

Report Definition Language Specification

SECOND VERSION
NOVEMBER 2005

Technical Documentation License Agreement

READ THIS! THIS IS A LEGAL AGREEMENT BETWEEN MICROSOFT CORPORATION ("MICROSOFT") AND THE RECIPIENT OF THESE MATERIALS, WHETHER AN INDIVIDUAL OR AN ENTITY ("YOU"). IF YOU HAVE ACCESSED THIS AGREEMENT IN THE PROCESS OF DOWNLOADING MATERIALS ("MATERIALS") FROM A WEB SITE, BY CLICKING "I ACCEPT", DOWNLOADING, USING OR PROVIDING FEEDBACK ON THE MATERIALS, YOU AGREE TO THESE TERMS. IF THIS AGREEMENT IS ATTACHED TO MATERIALS, BY ACCESSING, USING OR PROVIDING FEEDBACK ON THE ATTACHED MATERIALS, YOU AGREE TO THESE TERMS.

For good and valuable consideration, the receipt and sufficiency of which are acknowledged, You and Microsoft agree as follows:

1. You may review these Materials only (a) as a reference to assist You in planning and designing Your product, service or technology ("Product") as described in these Materials; and (b) to provide feedback on these Materials to Microsoft. All other rights are retained by Microsoft; this agreement does not give You rights under any Microsoft patents. You may not (i) duplicate any part of these Materials, (ii) remove this agreement or any notices from these Materials, or (iii) give any part of these Materials, or assign or otherwise provide Your rights under this agreement, to anyone else.
2. These Materials may contain preliminary information or inaccuracies, and may not correctly represent any associated Product as commercially released. All Materials are provided entirely "AS IS." To the extent permitted by law, MICROSOFT MAKES NO WARRANTY OF ANY KIND, DISCLAIMS ALL EXPRESS, IMPLIED AND STATUTORY WARRANTIES, AND ASSUMES NO LIABILITY TO YOU FOR ANY DAMAGES OF ANY TYPE IN CONNECTION WITH THESE MATERIALS OR ANY INTELLECTUAL PROPERTY IN THEM.
3. If You are an entity and (a) merge into another entity or (b) a controlling ownership interest in You changes, Your right to use these Materials automatically terminates and You must destroy them.
4. You have no obligation to give Microsoft any suggestions, comments or other feedback ("Feedback") relating to these Materials. However, any Feedback you voluntarily provide may be used in Microsoft's Products and related specifications or other documentation (collectively, "Microsoft Offerings") which in turn may be relied upon by other third parties to develop their own Products. Accordingly, if You do give Microsoft Feedback on any version of these Materials or the Microsoft Offerings to which they apply, You agree: (a) Microsoft may freely use, reproduce, license, distribute, and otherwise commercialize Your Feedback in any Microsoft Offering; (b) You also grant third parties, without charge, only those patent rights necessary to enable other Products to use or interface with any specific parts of an Microsoft Product that incorporate Your Feedback; and (c) You will not give Microsoft any Feedback (i) that You have reason to believe is subject to any patent, copyright or other intellectual property claim or right of any third party; or (ii) subject to license terms which seek to require any Microsoft Offering incorporating or derived from such Feedback, or other Microsoft intellectual property, to be licensed to or otherwise shared with any third party.
5. This agreement is governed by the laws of the State of Washington. Any dispute involving it must be brought in the federal or state superior courts located in King County, Washington, and You waive any defenses allowing the

dispute to be litigated elsewhere. If there is litigation, the losing party must pay the other party's reasonable attorneys' fees, costs and other expenses. If any part of this agreement is unenforceable, it will be considered modified to the extent necessary to make it enforceable, and the remainder shall continue in effect. This agreement is the entire agreement between You and Microsoft concerning these Materials; it may be changed only by a written document signed by both You and Microsoft.

Contents

About this Document	1
Audience.....	1
Introduction	2
Goals of Report Definition Language	2
What is a Report?	2
Report Definition Diagrams	3
Report Definition XML.....	9
XML Namespace and Versioning	9
Extending RDL	9
Element Definition Conventions	9
Element Definitions.....	11
Style Properties and ReportItem Types.....	87
Expressions.....	89
Expression Syntax	89
Custom Code References	89
Data Types.....	89
Global Collections.....	90
Aggregate Functions	97
Questions & Answers.....	107
Changes in this Version.....	113
Index.....	114

About this Document

This document specifies the structure, the semantics and the interpretation 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 entirely. 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 on 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** that the report such as author, parameters, images within the report, etc.

Report Definition Diagrams

This chapter presents diagrams that illustrate the schema of the Report Definition Language.

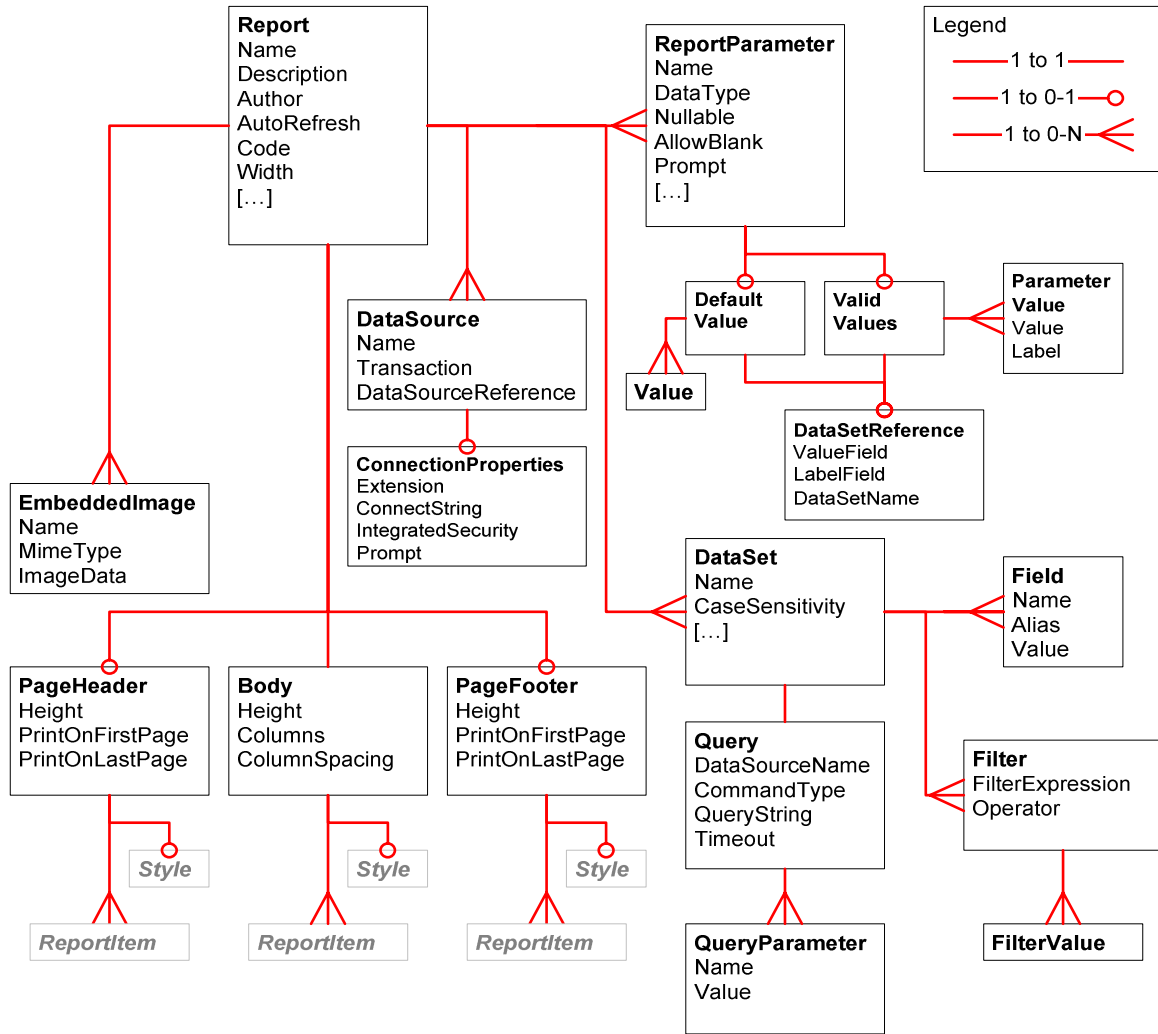


Figure 1 - Report

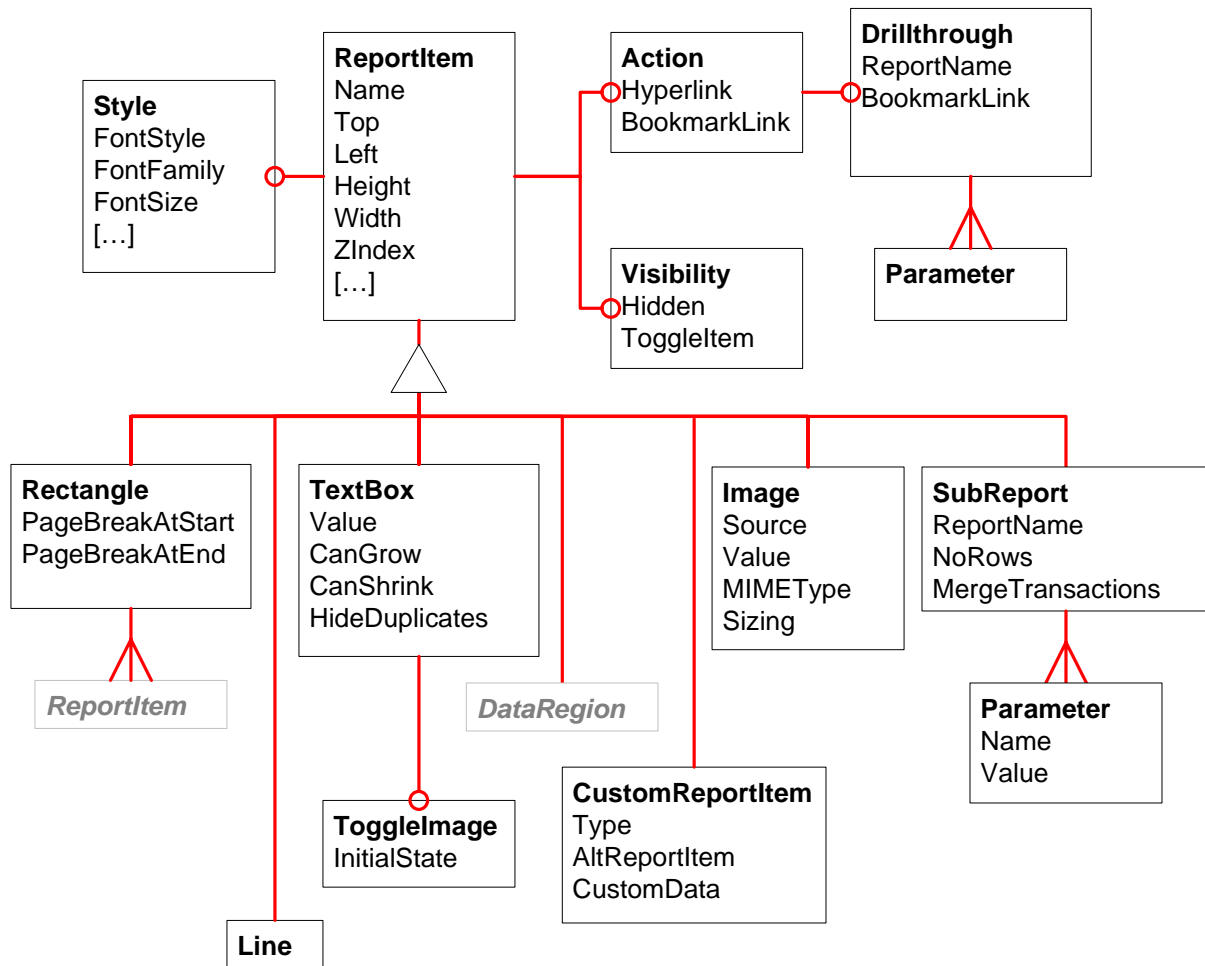


Figure 2 - Report Items

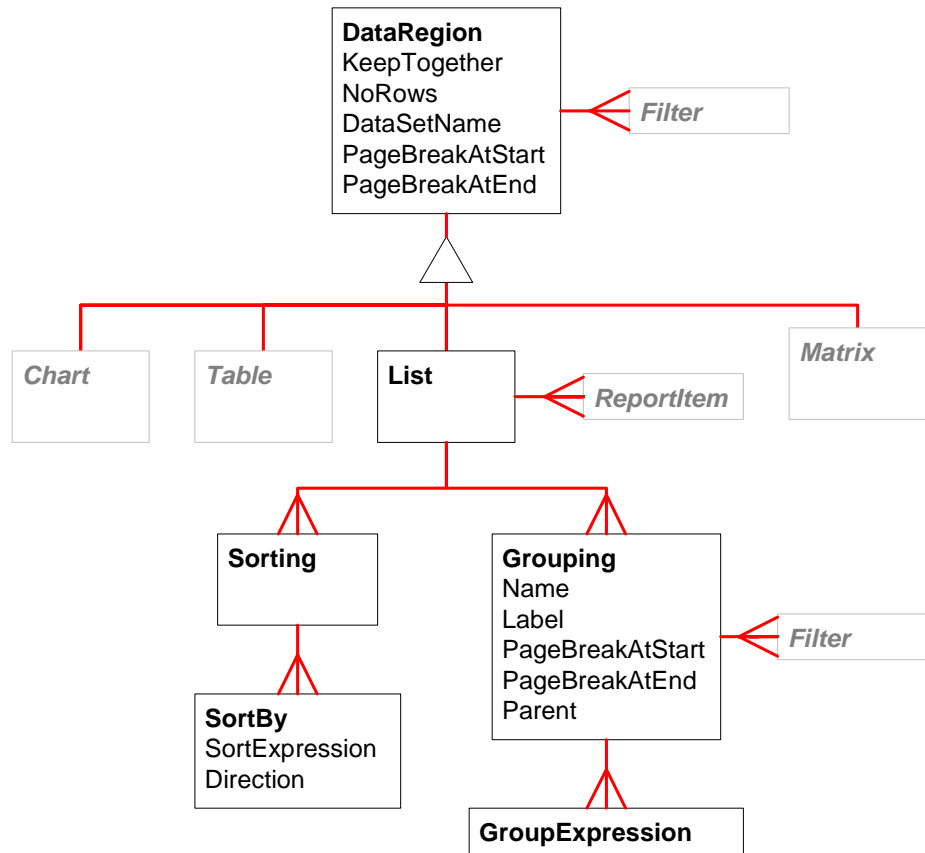


Figure 3 - Data Regions

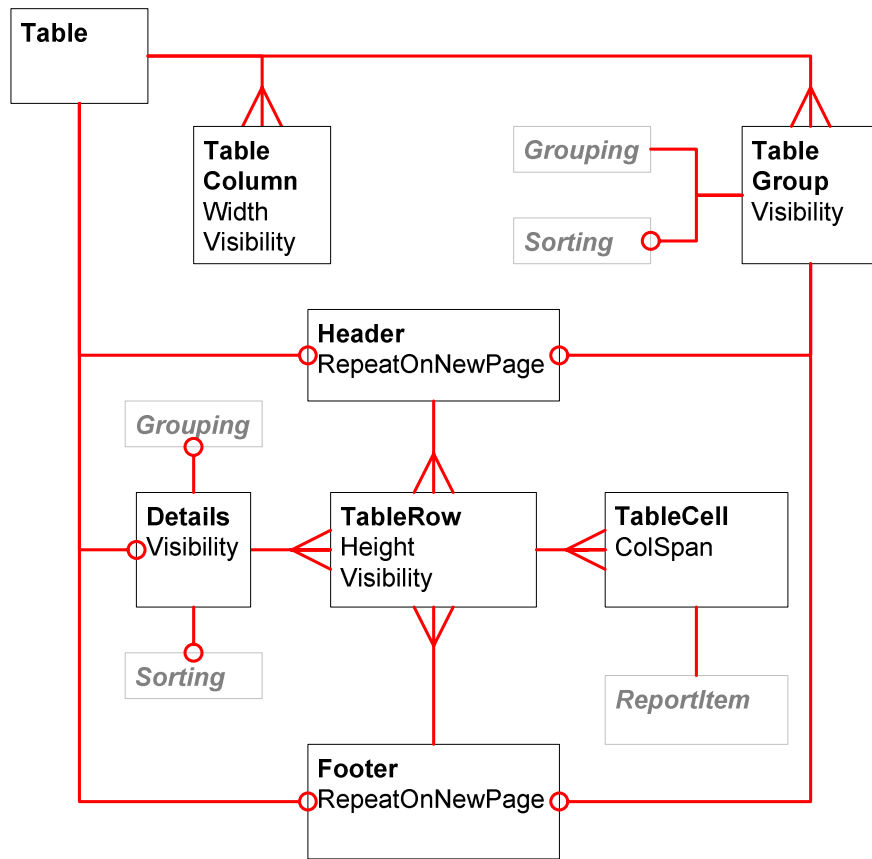


Figure 4 - Table

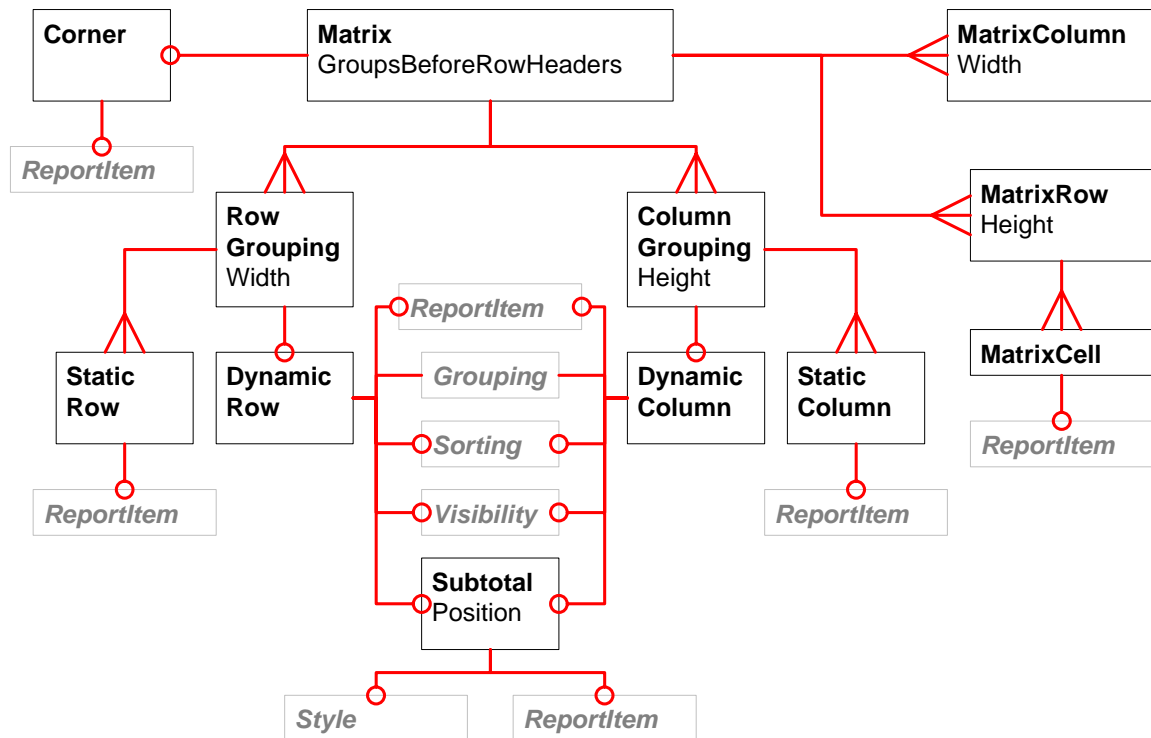


Figure 5 – Matrix

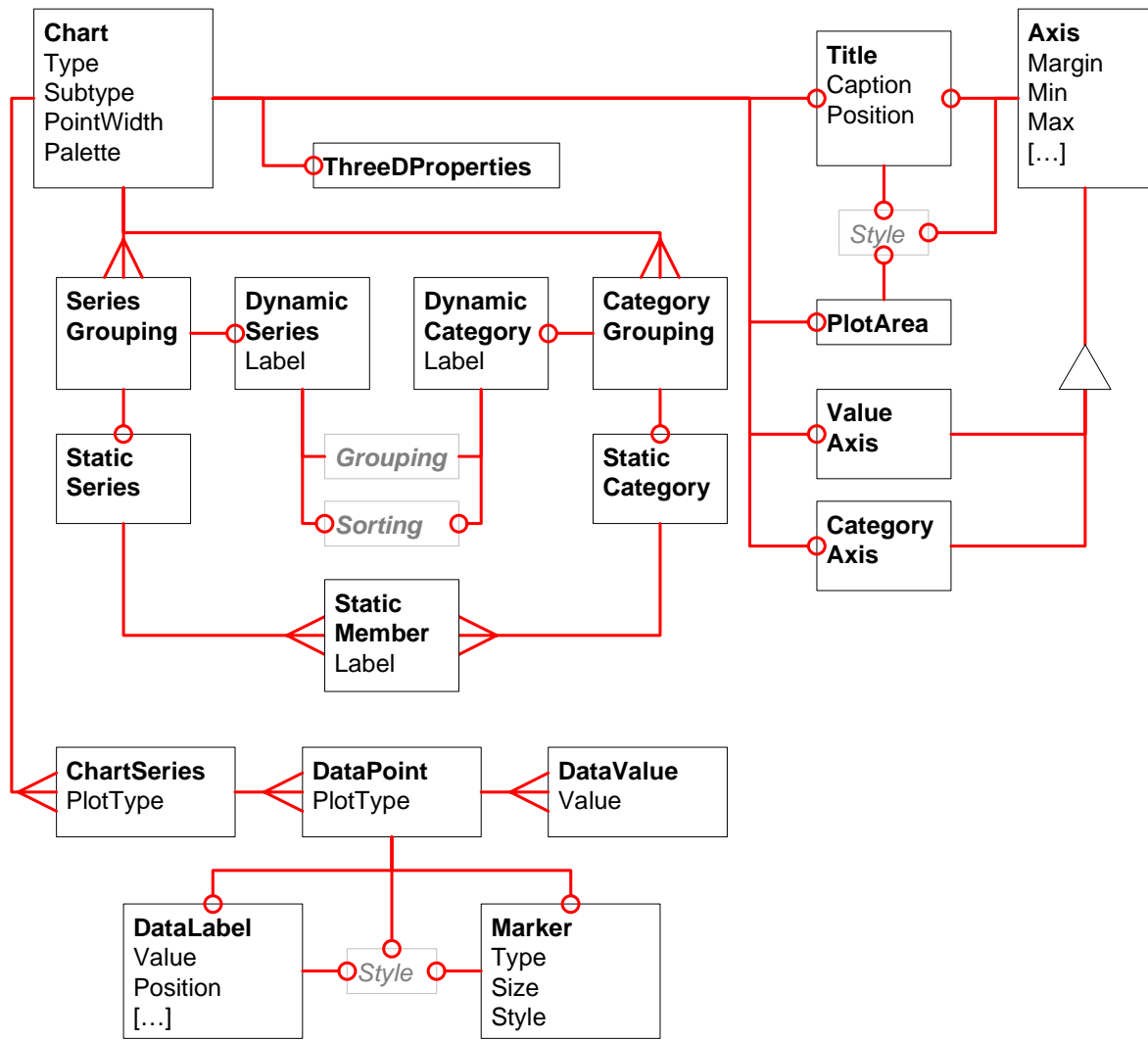


Figure 6 - Chart

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
```

This date format is used for versioning. As it will be unusual to update the namespace URI more than once in the same month, the RDL URI will use yyyy/mm.

Extending RDL

RDL is an open schema. It is expected that third parties will 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 (other than the contents of the Custom element).

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-insensitive CLS-compliant identifier (see <http://www.unicode.org/unicode/reports/tr15/tr15-18.html>)
- Element – A subelement (structure to be defined elsewhere in the document)
- String – A subelement or attribute with a string text value.
- This is an attribute if the name of the property is name and a subelement otherwise.

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

- Integer – A subelement with an integer (int32) value.
- 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.
- Size – A subelement with a size value. A size value is a floating-point number (with “.” 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 0.²
- Negative (absolute) sizes are only allowed where explicitly stated in the spec.
- Color – A subelement with a color value. A color value is either a color name or a hex HTML color string of the form #HHHHHH. See <http://msdn.microsoft.com/workshop/author/dhtml/reference/colors/colors.asp>.
- Expression – See the Expressions section below. Expected return type is listed below each expression.
- URL – A subelement with a string value that is a valid URL. See <http://www.ietf.org/rfc/rfc1738.txt>.
- 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 such as “en-us” for US English.³ The value must either be a specific language or a neutral language for which a default specific language is defined in the .NET framework.

² 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).

³ IDs follow the RFC 1766 standard in the format "<languagecode2>-<country/regioncode2>", where <languagecode2> is a lowercase two-letter code derived from ISO 639-1 and <country/regioncode2> is an uppercase two-letter code derived from ISO 3166. For example, U.S. English is "en-US". Some culture names have prefixes that specify the script; for example, "Cy-" specifies the Cyrillic script, "Lt-" specifies the Latin script. See <http://www.w3.org/TR/REC-html40/struct/dirlang.html#langcodes>.

Element Definitions

Report

The Report element contains property, data and layout information about the report (the top-level element).

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Description	0-1	String	Description of the report
Author	0-1	String	Author of the report
AutoRefresh	0-1	Integer	Rate at which the report page automatically refreshes, in seconds. 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
ReportParameters	0-1	Element	Parameters for the report
Custom	0-1	Element	Custom information to be handed to the report engine
Width	1	Size	Width of the report
PageHeader	0-1	Element	The header that is output at the top of each page of the report.
PageFooter	0-1	Element	The footer that is output at the bottom of each page of the report.
PageHeight	0-1	Size	Default height for the report in a

			physical-page oriented renderer. Default: 11 in. Must be greater than 0 in.
PageWidth	0-1	Size	Default width for 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. Default: 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. Default: 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
EmbeddedImages	0-1	Element	Images embedded within the report
Language	0-1	Expression (Language)	The primary language of the text. Default is server language.
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.
DataTransform	0-1	String	The location to a transformation to apply to a report data rendering. This can be a full folder path (e.g. “/xsl/xfrm.xsl”),

			relative path (e.g. “xfrm.xml”).
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 textboxes should render as elements or attributes. AttributeNormal (Default) ElementNormal

ReportParameters

The ReportParameters element contains an ordered list of parameters for the report. It is expected that a report execution engine would provide some sort of user interface that prompted the user for the set of necessary values before processing the report.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
ReportParameter	1-N	Element	Definition of a parameter for the report

ReportParameter

The ReportParameter element contains information about a parameter to the report.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
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 Boolean DateTime Integer Float String

Nullable	0-1	Boolean	Indicates the value for this parameter is allowed to be Null.
DefaultValue	0-1	Element	<p>Default value to use for the parameter (if not provided by the user)</p> <p>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).</p>
AllowBlank	0-1	Boolean	Indicates the value for this parameter is allowed to be the empty string. Ignored if DataType is not String.
Prompt	0-1	String	<p>The user prompt to display when asking for parameter values.</p> <p>If omitted, the user should not be prompted for or allowed to otherwise provide a value for this parameter.</p>
Hidden	0-1	Boolean	Indicates the parameter should not be displayed to the user (however, it will still be available for programmatic use with subreports, drillthrough reports etc.)
ValidValues	0-1	Element	Possible values for the parameter (for an end-user prompting interface)
MultiValue	0-1	Boolean	<p>Indicates this is a multi-value parameter (a parameter that can take a set of values).</p> <p>Multivalue parameters are accessed in expressions as zero-based arrays in the Value and Label properties (e.g. Parameters!Cities.Value(0) and Parameters!Cities.Label(0))</p> <p>Ignored for Boolean parameters</p>
UsedInQuery	0-1	Enum	<p>True False Auto (default)</p> <p>Indicates whether or not the parameter is used in a query in the report. This is</p>

			needed to determine if the queries need to be re-executed if the parameter changes. Auto indicates the UsedInQuery setting should be autodetected as follows: True if the parameter is referenced in any query value expression.
--	--	--	--

ValidValues

ValidValues contains the possible values for this parameter. It is used for populating UI selection lists for users to select a parameter value. Values must have one and only one of the following: DataSetReference or ParameterValues.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
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

DataSetReference

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

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
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

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
ParameterValue	1-N	Element	Possible value for the parameter

ParameterValue

A possible value for a parameter

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Value	0-1	Expression (Variant)	Possible value for the parameter. For Boolean parameters, use “true” and “false”. For DateTime parameters, use ISO 8601 compatible values. 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)	Label for the value to display to the user. 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

Contains the default value for a 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.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
DataSetReference	0-1	Element	The query to execute to obtain the default value(s) for the parameter. The default is the first value of the ValueField.
Values	0-1	Element	The default values for the parameter

Values

A set of values (used as defaults for a parameter).

For single-value parameters, only the first Value is used.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
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.

Note: Only one default value is allowed for Boolean parameters

DataSets

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

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
DataSet	1-N	Element	The sets of data for the report

DataSet

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

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Name	1	Name	Name of the data set Cannot be the same name as any data region or grouping
Fields	0-1	Element	The fields in the data set
Query	1	Element	Information about the data source, including connection information, query, etc. required to get the data from the data source.
CaseSensitivity	0-1	Enum	Indicates if the data is case sensitive True False Auto (Default) If Auto is specified, the application should attempt to derive the case sensitivity setting by querying the data provider. Defaults to False if the data provider does not support that method.
Collation	0-1	String	The locale to use for the collation sequence for sorting data. Uses the standard Microsoft SQL Server collation codes (see http://msdn.microsoft.com/library/en-us/tsqlref/ts_ca-co_2e95.asp). If no Collation is specified, the application should attempt to derive the collation setting by querying the data provider. Defaults to the application's locale settings if the data provider does not support that method or returns an unsupported or invalid value
AccentSensitivity	0-1	Enum	Indicates whether the data is accent sensitive True False Auto (Default) If Auto is specified, the application should attempt to derive the accent sensitivity setting by querying the data provider. Defaults to False if the data provider does not support that method.

KanatypeSensitivity	0-1	Enum	Indicates if the data is kanatype sensitive True False Auto (Default) If Auto is specified, the Application should attempt to derive the kanatype sensitivity setting by querying the data provider. Defaults to False if the data provider does not support that method.
WidthSensitivity	0-1	Enum	Indicates if the data is width sensitive True False Auto (Default) If Auto is specified, the Application should attempt to derive the width sensitivity setting by querying the data provider. Defaults to False if the data provider does not support that method.
Filters	0-1	Element	Filters to apply to each row of data in the data set.

Fields

The Fields element defines the fields in the data model.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Field	1-N	Element	Field within 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 (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>StoreState</DataField>
  </Field>
```



```

<Field Name="Category">
  <DataField>ProductCategory</DataField>
</Field>
<Field Name="Quarter">
  <DataField>Quarter</DataField>
</Field>
<Field Name="Sales">
  <DataField>StoreSales</DataField>
</Field>
</DataModel>

```

Field

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

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Name	1	Name	Name to use for the field within 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: DataField names do not need to be unique. Multiple fields can refer to the same data field.
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. Expressions are fully described in later in the specification.

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

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
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 Text (Default) StoredProcedure TableDirect
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 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

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
DataSource	1-N	Element	A source of data for the report

DataSource

The DataSource element contains information about a data source

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Name	1	String	The name of the data source Must be unique within the report
Transaction	0-1	Boolean	Indicates the data sets that use this data

			source should be executed in a single transaction.
ConnectionProperties	0-1	Element	Information about how to connect to the data source
DataSourceReference	0-1	String	The full path (e.g. “/salesreports/salesdatabase”) or relative path (e.g. “salesdatabase”) to a data source reference. Relative paths start in the same location as the report.

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

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
DataProvider	1	String	The type of the data source. This will determine the syntax of the ConnectString and CommandText. Supported types are listed below.
ConnectString	1	Expression (String)	The connection string for the data source
IntegratedSecurity	0-1	Boolean	Indicates that this data source should connected to using integrated security
Prompt	0-1	String	The prompt displayed to the user when prompting for database credentials for this data source

Supported Data Source Extensions

The following table lists the public set of data source extensions. Vendors may contribute additional extension names to be added to RDL or use custom extension names. RDL consumers must warn users when loading a report definition with an unknown or unsupported extension.

<i>Extension</i>	<i>Description</i>
SQL	Microsoft SQL Server
OleDb-MD	Microsoft SQL Server Analysis Services
OleDb	Microsoft OLE DB Provider
ODBC	Microsoft Open Database Connectivity Driver
Oracle	Oracle
DB2	IBM DB2
XML	XML Web Service

QueryParameters

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

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
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

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Name	1	String	Name of the parameter
Value	1	Expression (Variant or Variant Array)	<p>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.</p> <p>In the case of a parameter to a Values or DefaultValue query, the expression can only refer to report parameters that occur earlier in</p>

			the parameters list. The value for this query parameter is then taken from the user selection for that earlier report parameter.
--	--	--	--

CodeModules

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

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
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

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
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

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
ClassName	1	String	The name of the class
InstanceName	1	Name	The name of the variable to assign the class to. This variable can be used in expressions throughout the report.

Body

The Body element is the top-level container for all report items and defines the visual elements of the report, how the data is structured/grouped, and binds the visual elements to the data sets for the report.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
ReportItems	0-1	Element	The region that contains the elements of the report body
Height	1	Size	Height of the body
Columns	0-1	Integer	Number of columns for the report Default: 1. Min: 1. Max: 1000
ColumnSpacing	0-1	Size	Spacing between each column in multi-column output Default: 0.5 in
Style	0-1	Element	Default style information for the body ⁴

PageHeader

The PageHeader element defines the layout of report items to appear at the top of every page of the report.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Height	1	Size	Height of the page header
PrintOnFirstPage	0-1	Boolean	Indicates if the page header should be shown on the first page of the report
PrintOnLastPage	0-1	Boolean	Indicates if the page header should be shown on the last page of the report. Not used in single-page reports.
ReportItems	0-1	Element	The region that contains the elements of the header layout No data regions or subreports are allowed in the page header

⁴ The Body is generally treated as a rectangular ReportItem for the purposes of which Style properties apply, except for, for example, Direction, which controls overall alignment/scrollbar/etc.

Style	0-1	Element	Style information for the page header
-------	-----	---------	---------------------------------------

PageFooter

The PageFooter element defines the layout of report items to appear at the bottom of every page of the report.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Height	1	Size	Height of the page footer
PrintOnFirstPage	0-1	Boolean	Indicates if the page footer should be shown on the first page of the report. Not used in single-page reports.
PrintOnLastPage	0-1	Boolean	Indicates if the page footer should be shown on the last page of the report.
ReportItems	0-1	Element	The region that contains the elements of the footer layout No data regions or subreports are allowed in the page footer
Style	0-1	Element	Style information for the page footer ⁵

CustomProperties

The CustomProperties element allows report design tools to pass information to custom report renderers and custom report items. CustomProperties subsumes the functionality of the Custom element in the first version of the specification. Design tools should upgrade the Custom element into a single CustomProperty with Name="Custom".

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
CustomProperty	1-N	Element	A custom property for this report, report item or group.

⁵ The PageHeader and PageFooter are treated as rectangle ReportItems for the purposes of which Style properties apply.

CustomProperty

The content of CustomProperty is 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 likelihood of name collisions in the event that multiple applications are used for editing the same report definition. E.g. “msd:FormattedValue” rather than just “FormattedValue”

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
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 within the report definition. Once define, embedded images can be referenced by the Image report item.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
EmbeddedImage	1-N	Element	An image embedded within the report.

EmbeddedImage

The EmbeddedImage element is an image embedded within the report definition.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Name	1	Name	Name of the image.
MIMEType	1	String	The MIMEType for the image. Valid values are: 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 grouping.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
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 grouping. 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

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
FilterExpression	1	Expression (Variant)	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 grouping. See Filter Expression Restrictions below.
Operator	1	Enum	Equal Like NotEqual GreaterThan GreaterThanOrEqual LessThan LessThanOrEqual TopN BottomN TopPercent BottomPercent In Between Notes: Top and Bottom operators include ties in the resulting data. String comparisons are locale-dependent. Null equals Null.
FilterValues	1	Element	The values to compare to the FilterExpression. For Equal, Like, NotEqual, GreaterThan, GreaterThanOrEqual, LessThan, LessThanOrEqual, TopN, BottomN, TopPercent and BottomPercent, there must be

			<p>exactly one FilterValue.</p> <p>For TopN and BottomN, the FilterValue expression must evaluate to an integer.</p> <p>For TopPercent and BottomPercent, the FilterValue expression must evaluate to an integer or float.⁶</p> <p>For Between, there must be exactly two FilterValue elements.</p> <p>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.)</p> <p>Like uses the same special characters as the Visual Basic LIKE operator (e.g. “?” to represent a single character and “*” to represent any series of characters). See http://msdn.microsoft.com/library/en-us/vblr7/html/vaoprlike.asp.</p>
--	--	--	--

FilterValues

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

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
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 below.

⁶ 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.

Note: Multi-value 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.
- Grouping filter expressions/values cannot contain RunningValue or RowNumber.
- Grouping filter expressions/values cannot use the First or Last aggregate with anything but the default (current) scope.
- Failure when evaluating any filter expression or filter value causes a 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

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
ReportItem	1-N	Element	An element of the report layout (e.g. List, Textbox, Line)

ReportItem

A report item is one of the following types of objects: Line, Rectangle, Textbox, Image, Subreport, CustomReportItem or DataRegion. DataRegions are: List, Table, Matrix, and Chart. The ReportItem element itself is not used. Instead, specific report item element is used wherever ReportItem is allowed.

Common Attributes/Elements

The following attributes and elements are shared among all types of ReportItem elements (Note, however, that not all attributes/elements of ReportItem are necessarily meaningful for all types of report items).

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Name	1	Name	Name of the report item
Style	0-1	Element	Style information for the element
Action	0-1	Element	An action (e.g. a hyperlink) associated with the ReportItem
Top	0-1	Size	The distance of the item from the top of the containing object. ⁷ Defaults to 0 if omitted.
Left	0-1	Size	The distance of the item from the left of the containing object. ⁷ Defaults to 0 if omitted.
Height	0-1	Size	Height of the item. ⁷ 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. ⁸
Width	0-1	Size	Width of the item. Negative sizes allowed only for lines. ⁷ Defaults to the width of the containing object minus Left if omitted. ⁸
ZIndex	0-1	Integer	Drawing order of the report item within the

⁷ See “Automatic Position and Size Behavior”.

⁸ For Table and Matrix, the default Height and Width are instead derived from the sizes of the component parts (columns, rows, matrix cells).

			containing object. Items with lower indices are drawn first (appearing behind items with higher indices). Items with equal indices have an unspecified order. Default: 0 Min: 0 Max: 2147483647
Visibility	0-1	Element	Indicates if the item should be hidden. ⁹
ToolTip	0-1	Expression (String)	A textual label for the report item. Used for such things as including TITLE and ALT attributes in HTML reports.
Label	0-1	Expression (Variant)	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.
LinkToChild	0-1	String	The name of a report item contained directly within this report item that is the target location for the Document Map label (if any). Ignored if Label is not present. Used only for Rectangle.
Bookmark	0-1	Expression (String)	A bookmark that can be linked to via a Bookmark action
RepeatWith	0-1	String	The name of a data region that this report item should be repeated with if that data region spans multiple pages.

⁹ For List, Visibility controls whether each instance of the list is individually visible, not whether the entire set of list instances is visible as a whole.

			<p>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).</p> <p>Not allowed if this report item is a data region, subreport or rectangle that contains a data region or subreport.</p>
Custom	0-1	Element	Custom information to be handed to a report output component.
DataElement Name	0-1	String	<p>The name to use for the data element/attribute for this report item.</p> <p>Default: Name of the report item.</p> <p>Must be a CLS-compliant identifier.</p>
DataElement Output	0-1	Enum	<p>Indicates whether the item should appear in a data rendering.</p> <p>Output: Indicates the item should appear in the output</p> <p>NoOutput: Indicates the item should not appear in the output</p> <p>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 Lists.</p> <p>Auto (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 nontoggleable table column with Hidden set to non-expression True. Otherwise, acts as NoOutput for Textboxes with constant values, as ContentsOnly for Rectangles and as Output for all other items.</p>

Attributes and elements specific to each ReportItem element type are described below.

Automatic Position and Size Behavior

Items in a report may grow vertically or horizontally due to repeated sections, variable-size contents or other reasons.

When an item in a report grows, it pushes peer items (items within the same parent container) out of the way as follows:

- Each item moves down to maintain a minimum spacing between itself and all items that end above it. This spacing is the minimum of the original distances between the item and all items that end above it.
- Each item moves right to maintain a minimum spacing between itself and all items that end left of it. This spacing is the minimum of the original distances between the item and all items that end to the left of it.

When an item in the report grows, if it would extend beyond the bounds of the containing item, its container grows large enough to accommodate the contained item.

Which properties apply to which ReportItem types?

	Line	Rectangle	Textbox	Image	Subreport	CustomReport Item	List	Matrix	Table	Chart
Name	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Action	N	N	Y	Y	N	N	N	N	N	N
Top	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Left	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Height	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Width	Y	Y	Y	Y	Y	Y	Y	N	N	Y
ZIndex	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Visibility	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Tooltip	N	Y	Y	Y	Y	N	Y	Y	Y	Y
Label	Y	Y	Y	Y	Y	N	Y	Y	Y	Y
LinkToChild	N	Y	N	N	N	N	N	N	N	N
Bookmark	Y	Y	Y	Y	Y	N	Y	Y	Y	Y

RepeatWith	Y	Y	Y	Y	N	Y	N	N	N	N
Custom	Y	Y	Y	Y	Y	N	Y	Y	Y	Y
DataElementName	N	Y	Y	N	Y	N	Y	Y	Y	Y
DataElementOutput	N	Y	Y	N	Y	N	Y	Y	Y	Y

Action

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

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Hyperlink	0-1	Expression (URL)	An expression that evaluates to the URL of the hyperlink
Drillthrough	0-1	Element	The drillthrough report that should be executed by clicking on the hyperlink. Not supported in page headers/footers.
BookmarkLink	0-1	Expression (String)	An expression that evaluates to the ID of a bookmark within 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 one). Not supported in page headers/footers.

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

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
ReportName	1	URL	The path of the drillthrough report. Paths may be

			absolute or relative.
Parameters	0-1	Element	Parameters to the drillthrough report

Visibility

The Visibility element indicates the ReportItem should not be (initially) shown in the output report.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Hidden	0-1	Expression (Boolean)	Indicates if the item should be initially hidden.
ToggleItem	0-1	String	<p>The name of the textbox 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. If the Toggle item becomes hidden, this item should become hidden.¹⁰</p> <p>Must be a textbox in the same grouping scope as this item or in any containing (ancestor) grouping scope</p> <p>If omitted, no item will toggle the hidden state of this item.</p> <p>Not allowed on and cannot refer to report items contained in a page header or footer.</p> <p>Cannot refer to a report item contained within the current report item unless current grouping scope has a Parent.</p>

¹⁰ This cascading does not apply if the toggle item becomes hidden as a result of a containing table column or matrix cell becoming hidden.

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 generating the report. This means the report layout does not change due to the fact that the item is hidden (unlike hidden items that can be toggled or are conditionally hidden, thereby shifting layout to make room/remove empty space).

Using ToggleItem with a Recursive Hierarchy

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

Example:

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

Employee

VP

 Manager1

 Employee1

 Employee2

 Manager2

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

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

The grouping for the table details would look like this:

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

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

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
ReportItems	0-1	Element	Report items contained within the bounds of the rectangle.
PageBreakAtStart	0-1	Boolean	Indicates the report should page break at the start of the rectangle. ¹¹
PageBreakAtEnd	0-1	Boolean	Indicates the report should page break at the end of the rectangle. ¹¹

Textbox

The Textbox element defines a rectangular text region and has the following elements in addition to what it inherits from ReportItem.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Value	1	Expression (Variant)	An expression, the value of which is displayed in the text-box. This can be a constant expression for constant labels.
CanGrow	0-1	Boolean	Indicates the Textbox size can increase to

¹¹ Page breaks on rectangles are ignored if the rectangle has no report items.

			accommodate the contents
CanShrink	0-1	Boolean	Indicates the Textbox size 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 instance. The value of HideDuplicates is the name of a grouping or data set over which to apply the hiding. Each time a new instance of that group is encountered, the first instance of this report item will not be hidden. Rows on a previous page are ignored for the purposes of hiding duplicates. If the textbox is in a table or matrix cell, only the text will be omitted. The textbox will remain to provide background and border for the cell. Outside of a table/matrix cell, the background and borders are omitted as well. Ignored in matrix subtotals.
ToggleImage	0-1	Element	Indicates the initial state of a toggling image should one be displayed as a part of the textbox.
UserSort	0-1	Element	Indicates an end-user sort control should be displayed as a part of this textbox.
DataElementStyle	0-1	Enum	Indicates whether textbox value should render as an element or attribute: Auto (Default) AttributeNormal ElementNormal. Auto uses the setting on the Report element.

ToggleImage

Indicates the initial state of a toggle image should such an image be displayed as a part of the textbox. The image is always displayed if the textbox is a toggle item for another report item. If no ToggleImage element is present, no image should be displayed. Whenever the textbox/image is clicked on, the toggle image state flips and the image associated with the new state is displayed instead.

Attributes/Elements

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

UserSort

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

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
SortExpression	1	Expression (Variant)	<p>The expression on which to sort.</p> <p>Has the same restrictions as a Grouping 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.</p>
SortExpression Scope	0-1	String	<p>Name of the scope (data region or grouping) in which to evaluate the SortExpression.</p> <p>If omitted, the expression will be evaluated and the sort will be performed independently in each detail scope within the SortTarget.</p> <p>Must be a scope that is equal to or contained within the current scope. If the textbox has no current scope (i.e. it is not contained in any data region), SortExpressionScope must be equal to or contained within the SortTarget.</p> <p>The data set for the SortExpressionScope must be the same as the data set for the SortTarget.</p> <p>Sorting takes place within the group containing the SortExpressionScope. For example: In a table with a country group and a city group with UserSort on each header and SortExpressionScope</p>

			of the corresponding group, the country sort will sort the country groups within the table and the city sort will sort the city groups within each country group (without rearranging the country groups).
SortTarget	0-1	String	<p>Name of the data region, grouping or data set to apply the sort to.</p> <p>If omitted, the sort will apply to the instance of the current scope.</p> <p>Must be the current scope, an ancestor scope¹², or a peer scope which is a data region.</p>

Image

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

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Source	1	Enum	<p>Identifies the source of the image:</p> <p>External – The Value contains a constant or expression that evaluates to the location of the image. This could be a full path (e.g. “/images/logo.gif”), relative path (e.g. “logo.gif”) or URL (e.g. “http://myserver/images/logo.gif”).</p> <p>Embedded – The Value contains a constant or expression that evaluates to the name of an EmbeddedImage within the report.</p> <p>Database – The Value contains an expression (typically a field in the database) that evaluates to the binary data for the image.</p>

¹² Matrix groupings are only valid SortTargets from within matrix grouping scopes along the same matrix axis.

Value	1	Expression	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)	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.
Sizing	0-1	Enum	Defines the behavior if the image does not fit within the specified size. AutoSize = The borders should grow/shrink to accommodate the image (Default). Fit = The image is resized to exactly match the height and width of the image element. FitProportional = The image should be resized to fit, preserving aspect ratio. ¹³ Clip = The image should be clipped to fit. ¹³

Subreport

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

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
ReportName	1	String	The full path (e.g. “/salesreports/orderdetails”) or relative path (e.g. “orderdetails”) to a subreport. Relative paths start in the same folder as the current (parent) report.

¹³ Report output formats unable to support FitProportional or Clip should output as Fit instead.

			Cannot be an empty string (ignoring whitespace)
Parameters	0-1	Element	Parameters to the Subreport If the subreport is executed without parameters (and contains no Toggle elements), it will only be executed once (even if it appears inside of a list, table or matrix)
NoRows	0-1	Expression (String)	Message to display in the subreport (instead of the region layout ¹⁴) when no rows of data are available in any data set in 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.

Parameters

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

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Parameter	1-N	Element	Definition of a parameter passed into the subreport.

Parameter

The Parameter element contains information about a parameter.

¹⁴ If the subreport is in a table or matrix cell and does not have a NoRows property, the contents of the subreport will be omitted but the subreport's border properties will still apply to the cell.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
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.
Omit	0-1	Expression (Boolean)	Indicates the parameter should be skipped. Valid only for Drillthrough parameters.

DataRegion

A data region represents the layout of a data set in the report. Unlike other report items, data regions may have contents that repeat to accommodate the size of the data set results to which they are bound.

A DataRegion element is one of the following element types:

List, Matrix, Table, Chart

Common Attributes/Elements

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

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
KeepTogether	0-1	Boolean	Indicates the entire data region (all repeated sections) should be kept together on one page if possible.
NoRows	0-1	Expression (String)	Message to display in the DataRegion (instead of the region layout ¹⁵) when no rows of data are available. Note: Style information on the data region applies to this text.

¹⁵ If the data region is in a table or matrix cell and does not have a NoRows 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.

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.
PageBreakAtStart	0-1	Boolean	Indicates the report should page break at the start of the data region. ¹⁶
PageBreakAtEnd	0-1	Boolean	Indicates the report should page break at the end of the data region. ¹⁶
Filters	0-1	Element	Filters to apply to each row of data in the data region.

Attributes and elements specific to each DataRegion element type are described below.

List

The List element defines a region that repeats with each group in the rowset or each row in the rowset (if no group expression is defined).

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Grouping	0-1	Element	The expressions to group the data by. Required if there are any DataRegions contained within this List.
Sorting	0-1	Element	The expressions to sort the repeated list

¹⁶ Page breaks on lists are ignored if the list has no report items.

			regions by.
ReportItems	0-1	Element	The elements of the list layout.
DataInstanceName	0-1	String	The name to use for the data element for the each instance of this list. Ignored if there is a grouping for the list. Default: "Item"
DataInstanceElementOutput	0-1	Enum	Indicates whether the list instances should appear in a data rendering. Ignored if there is a grouping for the list. Output (Default): Indicates the list instances should appear in the output. NoOutput: Indicates the list instances should not appear in the output.

Grouping

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

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Name	1	Name	Name of the Grouping (for use in RunningValue and RowNumber) No two grouping elements may have the same name. No grouping element may have the same name as a data set or a data region
Label	0-1	Expression (String)	A label to identify an instance of the group within the client UI (to provide a user-friendly label for searching). See ReportItem.Label.
GroupExpressions	1	Element	The expressions to by which to group the

			data.
PageBreakAtStart	0-1	Boolean	Indicates the report should page break at the start of the group. ¹⁷ Not valid for column groupings in Matrix regions.
PageBreakAtEnd	0-1	Boolean	Indicates the report should page break at the end of the group. ¹⁷ Not valid for column groupings in Matrix regions.
Custom	0-1	Element	Custom information to be passed to the report output component.
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: <ul style="list-style-type: none"> 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

¹⁷ Grouping page breaks on table groups are ignored if the group has no header and no footer, no subgroups have a header or footer and there is no detail row in the table

			instance's parent, it is instead treated as a child of the parent's parent.
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.
DataCollectionName	0-1	String	The name to use for the data element for the collection of all instances of this group. Default: " <i>DataElementName_Collection</i> ". Must be a CLS-compliant identifier.
DataElementOutput	0-1	Enum	Indicates whether the group should appear in a data rendering. Output (Default): Indicates the group should appear in the output NoOutput: Indicates the group should not appear in the output

GroupExpressions

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

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
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. References to report items are not allowed.

Sorting

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

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
SortBy	1	Element	The expressions to sort the groups by. ¹⁸ This is an ordered list.

SortBy

The SortBy element defines an expression to sort the groups by

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
SortExpression	1	Expression (Variant)	The expression 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 Ascending (Default) Descending

Matrix

The Matrix element defines a grid of regions that repeats with each column group and row group in the dataset.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Corner	0-1	Element	The region that contains the elements of the upper left corner area of the matrix. If omitted, no report items are output in the corner.
ColumnGroupings	1	Element	The set of column groupings for the

¹⁸ Sorting preserves the order of rows from the original data for SortExpressions with identical values.

			matrix
RowGroupings	1	Element	The set of row groupings for the matrix
MatrixRows	1	Element	The rows contained in each detail cell of the matrix layout
MatrixColumns	1	Element	The columns contained in each detail cell of the matrix layout
LayoutDirection	0-1	Enum	Indicates whether the matrix columns grow left-to-right (with headers on the left) or right-to-left (with headers on the right). LTR (Default) RTL
GroupsBeforeRowHeaders	0-1	Integer	The number of instances of the outermost column group that should appear to the left of the row headers (right of the row headers for RTL matrixes). ¹⁹ Default is 0.
CellDataElementName	0-1	String	The name to use for the cell element. Default: "Cell" Must be a CLS-compliant identifier.
CellDataElementOutput	0-1	Enum	Indicates whether the cell contents should appear in a data rendering. Output (Default): Indicates the cell should appear in the output NoOutput: Indicates the cell should not appear in the output

Corner

The Corner element defines the layout and structure of the upper left-hand corner region of a Matrix.

¹⁹ For the purposes of GroupsBeforeRowHeaders, a subtotal column is considered a part of (and will move with) the rightmost or leftmost detail column, as appropriate.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
ReportItems	1	Element	The region that contains the elements of the corner layout This ReportItems collection must contain exactly one ReportItem. 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%. ²⁰

ColumnGroupings

The ColumnGroupings element defines the set of column groupings for a Matrix.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
ColumnGrouping	1-N	Element	The set of column groupings for the Matrix Each ColumnGrouping defines a row of column headings (the first ColumnGrouping is the outermost column headings).

ColumnGrouping

The ColumnGrouping element defines a column header region for a Matrix. A ColumnGrouping must have either DynamicColumns or StaticColumns, but not both.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Height	1	Size	Height of the column header.
DynamicColumns	0-1	Element	Dynamic column headings for this grouping
StaticColumns	0-1	Element	Static column headings for this grouping.

²⁰ 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.

FixedHeader	0-1	Boolean	Indicates the header for this grouping should be displayed on the page even when the user scrolls part of the matrix off the page. Fixed headers must be contiguous and must include the outermost grouping.
-------------	-----	---------	---

DynamicColumns

The DynamicColumns element defines a column header region that repeats with each column group in a Matrix's rowset.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Grouping	1	Element	The expressions to group the data by.
Sorting	0-1	Element	The expressions to sort the columns by.
Subtotal	0-1	Element	Indicates an automatic subtotal column should be included
ReportItems	1	Element	The elements of the column header layout This ReportItems collection must contain exactly one ReportItem. 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%.
Visibility	0-1	Element	Indicates if all of the dynamic columns for this grouping should be hidden and replaced with a subtotal column for this grouping scope

StaticColumns

The StaticColumns element defines a column header region with a fixed set of columns. Only one ColumnGrouping in each matrix may be static.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
StaticColumn	1-N	Element	Static column headings for this grouping.

StaticColumn

The StaticColumn element defines a fixed column header region in a Matrix.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
ReportItems	1	Element	The elements of the column header layout. This ReportItems collection must contain exactly one ReportItem. 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%.

RowGroupings

The Rows element defines the set of row groupings for a Matrix.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
RowGrouping	1-N	Element	The set of row groupings for the Matrix. Each Row defines a column of row headings (the first Row is the outermost row headings).

RowGrouping

The RowGrouping element defines a row header region for a Matrix. A RowGrouping must have either DynamicRows or StaticRows, but not both.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Width	1	Size	Width of the row header
DynamicRows	0-1	Element	Dynamic row headings for this grouping
StaticRows	0-1	Element	Static row headings for this grouping.

FixedHeader	0-1	Boolean	Indicates the header for this grouping should be displayed on the page even when the user scrolls part of the matrix off the page. Fixed headers must be contiguous and must include the outermost grouping.
-------------	-----	---------	---

DynamicRows

The DynamicRows element defines a row header region that repeats with each row group in a Matrix's rowset.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Grouping	1	Element	The expressions to group the data by
Sorting	0-1	Element	The expressions to sort the columns by
Subtotal	0-1	Element	Indicates an automatic subtotal row should be included
ReportItems	1	Element	The elements of the row header layout This ReportItems collection must contain exactly one ReportItem. 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%.
Visibility	0-1	Element	Indicates if all of the dynamic rows for this grouping should be hidden and replaced with a subtotal row for this grouping scope

StaticRows

The StaticRows element defines a row header region with a fixed set of rows.

Only one RowGrouping in each matrix may be static.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
StaticRow	1-N	Element	Static row headings for this grouping

StaticRow

The StaticRow element defines a fixed row header region in a Matrix.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
ReportItems	1	Element	The elements of the row header layout This ReportItems collection must contain exactly one ReportItem. 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%.

Subtotal

The Subtotals element defines a subtotal column/row to include in the Matrix.

In the subtotal column/row, the matrix cell contents are applied to the full set of data for the entire group.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
ReportItems	1	Element	The header cell for a subtotal column or row. This ReportItems collection must contain exactly one Textbox. 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%.
Style	0-1	Element	Style properties that override the style properties for all top-level report items contained in the subtotal column/row At Subtotal Column/Row intersections, Row style takes priority
Position	0-1	Enum	Before After (default) Indicates whether this subtotal column/row should appear before (left/above) or after (right/below) the detail columns/rows.
DataElementName	0-1	String	The name to use for this subtotal. Default: "Total" Must be a CLS-compliant identifier.
DataElementOutput	0-1	Enum	Indicates whether the subtotal should appear in

			a data rendering. Output: Indicates the subtotal element should appear in the output. NoOutput (Default): Indicates the subtotal should not appear in the output
--	--	--	--

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

MatrixColumns

The MatrixColumns element defines the set of columns in the detail section of a Matrix

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
MatrixColumn	1-N	Element	The set of columns in the detail section of the Matrix. There must be one MatrixColumn per StaticColumn in the Matrix (or exactly one MatrixColumn if there are no StaticColumns)

MatrixColumn

The MatrixRow element defines a column in the detail section of a Matrix

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Width	1	Size	Width of each detail cell in this column

MatrixRows

The MatrixRows element defines the set of rows in the detail section of a Matrix

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
MatrixRow	1-N	Element	The set of rows in the detail section of the Matrix There must be one MatrixRow per StaticRow in the Matrix (or exactly one MatrixRow if there are no StaticRows)

MatrixRow

The MatrixRow element defines the set of cells in a row of the detail section of a Matrix.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Height	1	Size	Height of each detail cell in this row.
MatrixCells	1	Element	The set of cells in a row in the detail section of the Matrix.

MatrixCells

The MatrixCells element defines the set of cells in a row of the detail section of a Matrix

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
MatrixCell	1-N	Element	The set of cells in a row in the detail section of the Matrix. There must be one MatrixCell per StaticColumn in the Matrix (or exactly one MatrixColumn if there are no StaticColumns).

MatrixCell

The MatrixCell element defines the contents of each detail cell in the matrix.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
ReportItems	1	Element	The report items contained in each detail cell of the matrix layout. This ReportItems collection must contain exactly one

			ReportItem. 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%.
--	--	--	--

Note: Page breaks on report items inside a MatrixCell are ignored.

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

Table

The Table element defines a tabular grouped layout of the data region.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
TableColumns	1	Element	The columns in the table
Header	0-1	Element	The table header rows
TableGroups	0-1	Element	The groups (group expressions/headers/footers) for the table
Details	0-1	Element	The details rows for the table The table must have at least one of: Details, Header or Footer
Footer	0-1	Element	The table footer rows
FillPage	0-1	Boolean	Indicates the table should expand to fill the page, pushing items below it to the bottom of the page.
DetailDataElementName	0-1	String	The name to use for the data element for instances of this group. Ignored if there is a grouping defined for the details. Default: "Details" Must be a CLS-compliant identifier.
DetailDataCollectionName	0-1	String	The name to use for the data element for the collection of all instances of

			this group. Default: “Details_Collection” Must be a CLS-compliant identifier.
DetailDataElementOutput	0-1	Enum	Indicates whether the details should appear in a data rendering. Output (Default): Indicates the details should appear in the output NoOutput: Indicates the details should not appear in the output

TableColumns

The TableColumns element defines the columns in the table.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
TableColumn	1-N	Element	A column in the table

TableColumn

The TableColumn element defines a column in the table.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Width	1	Size	Width of the column
Visibility	0-1	Element	Indicates if the column should be hidden
FixedHeader	0-1	Boolean	Indicates this column should be displayed on the page even when the user scrolls part of the table off the page. The fixed header columns in a table must be contiguous and must include the first or last column in the table.

Header

The Header element defines the header rows for a table or group.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
TableRows	1	Element	The header rows for the table or group
RepeatOnNewPage	0-1	Boolean	Indicates this header should be displayed on each page that the table (or group) is displayed
FixedHeader	0-1	Boolean	Indicates this header should be displayed on the page even when the user scrolls part of the table off the page. Allowed only on Table.Header.

TableRows

The TableRows element defines an ordered list of table rows.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
TableRow	1-N	Element	A header row of cells in the table.

TableRow

The TableRow element defines a row of cells in a table data region.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
TableCells	1	Element	Contents of the row. One cell per column
Height	1	Size	Height of the row
Visibility	0-1	Element	Indicates if the row should be hidden

Footer

The Footer element defines the footer rows for a table or group.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
TableRows	1	Element	The footer rows for the table or group
RepeatOnNewPage	0-1	Boolean	Indicates this footer should be displayed on

			each page that the table (or group) is displayed
--	--	--	--

TableGroups

The TableGroups element defines the groups in the table.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
TableGroup	1-N	Element	A group (group expression, header, footer) in the table

TableGroup

The TableGroup element defines a group in a table data region

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Grouping	1	Element	The expressions to group the data by.
Sorting	0-1	Element	The expressions to sort the data by.
Header	0-1	Element	A group header row.
Footer	0-1	Element	A group footer row.
Visibility	0-1	Element	Indicates if the group (and all groups embedded within it) should be hidden.

Details

The Details element defines the details rows for a table.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
TableRows	1	Element	The details rows for the table. The details rows cannot contain any DataRegions in any of their TableCells.
Grouping	0-1	Element	The expressions to group the detail data by
Sorting	0-1	Element	The expressions to sort the detail data by
Visibility	0-1	Element	Indicates if the details should be hidden

TableCells

The TableCells element defines the contents of a collection of cells in a table data region.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
TableCell	1-N	Element	Contents of the row. There must be one cell per column (ColSpan can allow multiple columns to be covered by one cell)

TableCell

The TableCell element defines the contents of a cell in a table data region

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
ReportItems	1	Element	An element of the report layout (e.g. List, Textbox, Line). This ReportItems collection must contain exactly one ReportItem. 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%. Pagebreaks on report items inside a TableCell are ignored.
ColSpan	0-1	Integer	Indicates the number of columns this cell spans. ²¹ A ColSpan of 1 is the same as not specifying a ColSpan

Chart

The Chart element defines a chart data region. The Chart is defined much like a Matrix, but instead of Columns, Rows, and Cells, the Chart has Categories, Series, and DataPoints.

²¹ 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.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Type	0-1	Enum	Type of the chart Column (Default) Bar Line Pie Scatter Bubble Area Doughnut Stock
Subtype	0-1	Enum	Subtype of the chart. Available subtypes (and default subtype) depends on Type <ul style="list-style-type: none">• Column: Plain (Default) Stacked PercentStacked• Bar: Plain (Default) Stacked PercentStacked• Line: Plain (Default) Smooth• Pie: Plain (Default) Exploded• Scatter: Plain (Default) Line SmoothLine• Bubble: Plain (Default)• Area: Plain (Default) Stacked PercentStacked• Doughnut: Plain (Default) Exploded• Stock: HighLowClose (Default) OpenHighLowClose Candlestick
SeriesGroupings	0-1	Element	Set of series groupings for the chart
CategoryGroupings	0-1	Element	Set of category (X) groupings for the chart
ChartData	0-1	Element	Defines the data values for the chart
Legend	0-1	Element	Defines the chart legend
CategoryAxis	0-1	Element	Defines the category axis
ValueAxis	0-1	Element	Defines the value axis
Title	0-1	Element	Defines a title for the chart
PointWidth	0-1	Integer	Non-zero Percent width for bars and

			columns. A value of 100 represents 100% of the distance between points (i.e. a value greater than 100 will cause columns to overlap each other).
Palette	0-1	Enum	Determines the color palette for the chart items. Values are Default (Default) EarthTones Excel GrayScale Light Pastel SemiTransparent
ThreeDProperties	0-1	Element	Properties for a 3D chart layout.
PlotArea	0-1	Element	Properties for the plot area
ChartElementOutput	0-1	Enum	Indicates whether a DataPoints element containing the chart data points should appear in a data rendering. Output (Default): Indicates the DataPoints should appear in the output NoOutput: Indicates the DataPoints should not appear in the output

A Chart must have either a SeriesGrouping, or a CategoryGrouping or both.

SeriesGroupings

The SeriesGroupings element defines the set of Series groupings for a Chart.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
SeriesGrouping	1-N	Element	The set of Series groupings for the Chart

SeriesGrouping

The SeriesGrouping element defines a Series level for a Chart SeriesGrouping. The labels for the series are displayed in the legend.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
DynamicSeries	0-1	Element	Dynamic Series headings for this grouping

StaticSeries	0-1	Element	Static Series headings for this grouping.
Style	0-1	Element	Defines border and background style properties for the series legend item(s) and data points in the series. When applied to DynamicSeries, style expressions are evaluated per group instance.

A SeriesGrouping must have either DynamicSeries or StaticSeries, but not both.

DynamicSeries

The DynamicSeries element defines a Series level that repeats with each Series group in a Chart's rowset.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Grouping	1	Element	The expressions to group the data by. Page breaks in the grouping are not allowed.
Sorting	0-1	Element	The expressions to sort the columns by
Label	1	Expression (String)	The label displayed on the legend.

StaticSeries

The StaticSeries element defines a series level with a fixed set of members.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
StaticMember	1-N	Element	The members for the static series

StaticMember

The StaticMember element defines the data label to display for a static series or static category member.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Label	1	Expression (Variant)	The label for the static member (displayed either on the category axis or legend, as appropriate).

CategoryGroupings

The CategoryGroupings element defines the set of Category groupings for a Chart

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
CategoryGrouping	1-N	Element	The set of Category groupings for the Chart

CategoryGrouping

The CategoryGrouping element defines a Category level for a Chart. The labels for the categories are displayed along the category axis.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
DynamicCategories	0-1	Element	Dynamic Category headings for this grouping
StaticCategories	0-1	Element	Static Category headings for this grouping

A CategoryGrouping must have either DynamicCategories or StaticCategories, but not both.

DynamicCategories

The DynamicCategories element defines a Category header region that repeats with each Category group in a Chart's rowset. The label displayed on the axis is defined on the Grouping.Label property.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Grouping	1	Element	The expressions to group the data by. Page breaks in the grouping are not allowed.

Sorting	0-1	Element	The expressions to sort the data by
Label	0-1	Expression (Variant)	The label displayed on the axis.

StaticCategories

The StaticCategories element defines a category level with a fixed set of members. There must be a corresponding DataPoint for each StaticCategories member.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
StaticMember	1-N	Element	The members of the static category level

Title

The Title element defines a title for the chart or for an axis

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Caption	0-1	Expression (string)	Caption of the title
Style	0-1	Element	Defines text, border and background style properties for the title. All Textbox properties apply.
Position	0-1	Enum	The position of the title Center (Default) Near Far Not used for chart title

Legend

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

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Visible	0-1	Boolean	Specifies whether a legend is displayed.

			Defaults to false.
Style	0-1	Element	Defines text, border and background style properties for the legend. All Textbox properties apply.
Position	0-1	Enum	The position of the legend TopLeft TopCenter TopRight LeftTop LeftCenter LeftBottom RightTop (Default) RightCenter RightBottom BottomRight BottomCenter BottomLeft.
Layout	0-1	Enum	The arrangement of labels within the legend Column (Default) Row Table
InsidePlotArea	0-1	Boolean	If true, draw legend inside plot area, otherwise draw outside plot area (default).

CategoryAxis

The CategoryAxis element defines the category (X) axis.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Axis	1	Element	Display properties for the category axis

ValueAxis

The ValueAxis element defines the data (Y) axis.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Axis	0-1	Element	Display properties for the value axis.

Axis

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

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Visible	0-1	Boolean	Whether the axis labels are displayed. Defaults to false.
Style	0-1	Element	Defines text style properties for the axis labels and line style properties for the axis line.
Title	0-1	Element	Defines a title for the axis
Margin	0-1	Boolean	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. Defaults to false.
MajorTickMarks	0-1	Enum	None (Default) Inside Outside Cross
MinorTickMarks	0-1	Enum	None (Default) Inside Outside Cross
MajorGridLines	0-1	Element	Indicates major gridlines should be displayed for this axis.
MinorGridLines	0-1	Element	Indicates minor gridlines should be displayed for this axis.
MajorInterval	0-1	Expression (Variant)	Unit for major gridlines/tickmarks If omitted, the axis is autodivided
MinorInterval	0-1	Expression (Variant)	Unit for minor gridlines/tickmarks If omitted, the axis is autodivided
Reverse	0-1	Boolean	If false (Default) the axis is plotted normally, if true its direction is reversed.
CrossAt	0-1	Expression (Variant)	Value at which to cross the other axis If omitted, uses the default behavior for the chart type.
Interlaced	0-1	Boolean	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.

			Defaults to False.
Scalar	0-1	Boolean	Indicates the values along this axis are scalar values (i.e. 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 grouping, if that grouping is static or has more than one group expression or if the axis values have a label.
Min	0-1	Expression (Variant)	Minimum value for the axis If omitted, the axis autoscales
Max	0-1	Expression (Variant)	Maximum value for the axis If omitted, the axis autoscales
LogScale	0-1	Boolean	Whether the axis is logarithmic. Default is false.

ChartData

The ChartData element defines the segmentation of the data into multiple series. There must be a corresponding ChartSeries for each StaticSeries element.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
ChartSeries	1-N	Element	Data points for each series in the chart

ChartSeries

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

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
DataPoints	1	Element	Data points within a series
PlotType	0-1	Enum	Indicates whether the series should be plotted as a line in a Column chart. If set to auto, should be plotted per the primary chart type. Auto (Default) Line

DataPoints

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

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
DataPoint	1-N	Element	Collection of data points

DataPoint

The DataPoint element defines a set of single data point for the chart. There must be a corresponding DataPoint for combination of StaticCategories and StaticSeries. A DataPoint may consist of a single value expression (for example in bar or line charts) or multiple value expressions (stock and bubble charts).

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
DataValues	1	Element	Data value set for the Y axis.
DataLabel	0-1	Element	Indicates the values should be marked with data labels.
Action	0-1	Element	Action to execute.
Style	0-1	Element	Defines border and background style properties for the data point.
Marker	0-1	Element	Defines marker properties. Markers do not apply to data points of pie, doughnut and any stacked chart types.
DataElementName	0-1	String	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.
DataElementOutput	0-1	Enum	Indicates whether the data point should appear in a data rendering. Output (default): Indicates the data point

			should appear in the output NoOutput: Indicates the data point should not appear in the output
--	--	--	--

DataValues

The DataValues element defines a collection of data values for a single data point in the chart.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
DataValue	1-N	Element	Set of DataValue expressions for a single DataPoint.

DataValue

The DataValue element defines a single value for the DataPoint in the chart.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Value	1	Expression (Variant)	Value expression. Same restrictions as the expressions in a matrix cell

DataLabel

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

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Style	0-1	Element	Defines text, border and background style properties for the labels. Supplied styles override Series styles.
Value	0-1	Expression (Variant)	Expression for the value labels. If omitted, values of in the ValueAxis are used for labels.
Visible	0-1	Boolean	Whether the data label is displayed on the chart. Defaults to False.
Position	0-1	Enum	Position of the label Auto (Default) Top TopLeft TopRight

			Left Center Right BottomRight Bottom BottomLeft
Rotation	0-1	Integer	Angle of rotation of the label text

Marker

The Marker element defines a marker for displayed chart values.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Type	0-1	Enum	Defines the marker type for values. None (Default) Square Circle Diamond Triangle Cross Auto
Size	0-1	Size	Represents the height and width of the plotting area of marker(s).
Style	0-1	Element	Defines the border and background style properties for the marker(s).

ThreeDProperties

The ThreeDProperties element defines properties for 3D layout.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Enabled	0-1	Boolean	Whether or not a chart is displayed in 3D. Default is False (2D).
ProjectionMode	0-1	Enum	The projection mode used for the 3D rendering Perspective (Default) Orthographic
Perspective	0-1	Integer	Represents the percent of perspective Applies only for Perspective projection
Rotation	0-1	Integer	Rotation angle
Inclination	0-1	Integer	Inclination angle
Shading	0-1	Enum	None (Default) Simple Real
WallThickness	0-1	Integer	Percent thickness of outer walls

DrawingStyle	0-1	Enum	Determines shape of chart data displayed Cylinder Cube (Default) Only applies to bar and column chart types.
Clustered	0-1	Boolean	Determines if data series are clustered (displayed along distinct rows). Only applies to bar and column chart types. Defaults to false.

PlotArea

The PlotArea element defines properties for the plot area (for chart types with X and Y axes).

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Style	0-1	Element	Defines borders and background for the plot area

MajorGridLines

The MajorGridLines element defines style properties for major gridlines along an axis

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
ShowGridLines	0-1	Boolean	Indicates the gridlines should be shown
Style	0-1	Element	Line style properties for the gridlines and tickmarks

MinorGridLines

The MinorGridLines element defines style properties for minor gridlines along an axis

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
ShowGridLines	0-1	Boolean	Indicates the gridlines should be shown
Style	0-1	Element	Line style properties for the gridlines and tickmarks

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 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.

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

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
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 below).
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 with no border.
CustomData	0-1	Element	Defines data to be passed to the custom report item control.

Example:

```
<CustomReportItem Name="Text5">
  <Type>RichTextControl</Type>
  <Top>2 in</Top>
  <Left>2 in</Left>
  <Height>1 in</Height>
  <Width>4 in</Width>
  <CustomProperties>
    <CustomProperty>
      <Name>ms:RTFValue</Name>
      <Value>{\b\i Pretty}{\b formatted text}</Value>
    </CustomProperty>
  </CustomProperties>
  <AltReportItem>
```



```

        <Textbox Name="Textbox6">
            <Value>Boring plain text</Value>
        </Textbox>
    </AltReportItem>
</CustomReportItem>

```

All report item Style properties apply to custom report items.

Note: ReportItems from the first version of RDL should become a rectangle inside AltReportItem.

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 (e.g. placement of data regions in page headers) apply to the AltReportItem as if it were a regular report item in the report.

Attributes/Elements

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

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 impacts such things as restricting placement within the report (i.e. not allowed in detail rows, page header or page footer), available scopes for expressions and RepeatWith.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
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
DataColumnGroupings	0-1	Element	The set of column groupings for the data
DataRowGroupings	0-1	Element	The set of row groupings for the data
DataRows	0-1	Element	The data values to calculate for each detail “cell” of the data

DataColumnGroupings

The DataColumnGroupings element defines the set of column groupings for the data

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
DataGroupings	1	Element	The set of column groupings

DataRowGroupings

The DataRowGroupings element defines the set of row groupings for the data

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
DataGroupings	1	Element	The set of row groupings.

DataGroupings

The DataGroupings element defines a set of groupings for the data.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
DataGrouping	1-N	Element	An ordered set of groupings.

DataGrouping

The DataGrouping element defines a grouping for the data.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Static	0-1	Boolean	Indicates this is a static grouping. Must be true if any ancestor grouping is a detail grouping.
Grouping	0-1	Element	The expressions by which to group the data Not allowed if Static is true. If omitted and Static is false, indicates this is a detail grouping (i.e. contains detail data).
Sorting	0-1	Element	The expressions by which to sort the group instances. Ignored if Static is true.
CustomProperties	0-1	Element	Custom properties for the grouping
DataGroupings	0-1	Element	Subgroupings contained within this grouping

Automatic Subtotals and Subgroups

When an automatic subtotal is requested, all subgroups are preserved in the subtotal, but are treated as static groups.

For example, consider the following custom report item:

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

The Year group and the Quarter group 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

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
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 (i.e. has no sub-groupings) DataGroupings in DataRowGroupings.

DataRow

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

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
DataCell	1-N	Element	A set of data values to pass to the custom report item. There must be as many DataCell elements as there are leaf-node (i.e. has no sub-groupings) DataGroupings in DataColumnGroupings.

DataCell

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

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
DataValue	1-N	Element	A data value to pass to the custom report item

DataValue

The DataValue element defines a data value to pass to the custom report item.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Name	0-1	Expression (String)	Identifier for the data value Note: Since this element is the same as the DataValue element in Chart, the Name property will need to be added to Chart. Note: Property names are not required to be unique
Value	1	Expression (Variant)	The value to pass to the custom report item

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-based Style elements evaluate to either the type listed or to *Nothing* (an error in evaluation is treated as *Nothing*). *Nothing* is equivalent to 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 above for the formats for each string type.

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
BorderColor	0-1	Element	Color of the border
BorderStyle	0-1	Element	Style of the border
BorderWidth	0-1	Element	Width of the border
BackgroundColor	0-1	Expression	Color of the background

		(Color)	If omitted, the background is transparent
BackgroundGradientType	0-1	Expression (Enum)	The type of background gradient None (Default) LeftRight TopBottom, Center DiagonalLeft DiagonalRight HorizontalCenter VerticalCenter
BackgroundGradientEndColor	0-1	Expression (Color)	End color for the background gradient. If omitted, there is no gradient.
BackgroundImage	0-1	Element	A background image for the report item. If omitted, there is no background image.
FontStyle	0-1	Expression (Enum)	Font style Normal (Default) Italic
FontFamily	0-1	Expression (String)	Name of the font family Default: Arial
FontSize	0-1	Expression (Size)	Point size of the font Default: 10 pt. Min: 1 pt. Max: 200 pt.
FontWeight	0-1	Expression (Enum)	Thickness of the font Lighter Normal (Default) Bold Bolder 100 200 300 400 500 600 700 800 900
Format	0-1	Expression (String)	.NET Framework formatting string ²² 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.

²² See <http://msdn.microsoft.com/library/en-us/cpguide/html/cpconformattingtypes.asp>

TextDecoration	0-1	Expression (Enum)	Special text formatting Underline Overline LineThrough None (Default)
TextAlign	0-1	Expression (Enum)	Horizontal alignment of the text Left Center Right General (Default)
VerticalAlign	0-1	Expression (Enum)	Vertical alignment of the text Top (Default) Middle Bottom
Color	0-1	Expression (Color)	The foreground color Default: Black
PaddingLeft	0-1	Expression (Size)	Padding between the left edge of the report item and its contents. ²³ Default: 0 pt. Max: 1000 pt.
PaddingRight	0-1	Expression (Size)	Padding between the right edge of the report item and its contents. ²³ Default: 0 pt. Max: 1000 pt.
PaddingTop	0-1	Expression (Size)	Padding between the top edge of the report item and its contents. ²³ Default: 0 pt. Max: 1000 pt.
PaddingBottom	0-1	Expression (Size)	Padding between the top edge of the report item and its contents, ²³ Default: 0 pt. Max: 1000 pt
LineHeight	0-1	Expression (Size)	Height of a line of text Default: Report output format determines line height based on font size Min: 1 pt. Max: 1000 pt.
Direction	0-1	Expression	Indicates whether text is written left-to-

²³ As both borders and padding are measured from the edge of the object, borders may overlap report item contents unless there is sufficient padding.

		(Enum)	<p>right or right-to-left.</p> <p>Does not impact the alignment of text unless using General alignment.</p> <p>LTR (Default) RTL</p>
WritingMode	0-1	Expression (Enum)	<p>Indicates whether text is written horizontally or vertically.</p> <p>lr-tb (default) tb-rl</p>
Language	0-1	Expression (Language)	<p>The primary language of the text.</p> <p>Default is Report.Language.</p>
UnicodeBiDi	0-1	Expression (Enum)	<p>Indicates the level of embedding with respect to the Bi-directional algorithm.</p> <p>Normal (default) Embed BiDi-Override</p>
Calendar	0-1	Expression (Enum)	<p>Indicates the calendar to use for formatting dates. Must be compatible in the .NET framework with the Language setting. Default is the default calendar for the Language of the report item.</p> <p>Gregorian Gregorian Arabic Gregorian Middle East French Gregorian Transliterated English Gregorian Transliterated French Gregorian US English Hebrew Hijri Japanese Korea Taiwan Thai Buddhist</p>
NumeralLanguage	0-1	Expression (Language)	<p>The digit format to use as described by its primary language. Any language is legal. Default is the Language property.</p>
NumeralVariant	0-1	Expression (Integer)	<p>The variant of the digit format to use. Currently defined values are:</p> <p>1: default, follow Unicode context rules</p> <p>2: 0123456789</p> <p>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.</p> <p>4: ko, ja, zh-CHS, zh-CHT only</p>

			5: ko, ja, zh-CHS, zh-CHT only 6: ko, ja, zh-CHS, zh-CHT only [Wide versions of regular digits] 7: ko only
--	--	--	--

BorderColor

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Default	0-1	Expression (Color)	Color of the border (unless overridden for a specific side). Default: Black.
Left	0-1	Expression (Color)	Color of the left border
Right	0-1	Expression (Color)	Color of the right border
Top	0-1	Expression (Color)	Color of the top border
Bottom	0-1	Expression (Color)	Color of the bottom border

BorderStyle

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Default	0-1	Expression (Enum)	Style of the border (unless overridden for a specific side) None (Default) Dotted Dashed Solid Double Groove Ridge Inset WindowInset Outset
Left	0-1	Expression (Enum)	Style of the left border
Right	0-1	Expression (Enum)	Style of the right border
Top	0-1	Expression (Enum)	Style of the top border
Bottom	0-1	Expression (Enum)	Style of the bottom border

BorderWidth

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Default	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
Left	0-1	Expression (Size)	Width of the left border. Max: 20 pt Min: 0.25 pt
Right	0-1	Expression (Size)	Width of the right border. Max: 20 pt Min: 0.25 pt
Top	0-1	Expression (Size)	Width of the top border. Max: 20 pt Min: 0.25 pt
Bottom	0-1	Expression (Size)	Width of the bottom border. Max: 20 pt Min: 0.25 pt

Overlapping Borders

In the event two borders overlap and an output format supports overlapping borders, one will be drawn over the other. If an output format does not support overlapping borders, one of the border styles will override (replace) the other.

- Container borders are drawn over / override borders on contents
Group headers/footers and table headers/footers are considered containers for the inner groups/details
- For peer report items, borders of higher Zindex items take precedence
- For output formats that do not support overlapping borders and containment and ZIndex cannot determine precedence, the overlapping border uses
 - The lower color number
 - The style closer to the end of the style Enum list
 - The wider border width

BackgroundImage

Attributes/Elements

<i>Name</i>	<i>Card</i>	<i>Type</i>	<i>Description</i>
Source	1	Enum	Identifies the source of the image: External – The Value contains a constant or expression that evaluates to for the location of the image. This can be a full folder path (e.g. “/images/logo.gif”), relative path (e.g. “logo.gif”) or URL (e.g. “http://myserver/images/logo.gif”). Embedded – The Value contains a constant or expression that evaluates to the name of an EmbeddedImage within the report Database – The Value contains an expression (a field in the database) that evaluates to the binary data for the image.
Value	1	Expression (String)	See Source. Expected datatype is string or binary, depending on Source. If the Value is null, no background image is displayed.
MIMEType	0-1	Expression (String)	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.
BackgroundRepeat	0-1	Expression (Enum)	Indicates how the background image should repeat to fill the available space: vertically (y), horizontally (x), both or neither Repeat (Default) NoRepeat RepeatX RepeatY

Style Properties and ReportItem Types

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

	Line	Rectangle ²⁴	Textbox	Image	Subreport ²⁵	List ²⁵	Matrix ²⁵	Table ²⁵	Chart ²⁶	Body	Sub-total
BorderColor	Y ²⁷	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
BorderStyle	Y ²⁷	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
BorderWidth	Y ²⁷	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
BackgroundColor	N	Y	Y	N	N	Y	Y	Y	Y	Y	Y
BackgroundGradientType	N	N	N	N	N	N	N	N	Y	N	N
BackgroundGradientEndColor	N	N	N	N	N	N	N	N	Y	N	N
BackgroundImage	N	Y	Y	N	N	Y	Y	Y	N	Y	Y
FontStyle	N	N	Y	N	N	N	N	N	N	N	Y
FontFamily	N	N	Y	N	N	N	N	N	N	N	Y
FontSize	N	N	Y	N	N	N	N	N	N	N	Y

²⁴ This includes PageHeader and PageFooter

²⁵ All Textbox properties apply in the event of NoRows or for subreports that fail to execute.

²⁶ Different elements within the Chart support different Style properties. These are described in the places where Style is referenced in the element definition.

²⁷ 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.

	Line	Rectangle ²⁴	Textbox	Image	Subreport ²⁵	List ²⁵	Matrix ²⁵	Table ²⁵	Chart ²⁶	Body	Sub-total
FontWeight	N	N	Y	N	N	N	N	N	N	N	Y
Format	N	N	Y	N	N	N	N	N	N	N	Y
TextDecoration	N	N	Y	N	N	N	N	N	N	N	Y
TextAlign	N	N	Y	N	N	N	N	N	N	N	Y
VerticalAlign	N	N	Y	N	N	N	N	N	N	N	Y
Color	N	N	Y	N	N	N	N	N	N	N	Y
PaddingLeft	N	N	Y	Y	N	N	N	N	N	N	Y
PaddingRight	N	N	Y	Y	N	N	N	N	N	N	Y
PaddingTop	N	N	Y	Y	N	N	N	N	N	N	Y
PaddingBottom	N	N	Y	Y	N	N	N	N	N	N	Y
LineHeight	N	N	Y	N	N	N	N	N	N	N	Y
CanSort	N	N	Y	N	N	N	N	N	N	N	Y
Direction	N	N	Y	N	N	N	N	N	N	N	Y
Language	N	N	Y	N	N	N	N	N	N	Y	N
UnicodeBiDi	N	N	Y	N	N	N	N	N	N	N	Y
Calendar	N	N	Y	N	N	N	N	N	N	N	N
NumeralLanguage	N	N	Y	N	N	N	N	N	N	N	N
NumeralVariant	N	N	Y	N	N	N	N	N	N	N	N
WritingMode	N	N	Y	N	N	N	N	N	N	N	Y

Expressions

Expression Syntax

All expressions in RDL begin with the character “=” and are defined in a Visual Basic .NET compatible syntax (see <http://msdn.microsoft.com/library/en-us/vbls7/html/vblrfvbspec9.asp>). Values that do not begin with “=” are 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 StartVisible property expects a Boolean, so 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 within the report are declared at the Report level via the CodeModules element.

Static methods in custom assemblies are accessible globally within the report.

ClassName.MethodName(...)

For example: MyCurrencyConverterClass.Convert(...)

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

Code.ClassName.MethodName(...)

For example: Code.CurrencyConverter.Convert(...)

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
Boolean	Boolean
Integer	Int16, Int32, Int64, UInt16, UInt32, UInt64, Byte,

	Sbyte, Timespan
DateTime	DateTime
Float	Single, Double, Decimal
Binary	Byte[]
Variant	Any of the above except Byte[]
VariantArray	Array of Variant

If an expression returns any other type or returns a type that is not permitted for expression element, an error/warning will be generated. In the case of Label, Textbox.Value and all Style expressions, expression errors are treated as warnings and null will be returned instead. If an RDL type is defined as the return type for a function described below, the underlined CLR type is used (unless otherwise specified).

Global Collections

There are five 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 ²⁸	ReportItem
Globals	Global variables	Variant
User	User-specific data	Variant

²⁸ 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.

DataSources	Data sources in the report ²⁹	DataSource
DataSets	Data sets the report	DataSet

Items in the collections are accessed by name using Visual Basic-style 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

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.
ReportFolder	String	Implementation-specific path of the folder containing the report, e.g. /salesreports/budgeting or c:\sales\budget
ReportName	String	Name of the report, e.g. currentbudget

²⁹ 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.

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 code 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 below. All other properties are of type Object.

Predefined Field Properties

Property	Type	Expected Values
Value	Object ³⁰	
IsMissing	Boolean	
UniqueName	String	
BackgroundColor	String	See Style.BackgroundColor
Color	String	See Style.Color
FontFamily	String	See Style.FontFamily
FontSize	String	See Style.FontSize
FontWeight	String	See Style.FontWeight
FontStyle	String	See Style.FontStyle
TextDecoration	String	See Style.TextDecoration
FormattedValue	String	
Key	Object	
LevelNumber	Integer	
ParentUniqueName	String	

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.

³⁰ 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/frlrfssystemdataidatreaderclassgetschematabletopic.asp>

- 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 of an aggregate) are undefined.

Report Items

Only textboxes appear in the ReportItems collection.

The ReportItem object has Value property that can be accessed via either property syntax or collection syntax (see Fields, above). The only property defined for ReportItem is Value. 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 of Aggregate functions.

The data type of Value is Object.

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.

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.

DataSources

The DataSource object has the following properties:

DataProvider – 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).

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
Page Header/Footer	No	At most one ³¹	Yes	Yes	No
Body	Yes ³²	Only those in current or ancestor scope ³³	Yes	No	Yes
Report Parameter	No	No	Only earlier parameters	No	No
Field	Yes	No	Yes	No	No
Query Parameter	No	No	Yes	No	No
Group Expression	Yes	No	Yes	No	Yes
Sort By	Yes	No	Yes	No	Yes
Code	No	No	Yes ³⁴	No	No

³¹ 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.

³² 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.

³³ 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.

³⁴ If methods defined in <Code> are used within Parameter initialization, the Parameters collection will be empty, as it has not yet been initialized.

Language	No	No	Yes	No	No
----------	----	----	-----	----	----

Note: Since references to items in global collections can be dynamic (e.g. “=ReportItems(Parameters!Param1.Value)”), all error checking will need to happen both during report publishing (to catch static disallowed references) and during report execution (to catch dynamic disallowed references).

Cyclic expressions (e.g. Textbox1.Value=Textbox2.Value+1; Textbox2.Value=Textbox1.Value+1) are not allowed, but processing implementations should check at execution time as authoring tools may not be able to check them.

Aggregate Functions

RDL supports the following list of aggregate functions.

<i>Function</i>	<i>Arguments</i>	<i>Type</i> ³⁵	<i>Description</i>
Sum	<i>Return</i>	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	Integer or Float	The expression to aggregate. Cannot contain any aggregate functions.
	Scope	String	Name of a DataSet or the name of a Grouping 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.
Avg	<i>Return</i>	Float	Returns the average of all non-null values of the expression within the scope See Sum regarding return type
	Expression	Integer or Float	See Sum
	Scope	String	See Sum
Max	<i>Return</i>	Variant	Returns the maximum of all non-null values of the expression within the scope Return type is the same as the expression type.

³⁵ 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.

	Expression	Variant	See Sum
	Scope	String	See Sum
Min	<i>Return</i>	Variant	Returns the minimum of all non-null values of the expression within the scope Return type is the same as the expression type.
	Expression	Variant	See Sum
	Scope	String	See Sum
Count	<i>Return</i>	Integer	Returns the count of all non-null values of the expression within the scope
	Expression	Variant or Binary	See Sum
	Scope	String	See Sum
CountDistinct	<i>Return</i>	Integer	Returns the count of all distinct non-null values of the expression within the scope
	Expression	Variant	See Sum
	Scope	String	See Sum
CountRows	<i>Return</i>	Integer	Returns the count of all rows within the scope Syntax: “CountRows(<i>Scope</i>)”
	Scope	String	See Sum
StDev	<i>Return</i>	Float	Returns the standard deviation of all non-null values of the expression within the scope
	Expression	Integer or Float	See Sum
	Scope	String	See Sum
StDevP	<i>Return</i>	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	Integer or Float	See Sum
	Scope	String	See Sum

Var	<i>Return</i>	Float	Returns the variance of all non-null values of the expression within the scope See Sum regarding return type
	Expression	Integer or Float	See Sum
	Scope	String	See Sum
VarP	<i>Return</i>	Float	Returns the population variance of all non-null values of the expression within the scope See Sum
	Expression	Integer or Float	See Sum
	Scope	String	See Sum
	Scope	String	

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

<i>Function</i>	<i>Arguments</i>	<i>Type</i>³⁶	<i>Description</i>
First	<i>Return</i>	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

³⁶ 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.

Last	<i>Return</i>	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	<i>Return</i>	Variant or Binary	Returns the value of the expression for the previous row of data. Returns Nothing if there is no previous row of data.
	Expression	Variant or Binary	The expression for which to retrieve the previous value. Cannot contain any aggregate functions. The functions Level() and InScope may not be used in the expression.
RunningValue	<i>Return</i>	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 Grouping 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. Optional. Default: Nothing

RowNumber	<i>Return</i>	Integer	The row number of the current row or group instance.
	<i>Scope</i>	String	See RunningValue
Aggregate	<i>Return</i>	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.
	<i>Expression</i>	N/A	The expression to aggregate. Must be a simple field reference (e.g. =Aggregate(Fields!Sales.Value,Year))
	<i>Scope</i>	String	See Sum All group expressions for the Scope (and all containing grouping 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.

If omitted, the scope is the innermost scope containing the report item in which the aggregate is used.³⁷ Aggregates that appear inside a DetailRow, therefore, use the innermost grouping as their default scope and in lists without grouping, the default scope is the entire list.

Specifying the keyword *Nothing* as the scope is equivalent to specifying the outermost data region containing the report item in which the aggregate is used.

For expressions outside of data regions (in the report body):

³⁷ For RunningValue and RowNumber used in a MatrixCell, the default is the innermost column scope. For other aggregates in a MatrixCell, the default scope is the cell itself (the intersection of the innermost row scope and innermost column scope).

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.

The keyword *Nothing* is not allowed.

For expressions in page headers and footers:

The scope argument is not allowed in page headers/footers (It is not needed, since expressions in page headers/footers cannot refer to fields)

Recursive

This is an optional final argument for each of the aggregate functions (other than First, Last, Level, Previous, RowNumber, RunningValue and Aggregate).

Type: Enum (Recursive | Simple). Default: Simple.

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:

```
<List>
  <Grouping Name="Employee">
    <GroupExpressions>
```

```

        <GroupExpression>=Fields!EmployeeID.Value</GroupExpression>
    </GroupExpressions>
    <Parent>=Fields!ManagerID.Value</Parent>
</Grouping>
<ReportItems>
    <Textbox><Value>=Fields!EmployeeID.Value</Value></Textbox>
    <Textbox><Value>=Fields!ManagerID.Value</Value></Textbox>
    <Textbox><Value>=Sum(Fields!Sales.Value)</Value></Textbox>
    <Textbox><Value>=Sum(Fields!EmployeeID,"Employee",Recursive)</Value></Tex
tbody>
    </ReportItems>
</List>

```

Recursive Depth

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

<i>Function</i>	<i>Arguments</i>	<i>Type</i> ³⁸	<i>Description</i>
Level	<i>Return</i>	Integer	A zero-based integer representing the current depth level of a recursive hierarchy. If the specified scope is a dataset, data region or grouping without Parent or the scope does not exist, Level returns 0.
	Scope	String	Optional. Defaults to the current scope.

³⁸ 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.

Restrictions on Aggregate Usage

Context	Running Value	Row Number	First / Last	Previous	Other Aggregates	Report Item Aggregates
Page Header/Footer	No	No	Yes	No	Yes	Yes
Body	Yes	Yes	Yes	Yes	Yes	No
Report Parameter	No	No	No	No	No	No
CalculatedField	No	No	No	No	No	No
Query Parameter	No	No	No	No	No	No
Group Expression	No	Yes	No	No	No	No
Sort By	No	No	No	No	Yes	No
Matrix Cell	Yes ³⁹	Yes ³⁹	Yes	Yes	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

³⁹ Within a MatrixCell, the Scope argument for RunningValue and RowNumber must refer to either a Column Grouping or a Row Grouping for the Matrix or to a scope contained within this MatrixCell. The scope of the running value/row number defines the direction of the running value. RunningValue and RowNumber in a matrix may use either column scopes or row scopes but not both.

- First, Last, Previous, RunningValue and RowNumber are applied after containing group and data region filters are applied

Dynamic Scoping

Report items contained within a cell of a matrix with subtotals have dynamic scoping.

For example, consider a matrix that has a year column grouping and a product row grouping (with a subtotal on each). If the value of the textbox in the detail cell is =Sum(Fields!Sales.Value), each detail cell will be grouped on both year and product. However, the year subtotal 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:

<i>Function</i>	<i>Arguments</i>	<i>Type</i>	<i>Description</i>
InScope	<i>Return</i>	Boolean	True if the current instance is within the specified scope
	<i>Scope</i>	String	Name of a DataSet, Grouping or DataRegion

A typical usage for the InScope function is to construct parameters for drill-through reports which will work even in both matrix details cells and 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:

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

Questions & Answers

Q: How does RDL compare to banded report definitions?

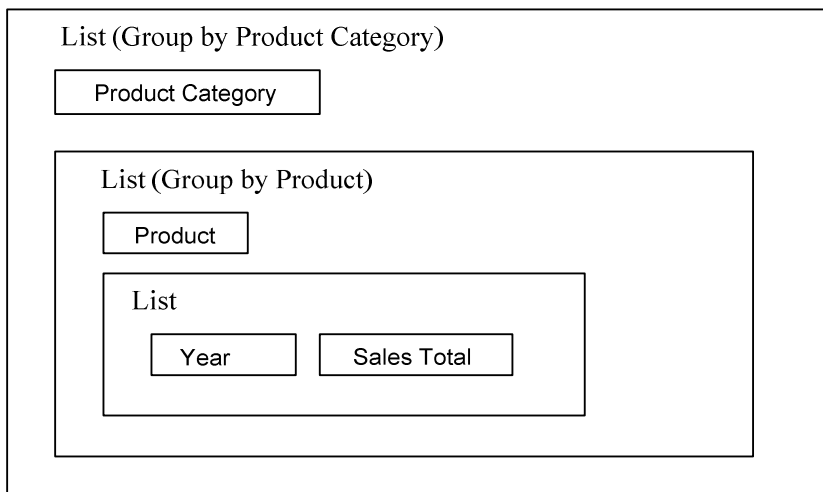
A: The following chart and diagrams illustrate how RDL is a superset of banded report definitions.

Feature	Banded Reports	RDL
Group	Band	Data region (group)
Details	Band	Data region (list)
Graph	Essentially a subreport	Data region (graph)
Crosstabs	N/A	Data region (matrix)
Tables	Manual layout	Data region (table)
Delimited sublayout	Subreport	Data region or subreport
Side-by-side layout	Must embed in subreports	Multiple data regions
Multiple use of single dataset	Must embed in subreports	Multiple data regions
Independent one-to-many	Subreport	Subreport

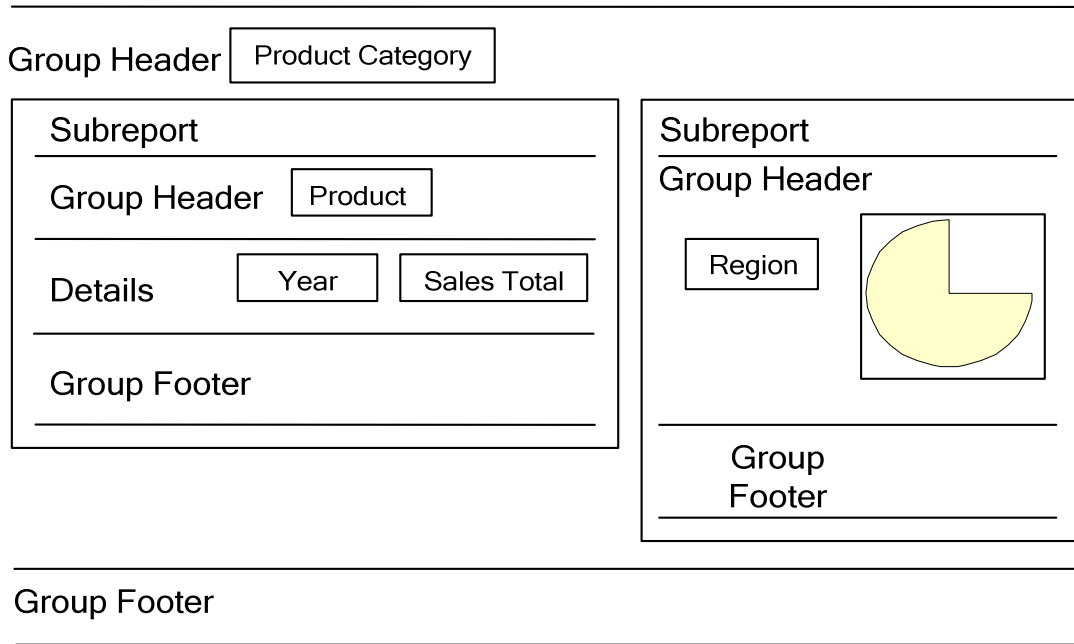
Banded Sample Grouping Model

Group Header	Product Category	
Group Header	Product	
Details	Year	Sales Total
Group Footer		
Group Footer		

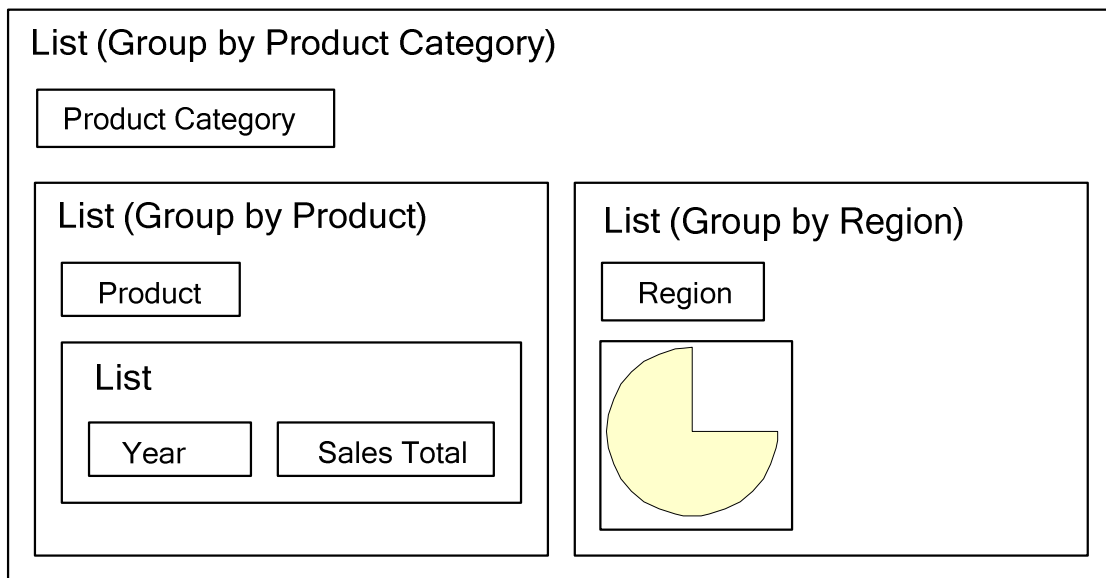
RDL Sample Grouping Model



Banded Side-By-Side Example



RDL Side-By-Side Example



Q: How does drilldown work?

A: In a drilldown layout, the query contains the drilled-down items. The layout contains drilled-down items, which are marked as initially hidden.

Q: How do I page break on a specific number of rows?

A: Use a Grouping with a PageBreakAtEnd property. For your grouping expression, use a count running value to derive a unique group value for each block of N rows. (e.g. =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 (e.g. =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 matrixes?

A: The Scope argument for RunningValue and RowNumber can refer to either a column grouping or a row grouping. 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 matrix with two groupings 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 matrix growth keeps pushing it off the page. How do I stop this?

A: Group the matrix with the blank space to its right by using a rectangle. Since growth only pushes peer items out of the way, the matrix 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 table should grow to fill the blank space below it (rather than preserving the blank space)?

A: Group the table with the blank space below it by using a rectangle. Since growth only pushes peer items out of the way, the table in the rectangle has no items to push down below it, so it will consume the blank space until it fills the rectangle.

Changes in this Version

This section summarizes the changes since the initial release of the RDL specification in December, 2003.

Major New Elements

- Multi-value parameters
- Extended field properties
- End-user sort
- Fixed headers
- Hidden parameters
- Custom report items

Minor New Elements

- Allow expressions for data set connection string
- Allow expressions for additional Chart axis properties
- Added Style to SeriesGrouping in Chart
- Added GUID as string data type
- Added DataSources collection
- Added DataSets collection
- Added InteractiveHeight and InteractiveWidth to report

Minor Changes

- Restricted DataElementName properties to CLS-compliant identifiers
- Restricted PageWidth and PageHeight to nonzero values
- Restricted bookmark links and drillthrough actions in page header/footer
- Modified behavior of DataElementOutput = Auto

Index

Action, 35
Axis, 68
BackgroundImage, 86
Body, 25
BorderColor, 84
BorderStyle, 84
BorderWidth, 85
CategoryAxis, 68
CategoryGrouping, 66
CategoryGroupings, 66
Chart, 62
ChartData, 70
ChartSeries, 70
Class, 24
Classes, 24
CodeModules, 24
ColumnGrouping, 51
ColumnGroupings, 51
ConnectionProperties, 22
Corner, 51
DataLabel, 72
DataPoint, 71
DataPoints, 71
DataRegion, 44
DataSet, 18
DataSetReference, 15
DataSets, 17
DataSource, 21
DataSources, 21
DataValue, 72
DataValues, 72
DefaultValue, 17
Details, 61
Drillthrough, 36
DynamicCategories, 66
DynamicColumns, 52
DynamicRows, 54
DynamicSeries, 65
EmbeddedImage, 27
EmbeddedImages, 27
Field, 20
Fields, 19
Filter, 28
Filters, 28
FilterValues, 29
Footer, 60
GroupExpressions, 48
Grouping, 46
Header, 59
HeaderRow, 60
HeaderRows, 60
Image, 41

Legend, 67	RowGrouping, 53
Line, 38	RowGroupings, 53
List, 46	SeriesGrouping, 64
MajorGridLines, 74	SeriesGroupings, 64
Marker, 73	SortBy, 49
Matrix, 49	Sorting, 48
MatrixCell, 57	StaticCategories, 67
MatrixCells, 57	StaticColumn, 53
MatrixColumn, 56	StaticColumns, 52
MatrixColumns, 56	StaticMember, 65
MatrixRow, 57	StaticRow, 54
MatrixRows, 56	StaticRows, 54
MinorGridLines, 74	StaticSeries, 65
PageFooter, 26	Style, 80
PageHeader, 25	Subreport, 43
Parameter, 44	Subtotal, 55
Parameters, 44	Table, 58
ParameterValue, 16	TableCell, 62
ParameterValues, 16	TableCells, 62
PlotArea, 74	TableColumn, 59
Query, 21	TableColumns, 59
QueryParameter, 23	TableGroup, 61
QueryParameters, 23	TableGroups, 61
Rectangle, 38	Textbox, 38
Report, 11	ThreeDProperties, 73
ReportItem, 31	Title, 67
ReportItems, 30	ToggleImage, 40
ReportParameter, 13	ValidValues, 15
ReportParameters, 13	ValueAxis, 68

Values, 17

Visibility, 36