCircularNE2
			CustomQuarterCircularNE3
			CustomQuarterCircularNE4
			CustomQuarterCircularNW1
			CustomQuarterCircularNW2
			CustomQuarterCircularNW3
			CustomQuarterCircularNW4
			CustomQuarterCircularSE1
			CustomQuarterCircularSE2
			CustomQuarterCircularSE3
			CustomQuarterCircularSE4
			CustomQuarterCircularSW1
			CustomQuarterCircularSW2 |
			CustomQuarterCircularSW3
			CustomQuarterCircularSW4
FrameWidth	0-1	Expression (Float)	Width (thickness) of the frame. It is measured as a percentage of the width or height of the gauge or gauge container, whichever is smaller. Must be between 0 and 50. Default: 8
GlassEffect	0-1	Expression (Enum)	The glass effect applied to the frame. **Value** | **Description** None | Default Simple |

FrameBackground	0-1	Element	Appearance properties for the frame background.
FrameImage	0-1	Element	Image to use for the frame background.

FrameBackground

The FrameBackground element defines the background of the frame for a gauge or gauge panel.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Style	0-1	Element	Style properties for the background.

FrameImage

The FrameImage element defines an image to be used for the frame background of a gauge or gauge panel.
It has the following attributes and elements in addition to what it inherits from BaseGaugeImage:

Attributes/Elements

Name	*Card*	*Type*	*Description*
HueColor	0-1	Expression (Color)	Color with which to tint the image.
Transparency	0-1	Expression (Float)	Percent transparency for the image. Ignored within IndicatorState.
ClipImage	0-1	Expression (Boolean)	Indicates the image should be clipped to the frame outline.

CustomReportItem

CustomReportItem describes a report item that is not natively defined in RDL. Extended information about the custom report item should be placed within the CustomProperties element by using a namespace prefix specific to the tool or server that supports the type. Tools and servers that do not support the type use the AltReportItem instead.
CustomReportItem is not allowed in the PageHeader or PageFooter.
The CustomReportItem element has the following attributes/elements in addition to what it inherits from ReportItem:

Attributes/Elements

Name	***Card***	***Type***	***Description***
Type	1	String	The type of the custom report item. Interpreted by the report design tool or server. Unsupported types generate a warning (see ReportItems later in this document).
AltReportItem	0-1	Element	Report item to render instead of the custom report item, if the custom item type is not supported natively. If not supplied, the AltReportItem will be treated as an empty rectangle without a border.
CustomData	0-1	Element	Defines data to be passed to the custom report item control.

Example:

```
<CustomReportItem Name=”Text5”>
        <Type>ThreeDTextControl</Type>
        <Top>2 in</Top>
        <Left>2 in</Left>
        <Height>1 in</Height>
        <Width>4 in</Width>
        <CustomProperties>
                <CustomProperty>
                        <Name>ms:ThreeDTextValue</Name>
                        <Value>{\r90\a45 Pretty}{\r90\d2 formatted text}</Value>
                </CustomProperty>
        </CustomProperties>
        <AltReportItem>
                <Textbox Name=”Textbox6”>
                        <Paragraphs>
                                <Paragraph>
                                        <TextRuns>
                                                <TextRun>
                                                <Value>Boring plain text</Value>
                                                </TextRun>
                                        </TextRuns>
                                </Paragraph>
                        <Paragraphs>
                </Textbox>
        </AltReportItem>
</CustomReportItem>
```

All report item Style properties apply to custom report items.

AltReportItem

The AltReportItem element defines a report item to render instead of the custom report item, if the custom item type is not supported natively. All restrictions on report items (for example, placement of data regions in page headers) apply to the AltReportItem as if it were a regular report item in the report.

Attributes/Elements

Name	*Card*	*Type*	*Description*
ReportItem	1	Element	The report item to render instead of the custom report item. Position properties of the ReportItem (Top, Left, Height, Width, ZIndex) are ignored because they are taken from the CustomReportItem instead. Can not contain a CustomReportItem. ReportItems within AltReportItem are available in the ReportItems global collection only if the custom report item type is not supported natively.

CustomData

The CustomData element defines the data to be handed to a custom report item and the way that data should be grouped, sorted, filtered and aggregated. Custom report items with a CustomData element are considered to be DataRegions. This affects such things as restricting placement in the report (that is, not supported in detail rows, page header or page footer), available scopes for expressions and RepeatWith.

Attributes/Elements

Name	*Card*	*Type*	*Description*
DataSetName	1	String	Indicates which data set to use as the source data for this custom report item.
Filters	0-1	Element	Filters to apply to each row of data.
DataColumnHierarchy	0-1	Element	The hierarchy of column members for the data.
DataRowHierarchy	0-1	Element	The hierarchy of row members for the data.
DataRows	0-1	Element	The data values to calculate for each detail “cell” of the data.

DataHierarchy

The virtual DataHierarchy element defines a hierarchy of groups for the data.

Attributes/Elements

Name	*Card*	*Type*	*Description*
DataMembers	1	Element	The list of members at the base of the hierarchy.

DataColumnHierarchy

The DataColumnHierarchy element has no additional attributes/elements beyond what it inherits from DataHierarchy.

DataRowHierarchy

The DataRowHierarchy element has no additional attributes/elements beyond what it inherits from DataHierarchy.

DataMembers

The DataMembers element defines a list of members of a custom data hierarchy.

Attributes/Elements

Name	*Card*	*Type*	*Description*
DataMember	1-N	Element	An ordered list of members.

DataMember

The DataMember element defines a member of a custom data hierarchy.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Group	0-1	Element	The expressions by which to group the data. If omitted, this is a static group. Not allowed if any ancestor group is a detail group.
SortExpressions	0-1	Element	The expressions by which to sort the group instances. Not allowed if Group is omitted.
CustomProperties	0-1	Element	Custom properties for the member.
DataMembers	0-1	Element	Submembers contained within this member.
Subtotal	0-1	Boolean	Indicates an automatic subtotal should be calculated for this member.

Automatic Subtotals and Submembers
When an automatic subtotal is requested, all submembers are preserved in the subtotal, but are treated as static members.
For example, consider the following custom report item (drawn as a Tablix for clarity):

	=Year	
	=Quarter	Growth
=Product	=Sum(Sales)	=Avg(Growth)

The Year member and the Quarter member are both marked to generate subtotals.

When expanded with data, the rendering object model structure would look something like this:

	2003						2004				Total	
	Q1	Q2	Q3	Q4	Total	Growth	Q1	Q2	Total	Growth	Total	Growth
Table	2	2	2	2	8	0%	2	4	6	50%	14	20%
Chair	1	2	4	8	15	100%	8	8	16	0%	31	60%

Notice the Year subtotal contains two subcolumns: One for quarter (which is a total) and one for growth.

DataRows

The DataRows element defines the rows of data to pass to the custom report item.

Attributes/Elements

Name	***Card***	***Type***	***Description***
DataRow	1-N	Element	A row of data values to pass to the custom report item. There must be as many DataRow elements as there are leaf-node (that is, has no sub-groups) DataGroups in DataRowGroups.

DataRow

The DataRow element defines the columns of data within each row to pass to the custom report item.

Attributes/Elements

Name	*Card*	*Type*	*Description*
DataCell	1-N	Element	A list of data values to pass to the custom report item. There must be as many DataCell elements as there are leaf-node (that is, has no sub-groups) DataGroups in DataColumnGroups.

DataCell

The DataCell element defines the list of data values to pass to the custom report item for a specific combination of leaf-node groups in CustomData.

Attributes/Elements

Name	*Card*	*Type*	*Description*
DataValue	1-N	Element	A data value to pass to the custom report item.

Tablix

The Tablix element defines a flexible layout grid with nested repeating column groups and row groups.

Attributes/Elements

Name	*Card*	*Type*	*Description*
TablixCorner	0-1	Element	The region that contains the elements of the upper left corner area of the tablix.
TablixBody	1	Element	The bottom right region that contains the data elements of the tablix.
TablixColumnHierarchy	1	Element	The hierarchy of column members for the tablix.
TablixRowHierarchy	1	Element	The hierarchy of row members for the tablix.
LayoutDirection	0-1	Enum	Indicates the overall direction of the tablix layout. **Value** / **Description** LTR — Default Dynamic tablix columns grow left-to-right (with headers on the left). RTL — Dynamic tablix columns grow right-to-left (with headers on the right) and the order of peer groups is reversed.
GroupsBeforeRowHeaders	0-1	Integer	The number of instances of the leftmost outer column member that should appear to the left of the row headers (right of the row headers for

			RTL tablixes). Ignored if the leftmost outer column member is a static member. Default: 0 Min: 0 Max: 2147483647
RepeatColumnHeaders	0-1	Boolean	Indicates the column headers should be repeated on each page on which a portion of the Tablix appears.
RepeatRowHeaders	0-1	Boolean	Indicates the column headers should be repeated on each page on which a portion of the Tablix appears.
FixedColumnHeaders	0-1	Boolean	Indicates the column headers should be displayed on the page even when the user scrolls part of the Tablix off the page.
FixedRowHeaders	0-1	Boolean	Indicates the row headers should be displayed on the page even when the user scrolls part of the Tablix off the page.
OmitBorderOnPageBreak	0-1	Boolean	Indicates the borders should not appear at locations where the tablix spans multiple pages. Also causes repeated background images to continue rather than restart after a page break.
KeepTogether	0-1	Boolean	Indicates the whole tablix (all repeated sections) should be kept together on one page if possible.

TablixCorner

The TablixCorner element defines the layout and structure of the upper left-hand corner region of a Tablix[32].

Attributes/Elements

Name	***Card***	***Type***	***Description***
TablixCornerRows	1	Element	The list of rows in the corner section of the Tablix. There must be as many TablixCornerRow elements as there are unique total heights (sum of the sizes of the TablixHeaders of the member and all ancestor members) of TablixMembers in the TablixColumnHierarchy.

[32] The height of the corner is the sum of the heights of the column headers. The width of the corner is the sum of the widths of the row headers.

TablixCornerRows

The TablixCornerRows element defines the list of rows in the TablixCorner.

Attributes/Elements

Name	*Card*	*Type*	*Description*
TablixCornerRow	1-N	Element	The list of rows in the corner section of the Tablix.

TablixCornerRow

The TablixCornerRow element defines the list of cells in a row of the corner section of a Tablix. The height of the row is equal to the height of the corresponding column TablixHeader[33].

Attributes/Elements

Name	*Card*	*Type*	*Description*
TablixCornerCell	1-N	Element	The list of cells in a row in the corner section of the Tablix. There must be as many TablixCornerCell elements as there are unique total widths (sum of the sizes of the TablixHeaders of the member and all ancestor members) of TablixMembers in the TablixRowHierarchy.

[33] The deepest nesting of TablixGroupings is used, ignoring DataGroupings with no TablixHeader.

TablixCornerCell

The TablixCornerCell element defines the contents of each corner cell in the Tablix.
The width of the each column is equal to the width of the corresponding row TablixHeader.

Attributes/Elements

Name	*Card*	*Type*	*Description*
CellContents	0-1	Element	Page breaks in this ReportItem are ignored. Must be omitted if the position in the corner is covered by a span from another cell. Required otherwise.

TablixHierarchy

The virtual TablixHierarchy element defines a hierarchy of members for the tablix

Attributes/Elements

Name	*Card*	*Type*	*Description*
TablixMembers	1	Element	The list of members at the base of the hierarchy

For each leaf member in the hierarchy, the total size (that is, sum of the sizes of the TablixHeaders of the member and all ancestor members) must be the same.

TablixRowHierarchy

The TablixRowHierarchy element has no attributes/elements beyond what it inherits from TablixHierarchy.

TablixColumnHierarchy

The TablixColumnHierarchy element has no attributes/elements beyond what it inherits from TablixHierarchy.

TablixMembers

The TablixMembers element defines a list of members in a Tablix hierarchy.

Attributes/Elements

Name	*Card*	*Type*	*Description*
TablixMember	1-N	Element	An ordered list of members of a tablix hierarchy.

TablixMember

The TablixMember element defines a member of a tablix hierarchy.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Group	0-1	Element	The expressions by which to group the data. If omitted, this is a static member (otherwise, this is a dynamic member). Not allowed if any ancestor group is a detail group.

SortExpressions	0-1	Element	The expressions by which to sort the group instances. Not allowed if Group is omitted.
TablixHeader	0-1	Element	Defines the header cell for the member.
TablixMembers	0-1	Element	Submembers contained within this member.
CustomProperties	0-1	Element	Custom properties for the member.
FixedData	0-1	Boolean	Indicates the whole member (including its body cells) should be displayed on the page even when the user scrolls part of the Tablix off the page. Can be true only on the outermost members on the hierarchy (row or column) of the Tablix. Members with FixedData=True must be contiguous with all other FixedData members on the hierarchy. Not allowed to be true if the Tablix has headers on the opposite hierarchy (FixedRowHeaders or FixedColumnHeaders). Not allowed to be true on the leftmost column member if GroupsBeforeRowHeaders is set. Not allowed to be true on row members unless set on the first row member. Not allowed to be true if a corresponding body cell is part of a span and the FixedData for a peer TablixMember corresponding to another cell in the spanned area is false.
Visibility	0-1	Element	Indicates whether instances of this member should be hidden. If all instances of all submembers of a particular member instance are hidden, that member instance is automatically hidden[34]. See below for details on when hidden instances are replaced with a subtotal instance.

[34] This automatic hiding does not trigger cascaded hiding due to ToggleItem

<table>
<tr><td>HideIfNoRows</td><td>0-1</td><td>Boolean</td><td>Indicates whether this static member should be hidden if the Tablix contains no rows of data.
Ignored for dynamic members. A member hidden in this way will ignore Visibility properties (including ToggleItem).</td></tr>
<tr><td>KeepWithGroup</td><td>0-1</td><td>Enum</td><td>Indicates whether this static member should be kept on the page (if possible) with the closest non-hidden instance of the previous/following[35] sibling dynamic member.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>None</td><td>Default
Do not keep this member with another member.</td></tr>
<tr><td>Before</td><td>Keep this member with the previous dynamic member.</td></tr>
<tr><td>After</td><td>Keep this member with the following dynamic member.</td></tr>
</table>
Each sibling member between this member and the target dynamic member must have the same value for KeepWithGroup as this member.
Must be None on column members, dynamic members or members with dynamic descendants.</td></tr>
<tr><td>RepeatOnNewPage</td><td>0-1</td><td>Boolean</td><td>Indicates whether this static member should be repeated on every page on which appears at least one complete instance of the dynamic member referred to via KeepWithGroup or one of that member’s descendents (excepting descendents with KeepWithGroup not equal to None).
Ignored if KeepWithGroup = None.
Each peer member between this member and the target dynamic member must have the same value for RepeatOnNewPage as this member.
Must be False on column members.</td></tr>
<tr><td>DataElementName</td><td>0-1</td><td>String</td><td>The name to use for the data element for this member.
Must be a CLS-compliant identifier.
Default for dynamic members:
[Group.Name] Collection
Default for static members:
[TablixHeader.ReportItem.DataElementName]
(Null for static members with no header or a header with no report item.)</td></tr>
<tr><td>DataElementOutput</td><td>0-1</td><td>Enum</td><td>Indicates whether the member should appear in a</td></tr>
</table>

[35] The closest preceding or following dynamic grouping. If there is none, the setting is ignored.

<table>
<tr><td></td><td></td><td></td><td>data rendering.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Auto</td><td>Default
Behaves as Output for dynamic members and for static members with headers.
Behaves as ContentsOnly[36] for static members without headers.</td></tr>
<tr><td>Output</td><td>Indicates the member should appear in the output.</td></tr>
<tr><td>NoOutput</td><td>Indicates the member should not appear in the output.</td></tr>
</table></td></tr>
<tr><td>KeepTogether</td><td>0-1</td><td>Boolean</td><td>Indicates the entire Tablix member should be kept together on one page if possible.</td></tr>
</table>

Automatic Subtotals and Subgroups

If a dynamic member has toggleable visibility, an automatic subtotal is calculated for that member unless it has peer members without conditional or toggleable visibility[37]. If all of the member's instances and all of its siblings are hidden, the subtotal values are displayed instead. Note: All static submembers are preserved in the subtotal. Headers of dynamic members hidden in this way are merged in with their parents unless prevented from doing so by the existence of a peer static member or because it is the outermost member on the axis. In this case, the dynamic member is converted into a static member whose header is a rectangle which has its style properties copied from the original dynamic member's header. Note: Part of the space taken by such a header will be merged in with its parent if it has only dynamic peer members but one or more of them have static descendants which prevent full merging.

Consider the following tablix:

<table>
<tr><td></td><td colspan="2">=Region</td></tr>
<tr><td></td><td colspan="2">=Year</td></tr>
<tr><td></td><td>=Quarter</td><td>Growth</td></tr>
<tr><td>=Product</td><td>=Sum(Sales)</td><td>=Avg(Growth)</td></tr>
</table>

The Region and Year members generate automatic subtotals because they have no peer members, whereas the Quarter member does not.

When expanded with data, the tablix structure would look something like this:

[36] See ReportItem.DataElementOutput for a description of ContentsOnly

[37] In other words, an automatic subtotal is created in cases where it is possible for everything to disappear, leaving the member instance collection empty.

	West						East					
	2003			2004			2003			2004		
	Q3	Q4	Growth	Q1	Q2	Growth	Q3	Q4	Growth	Q1	Q2	Growth
Table	2	2	0%	2	4	50%	2	2	0%	2	4	50%
Chair	4	8	100%	8	8	0%	4	8	100%	8	8	0%

Toggling the quarter member instances would generate this:

	West		East	
	2003	2004	2003	2004
	Growth	Growth	Growth	Growth
Table	0%	50%	0%	50%
Chair	100%	0%	100%	0%

Toggling the West year member instances would generate this:

	West	East	
		2003	2004
	Growth	Growth	Growth
Table	25%	0%	50%
Chair	50%	100%	0%

Note: If a tablix cell contains an aggregate that uses an explicit group scope contained within the scope of the aggregate, that scope will be automatically adjusted to equal the current scope. For example, if a text box in a tablix cell contains the expression =Sum(Fields!Sales.Value)/Sum(Fields!Sales.Value,"State") and this subtotal is applied at the Country group scope, the expression will evaluate as =Sum(Fields!Sales.Value)/Sum(Fields!Sales.Value,"Country"), because the State group scope is contained within the country group scope.

TablixHeader

The TablixHeader element defines the ReportItem to use as the header for the group.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Size	1	Size	The height (if this is a column member) or width (if this is a row member) of the group header.
CellContents	1	Element	Page breaks in this ReportItem are ignored.

CellContents

The CellContents element defines the report item contained in a body, header or corner cell of a Tablix.

Attributes/Elements

Name	*Card*	*Type*	*Description*
ReportItem	0-1	Element	The Top, Left, Height and Width for this ReportItem are ignored. The position is taken to be 0, 0 and the size to be 100%, 100%.
ColSpan	0-1	Integer	Indicates the number of columns this cell spans[38] Default: 1 Must be 1 for CellContents inside TablixCell unless all column groups are static between the spanned cells and the common ancestor of those cells. Ignored for CellContents inside TablixHeader.
RowSpan	0-1	Integer	Indicates the number of rows this cell spans Default: 1 Must be 1 for CellContents inside TablixCell. Ignored for CellContents inside TablixHeader.

[38] For the purposes of hidden columns, this cell is considered to occupy the first visible column it appears in. Hiding a spanned column will reduce the effective number of columns spanned. The cell will remain visible, spanning the nonhidden columns, unless all spanned columns are hidden.

TablixBody

The TablixBody element defines the layout and structure of the bottom right region that contains the data elements of the Tablix.

Attributes/Elements

Name	*Card*	*Type*	*Description*
TablixColumns	1	Element	The list of columns in the body section of the Tablix.
TablixRows	1	Element	The list of rows in the body section of the Tablix.

TablixColumns

The TablixColumns element defines the set of columns in the body section of a Tablix.

Attributes/Elements

Name	*Card*	*Type*	*Description*
TablixColumn	1-N	Element	The list of columns in the body section of the Tablix. There must be as many TablixColumn elements as there are leaf-node (that is, has no sub-groups) TablixMembers in TablixColumnHierarchy.

TablixColumn

The TablixColumn element defines a column in the body section of a Tablix.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Width	1	Size	Width of each cell in this column

TablixRows

The TablixRows element defines the list of rows in the body section of a Tablix.

Attributes/Elements

Name	*Card*	*Type*	*Description*
TablixRow	1-N	Element	The list of rows in the body section of the Tablix. There must be as many TablixRow elements as there are leaf-node (that is, has no sub-groups) TablixMembers in TablixRowHierarchy.

TablixRow

The TablixRow element defines a list of cells in a row of the body section of a Tablix.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Height	1	Size	Height of each cell in this row
TablixCells	1	Element	The list of cells in a row in the detail section of the Tablix.

TablixCells

The TablixCells element defines the list of cells in a row of the body section of a Tablix.

Attributes/Elements

Name	*Card*	*Type*	*Description*
TablixCell	1-N	Element	The list of cells in a row in the body section of the Tablix. There must be as many TablixCell elements as there are leaf-node (that is, has no sub-groups) TablixMembers in TablixColumnHierarchy.

TablixCell

The TablixCell element defines the contents of each cell in the body section of a Tablix.

Attributes/Elements

<table>
<tr><th>Name</th><th>Card</th><th>Type</th><th>Description</th></tr>
<tr><td>CellContents</td><td>0-1</td><td>Element</td><td>Page breaks in this ReportItem are ignored if the tablix has any row headers, any dynamic column groups or more than one column.
Must be omitted if the position in the body is covered by a span from another cell.
Required otherwise.</td></tr>
<tr><td>DataElementName</td><td>0-1</td><td>String</td><td>The name to use for the cell element.
Default: “Cell”
Must be a CLS-compliant identifier.
Ignored if CellContents is omitted.</td></tr>
<tr><td>DataElementOutput</td><td>0-1</td><td>Enum</td><td>Indicates whether the cell contents should appear in a data rendering.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>ContentsOnly</td><td>Default
Indicates the cell should not appear in the output, but its contents should be rendered as if they were in the cell’s container.</td></tr>
<tr><td>Output</td><td>Indicates the cell should appear in the output.</td></tr>
<tr><td>NoOutput</td><td>Indicates the cell should not appear in the output.</td></tr>
</table>
Ignored if CellContents is omitted.</td></tr>
</table>

Note: For the purposes of Visibility.ToggleItem, a TablixCell is considered to be in the same scope as the Tablix. This means report items contained within a TablixCell cannot have their visibility toggled by report items in the tablix row or column headers.

Style

The Style element contains information about the style of a report item.
Where possible, the style property names and values match standard HTML/CSS properties.

All expression Style elements evaluate to either the type listed or to *Nothing* (an error in evaluation is treated as *Nothing*). *Nothing* is the same as not specifying the style, thereby indicating it should use the default. The expression must evaluate to a Boolean for Boolean properties, an enum value for enum properties, an integer for integer properties and a (formatted) string for all other properties. See the Data Types section earlier in this document for the formats for each string type.

Attributes/Elements

<table>
<tr><th>Name</th><th>Card</th><th>Type</th><th>Description</th></tr>
<tr><td>Border</td><td>0-1</td><td>Element</td><td>Default border properties.</td></tr>
<tr><td>TopBorder</td><td>0-1</td><td>Element</td><td>Properties of the top border.</td></tr>
<tr><td>BottomBorder</td><td>0-1</td><td>Element</td><td>Properties of the bottom border.</td></tr>
<tr><td>LeftBorder</td><td>0-1</td><td>Element</td><td>Properties of the left border.</td></tr>
<tr><td>RightBorder</td><td>0-1</td><td>Element</td><td>Properties of the right border.</td></tr>
<tr><td>BackgroundColor</td><td>0-1</td><td>Expression (Color)</td><td>Color of the background.
If omitted, the background is transparent.</td></tr>
<tr><td>BackgroundGradientType</td><td>0-1</td><td>Expression (Enum)</td><td>The type of background gradient.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Default</td><td>Default
Treated as None</td></tr>
<tr><td>None</td><td>No gradient</td></tr>
<tr><td>LeftRight</td><td>Horizontal gradient</td></tr>
<tr><td>TopBottom</td><td>Vertical gradient</td></tr>
<tr><td>Center</td><td>Circular gradient</td></tr>
<tr><td>DiagonalLeft</td><td>Diagonal left-to-right gradient</td></tr>
<tr><td>DiagonalRight</td><td>Diagonal right-to-left gradient</td></tr>
<tr><td>HorizontalCenter</td><td>Center-out horizontal gradient</td></tr>
<tr><td>VerticalCenter</td><td>Center-out vertical gradient</td></tr>
</table></td></tr>
<tr><td>BackgroundGradientEndColor</td><td>0-1</td><td>Expression (Color)</td><td>End color for the background gradient.</td></tr>
<tr><td>BackgroundImage</td><td>0-1</td><td>Element</td><td>A background image for the object.
If omitted, there is no background image.</td></tr>
<tr><td>FontStyle</td><td>0-1</td><td>Expression (Enum)</td><td>Font style
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Default</td><td>Default
Treated as Normal</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>Normal</td><td>Non-Italic font</td></tr>
<tr><td>Italic</td><td>Italic font</td></tr>
</table>
</td></tr>
<tr><td>FontFamily</td><td>0-1</td><td>Expression (String)</td><td>Name of the font family.
Default: Arial</td></tr>
<tr><td>FontSize</td><td>0-1</td><td>Expression (Size)</td><td>Point size of the font.
Default: 10 pt. Min: 1 pt. Max: 200 pt.</td></tr>
<tr><td>FontWeight</td><td>0-1</td><td>Expression (Enum)</td><td>Thickness of the font.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Default</td><td>Default
Treated as Normal</td></tr>
<tr><td>Thin</td><td>Matches CSS font weight 100</td></tr>
<tr><td>ExtraLight</td><td>Matches CSS font weight 200</td></tr>
<tr><td>Light</td><td>Matches CSS font weight 300</td></tr>
<tr><td>Normal</td><td>Matches CSS font weight 400</td></tr>
<tr><td>Medium</td><td>Matches CSS font weight 500</td></tr>
<tr><td>SemiBold</td><td>Matches CSS font weight 600</td></tr>
<tr><td>Bold</td><td>Matches CSS font weight 700</td></tr>
<tr><td>ExtraBold</td><td>Matches CSS font weight 800</td></tr>
<tr><td>Heavy</td><td>Matches CSS font weight 900</td></tr>
</table>
</td></tr>
<tr><td>Format</td><td>0-1</td><td>Expression (String)</td><td>.NET Framework formatting string[39].
Note: Locale-dependent currency formatting (format code “C”) is based on the language setting for the report item.
Locale-dependent date formatting is supported and should be based on the language property of the ReportItem.
Default: No formatting.</td></tr>
<tr><td>TextDecoration</td><td>0-1</td><td>Expression (Enum)</td><td>Special text formatting.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Default</td><td>Default
Treated as None</td></tr>
<tr><td>None</td><td>No text decoration</td></tr>
<tr><td>Underline</td><td>Underline the text</td></tr>
<tr><td>Overline</td><td>Overline the text</td></tr>
<tr><td>LineThrough</td><td>Strike through the text</td></tr>
</table>
</td></tr>
<tr><td>TextAlign</td><td>0-1</td><td>Expression (Enum)</td><td>Horizontal alignment of the text.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Default</td><td>Default
Treated as General</td></tr>
<tr><td>General</td><td>Text alignment is derived from the</td></tr>
</table>
</td></tr>
</table>

[39] See http://msdn2.microsoft.com/en-us/library/fbxft59x.aspx

<table>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td></td><td>data type and Direction</td></tr>
<tr><td>Left</td><td>Left aligned text</td></tr>
<tr><td>Center</td><td>Centered text</td></tr>
<tr><td>Right</td><td>Right aligned text</td></tr>
</table>
</td></tr>
<tr><td>VerticalAlign</td><td>0-1</td><td>Expression (Enum)</td><td>Vertical alignment of the text.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Default</td><td>Default
Treated as Top</td></tr>
<tr><td>Top</td><td>Top aligned text</td></tr>
<tr><td>Middle</td><td>Vertically centered text</td></tr>
<tr><td>Bottom</td><td>Bottom aligned text</td></tr>
</table>
</td></tr>
<tr><td>Color</td><td>0-1</td><td>Expression (Color)</td><td>The foreground color.
Default: Black (except within ChartDataPoint and ChartSeries, where the default is to use the palette colors).</td></tr>
<tr><td>PaddingLeft</td><td>0-1</td><td>Expression (Size)</td><td>Padding between the left edge of the report item and its contents[40].
Default: 0 pt. Max: 1000 pt.</td></tr>
<tr><td>PaddingRight</td><td>0-1</td><td>Expression (Size)</td><td>Padding between the right edge of the report item and its contents.
Default: 0 pt. Max: 1000 pt.</td></tr>
<tr><td>PaddingTop</td><td>0-1</td><td>Expression (Size)</td><td>Padding between the top edge of the report item and its contents.
Default: 0 pt. Max: 1000 pt.</td></tr>
<tr><td>PaddingBottom</td><td>0-1</td><td>Expression (Size)</td><td>Padding between the bottom edge of the report item and its contents.
Default: 0 pt. Max: 1000 pt</td></tr>
<tr><td>LineHeight</td><td>0-1</td><td>Expression (Size)</td><td>Height of a line of text.
Default: Renderer determines line height based on font size.
Min: 1 pt. Max: 1000 pt.</td></tr>
<tr><td>Direction</td><td>0-1</td><td>Expression (Enum)</td><td>Indicates whether text is written left-to-right or right-to-left.
Does not affect the alignment of text unless using General alignment.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Default</td><td>Default
Treated as LTR</td></tr>
<tr><td>LTR</td><td>Left-to-right text</td></tr>
<tr><td>RTL</td><td>Right-to-left text</td></tr>
</table>
</td></tr>
</table>

[40] As both borders and padding are measured from the edge of the object, borders may overlap report item contents unless there is sufficient padding.

<table>
<tr><td>WritingMode</td><td>0-1</td><td>Expression (Enum)</td><td>Indicates whether text is written horizontally or vertically.

<table><tr><th>Value</th><th>Description</th></tr><tr><td>Default</td><td>Default
Treated as Horizontal</td></tr><tr><td>Horizontal</td><td>Horizontal text</td></tr><tr><td>Vertical</td><td>Vertical text – For east Asian text, characters are not rotated. For other text, the entire text is rotated 90 degrees.</td></tr></table></td></tr>
<tr><td>Language</td><td>0-1</td><td>Expression (Language)</td><td>The primary language of the text. Default is Report language.
Used for text formatting operations for: Textbox.Value, DataLabel.Value, ChartMember.Label and DataValue.Value</td></tr>
<tr><td>Calendar</td><td>0-1</td><td>Expression (Enum)</td><td>Indicates the calendar to use for formatting dates. Must be compatible in .NET Framework with the Language setting.

<table><tr><th>Value</th><th>Description</th></tr><tr><td>Default</td><td>Default
Uses the .NET Framework default calendar for the Language of the report item.</td></tr><tr><td>Gregorian</td><td>Gregorian</td></tr><tr><td>GregorianArabic</td><td>Gregorian Arabic</td></tr><tr><td>Gregorian MiddleEastFrench</td><td>Gregorian Middle East French</td></tr><tr><td>GregorianTransliteratedEnglish</td><td>Gregorian Transliterated English</td></tr><tr><td>GregorianTransliteratedFrench</td><td>Gregorian Transliterated French</td></tr><tr><td>GregorianUSEnglish</td><td>Gregorian US English</td></tr><tr><td>Hebrew</td><td>Hebrew</td></tr><tr><td>Hijri</td><td>Hijri</td></tr><tr><td>Japanese</td><td>Japanese</td></tr><tr><td>Korean</td><td>Korean</td></tr><tr><td>Taiwan</td><td>Taiwan</td></tr><tr><td>ThaiBuddhist</td><td>Thai Buddhist</td></tr></table></td></tr>
<tr><td>NumeralLanguage</td><td>0-1</td><td>Expression</td><td>The digit format to use as described by its primary</td></tr>
</table>

		(Language)	language. Any language is valid. Default is the Language property.
NumeralVariant	0-1	Expression (Integer)	The variant of the digit format to use. Currently defined values are as follows: 1: default, follow Unicode context rules 2: 0123456789 3: traditional digits for the script as defined in GDI+. Currently supported for: ar | bn | bo | fa | gu | hi | kn | kok | lo | mr | ms | or | pa | sa | ta | te | th | ur and variants. 4: ko, ja, zh-CHS, zh-CHT only 5: ko, ja, zh-CHS, zh-CHT only 6: ko, ja, zh-CHS, zh-CHT only [Wide versions of regular digits] 7: ko only

<table>
<tr><td>TextEffect</td><td>0-1</td><td>Expression (Enum)</td><td>Effect to apply to the text

<table><tr><th>Value</th><th>Description</th></tr><tr><td>Default</td><td>Default
Treated as None</td></tr><tr><td>None</td><td>No special effect applied</td></tr><tr><td>Shadow</td><td>Shadow the text</td></tr><tr><td>Emboss</td><td>Emboss the text</td></tr><tr><td>Embed</td><td>Apply embedding effect to the text</td></tr><tr><td>Frame</td><td>Add a frame around the text</td></tr></table></td></tr>
<tr><td>BackgroundHatchType</td><td>0-1</td><td>Expression (Enum)</td><td>Hatch type. If not set to None, BackgroundGradientType is ignored.

<table><tr><th>Value</th><th>Description</th></tr><tr><td>Default</td><td>Default
Treated as None.</td></tr><tr><td>None</td><td>No background hatching.</td></tr><tr><td>BackwardDiagonal</td><td></td></tr><tr><td>Cross</td><td></td></tr><tr><td>DarkDownwardDiagonal</td><td></td></tr><tr><td>DarkHorizontal</td><td></td></tr><tr><td>DarkUpwardDiagonal</td><td></td></tr><tr><td>DarkVertical</td><td></td></tr><tr><td>DashedDownwardDiagonal</td><td></td></tr><tr><td>DashedHorizontal</td><td></td></tr><tr><td>DashedUpwardDiagonal</td><td></td></tr><tr><td>DashedVertical</td><td></td></tr><tr><td>DiagonalBrick</td><td></td></tr><tr><td>DiagonalCross</td><td></td></tr><tr><td>Divot</td><td></td></tr><tr><td>DottedDiamond</td><td></td></tr><tr><td>DottedGrid</td><td></td></tr><tr><td>ForwardDiagonal</td><td></td></tr><tr><td>Horizontal</td><td></td></tr><tr><td>HorizontalBrick</td><td></td></tr><tr><td>LargeCheckerBoard</td><td></td></tr><tr><td>LargeConfetti</td><td></td></tr><tr><td>LargeGrid</td><td></td></tr><tr><td>LightDownwardDiagonal</td><td></td></tr><tr><td>LightHorizontal</td><td></td></tr><tr><td>LightUpwardDiagonal</td><td></td></tr><tr><td>LightVertical</td><td></td></tr></table></td></tr>
</table>

			NarrowHorizontal NarrowVertical OutlinedDiamond Percent05 Percent10 Percent20 Percent25 Percent30 Percent40 Percent50 Percent60 Percent70 Percent75 Percent80 Percent90 Plaid Shingle SmallCheckerBoard SmallConfetti SmallGrid SolidDiamond Sphere Trellis Vertical Wave Weave WideDownwardDiagonal WideUpwardDiagonal ZigZag
ShadowColor	0-1	Expression (Color)	Color of the shadow for the item. Default: #0000007F
ShadowOffset	0-1	Expression (Size)	Size of the shadow for the item. Default: 0

Border

The default border properties for the object.

Attributes/Elements

Name	*Card*	*Type*	*Description*
Color	0-1	Expression (Color)	Color of the border (unless overridden for a specific side). Default: Black
Style	0-1	Expression (Enum)	Style of the border (unless overridden for a specific side). **Value** / **Description** Default / Default Treated as Solid within Line and None elsewhere. None / Do not draw a border. Dotted / Dotted line Dashed / Dashed line Solid / Solid line Double / Double solid line DashDot / Dash-dot line Allowed only within Chart DashDotDot / Dash-dot-dot line Allowed only within Chart
Width	0-1	Expression (Size)	Width of the border (unless overridden for a specific side) Borders are centered on the edge of the object. Default: 1 pt Max: 20 pt Min: 0.25 pt

TopBorder, BottomBorder, LeftBorder, RightBorder

Each of these elements contains the following properties:

Attributes/Elements

Name	*Card*	*Type*	*Description*
Color	0-1	Expression (Color)	If specified, overrides the default border color.
Style	0-1	Expression (Enum)	If not Default, overrides the default border style. **Value** / **Description** Default / Default Uses Border.Style None / Do not draw a border Dotted / Dotted line Dashed / Dashed line Solid / Solid line Double / Double solid line
Width	0-1	Expression (Size)	Max: 20 pt Min: 0.25 pt If specified, overrides the default border width.

BackgroundImage

Attributes/Elements

<table>
<tr><th>Name</th><th>Card</th><th>Type</th><th>Description</th></tr>
<tr><td>Source</td><td>1</td><td>Enum</td><td>Identifies the source of the image:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>External</td><td>The Value contains a constant or expression that evaluates to the location of the image. This can be a full folder path (for example, “/images/logo.gif”), relative path (for example, “logo.gif”) or URL (for example, “http://reportserver/images/logo.gif”). Relative paths start in the same folder as the report.</td></tr>
<tr><td>Embedded</td><td>The Value contains a constant or expression that evaluates to the name of an EmbeddedImage in the report</td></tr>
<tr><td>Database</td><td>The Value contains an expression (a field in the database) that evaluates to the binary data for the image.</td></tr>
</table></td></tr>
<tr><td>Value</td><td>1</td><td>Expression (String)</td><td>See Source. Expected datatype is string or binary, depending on Source. If the Value is null, no background image is displayed.</td></tr>
<tr><td>MIMEType</td><td>0-1</td><td>Expression (String)</td><td>The MIMEType for the image.
Valid values are: image/bmp, image/jpeg, image/gif, image/png, image/x-png
Required if Source = Database. Ignored otherwise.</td></tr>
<tr><td>BackgroundRepeat</td><td>0-1</td><td>Expression (Enum)</td><td>Indicates how the background image should fill the available space.
Default is treated as Fit within Chart and Repeat elsewhere (unless another default behavior is specified in the definition of the containing element).
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Default</td><td>Default
Treated as Fit within Chart and Repeat elsewhere</td></tr>
<tr><td>Repeat</td><td>Repeat the image both horizontally and vertically to fill the space.</td></tr>
<tr><td>RepeatX</td><td>Repeat the image horizontally to fill the space.
Not allowed within Chart</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>RepeatY</td><td>Repeat the image horizontally to fill the space.
Not allowed within Chart</td></tr>
<tr><td>Fit</td><td>Stretch the image to fill the space.
Allowed only within Chart.</td></tr>
<tr><td>Clip</td><td>Clip the image to the available space.</td></tr>
</table>
</td></tr>
<tr><td>TransparentColor</td><td>0-1</td><td>Expression (Color)</td><td>Defines a color to treat as transparent in the background image.
Used only for background images within Chart.</td></tr>
<tr><td>Position</td><td>0-1</td><td>Expression (Enum)</td><td>Indicates where a background image with BackgroundRepeat = Clip should be drawn.
Used only for background images within Chart.
.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Default</td><td>Default
Treated as Center</td></tr>
<tr><td>Top</td><td>Draw image at top center</td></tr>
<tr><td>TopLeft</td><td>Draw image at top left</td></tr>
<tr><td>TopRight</td><td>Draw image at top right</td></tr>
<tr><td>Left</td><td>Draw image at left center</td></tr>
<tr><td>Center</td><td>Draw image in center</td></tr>
<tr><td>Right</td><td>Draw image at right center</td></tr>
<tr><td>BottomRight</td><td>Draw image at bottom right</td></tr>
<tr><td>Bottom</td><td>Draw image at bottom center</td></tr>
<tr><td>BottomLeft</td><td>Draw image at bottom left</td></tr>
</table>
</td></tr>
</table>

Style Properties and ReportItem Types

The following table describes which style properties apply to which types of ReportItems.

	Line	Rectangle [41]	Textbox	Paragraph	TextRun	Image	SubReport [42]	Tablix	Chart [43]	Body	Page Section	Page
Border	X[44]	X	X			X	X	X	X	X	X	X
{Top|Bottom|Left|Right} Border	X	X	X			X	X	X	X	X	X	X
BackgroundColor		X	X					X	X	X	X	X
BackgroundGradient {Type,EndColor}									X			
BackgroundImage		X	X					X		X	X	X
FontStyle					X				X			
FontFamily					X				X			
FontSize					X				X			
FontWeight					X				X			
Format					X				X			
TextDecoration					X							
TextAlign				X								
VerticalAlign			X									
Color					X							
Padding {Left,Right,Top,Bottom}			X			X						
LineHeight				X								
Direction			X									
Language					X				X			
Calendar					X				X			
NumeralLanguage					X				X			
NumeralVariant					X				X			
WritingMode			X									

[41] This includes PageHeader and PageFooter

[42] All Textbox properties apply in the event of NoRowsMessage being used or for subreports that fail to execute.

[43] Different elements within the Chart support different Style properties. These are described in the places where Style is referenced in the element definition.

[44] The following border styles apply to Lines: Dotted | Dashed | Solid. All others (including None) are treated as Solid. Lines use only the Default property for BorderStyle, BorderWidth and BorderColor. The Top, Left, Bottom and Right properties are unused.

The following table describes which style properties apply to which subelements of Chart.

	Chart	Chart Area	Chart Border Skin	Chart Title	Chart Legend	Chart Legend Title	Chart Axis	Chart Axis Title	Chart Axis Scale Break	Chart Series	Chart Data Point	Chart Empty Points	Chart Data Label	Chart Marker	Chart Gridlines Chart Tickmarks	Chart Strip Line
Border	X	X	X	X	X	X	X		X	X[45]	X	X	X	Color only	X	X
{Top|Bottom| Left|Right} Border																
BackgroundColor	X	X	X	X	X	X				X	X	X	X	X		X
BackgroundGradient {Type,EndColor}	X	X	X	X	X	X				X	X	X	X			X
Background HatchType	X	X	X		X	X				X	X	X	X			X
BackgroundImage	X	X									X	X		X		X
TransparentColor	X	X												X		X
Background Repeat	X	X														X
Position	X	X														X
Font {Style,Family, Size,Weight}				X	X	X	X	X					X			X
Format				X	X	X	X	X					X			X
TextDecoration				X	X	X	X	X					X			X
TextAlign				X	X	X	X	X					X			X
VerticalAlign				X	X	X	X	X					X			X
Color				X	X	X	X	X				X	X			X
LineHeight																
Direction	X															
WritingMode																
Language	X															
Calendar	X															
NumeralLanguage	X															
NumeralVariant	X															
TextEffect				X												
ShadowColor		X		X	X					X						
ShadowOffset		X		X	X					X						
Padding {Left,Right,Top,Bottom}																

[45] Only applies for DerivedSeries

The following table describes which style properties apply to which subelements of GaugePanel.

	Gauge	Gauge Panel	Gauge Scale	Gauge Pointer	Scale Range	Thermo meter	Pointer Cap	Scale Labels	Custom Label	Custom TickMarks	Pin Label	Back Frame	Frame Background
Border		X	X	X	X					X		X	
{Top|Bottom| Left|Right} Border		X											
BackgroundColor		X	X	X	X	X	X			X		X	X
BackgroundGradient Type			X	X		X	X					X	X
BackgroundGradient EndColor			X	X	X	X	X					X	X
BackgroundHatchType			X	X	X	X	X					X	X
BackgroundImage													
TransparentColor													
BackgroundRepeat													
Position													
Font {Style,Family, Size,Weight}								X	X		X		
Format								X					
TextDecoration													
TextAlign													
VerticalAlign													
Color								X	X		X		
LineHeight													
Direction	X												
WritingMode													
Language	X												
Calendar	X												
NumeralLanguage	X												
NumeralVariant	X												
TextEffect													
ShadowColor													
ShadowOffset			X	X	X								
Padding{L,R,T,B}													

Expressions

Expression Syntax

All expressions in RDL begin with the character “=”and are defined in a Visual Basic compatible syntax (see http://msdn.microsoft.com/library/en-us/vbls7/html/vblrfvbspec9.asp).Values that do not begin with “=” will be treated as constants of the type expected by the property if that type is Boolean, String or Integer (see http://www.w3.org/TR/xmlschema-2/). For example, the Hidden property expects a Boolean, therefore strings *true* and *false* will be treated as Boolean constants.

For properties that take a Variant, all values that do not start with "=" will be treated as string constants.

Custom Code References

Custom assemblies that are referenced in the report are declared at the Report level via the CodeModules element.

Static methods in custom assemblies can be accessed globally within the report.

ClassName.MethodName(…)
For example: MyCurrencyConverterClass.Convert(…)

Instance-based methods are instantiated though the Classes element and accessed via a globally defined Code member.

Code.*InstanceName.MethodName*(…)
For example: Code.CurrencyConverter.Convert(…)

Built-in References

In addition to user-defined references, the following namespaces and classes are always available: Microsoft.VisualBasic, System.Convert, System.Math.

Data Types

Every expression used in an expression element or as an argument to an RDL function must return one of the following types:

RDL Type	CLR Types
String	String, Char, GUID, Timespan
Boolean	Boolean
Integer	Int16, Int32, Int64,Uint16, Uint32, Uint64, Byte, Sbyte
DateTime	DateTime, DateTimeOffset
Float	Single, Double, Decimal
Binary	Byte[]
Variant	Any of the above except Byte[]
VariantArray	Array of Variant

"Numeric" refers to either Integer or Float
"Scalar" refers to Integer, Float or DateTime
If an expression returns any other type or returns a type that is not permitted for that expression element, an error will be generated. In the case of Label, TextRun.Value and all Style expressions, expression errors are treated as warnings and null is returned instead. Note: For TextRun value and group labels, the FormattedValue property in the rendering object model will be "#Error". See the rendering object model spec for details. If an RDL type is defined as the return type for a function described below, the underlined CLR type is used (unless otherwise specified). Expressions that evaluate to DBNull will instead return null.

Error Handling

Errors that occur during expression evaluation fall into two categories: critical errors and non-critical errors. A non-critical error will not cause the report to fail to render but instead will register a warning and fall back to a default error handling behavior. For example, an error in TextRun.Value will result in the TextRun containing the string "#Error". Critical errors will result in the report (or currently requested page to fail to render, returning an error message instead.
Errors in evaluation of the following properties are considered critical errors: FilterExpression, FilterValue, GroupExpression, Variable.Value, Visibility.Hidden, SortExpression.Value.

Global Collections

There are several global object collections available within report expressions.

Collection	Description	Item Data Type
Fields	Fields in the current data set	Field
Parameters	Report parameters	Parameter
ReportItems	All textboxes in the report[46]	ReportItem
Globals	Global variables	Variant
User	User-specific data	Variant
DataSources	Data sources in the report[47]	DataSource
DataSets	Data sets the report	DataSet
Variables	Variables defined on the report or in groups	Variant

Items in the collections are accessed by name using standard Visual Basic collection syntax:
Collection!*ObjectName* or *Collection.Item*("*ObjectName*") or *Collection*("*ObjectName*")
For example: User!Language
Items in the Globals and User collection can also be accessed via property syntax:
Collection.ObjectName
For example: Globals.PageNumber
Circular references involving items in global collections are errors.

Globals

Members of the Globals collection are variants, but have known types.

Name	Type	Description
PageNumber	Integer	Current page number Available only in the page header/footer of the report.
TotalPages	Integer	Total number of pages in the report Available only in the page header/footer of the report.
ExecutionTime	DateTime	The date/time the report began executing. The value of Now() stored at the start of the report execution.
ReportServerUrl	String	URL to the report server. For example, http://reportserver/reports
ReportFolder	String	Full path on the report server to the folder containing the report. For example, for the report

[46] References to report items outside of the current (or any ancestor) scope are ambiguous and have an undefined value (which may be Nothing or an actual report item, depending on the context). The scope for expressions in page headers/footers is considered to be items on the current page.

[47] Only data sources and data sets used in the body of the report will be included in the DataSources and DataSets collections. Data sets and data sources used only in parameter valid values and default values properties will not be included.

		http://reportserver/reports/salesreports/budgeting/currentbudget, the ReportFolder is /salesreports/budgeting.
ReportName	String	Name of the report in the report catalog. For example, for the report http://reportserver/reports/salesreports/budgeting/currentbudget, the ReportName is currentbudget.

User

Members of the User collection are variants, but have known types.

Name	Type	Description
UserID	String	ID of the user executing the report
Language	String	Language ID of the client executing the report

Fields

The Field object has a set of predefined properties that can be accessed via either property syntax:

Fields!*FieldName*.*PropertyName*

For example: Fields!Region.BackgroundColor

Or collection syntax:

Fields!*FieldName*!*PropertyName*

Fields!*FieldName*(“*PropertyName*”)

Fields!*FieldName*.Properties(“*PropertyName*”)

When a report is executed, queries may return a different set of fields than were originally defined in the report. The IsMissing property indicates whether or not the field was found in the resulting data set. The Value property of missing fields is NULL.

In addition, data providers that support field properties can provide additional properties, which can be accessed only via collection syntax. If the data provider does not support the requested property or the field is not found when the query is executed, the default value returned is null for String and Object properties and 0 for Integer properties. Predefined field properties have types as specified in the following table. All other properties are of type Object.

Predefined Field Properties

Property	Type	Expected Values	Corresponding Analysis Services Property
Value	Object[48]		Member_Caption
IsMissing	Boolean		
UniqueName	String		Member_Unique_Name
BackgroundColor	String	See Style.BackgroundColor	Back_Color
Color	String	See Style.Color	Fore_Color
FontFamily	String	See Style.FontFamily	Font_Name
FontSize	String	See Style.FontSize	Font_Size
FontWeight	String	See Style.FontWeight	Font_Flags
FontStyle	String	See Style.FontStyle	Font_Flags
TextDecoration	String	See Style.TextDecoration	Font_Flags
FormattedValue	String		Formatted_Value
Key	Object		Member_Key
LevelNumber	Integer		Level_Number
ParentUniqueName	String		Parent_Unique_Name

Fields Collection in reports with multiple DataSets

When a report contains multiple data sets, there are multiple virtual Fields collections in the report.

Which of these is accessed via Fields depends on the context.

- Inside of an aggregate, if the scope argument refers to a data set, Fields refers to the fields in that data set.
- Within a data region, the Fields collection refers to the fields in the data set for that region.
- Outside of a data region, direct references to fields (outside an aggregate) are undefined.

Report Items

Only textboxes appear in the ReportItems collection.

The ReportItem object has a set of predefined properties that can be accessed via either property syntax or collection syntax (see Fields above). The only property currently defined for ReportItem is Value. The data type of Value is Object.

The value of the current ReportItem can be referenced in property expressions using Me.Value or simply Value. Neither Me.Value nor Value is supported inside Aggregate functions.

Where detectable during report validation (that is, simple references) the use of report items which are not defined at or above the current grouping scope (for example, peer or descendent

[48] Field values have CLR data types defined by the Data Provider API (with the exception that DBNull is returned as null).
See http://msdn.microsoft.com/library/en-us/cpref/html/frlrfsystemdataidatareaderclassgetschematabletopic.asp

grouping scopes) is not allowed. Where not detectable, the Value property of such references is null.

Parameters

The Parameter object has a set of predefined properties that can be accessed via either property syntax or collection syntax (see Fields above). The only properties currently defined for Parameter are Value, Label, Count and IsMultiValue. The label is determined by the value of the Label or LabelField element of the selected parameter value in the ValidValues list. If there is no ValidValues list for a parameter or no Label or LabelField specified for a Value, the Label is the same as the Value[49].

The data type of Value is Variant for single-value parameters and VariantArray for multi-value parameters. The data type of Label is String for single-value parameters and StringArray for multi-value parameters.

The IsMultiValue property is a Boolean and indicates whether the parameter is defined as multi-value. The Count property indicates the number of values and labels there are (1 if it is not a multi-value parameter). The Count property may be 0 if the parameter does not have a value which is valid (or values have not yet been supplied).

DataSources

The DataSource object has the following properties:

Type – Type of data provider for the data source
DataSourceReference – Path to the data source (Nothing for embedded data sources)

DataSets

The DataSet object has the following properties:

CommandText – The CommandText of the data set. If the CommandText is an expression, this is the result of evaluating the expression.
RewrittenCommandText – The CommandText of the data set after being handed back to the data extension for rewriting (Typically, this involves expanding the parameter values into constants in the query).

[49] Like all other language-dependent parameter operations, User.Language is used as the culture if a conversion to string is required.

Variables

The members of the Variables collection are taken from the Variables defined in the report. The Variable object has a set of predefined properties that can be accessed via either property syntax or collection syntax (see Fields above). The only property currently defined for Variable is Value. The data type of Value is Variant.
Where detectable during report validation (that is, simple references) the use of variables which are not defined at or above the current scope (for example, peer or descendent scopes) is not allowed. Where not detectable, the Value property of such references is null.

Restrictions on Use of Global Collections

The Fields, Parameters, ReportItems and Globals collections have restrictions on the contexts in which they can be used in expressions. The following table summarizes where these global collections can and cannot be used.

Context	Fields	ReportItems	Parameters	PageNumber TotalPages	DataSource DataSet	Variables
Page Header / Footer	Yes	At most one[50]	Yes	Yes	Yes	Yes
Body	Yes[51]	Only those in current or ancestor scope[52]	Yes	No	Yes	Yes
Report Parameter	No	No	Only earlier parameters[53]	No	No	No
Field	Yes	No	Yes	No	No	No
Query Parameter	No	No	Yes	No	No	No
Group Expression	Yes	No	Yes	No	Yes	No
Sort Expression	Yes	No	Yes	No	Yes	Yes[54]

[50] If a report item does not appear on a page, the value for that report item for that page is null. Expressions in page headers and footers should generally only use report items that will actually appear on each page of the report. If a report item is conditionally hidden on a page, it is not considered when evaluating the page header and footer when it is not visible.

[51] If an expression in a non-detail section refers to a field that is not in the group expression for the group (or any ancestor grouping scope), which specific value is used is not defined. Report designers should use the First() and Last() aggregate functions if they wish to guarantee which row of data is used. Note, however, that not using an aggregate function is perfectly reasonable if the semantics of the query and grouping result in there being only one value in all rows for the field in question

[52] Expressions that refer to report items can only refer to values of peer report items (those in the same grouping scope) or report items in any containing (ancestor) grouping scope. Although using a report item value in a child grouping scope in an aggregate function is meaningful, it is not currently supported.

[53] For parameter properties which may be evaluated prior to any user selection of parameter values, the earlier parameter values may be null. For example, consider a Region parameter with Prompt expression =iif(Parameters!Country.Value=”USA”,”State”,”Region”). If the Country parameter has no default value, the prompt for the region parameter will be initially “Region” since Parameters!Country.Value will evaluate to null until the user selects a value.

[54] Member sorts only. Not allowed in DataRegion.SortExpressions

Code	No	No	Yes[55]	No	No	No
Report.Language	No	No	Yes	No	No	No
Variables	Yes	No	Yes	No	Yes	Current or ancestor scope
Aggregates	Yes	Only in page header/footer	Yes	Only in report item aggregates	Yes	No

Note: Since references to items in global collections can be dynamic (for example, "=ReportItems(Parameters!Param1.Value)"), all error checking must occur both during report publishing (to catch static disallowed references) and during report execution (to catch dynamic disallowed references).

Cyclic expressions (for example, TextBox1=TextBox2+1; TextBox2=TextBox1+1) are not allowed, but will only be caught at report execution time.

[55] If methods defined in <Code> are used within Parameter initialization, the Parameters collection will be empty, as it has not yet been initialized.

Aggregate Functions

RDL supports the following list of standard aggregate functions:

Function	***Arguments***	***Type***[56]	***Description***
Sum	*Return*[57]	Float	Returns the sum of all values of the expression within the scope. Return type is decimal for decimal expressions and double for all other expressions.
	Expression	Numeric	The expression to aggregate. Cannot contain any aggregate functions.
	Scope	String	Name of a DataSet or the name of a Group or DataRegion that contains (directly or indirectly) the report item that the aggregate function is used in. Indicates the aggregate should apply to the entire data set, all of the data in the current group, or all of the data in the current data region. May only be a constant, not an expression. See notes on Scope later in this document.
	Recursive	Enum	Recursive | Simple (Default). Indicates whether the aggregate should be calculated recursively. Optional. See notes on Recursive later in this document.
Avg	*Return*	Float	Returns the average of all nonnull values of the expression within the scope. See Sum regarding return type.
	Expression	Numeric	See Sum
	Scope	String	See Sum
	Recursive	Enum	See Sum
Max	*Return*	Variant	Returns the maximum of all nonnull values of the expression within the scope. Return type is the same as the expression type.
	Expression	Variant	See Sum
	Scope	String	See Sum
	Recursive	Enum	See Sum

[56] For all aggregates other than First, Last and Count, the data type of the aggregated expression is expected to be fixed. If values (other than null) are encountered of multiple data types, it is an error.

[57] All aggregates other than Count and CountDistinct return null if there is not enough data to aggregate (0 or 1 rows for StDev and Var, 0 rows for all others)". Count and CountDistinct return 0 if there is no data to aggregate. This includes aggregates in the page header or footer that do not refer to any report items.

Min	*Return*	Variant	Returns the minimum of all nonnull values of the expression within the scope. Return type is the same as the expression type.
	Expression	Variant	See Sum
	Scope	String	See Sum
	Recursive	Enum	See Sum
Count	*Return*	Integer	Returns the count of all nonnull values of the expression within the scope.
	Expression	Variant or Binary	See Sum
	Scope	String	See Sum
	Recursive	Enum	See Sum
CountDistinct	*Return*	Integer	Returns the count of all distinct nonnull values of the expression within the scope.
	Expression	Variant	See Sum
	Scope	String	See Sum
	Recursive	Enum	See Sum
CountRows	*Return*	Integer	Returns the count of all rows within the scope. Syntax: "CountRows(*Scope*)"
	Scope	String	See Sum
	Recursive	Enum	See Sum
StDev	*Return*	Float	Returns the standard deviation of all nonnull values of the expression within the scope.
	Expression	Numeric	See Sum
	Scope	String	See Sum
	Recursive	Enum	See Sum
StDevP	*Return*	Float	Returns the population standard deviation of all nonnull values of the expression within the scope Return type is the same as the expression type.
	Expression	Numeric	See Sum
	Scope	String	See Sum
	Recursive	Enum	See Sum
Var	*Return*	Float	Returns the variance of all nonnull values of the expression within the scope. See Sum regarding return type.
	Expression	Numeric	See Sum
	Scope	String	See Sum
	Recursive	Enum	See Sum
VarP	*Return*	Float	Returns the population variance of all nonnull values of the expression within the scope. See Sum
	Expression	Numeric	See Sum
	Scope	String	See Sum
	Recursive	Enum	See Sum

In addition, RDL supports the following list of advanced aggregate functions:

Function	***Arguments***	***Type***[58]	***Description***
First	*Return*	Variant or Binary	Returns the first value of the expression within the scope (after all sorting up through the Scope has been applied) Return type is the same as the expression type.
	Expression	Variant or Binary	See Sum
	Scope	String	See Sum
Last	*Return*	Variant or Binary	Returns the last value of the expression within the scope (after all sorting up through the Scope has been applied) Return type is the same as the expression type.
	Expression	Variant or Binary	See Sum
	Scope	String	See Sum
Previous	*Return*	Variant or Binary	Returns the value of the expression for the previous instance of the PreviousScope[59] or (if the expression is an aggregate) the value of the aggregate expression as applied to the previous instance of the PreviousScope corresponding[60] to the current instance of the Scope of the aggregate function. Returns Nothing if there is no corresponding previous instance.

[58] For all aggregates other than First, Last and Count, the data type of the aggregated expression is expected to be fixed. If values (other than null) are encountered of multiple data types, it is an error.

[59] For non-aggregate expressions where the PreviousScope is not Nothing, the value for the corresponding row in the previous instance of PreviousScope is returned.

[60] In cases where there are grouping scopes in between the PreviousScope and the aggregate's Scope, all instances must match to be considered "corresponding". For example. Previous(Sum(Fields!Sales.Value,"Day"),"Year"), should return the total sales for the same Day and Month in the previous Year. If the aggregate is within a detail scope, "corresponding" is defined by row number within the containing scope.

	Expression	Variant or Binary	The expression for which to retrieve the previous value. If the expression contains an aggregate, Previous aggregates the data within the previous instance of the PreviousScope that corresponds to the current instance of the aggregate's Scope. An aggregate Scope equal to the PreviousScope indicates all of the data in the PreviousScope should be aggregated. The scope of the aggregate must be contained by (or equal to) PreviousScope. The aggregate function cannot be Aggregate or Previous. The aggregate may not be recursive. The functions Level() and InScope may not be used in the expression.
	PreviousScope	String	Name of a Group or DataRegion that contains (directly or indirectly) the report item that the aggregate function is used in. Previous retrieves the data in the previous instance of the group/data region. A PreviousScope of Nothing indicates Previous should retrieve the value of the expression for the previous detail row of data. May only be a constant, not an expression. Optional. Default: The current scope (Nothing if in a detail scope).
RunningValue	*Return*	See Function	A running aggregate of the expression, using the specified aggregate function.
	Expression	See Function	The expression to aggregate. Cannot contain any aggregate functions.
	Function	Enum	Name of an aggregate function for which to calculate a running value (Cannot be CountRows, RunningValue, RowNumber or Aggregate). Expression type and Return type are determined by the aggregate function used.

	Scope	String	Name of a Group or DataRegion that contains (directly or indirectly) the report item that the aggregate function is used in. Indicates the running value resets whenever the group expression changes or resets with each new instance of the data region. A value of Nothing indicates the running value never resets. May only be a constant, not an expression.
RowNumber	*Return*	Integer	The row number of the current row or group instance.
	Scope	String	See RunningValue
Aggregate	*Return*	Determined by data provider (see Field.Value)	Calculates a custom (data provider defined) aggregate for the expression at the given scope. If the data provider does not support this function or if the data is not available for the given expression or scope, Nothing is returned.
	Expression	N/A	The expression to aggregate. Must be a simple field reference (for example, =Aggregate(Fields!Sales.Value,*Year*))
	Scope	String	See Sum All group expressions for the Scope (and all containing group scopes) must be simple field references or (non-expression) constants.

Scope

Scope may only be a constant, not an expression.

For expressions inside data regions:

Within a data region, the Scope argument is optional for all aggregates other than RunningValue and RowNumber.

If omitted, the scope is the innermost scope containing the report item in which the aggregate is used[61].*For expressions outside of data regions (in the report body):*

When used outside of a data region, the scope argument can only refer to a data set name.

If there exists more than one data set in the report, the Scope argument is required.

If there exists exactly one data set in the report, the Scope argument is optional.

If omitted, the scope is the only data set in the report.

Aggregates are not allowed if there are no data sets.

For expressions in page headers and footers:

If the scope argument is omitted in page headers/footers, the scope is the data on the current page. In this case, report items can be used in the aggregate expression, but fields cannot be. If a scope is specified, however, fields can be used but report items cannot be (this is identical to an aggregate in the body of the report).

[61] For aggregates in a TablixCell, the default scope is the cell itself (the intersection of the innermost row scope and innermost column scope).

Recursive
Recursive indicates that the aggregate should apply to all data in the current instance of the given scope and all descendant instances of the current instance. Recursive is ignored if the scope has no Parent property.

For example:

EmployeeID	ManagerID	Sales	AllSales
1	NULL	10	70
1a	1	10	30
1a1	1a	10	10
1a2	1a	10	10
1b	1	10	30
1b1	1b	10	10
1b2	1b	10	10

The data above would be generated by this (simplified) RDL snippet:

```
<Group Name="Employee">
       <GroupExpressions>
              <GroupExpression>=Fields!EmployeeID.Value</GroupExpression>
       </GroupExpressions>
       <Parent>=Fields!ManagerID.Value</Parent>
</Group>
<ReportItems>
       <Textbox>
       <Paragaphs><Paragraph><TextRuns>
       <TextRun><Value>=Fields!EmployeeID.Value</Value></TextRun>
       </TextRuns></Paragraph></Paragraphs>
       </Textbox>
       <Textbox>
       <Paragaphs><Paragraph><TextRuns>
       <TextRun><Value>=Fields!ManagerID.Value</Value></TextRun>
       </TextRuns></Paragraph></Paragraphs>
       </Textbox>
       <Textbox>
       <Paragaphs><Paragraph><TextRuns>
       <TextRun><Value>=Sum(Fields!Sales.Value)</Value>
       </TextRun></TextRuns></Paragraph>
       </Paragraphs>
       </Textbox>
       <Textbox><Paragaphs><Paragraph><TextRuns>
       <TextRun>
       <Value>=Sum(Fields!EmployeeID,"Employee",Recursive)</Value>
       </TextRun>
       </TextRuns></Paragraph></Paragraphs>
       </Textbox>
</ReportItems>
```

Recursive Depth
In recursive hierarchies, the function Level can be used to determine the current depth of the recursive hierarchy.

Function	*Arguments*	*Type*[62]	*Description*
Level	*Return*	Integer	A zero-based integer representing the current depth level of a recursive hierarchy. If the specified scope is a dataset, data region or group without Parent or the scope does not exist, Level returns 0.
	Scope	String	Optional. Defaults to the current scope.

Restrictions on Aggregate Usage

Context	Running Value	Row Number	First / Last	Previous	Other Aggregates	ReportItem Aggregates
PageHeader/Footer	No	No	Yes	No	Yes	Yes
Body	Yes	Yes	Yes	Yes	Yes	No
ReportParameter	No	No	No	No	No	No
Calculated Field	No	No	No	No	No	No
Query Parameter	No	No	No	No	No	No
Group Expression	No	Yes	No	No	No	No
Sort Expression	No	No	No	No	Yes[63]	No
Tablix Cell	Yes[64]	Yes	Yes	Yes	Yes	No
Group Variables	No	No	No	No	Yes	No
Report Variables	No	No	No	No	Yes	No

Filtering and Aggregates

- Aggregates using data set scopes are applied after the data set filter (if any) is applied to the data.
- Aggregates using data region scopes are applied after the data region filter (if any) is applied to the data.
- Group filters are ignored for the purposes of calculating aggregates.
- The aggregate function "Aggregate" cannot be used in a report that contains any Filter elements.
- First, Last, Previous, RunningValue and RowNumber are applied after containing data region and group filters are applied.

[62] For all aggregates other than First, Last and Count, the data type of the aggregated expression is expected to be fixed. If values (other than null) are encountered of multiple data types, it is an error.

[63] Aggregates are not allowed in sort expressions on DataRegion.

[64] Within a TablixCell in the scope of both a dynamic row and a dynamic column of the Tablix, the Scope argument for RunningValue and RowNumber must refer to either a Column Grouping or a Row Grouping for the Tablix or to a scope contained within this TablixCell. The scope of the running value/row number defines the direction of the running value. HideDuplicates in a TablixCell has the same restrictions/behavior. RunningValue, RowNumber and HideDuplicates in cells of a tablix may use either column scopes or row scopes but not both.

Dynamic Scoping

Report items contained within a cell of a tablix with automatic subtotals (due to drilldown) have dynamic scoping.

For example, consider a tablix that has a year column group and a product row group. If the value of the text box in the detail cell is =Sum(Fields!Sales.Value), each detail cell will be grouped on both year and product. However, the year subtotal (shown when the year group is hidden) will only be grouped on product and the product subtotal will only be grouped on year (and the grand total will not be grouped on either).

The function InScope can be used to determine what the current instance is being grouped on:

Function	***Arguments***	***Type***	***Description***
InScope	*Return*	Boolean	True if the current instance is within the specified scope.
	Scope	String	Name of a DataSet, Group or DataRegion.

A typical usage for the InScope function is to construct links to drill-through reports which will work in both tablix details cells and automatic subtotal cells.

For example:

```
<Drillthrough>
        <ReportName>=iif(InScope("Month"),"Transactions","ProductTotByYear")</ReportName>
        <Parameters>
                <Parameter Name=Year>
                        <Value>=Fields!Year</Value>
                        <Omit>=Not(InScope("Year"))</Omit>
                </Parameter>
                <Parameter Name=Month>
                        <Value>=Fields!Month</Value>
                        <Omit>=Not(InScope("Month"))</Omit>
                </Parameter>
                <Parameter Name=Product>
                        <Value>=Fields!Product</Value>
                        <Omit>=Not(InScope("Product"))</Omit>
                </Parameter>
        </Parameters>
</Drillthrough>
```

Semantic Query Drillthrough

For reports with semantic queries that utilize automatic drillthrough query rewriting, the following function is available to generate a default drill-through context:

Function	***Arguments***	***Type***	***Description***
CreateDrillthroughContext	*Return*	String	A DrillthroughContext parameter value that describes the current drillthrough context, including semantic query fields from the current scope referenced in the value property of the current object (textbox, image or chart data point) and semantic query field values for the current group scopes.

This function can only be used in a drillthrough parameter value expression.

Questions & Answers (FAQ)

Q: How do I create a page break on a specific number of rows?
A: Use a Group with a PageBreak property set to Between. For your group expression, use a count running value to derive a unique group value for each block of N rows. (for example, = Ceiling(RowNumber(Nothing)/20))

Q: How do I make a green-bar report?
A: Use a conditional background color based on RowNumber in each item in the row that should alternate color (for example, =iif(RowNumber(Nothing) Mod 2, “Green”, “White”))

Q: How do I include a global constant in my report definition?
A: Add a parameter to your report with a value but without a prompt. Since there is no prompt, users will not be prompted to enter a new value.

Q: How do RowNumber and RunningValue work in tablixes with both dynamic rows and dynamic columns?
A: The Scope argument for RunningValue and RowNumber can refer to either a column group or a row group. This defines both the direction of the running value (along rows or along columns) and when the running value resets. For example, here is RowNumber used with various Scope arguments in a tablix with two groups on each axis:

RowNumber ("Country")		1999				2000			
		Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
US	West	1	2	3	4	5	6	7	8
	East	9	10	11	12	13	14	15	16
	South	17	18	19	20	21	22	23	24
Canada	West	1	2	3	4	5	6	7	8
	East	9	10	11	12	13	14	15	16
	Central	17	18	19	20	21	22	23	24

RowNumber ("Region")		1999				2000			
		Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
US	West	1	2	3	4	5	6	7	8
	East	1	2	3	4	5	6	7	8
	South	1	2	3	4	5	6	7	8
Canada	West	1	2	3	4	5	6	7	8
	East	1	2	3	4	5	6	7	8
	Central	1	2	3	4	5	6	7	8

RowNumber ("Year")		1999				2000			
		Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
US	West	1	7	13	19	1	7	13	19
	East	2	8	14	20	2	8	14	20
	South	3	9	15	21	3	9	15	21
Canada	West	4	10	16	22	4	10	16	22
	East	5	11	17	23	5	11	17	23
	Central	6	12	18	24	6	12	18	24

RowNumber ("Quarter")		1999				2000			
		Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
US	West	1	1	1	1	1	1	1	1
	East	2	2	2	2	2	2	2	2
	South	3	3	3	3	3	3	3	3
Canada	West	4	4	4	4	4	4	4	4
	East	5	5	5	5	5	5	5	5
	Central	6	6	6	6	6	6	6	6

Q: I have two vertical sections in my report. How do I stop items in one section from pushing items in the other section downward?
A: Group the items in each section using a rectangle. Since growth only pushes peer items out of the way, the items in each rectangle can't push each other around.

Q: I have an image I'd like to place along the right edge of the page, but the tablix growth keeps pushing it off the page. How do I stop this?
A: Group the tablix with the blank space to its right by using a rectangle. Since growth only pushes peer items out of the way, the tablix in the rectangle can't push the image to the right (until it runs out of room and forces the rectangle to grow).

Q: How can I indicate that my tablix should grow to fill the blank space below it (rather than preserving the blank space)?
A: Group the tablix with the blank space below it by using a rectangle. Since growth only pushes peer items out of the way, the tablix in the rectangle has no items to push down below it, so it will consume the blank space until it fills the rectangle.

Q: What are the major changes between the Second (2005) and Third (2008) versions of RDL?
A:

- Replaced Table, Matrix and List with Tablix
- Aligned structure of Chart and CustomReportItem with Tablix
- Significant additions to Chart
- Added Gauge
- Added Rich Text capabilities to Textbox
- Renamed numerous elements and enum values for consistency and clarity
- Added KeepTogether to several objects
- Added scope to Previous function
- Added Variables
- Added constant datatype and nullability to some elements
- Misc structural refactoring