

# Schema documentation for ApplicationComponentsConfig.xsd

december 28, 2018

## Table of Contents

Namespace: "" .....	7
Schema(s) .....	7
Main schema ApplicationComponentsConfig.xsd .....	7
Element(s) .....	7
Element Component .....	7
Element Identification .....	7
Element Documentation .....	8
Element Versions .....	8
Element Version .....	8
Element UpgradeInformation .....	9
Element Configuration .....	9
Element ConfigurationData .....	10
Element Header .....	10
Element Heading .....	11
Element Namespace .....	11
Element Modules .....	11
Element Module .....	12
Element ReportDefaults .....	12
Element ReportDefaults / Tables .....	13
Element ReportDefaults / Tables / Table .....	13
Element ReportDefaults / Tables / Table / Keys .....	14
Element ReportDefaults / Tables / Table / Keys / Key .....	14
Element ReportDefaultColumnType / Fieldname .....	15
Element ReportDefaultColumnType / Fieldtype .....	15
Element ReportDefaultColumnType / Fieldlength .....	16
Element ReportDefaultColumnType / Nullable .....	16
Element ReportDefaults / Tables / Table / Attributes .....	16
Element ReportDefaults / Tables / Table / Attributes / Attribute .....	17
Element UIVersions .....	17
Element kw_UIVersions_type / UIVersion .....	18
Element Header / DataModelURI .....	18
Element Header / SaveURI .....	18
Element Header / Dependencies .....	18
Element Header / Dependencies / Dependency .....	18
Element Header / Dependencies / Dependency / Element .....	19
Element Elements .....	21
Element elLabel .....	21
Element UIVersionElements .....	22
Element kw_UIVersionElements_type / UIElement .....	22
Element kw_elLabel_type / Display .....	23
Element Label .....	24
Element kw_elLabel_type / Display / Xform .....	25
Element kw_elLabel_type / Display / Format .....	25
Element kw_elLabel_type / Display / Element .....	25
Element kw_elLabel_type / Display / Condition .....	27
Element kw_elLabel_type / Display / Condition / Element .....	28
Element kw_elLabel_type / Display / Condition / Value .....	29
Element kw_elLabel_type / Display / Condition / Element2 .....	29
Element Help .....	31
Element elField .....	31
Element kw_elField_type / Display .....	32
Element kw_elField_type / Display / Format .....	33
Element kw_elField_type / Display / Xform .....	33
Element kw_elField_type / Display / Condition .....	34
Element Condition_type / Element .....	35
Element Condition_type / ReferenceElement .....	36
Element kw_elField_type / EntryMode .....	37
Element kw_elField_type / EntryMode / Default .....	38
Element Default_type / Value .....	38
Element Default_type / InitiatialValue .....	39

Element Calc .....	39
Element CalcType / ColumnTotal .....	40
Element fnColumnTotal / Calculation .....	41
Element fnColumnTotal / RepeatID .....	42
Element fnColumnTotal / Element .....	42
Element fnColumnTotal / Filters .....	44
Element fnColumnTotal / Filters / Filter .....	44
Element Kwantu_Filter_type / Parameters .....	45
Element Kwantu_Filter_type / Parameters / Element .....	45
Element Kwantu_Filter_type / Parameters / Constant .....	47
Element Kwantu_Filter_type / Condition .....	47
Element Kwantu_Filter_type / Element .....	48
Element CalcType / DateDiff .....	50
Element fnDateDiff / Calculation .....	50
Element fnDateDiff / Element .....	50
Element CalcType / ElementCalc .....	52
Element fnElementCalc / Calculation .....	52
Element fnElementCalc / Element .....	53
Element CalcType / RowTotal .....	54
Element fnRowCalc / Calculation .....	54
Element fnRowCalc / Element .....	55
Element CalcType / ElementsList .....	57
Element fnElementsList / Element .....	57
Element fnElementsList / Calculation .....	59
Element CalcType / CalcFinYearDates .....	59
Element fnCalcFinYearDates / Calculation .....	60
Element fnCalcFinYearDates / Element .....	60
Element CalcType / CalcHistory .....	62
Element fnHistoryTotal / Calculation .....	62
Element fnHistoryTotal / Element .....	63
Element fnHistoryTotal / Filters .....	64
Element fnHistoryTotal / Filters / Filter .....	65
Element fnHistoryTotal / Filters / Filter / Parameters .....	66
Element fnHistoryTotal / Filters / Filter / Parameters / Element .....	66
Element fnHistoryTotal / Filters / Filter / Condition .....	68
Element Default_type / ElementValue .....	69
Element kw_elField_type / EntryMode / AssignElement .....	70
Element kw_elField_type / EntryMode / AssignSubElements .....	70
Element AssignSubElement_type / AssignCode .....	71
Element AssignSubElement_type / AssignName .....	71
Element AssignSubElement_type / AssignSubElement .....	72
Element ReadOnlyIf .....	73
Element ReadOnlyIf / Condition .....	73
Element Validations .....	74
Element kw_Validations_type / Required .....	75
Element DataType .....	75
Element Validation .....	75
Element kw_Validations_type / ErrorMessage .....	76
Element Reports .....	76
Element Column .....	77
Element ReportColumnType / Fieldname .....	78
Element ReportColumnType / Fieldtype .....	78
Element ReportColumnType / Fieldlength .....	78
Element ReportColumnType / Nullable .....	79
Element elSelect .....	79
Element kw_elSelect_type / Display .....	80
Element kw_elSelect_type / Display / Format .....	81
Element kw_elSelect_type / Display / Xform .....	81
Element Condition .....	82
Element kw_Condition_type / Element .....	82
Element kw_Condition_type / Value .....	84
Element kw_Condition_type / Element2 .....	84
Element kw_elSelect_type / EntryMode .....	85
Element kw_elSelect_type / EntryMode / Select .....	87
Element Codelist .....	88
Element Codelist / Taxonomy .....	89
Element kw_elSelect_type / EntryMode / Select / SharedData .....	89
Element kw_elSelect_type / EntryMode / Select / SharedData / Collection .....	90
Element kw_elSelect_type / EntryMode / Select / SharedData / TopElement .....	90
Element kw_elSelect_type / EntryMode / Select / SharedData / KeyValue .....	91
Element kw_elSelect_type / EntryMode / Select / SharedData / Filter .....	91
Element kw_elSelect_type / EntryMode / Select / SharedData / Filter / FilterElement .....	92

Element kw_elSelect_type / EntryMode / Select / SharedData / Filter / FilterValue .....	92
Element kw_elSelect_type / EntryMode / Select / SharedData / Filter / FilterValue / Element .....	92
Element kw_elSelect_type / EntryMode / Select / SharedData / Filter / FilterValue / Value .....	94
Element kw_elSelect_type / EntryMode / Select / Rest .....	94
Element RestType / HostID .....	94
Element RestType / ServiceURL .....	95
Element RestType / Parameters .....	95
Element RestType / Parameters / Parameter .....	96
Element RestType / Parameters / Parameter / Element .....	96
Element RestType / Parameters / Parameter / Constant .....	98
Element RestType / Return .....	98
Element RestType / Return / Result .....	99
Element RestType / Return / Status .....	100
Element RestType / Return / Status / StatusCode .....	101
Element RestType / Return / Status / StatusMessage .....	101
Element RestType / Return / Assign .....	101
Element RestType / IfError .....	102
Element kw_elSelect_type / EntryMode / Select / Taxonomy .....	103
Element kw_elSelect_type / EntryMode / Select / Refresh .....	103
Element kw_elSelect_type / EntryMode / Select / Refresh / ElementID .....	103
Element kw_elSelect_type / EntryMode / AssignSubElements .....	104
Element kw_elSelect_type / EntryMode / Assigns .....	104
Element Assigns_type / Assign .....	105
Element kw_elSelect_type / EntryMode / Default .....	106
Element Repeat .....	106
Element RepeatDef .....	107
Element RepeatDef / AddRows .....	110
Element RepeatDef / AddRows / UserAdd .....	113
Element Add .....	113
Element Delete .....	114
Element RepeatDef / AddRows / UserAdd / MaxIterations .....	114
Element RepeatDef / AddRows / UserAdd / MinIterations .....	114
Element RepeatDef / AddRows / UserAdd / AddLinesMode .....	114
Element RepeatDef / AddRows / AutoLoad .....	115
Element RepeatDef / AddRows / AutoLoad / BlankKlines .....	116
Element RepeatDef / AddRows / AutoLoad / Codelist .....	117
Element RepeatDef / AddRows / AutoLoad / Codelist / Taxonomy .....	117
Element RepeatDef / AddRows / AutoLoad / Periods .....	117
Element RepeatDef / AddRows / AutoLoad / Periods / StartDate .....	118
Element RepeatDef / AddRows / AutoLoad / Periods / StartDate / Element .....	119
Element RepeatDef / AddRows / AutoLoad / Periods / EndDate .....	120
Element RepeatDef / AddRows / AutoLoad / Periods / EndDate / Element .....	120
Element RepeatDef / AddRows / AutoLoad / Component .....	122
Element RepeatDef / AddRows / AutoLoad / Component / Filter .....	123
Element RepeatDef / AddRows / AutoLoad / Component / Filter / Value .....	123
Element RepeatDef / AddRows / AutoLoad / Component / Filter / ElementID .....	124
Element RepeatDef / AddRows / AutoLoad / Component / Filter / ElementCalc .....	125
Element RepeatDef / AddRows / AutoLoad / Component / Filter / ElementsList .....	125
Element RepeatDef / HeaderRow .....	126
Element SummaryRow .....	126
Element RepeatDef / Columns .....	127
Element RepeatDef / Columns / Column .....	127
Element KeyValue .....	128
Element RepeatHeader .....	128
Element RepeatHeader / rptColumn .....	129
Element elCalcField .....	130
Element elCalcField / Display .....	131
Element elCalcField / Display / Format .....	132
Element elCalcField / Display / Xform .....	132
Element elCalcField / EntryMode .....	132
Element elCalcField / EntryMode / Default .....	133
Element elCalcField / EntryMode / CalcField .....	134
Element elGroup .....	134
Element RepeatData .....	135
Element RepeatData / rptColumn .....	135
Element RepeatSummary .....	136
Element rptColumn .....	137
Element Hierarchy .....	138
Element HierarchyNode .....	138
Element elSelectGroup .....	139

Element elSelectGroup / SelectGroupKey .....	140
Element elSelectGroup / SelectGroup .....	141
Element TableName .....	142
Element Signature .....	142
Element NodeCount .....	142
Element Nodes .....	143
Element Node .....	143
Element FieldName .....	143
Element NumberColumns .....	144
Element CalcProcessType / ValidDate .....	144
Element fnProcessPeriodValidDate / Calculation .....	144
Element fnProcessPeriodValidDate / Number .....	145
Element fnProcessPeriodValidDate / Unit .....	145
Element CalcProcessType / DueDate .....	146
Element fnElementValue / Calculation .....	146
Element fnElementValue / Element .....	147
Element fnProcessPeriodDueDate / CalcDueDate .....	148
Element fnProcessPeriodDueDate / CalcDueDate / DueDate .....	149
Element fnProcessPeriodDueDate / CalcDueDate / DueDate / Calculation .....	150
Element fnProcessPeriodDueDate / CalcDueDate / DueDate / Number .....	151
Element fnProcessPeriodDueDate / CalcDueDate / DueDate / Unit .....	151
Element fnProcessPeriodDueDate / CalcDueDate / DueDate / Reference .....	151
Element fnProcessScheduleDueDate / CalcDueDate .....	152
Element fnProcessScheduleDueDate / CalcDueDate / DueDate .....	153
Element fnProcessScheduleDueDate / CalcDueDate / DueDate / Calculation .....	155
Element fnProcessScheduleDueDate / CalcDueDate / DueDate / Number .....	155
Element fnProcessScheduleDueDate / CalcDueDate / DueDate / Unit .....	156
Element fnProcessScheduleDueDate / CalcDueDate / DueDate / Reference .....	156
Element fnProcessScheduleDueDate / CalcDueDate / DueDate / Element .....	157
Element fnProcessScheduleDueDate / CalcDueDate / DueDate / ProcessID .....	158
Element kw_Label_type / Int .....	158
Complex Type(s) .....	159
Complex Type ReportDefaultColumnType .....	159
Complex Type kw_UIVersions_type .....	159
Complex Type KwantuElement_type .....	160
Complex Type kw_elLabel_type .....	161
Complex Type kw_UIVersionElements_type .....	162
Complex Type kw_Xform_type .....	163
Complex Type kw_Help_type .....	163
Complex Type kw_elField_type .....	164
Complex Type Condition_type .....	166
Complex Type Default_type .....	166
Complex Type CalcType .....	167
Complex Type fnColumnTotal .....	168
Complex Type Kwantu_Filter_type .....	169
Complex Type fnDateDiff .....	171
Complex Type fnElementCalc .....	171
Complex Type fnRowCalc .....	172
Complex Type fnElementsList .....	172
Complex Type fnCalcFinYearDates .....	173
Complex Type fnHistoryTotal .....	174
Complex Type AssignSubElement_type .....	175
Complex Type kw_Validations_type .....	176
Complex Type ReportColumnType .....	176
Complex Type kw_elSelect_type .....	177
Complex Type kw_Condition_type .....	180
Complex Type RestType .....	181
Complex Type Assigns_type .....	183
Complex Type kw_Repeat_type .....	184
Complex Type kw_elGroup_type .....	184
Complex Type kwElementMap .....	185
Complex Type GeoPoint_type .....	186
Complex Type NodeType .....	186
Complex Type CalcProcessType .....	187
Complex Type fnProcessPeriodValidDate .....	187
Complex Type fnElementValue .....	188
Complex Type fnProcessPeriodDueDate .....	188
Complex Type fnProcessScheduleDueDate .....	189
Complex Type kw_Label_type .....	191
Simple Type(s) .....	191
Simple Type ColumnDefReportFieldtype .....	191
Simple Type Source_type .....	191

Simple Type kw_ElementScope_type .....	192
Simple Type KwantuForm DataTypes .....	193
Simple Type kw_Xpath_type .....	193
Simple Type kw_Show_type .....	193
Simple Type DisplayFormat_type .....	194
Simple Type kwantu_comparison_types .....	194
Simple Type FilterElement_type .....	195
Simple Type ColumnFieldtype .....	195
Attribute(s) .....	196
Attribute Version / @Number .....	196
Attribute Identification / @BaseComponent .....	196
Attribute Identification / @CurrentVersion .....	196
Attribute Identification / @DevVersion .....	196
Attribute Identification / @ID .....	196
Attribute ReportDefaultColumnType / @Source .....	197
Attribute ReportDefaults / Tables / Table / @Type .....	197
Attribute ReportDefaults / Tables / Table / @Name .....	197
Attribute kw_UIVersions_type / UIVersion / @Name .....	197
Attribute KwantuElement_type / @Seq .....	197
Attribute KwantuElement_type / @Id .....	198
Attribute KwantuElement_type / @ID .....	198
Attribute KwantuElement_type / @Component .....	198
Attribute KwantuElement_type / @Element .....	198
Attribute KwantuElement_type / @SubElement .....	198
Attribute KwantuElement_type / @acSequenceRule .....	198
Attribute KwantuElement_type / @RepeatKeyRule .....	199
Attribute KwantuElement_type / @Scope .....	199
Attribute KwantuElement_type / @DataType .....	199
Attribute KwantuElement_type / @Xpath .....	200
Attribute Header / Dependencies / Dependency / @ID .....	200
Attribute Header / Dependencies / Dependency / @ComponentID .....	200
Attribute kw_UIVersionElements_type / UIElement / @Name .....	200
Attribute kw_UIVersionElements_type / UIElement / @Type .....	200
Attribute @Show .....	201
Attribute kw_Xform_type / @Appearance .....	201
Attribute kw_Xform_type / @CSS .....	201
Attribute kw_elLabel_type / Display / Condition / @Operator .....	202
Attribute kw_elLabel_type / Display / @Span .....	202
Attribute kw_Help_type / @Show .....	202
Attribute @ID .....	203
Attribute kw_elLabel_type / @Type .....	203
Attribute kw_elField_type / Display / Xform / @Appearance .....	203
Attribute kw_elField_type / Display / Xform / @CSS .....	204
Attribute Condition_type / @Operator .....	204
Attribute Condition_type / @Value .....	204
Attribute Condition_type / @DataType .....	204
Attribute Default_type / Value / @DataType .....	205
Attribute Default_type / InitiatialValue / @DataType .....	205
Attribute Kwantu_Filter_type / Parameters / Constant / @DataType .....	206
Attribute Kwantu_Filter_type / @Seq .....	206
Attribute Kwantu_Filter_type / @Type .....	206
Attribute Kwantu_Filter_type / @Parameters .....	207
Attribute Kwantu_Filter_type / @Reference .....	207
Attribute Kwantu_Filter_type / @FilterElement .....	207
Attribute fnHistoryTotal / Filters / Filter / @Seq .....	207
Attribute fnHistoryTotal / Filters / Filter / @Type .....	207
Attribute fnHistoryTotal / Filters / Filter / @Parameters .....	208
Attribute fnHistoryTotal / Filters / Filter / @Reference .....	208
Attribute fnHistoryTotal / Filters / Filter / @FilterElement .....	208
Attribute CalcType / @Seq .....	208
Attribute AssignSubElement_type / AssignCode / @SubElementID .....	208
Attribute AssignSubElement_type / AssignCode / @DataType .....	208
Attribute AssignSubElement_type / AssignName / @SubElementID .....	209
Attribute AssignSubElement_type / AssignName / @DataType .....	209
Attribute AssignSubElement_type / AssignSubElement / @Action .....	209
Attribute AssignSubElement_type / AssignSubElement / @SubElementID .....	210
Attribute AssignSubElement_type / AssignSubElement / @DataType .....	210
Attribute kw_elField_type / EntryMode / @Mode .....	210
Attribute ReportColumnType / @TableName .....	211
Attribute ReportColumnType / @SubElement .....	211
Attribute ReportColumnType / @Source .....	211
Attribute Reports / @Include .....	211

Attribute kw_elField_type / @Type .....	211
Attribute kw_elSelect_type / Display / Xform / @Appearance .....	212
Attribute kw_elSelect_type / Display / Xform / @CSS .....	212
Attribute kw_Condition_type / @Operator .....	213
Attribute Codelist / Taxonomy / @Type .....	213
Attribute Codelist / @SaveCode .....	213
Attribute RestType / ServiceURL / @Source .....	214
Attribute RestType / Parameters / Parameter / @Seq .....	214
Attribute RestType / Parameters / Parameter / @Name .....	214
Attribute RestType / Parameters / Parameter / @Format .....	214
Attribute RestType / Return / Status / StatusCode / @Value .....	214
Attribute RestType / Return / Status / @Type .....	215
Attribute RestType / Return / Assign / @AssignTo .....	215
Attribute RestType / Return / Assign / @AssignToID .....	215
Attribute RestType / Return / Assign / @Type .....	215
Attribute RestType / @ID .....	216
Attribute kw_elSelect_type / EntryMode / Select / Taxonomy / @Type .....	216
Attribute Assigns_type / Assign / @Action .....	216
Attribute Assigns_type / Assign / @Scope .....	216
Attribute Assigns_type / Assign / @ElementID .....	217
Attribute Assigns_type / Assign / @SubElementID .....	217
Attribute Assigns_type / Assign / @RepeatKeyValue .....	217
Attribute kw_elSelect_type / EntryMode / @Mode .....	217
Attribute kw_elSelect_type / EntryMode / @Source .....	218
Attribute kw_elSelect_type / @Type .....	218
Attribute RepeatDef / AddRows / AutoLoad / Periods / @PeriodType .....	218
Attribute RepeatDef / AddRows / AutoLoad / Component / Filter / @ElementID .....	219
Attribute RepeatDef / AddRows / AutoLoad / Component / Filter / @SubElement .....	219
Attribute RepeatDef / AddRows / AutoLoad / Component / @ComponentName .....	219
Attribute RepeatDef / AddRows / AutoLoad / Component / @RepeatID .....	219
Attribute RepeatDef / HeaderRow / @LabelsFromData .....	219
Attribute @Seq .....	220
Attribute RepeatDef / Columns / Column / @Width .....	220
Attribute RepeatDef / Columns / @NumberColumns .....	220
Attribute elCalcField / EntryMode / @Mode .....	220
Attribute elCalcField / @Type .....	220
Attribute BaseElements / @ID .....	221
Attribute kw_elGroup_type / @ID .....	221
Attribute RepeatHeader / rptColumn / @ColumnNo .....	221
Attribute RepeatHeader / rptColumn / @Span .....	221
Attribute RepeatHeader / @GroupName .....	221
Attribute RepeatHeader / @ID .....	221
Attribute RepeatData / rptColumn / @ColumnNo .....	222
Attribute RepeatData / rptColumn / @Span .....	222
Attribute RepeatData / @GroupName .....	222
Attribute RepeatData / @ID .....	222
Attribute rptColumn / @ColumnNo .....	222
Attribute rptColumn / @Span .....	222
Attribute RepeatSummary / @GroupName .....	222
Attribute RepeatSummary / @ID .....	222
Attribute kw_Repeat_type / @Type .....	223
Attribute HierarchyNode / @Level .....	223
Attribute HierarchyNode / @Parameter .....	223
Attribute HierarchyNode / @SelectElement .....	223
Attribute Hierarchy / @Levels .....	223
Attribute Hierarchy / @Taxonomy .....	223
Attribute Hierarchy / @TopLevel .....	224
Attribute Hierarchy / @Type .....	224
Attribute elSelectGroup / SelectGroupKey / @KeyID .....	224
Attribute elSelectGroup / @Type .....	224
Attribute ConfigurationData / @Version .....	225
Attribute Node / @Action .....	225
Attribute Node / @Seq .....	225
Attribute @No .....	225
Attribute @Source .....	225
Attribute kwElementMap / @LocalacElementName .....	225
Attribute GeoPoint_type / @LongSeconds .....	225
Attribute GeoPoint_type / @LongMinutes .....	226
Attribute GeoPoint_type / @LongDegrees .....	226
Attribute GeoPoint_type / @Longitude .....	226
Attribute GeoPoint_type / @LattSeconds .....	226
Attribute GeoPoint_type / @LattMinutes .....	226

Attribute GeoPoint_type / @LattDegrees .....	226
Attribute GeoPoint_type / @Latitude .....	226
Attribute NodeType / @Seq .....	226
Attribute fnProcessPeriodDueDate / CalcDueDate / DueDate / @Seq .....	227
Attribute fnProcessPeriodDueDate / CalcDueDate / @Type .....	227
Attribute fnProcessScheduleDueDate / CalcDueDate / DueDate / ProcessID / @Aspect .....	227
Attribute fnProcessScheduleDueDate / CalcDueDate / DueDate / @Seq .....	227
Attribute fnProcessScheduleDueDate / CalcDueDate / @Type .....	227
Attribute kw_Label_type / Int / @lang .....	228
Element Group(s) .....	228
Element Group BaseElements .....	228

## Namespace: ""

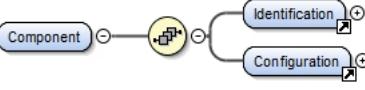
### Schema(s)

#### Main schema ApplicationComponentsConfig.xsd

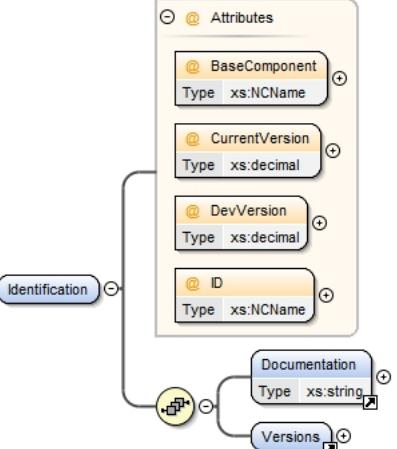
Namespace	No namespace
Properties	attribute form default: unqualified element form default: unqualified

### Element(s)

#### Element Component

Namespace	No namespace
Diagram	
Properties	content: complex
Model	Identification , Configuration
Children	Configuration, Identification
Instance	<pre>&lt;Component&gt;   &lt;Identification BaseComponent="" CurrentVersion="" DevVersion="" ID=""&gt;{1,1}&lt;/Identification&gt;   &lt;Configuration&gt;{1,1}&lt;/Configuration&gt; &lt;/Component&gt;</pre>
Source	<pre>&lt;xs:element name="Component"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element ref="Identification"/&gt;       &lt;xs:element ref="Configuration"/&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>

#### Element Identification

Namespace	No namespace
Diagram	

Properties	content: complex		
Used by	Element Component		
Model	Documentation , Versions		
Children	Documentation, Versions		
Instance	<pre>&lt;Identification BaseComponent="" CurrentVersion="" DevVersion="" ID=""&gt;   &lt;Documentation&gt;{1,1}&lt;/Documentation&gt;   &lt;Versions&gt;{1,1}&lt;/Versions&gt; &lt;/Identification&gt;</pre>		
Attributes	QName	Type	Use
	<b>BaseComponent</b>	xs:NCName	required
	<b>CurrentVersion</b>	xs:decimal	required
	<b>DevVersion</b>	xs:decimal	required
	<b>ID</b>	xs:NCName	required
Source	<pre>&lt;xss:element name="Identification"&gt;   &lt;xss:complexType&gt;     &lt;xss:sequence&gt;       &lt;xss:element ref="Documentation"/&gt;       &lt;xss:element ref="Versions"/&gt;     &lt;/xss:sequence&gt;     &lt;xss:attribute name="BaseComponent" use="required" type="xs:NCName"/&gt;     &lt;xss:attribute name="CurrentVersion" use="required" type="xs:decimal"/&gt;     &lt;xss:attribute name="DevVersion" use="required" type="xs:decimal"/&gt;     &lt;xss:attribute name="ID" use="required" type="xs:NCName"/&gt;   &lt;/xss:complexType&gt; &lt;/xss:element&gt;</pre>		

## Element Documentation

Namespace	No namespace
Diagram	<p>The diagram shows a class named "Documentation" with an attribute "Type" of type "xs:string". A callout box points to the "xs:string" type, stating: "Built-in primitive type. The string datatype represents character strings in XML."</p>
Type	xs:string
Properties	content: simple
Used by	Elements Identification, Version
Source	<pre>&lt;xss:element name="Documentation" type="xs:string"/&gt;</pre>

## Element Versions

Namespace	No namespace
Diagram	<p>The diagram shows a class named "Versions" with a child element "Version" indicated by a multiplicity of "1..∞".</p>
Properties	content: complex
Used by	Element Identification
Model	Version+
Children	Version
Instance	<pre>&lt;Versions&gt;   &lt;Version Number=""&gt;{1,unbounded}&lt;/Version&gt; &lt;/Versions&gt;</pre>
Source	<pre>&lt;xss:element name="Versions"&gt;   &lt;xss:complexType&gt;     &lt;xss:sequence&gt;       &lt;xss:element ref="Version" maxOccurs="unbounded"/&gt;     &lt;/xss:sequence&gt;   &lt;/xss:complexType&gt; &lt;/xss:element&gt;</pre>

## Element Version

Namespace	No namespace
-----------	--------------

Diagram	<pre> classDiagram     class Version {         @ Number Type xs:decimal         @ UpgradeInformation Type xs:string     }     class UpgradeInformation {         @ Documentation Type xs:string     }     </pre>						
Properties	content: complex						
Used by	Element Versions						
Model	UpgradeInformation , Documentation						
Children	Documentation, UpgradeInformation						
Instance	<pre> &lt;Version Number=""&gt;   &lt;UpgradeInformation&gt;{1,1}&lt;/UpgradeInformation&gt;   &lt;Documentation&gt;{1,1}&lt;/Documentation&gt; &lt;/Version&gt; </pre>						
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">QName</th> <th style="text-align: left; padding: 2px;">Type</th> <th style="text-align: left; padding: 2px;">Use</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">Number</td> <td style="padding: 2px;">xs:decimal</td> <td style="padding: 2px;">required</td> </tr> </tbody> </table>	QName	Type	Use	Number	xs:decimal	required
QName	Type	Use					
Number	xs:decimal	required					
Source	<pre> &lt;xs:element name="Version"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element ref="UpgradeInformation" /&gt;       &lt;xs:element ref="Documentation" /&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute name="Number" use="required" type="xs:decimal" /&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>						

## Element UpgradeInformation

Namespace	No namespace
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple
Used by	Element Version
Source	<pre> &lt;xs:element name="UpgradeInformation" type="xs:string" /&gt; </pre>

## Element Configuration

Namespace	No namespace
Diagram	
Properties	content: complex
Used by	Element Component
Model	ConfigurationData
Children	ConfigurationData
Instance	<pre> &lt;Configuration&gt;   &lt;ConfigurationData Version=""&gt;{1,1}&lt;/ConfigurationData&gt; &lt;/Configuration&gt; </pre>
Source	<pre> &lt;xs:element name="Configuration"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element ref="ConfigurationData" /&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>

| </xs:element> |

## Element ConfigurationData

Namespace	No namespace								
Diagram	<pre> classDiagram     class ConfigurationData {         @ Version         Type xs:decimal         Header         Elements     }     class Header     class Elements     ConfigurationData "1..1" o--&gt; Header     ConfigurationData "1..1" o--&gt; Elements     Header --o--&gt; Elements   </pre>								
Properties	content: complex								
Used by	Element Configuration								
Model	Header , Elements								
Children	Elements, Header								
Instance	<pre> &lt;ConfigurationData Version=""&gt;   &lt;Header&gt;{1,1}&lt;/Header&gt;   &lt;Elements&gt;{1,1}&lt;/Elements&gt; &lt;/ConfigurationData&gt;   </pre>								
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>Version</td> <td>xs:decimal</td> <td>required</td> </tr> </tbody> </table>	QName	Type	Use	Version	xs:decimal	required		
QName	Type	Use							
Version	xs:decimal	required							
Source	<pre> &lt;xs:element name="ConfigurationData"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element ref="Header"/&gt;       &lt;xs:element ref="Elements"/&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute name="Version" use="required" type="xs:decimal"/&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;   </pre>								

## Element Header

Namespace	No namespace		
Diagram	<pre> classDiagram     class Header {         Heading         Type xs:string         Namespace         Type xs:NCName         Modules         ReportDefaults         UIVersions         Type kw_UIVersions_type         DataModelURI         SaveURI         Dependencies     }     class Dependencies     Header "1..1" o--&gt; Dependencies   </pre>		
Properties	content: complex		
Used by	Element ConfigurationData		
Model	Heading , Namespace , Modules , ReportDefaults , UIVersions , DataModelURI , SaveURI , Dependencies{0,1}		
Children	DataModelURI, Dependencies, Heading, Modules, Namespace, ReportDefaults, SaveURI, UIVersions		
Instance	<pre> &lt;Header&gt;   &lt;Heading&gt;{1,1}&lt;/Heading&gt;   &lt;Namespace&gt;{1,1}&lt;/Namespace&gt;   &lt;Modules&gt;{1,1}&lt;/Modules&gt;   &lt;ReportDefaults&gt;{1,1}&lt;/ReportDefaults&gt;   &lt;UIVersions&gt;{1,1}&lt;/UIVersions&gt;   &lt;DataModelURI&gt;{1,1}&lt;/DataModelURI&gt;   </pre>		

	<pre>&lt;SaveURI&gt;{1,1}&lt;/SaveURI&gt; &lt;Dependencies&gt;{0,1}&lt;/Dependencies&gt; &lt;/Header&gt;</pre>
Source	<pre>&lt;xs:element name="Header"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element ref="Heading" /&gt;       &lt;xs:element ref="Namespace" /&gt;       &lt;xs:element ref="Modules" /&gt;       &lt;xs:element ref="ReportDefaults" /&gt;       &lt;xs:element ref="UIVersions" /&gt;       &lt;xs:element name="DataModelURI" /&gt;       &lt;xs:element name="SaveURI" /&gt;       &lt;xs:element minOccurs="0" name="Dependencies"&gt;         &lt;xs:complexType&gt;           &lt;xs:sequence&gt;             &lt;xs:element maxOccurs="unbounded" name="Dependency" &gt;               &lt;xs:complexType&gt;                 &lt;xs:sequence&gt;                   &lt;xs:element name="Element" type="KwantuElement_type" /&gt;                 &lt;/xs:sequence&gt;                 &lt;xs:attribute name="ID" /&gt;                 &lt;xs:attribute name="ComponentID" /&gt;               &lt;/xs:complexType&gt;             &lt;/xs:element&gt;           &lt;/xs:sequence&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>

## Element Heading

Namespace	No namespace
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple
Used by	Element Header
Source	<pre>&lt;xs:element name="Heading" type="xs:string" /&gt;</pre>

## Element Namespace

Namespace	No namespace
Diagram	<p>Built-in derived type. NCName represents XML "non-colonized" Names. The base type of NCName is Name.</p>
Type	xs:NCName
Properties	content: simple
Used by	Element Header
Source	<pre>&lt;xs:element name="Namespace" type="xs:NCName" /&gt;</pre>

## Element Modules

Namespace	No namespace
Diagram	
Properties	content: complex
Used by	Element Header
Model	Module+

Children	Module
Instance	<pre>&lt;Modules&gt;   &lt;Module&gt;{1,unbounded}&lt;/Module&gt; &lt;/Modules&gt;</pre>
Source	<pre>&lt;xs:element name="Modules"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element maxOccurs="unbounded" ref="Module" /&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>

## Element Module

Namespace	No namespace
Diagram	
Type	xs:NCName
Properties	content: simple
Used by	Element Modules
Source	<pre>&lt;xs:element name="Module" type="xs:NCName" /&gt;</pre>

## Element ReportDefaults

Namespace	No namespace
Diagram	
Properties	content: complex
Used by	Element Header
Model	Tables
Children	Tables
Instance	<pre>&lt;ReportDefaults&gt;   &lt;Tables&gt;{1,1}&lt;/Tables&gt; &lt;/ReportDefaults&gt;</pre>
Source	<pre>&lt;xs:element name="ReportDefaults"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="Tables"&gt;         &lt;xs:complexType&gt;           &lt;xs:sequence&gt;             &lt;xs:element name="Table" maxOccurs="unbounded"&gt;               &lt;xs:complexType&gt;                 &lt;xs:sequence&gt;                   &lt;xs:element name="Keys"&gt;                     &lt;xs:complexType&gt;                       &lt;xs:sequence&gt;                         &lt;xs:element name="Key" type="ReportDefaultColumnType" maxOccurs="unbounded" /&gt;                       &lt;/xs:sequence&gt;                     &lt;/xs:complexType&gt;                   &lt;/xs:element&gt;                 &lt;xs:element name="Attributes"&gt;                   &lt;xs:complexType&gt;                     &lt;xs:sequence&gt;                       &lt;xs:element name="Attribute" type="ReportDefaultColumnType" maxOccurs="unbounded" /&gt;                     &lt;/xs:sequence&gt;                   &lt;/xs:complexType&gt;                 &lt;/xs:element&gt;               &lt;/xs:sequence&gt;             &lt;xs:attribute name="Type"&gt;               &lt;xs:simpleType&gt;                 &lt;xs:restriction base="xs:NCName"&gt;                   &lt;xs:enumeration value="Summary"/&gt;                   &lt;xs:enumeration value="Detail"/&gt;                 &lt;/xs:restriction&gt;               &lt;/xs:simpleType&gt;             &lt;/xs:attribute&gt;           &lt;/xs:sequence&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>

```

        </xs:attribute>
        <xs:attribute name="Name" type="xs:ID" />
    </xs:complexType>
    </xs:element>
    </xs:sequence>
    </xs:complexType>
    </xs:element>
    </xs:sequence>
    </xs:complexType>
</xs:element>

```

### Element ReportDefaults / Tables

Namespace	No namespace
Diagram	
Properties	content: complex
Model	Table+
Children	Table
Instance	<pre> &lt;Tables&gt;     &lt;Table Name="" Type=""&gt;{1,unbounded}&lt;/Table&gt; &lt;/Tables&gt; </pre>
Source	<pre> &lt;xs:element name="Tables"&gt;     &lt;xs:complexType&gt;         &lt;xs:sequence&gt;             &lt;xs:element name="Table" maxOccurs="unbounded"&gt;                 &lt;xs:complexType&gt;                     &lt;xs:sequence&gt;                         &lt;xs:element name="Keys"&gt;                             &lt;xs:complexType&gt;                                 &lt;xs:sequence&gt;                                     &lt;xs:element name="Key" type="ReportDefaultColumnType" maxOccurs="unbounded" /&gt;                                 &lt;/xs:sequence&gt;                             &lt;/xs:complexType&gt;                         &lt;/xs:element&gt;                     &lt;xs:element name="Attributes"&gt;                         &lt;xs:complexType&gt;                             &lt;xs:sequence&gt;                                 &lt;xs:element name="Attribute" type="ReportDefaultColumnType" maxOccurs="unbounded" /&gt;                             &lt;/xs:sequence&gt;                         &lt;/xs:complexType&gt;                     &lt;/xs:element&gt;                 &lt;/xs:sequence&gt;             &lt;/xs:element&gt;         &lt;/xs:sequence&gt;     &lt;xs:attribute name="Type"&gt;         &lt;xs:simpleType&gt;             &lt;xs:restriction base="xs:NCName"&gt;                 &lt;xs:enumeration value="Summary" /&gt;                 &lt;xs:enumeration value="Detail" /&gt;             &lt;/xs:restriction&gt;         &lt;/xs:simpleType&gt;     &lt;/xs:attribute&gt;     &lt;xs:attribute name="Name" type="xs:ID" /&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>

### Element ReportDefaults / Tables / Table

Namespace	No namespace
Diagram	

Properties	content: complex maxOccurs: unbounded		
Model	Keys , Attributes		
Children	Attributes, Keys		
Instance	<Table Name="" Type=""> <Keys>{1,1}</Keys> <Attributes>{1,1}</Attributes> </Table>		
Attributes	QName	Type	Use
	Name	xs:ID	optional
	Type	restriction of xs:NCName	optional
Source	<pre> &lt;xss:element name="Table" maxOccurs="unbounded"&gt;   &lt;xss:complexType&gt;     &lt;xss:sequence&gt;       &lt;xss:element name="Keys"&gt;         &lt;xss:complexType&gt;           &lt;xss:sequence&gt;             &lt;xss:element name="Key" type="ReportDefaultColumnType" maxOccurs="unbounded"/&gt;           &lt;/xss:sequence&gt;         &lt;/xss:complexType&gt;       &lt;/xss:element&gt;       &lt;xss:element name="Attributes"&gt;         &lt;xss:complexType&gt;           &lt;xss:sequence&gt;             &lt;xss:element name="Attribute" type="ReportDefaultColumnType" maxOccurs="unbounded"/&gt;           &lt;/xss:sequence&gt;         &lt;/xss:complexType&gt;       &lt;/xss:element&gt;     &lt;/xss:sequence&gt;     &lt;xss:attribute name="Type"&gt;       &lt;xss:simpleType&gt;         &lt;xss:restriction base="xs:NCName"&gt;           &lt;xss:enumeration value="Summary"/&gt;           &lt;xss:enumeration value="Detail"/&gt;         &lt;/xss:restriction&gt;       &lt;/xss:simpleType&gt;     &lt;/xss:attribute&gt;     &lt;xss:attribute name="Name" type="xs:ID"/&gt;   &lt;/xss:complexType&gt; &lt;/xss:element&gt; </pre>		

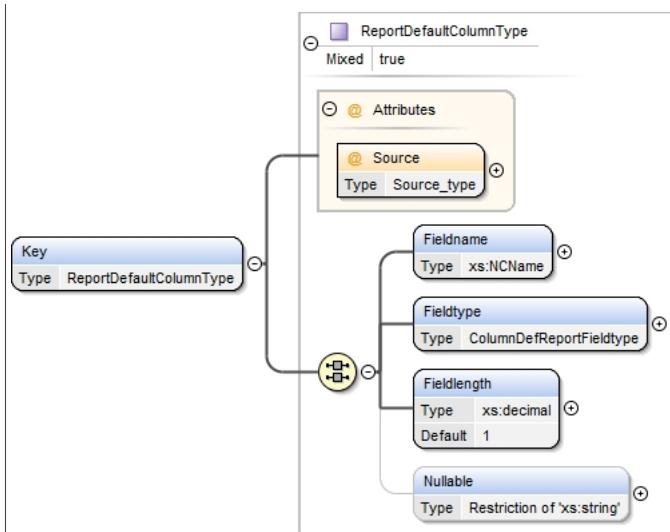
### Element ReportDefaults / Tables / Table / Keys

Namespace	No namespace
Diagram	<pre> classDiagram     class Keys     class Key {         Type         ReportDefaultColumnType     }     Keys "1..infinity" -- "0..1" Key </pre>
Properties	content: complex
Model	Key+
Children	Key
Instance	<Keys> <Key Source="">{1,unbounded}</Key> </Keys>
Source	<pre> &lt;xss:element name="Keys"&gt;   &lt;xss:complexType&gt;     &lt;xss:sequence&gt;       &lt;xss:element name="Key" type="ReportDefaultColumnType" maxOccurs="unbounded"/&gt;     &lt;/xss:sequence&gt;   &lt;/xss:complexType&gt; &lt;/xss:element&gt; </pre>

### Element ReportDefaults / Tables / Table / Keys / Key

Namespace	No namespace
-----------	--------------

Diagram



Type `ReportDefaultColumnType`

Properties content: complex  
maxOccurs: unbounded  
mixed: true

Model `ALL(Fieldname Fieldtype Fieldlength Nullable{0,1})`

Children `Fieldlength, Fieldname, Fieldtype, Nullable`

Instance `<Key Source="">`  
`<Fieldname>{1,1}</Fieldname>`  
`<Fieldtype>{1,1}</Fieldtype>`  
`<Fieldlength>{1,1}</Fieldlength>`  
`<Nullable>{0,1}</Nullable>`  
`</Key>`

QName	Type	Use
<code>Source</code>	<code>Source_type</code>	required

Source `<xss:element name="Key" type="ReportDefaultColumnType" maxOccurs="unbounded"/>`

### Element `ReportDefaultColumnType / Fieldname`

Namespace No namespace

Diagram   
A UML class diagram showing `Fieldname` as a derived type of `xs:NCName`. A callout box explains: "Built-in derived type. NCName represents XML 'non-colonized' Names. The base type of NCName is Name."

Type `xs:NCName`

Properties content: simple

Source `<xss:element name="Fieldname" type="xs:NCName"/>`

### Element `ReportDefaultColumnType / Fieldtype`

Namespace No namespace

Diagram   
A UML class diagram showing `Fieldtype` as a derived type of `ColumnDefReportFieldtype`.

Type `ColumnDefReportFieldtype`

Properties content: simple

Facets enumeration Varchar  
enumeration Char  
enumeration Date

	enumeration	Time
	enumeration	DateTime
	enumeration	Numeric
	enumeration	Integer
	enumeration	Memo
	enumeration	Boolean
	enumeration	Int
Source	<code>&lt;xs:element name="Fieldtype" type="ColumnDefReportFieldtype"/&gt;</code>	

### Element ReportDefaultColumnType / Fieldlength

Namespace	No namespace				
Diagram					
Type	xs:decimal				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>default:</td> <td>1</td> </tr> </table>	content:	simple	default:	1
content:	simple				
default:	1				
Source	<code>&lt;xs:element default="1" name="Fieldlength" type="xs:decimal"/&gt;</code>				

### Element ReportDefaultColumnType / Nullable

Namespace	No namespace				
Diagram					
Type	restriction of xs:string				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Facets	<table border="1"> <tr> <td>enumeration</td> <td>Not Null</td> </tr> <tr> <td>enumeration</td> <td>Nullable</td> </tr> </table>	enumeration	Not Null	enumeration	Nullable
enumeration	Not Null				
enumeration	Nullable				
Source	<pre> &lt;xs:element minOccurs="0" name="Nullable"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:string"&gt;       &lt;xs:enumeration value="Not Null"/&gt;       &lt;xs:enumeration value="Nullable"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:element&gt; </pre>				

### Element ReportDefaults / Tables / Table / Attributes

Namespace	No namespace
Diagram	
Properties	content: complex
Model	Attribute+
Children	Attribute
Instance	<pre> &lt;Attributes&gt;   &lt;Attribute Source=""&gt;{1,unbounded}&lt;/Attribute&gt; &lt;/Attributes&gt; </pre>
Source	<pre> &lt;xs:element name="Attributes"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt; </pre>

```

<xs:element name="Attribute" type="ReportDefaultColumnType" maxOccurs="unbounded" />
</xs:sequence>
</xs:complexType>
</xs:element>

```

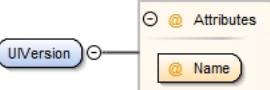
### Element ReportDefaults / Tables / Table / Attributes / Attribute

Namespace	No namespace						
Diagram							
Type	ReportDefaultColumnType						
Properties	content: complex maxOccurs: unbounded mixed: true						
Model	ALL(Fieldname Fieldtype Fieldlength Nullable{0,1})						
Children	Fieldlength, Fieldname, Fieldtype, Nullable						
Instance	<pre> &lt;Attribute Source=""&gt;   &lt;Fieldname&gt;{1,1}&lt;/Fieldname&gt;   &lt;Fieldtype&gt;{1,1}&lt;/Fieldtype&gt;   &lt;Fieldlength&gt;{1,1}&lt;/Fieldlength&gt;   &lt;Nullable&gt;{0,1}&lt;/Nullable&gt; &lt;/Attribute&gt; </pre>						
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>Source</td> <td>Source_type</td> <td>required</td> </tr> </tbody> </table>	QName	Type	Use	Source	Source_type	required
QName	Type	Use					
Source	Source_type	required					
Source	<pre>&lt;xs:element name="Attribute" type="ReportDefaultColumnType" maxOccurs="unbounded" /&gt;</pre>						

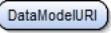
### Element UIVersions

Namespace	No namespace
Diagram	
Type	kw_UIVersions_type
Properties	content: complex
Used by	Element Header
Model	UIVersion+
Children	UIVersion
Instance	<pre> &lt;UIVersions&gt;   &lt;UIVersion Name=""&gt;{1, unbounded}&lt;/UIVersion&gt; &lt;/UIVersions&gt; </pre>
Source	<pre>&lt;xs:element name="UIVersions" type="kw_UIVersions_type" /&gt;</pre>

### **Element kw\_UIVersions\_type / UIVersion**

Namespace	No namespace						
Diagram	 <pre> graph LR     UIVersion[UIVersion] --&gt; Attributes[Attributes]     Attributes --&gt; Name[@Name]   </pre>						
Properties	content: complex maxOccurs: unbounded						
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td></td> <td>required</td> </tr> </tbody> </table>	QName	Type	Use	Name		required
QName	Type	Use					
Name		required					
Source	<pre> &lt;xs:element maxOccurs="unbounded" name="UIVersion"&gt;   &lt;xs:complexType&gt;     &lt;xs:attribute name="Name" use="required"/&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;   </pre>						

### **Element Header / DataModelURI**

Namespace	No namespace
Diagram	 <pre> graph LR     DataModelURI[DataModelURI]   </pre>
Source	<pre> &lt;xs:element name="DataModelURI" /&gt;   </pre>

### **Element Header / SaveURI**

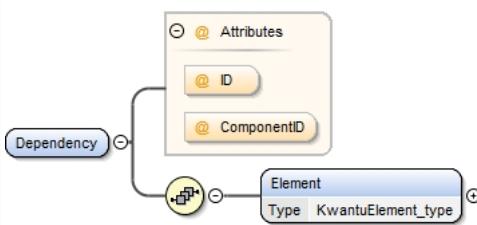
Namespace	No namespace
Diagram	 <pre> graph LR     SaveURI[SaveURI]   </pre>
Source	<pre> &lt;xs:element name="SaveURI" /&gt;   </pre>

### **Element Header / Dependencies**

Namespace	No namespace
Diagram	 <pre> graph LR     Dependencies[Dependencies] --&gt; ManyToMany[Many-to-Many]     ManyToMany --&gt; Dependency[Dependency]   </pre>
Properties	content: complex minOccurs: 0
Model	Dependency+
Children	Dependency
Instance	<pre> &lt;Dependencies&gt;   &lt;Dependency ComponentID="" ID=""&gt;{1,unbounded}&lt;/Dependency&gt; &lt;/Dependencies&gt;   </pre>
Source	<pre> &lt;xs:element minOccurs="0" name="Dependencies"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element maxOccurs="unbounded" name="Dependency"&gt;         &lt;xs:complexType&gt;           &lt;xs:sequence&gt;             &lt;xs:element name="Element" type="KwantuElement_type"/&gt;           &lt;/xs:sequence&gt;           &lt;xs:attribute name="ID"/&gt;           &lt;xs:attribute name="ComponentID"/&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;   </pre>

### **Element Header / Dependencies / Dependency**

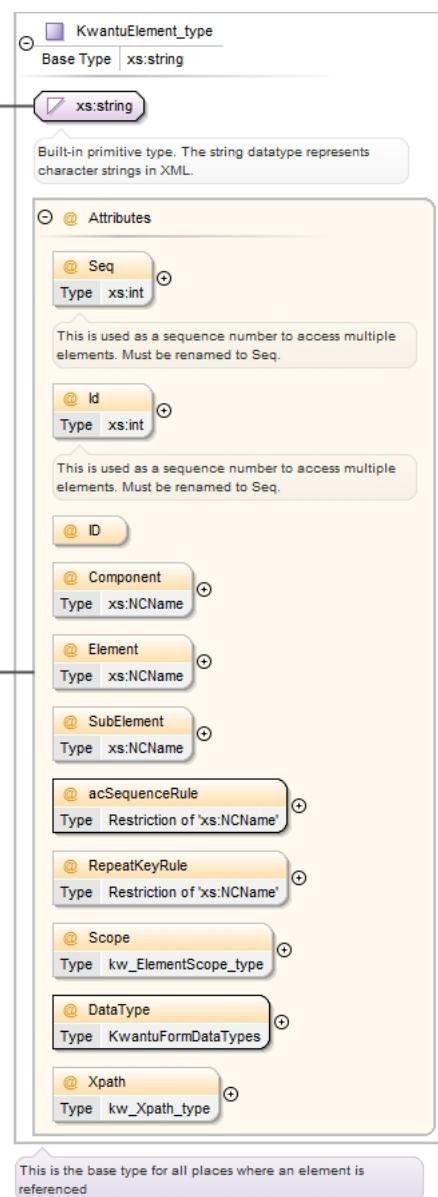
Namespace	No namespace
-----------	--------------

Diagram										
Properties	<p>content: complex</p> <p>maxOccurs: unbounded</p>									
Model	Element									
Children	Element									
Instance	<pre>&lt;Dependency ComponentID="" ID=""&gt;   &lt;Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE...&gt; &lt;/Dependency&gt;</pre>									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>ComponentID</td> <td></td> <td>optional</td> </tr> <tr> <td>ID</td> <td></td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Use	ComponentID		optional	ID		optional
QName	Type	Use								
ComponentID		optional								
ID		optional								
Source	<pre>&lt;xs:element maxOccurs="unbounded" name="Dependency"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="Element" type="KwantuElement_type"/&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute name="ID" /&gt;     &lt;xs:attribute name="ComponentID" /&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>									

## Element Header / Dependencies / Dependency / Element

Namespace	No namespace
-----------	--------------

Diagram



Type	KwantuElement_type
------	--------------------

Properties	content: complex
------------	------------------

Attributes	QName	Type	Use
	<b>Component</b>	<code>xs:NCName</code>	optional
	<b>DataType</b>	<code>KwantuFormDataTypes</code>	required
	<b>Element</b>	<code>xs:NCName</code>	optional
	<b>ID</b>		optional
	<b>Id</b>	<code>xs:int</code>	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>RepeatKeyRule</b>	<code>restriction of xs:NCName</code>	optional
	<b>Scope</b>	<code>kw_ElementScope_type</code>	optional
	<b>Seq</b>	<code>xs:int</code>	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>SubElement</b>	<code>xs:NCName</code>	optional
	<b>Xpath</b>	<code>kw_Xpath_type</code>	optional

	QName	Type	Use	
	acSequenceRule	restriction of xs:NCName	required	
Source	<xs:element name="Element" type="KwantuElement_type" />			

## Element Elements

Namespace	No namespace
Diagram	<p>Root element for the structure that defines the repeat or table structure</p>
Properties	content: complex
Used by	Element ConfigurationData
Model	elLabel   elField   elSelect   Repeat   elGroup   elCalcField   Hierarchy   elSelectGroup
Children	Hierarchy, Repeat, elCalcField, elField, elGroup, elLabel, elSelect, elSelectGroup
Instance	<pre> &lt;Elements&gt;   &lt;elLabel ID="" Type=""&gt;{1,1}&lt;/elLabel&gt;   &lt;elField ID="" Type=""&gt;{1,1}&lt;/elField&gt;   &lt;elSelect ID="" Type=""&gt;{1,1}&lt;/elSelect&gt;   &lt;Repeat ID="" Type=""&gt;{1,1}&lt;/Repeat&gt;   &lt;elGroup ID=""&gt;{1,1}&lt;/elGroup&gt;   &lt;elCalcField ID="" Type=""&gt;{1,1}&lt;/elCalcField&gt;   &lt;Hierarchy Levels="" Taxonomy="" TopLevel="" Type=""&gt;{1,1}&lt;/Hierarchy&gt;   &lt;elSelectGroup ID="" Type=""&gt;{1,1}&lt;/elSelectGroup&gt; &lt;/Elements&gt; </pre>
Source	<pre> &lt;xs:element name="Elements"&gt;   &lt;xs:complexType&gt;     &lt;xs:choice maxOccurs="unbounded"&gt;       &lt;xs:element ref="elLabel"/&gt;       &lt;xs:element ref="elField"/&gt;       &lt;xs:element ref="elSelect"/&gt;       &lt;xs:element ref="Repeat"/&gt;       &lt;xs:element ref="elGroup"/&gt;       &lt;xs:element ref="elCalcField"/&gt;       &lt;xs:element ref="Hierarchy"/&gt;       &lt;xs:element ref="elSelectGroup"/&gt;     &lt;/xs:choice&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>

## Element elLabel

Namespace	No namespace
-----------	--------------

Diagram	<pre> classDiagram     elLabel "Type kw_elLabel_type"     elLabel --&gt; UIVersionElements :      elLabel --&gt; Display :      elLabel --&gt; Help :      elLabel &lt; -- ID : xs:ID     elLabel &lt; -- Type : restriction of xs:NCName   </pre>												
Type	kw_elLabel_type												
Properties	content: complex												
Used by	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Element</td><td>Elements</td></tr> <tr> <td>Element Group</td><td>BaseElements</td></tr> </table>	Element	Elements	Element Group	BaseElements								
Element	Elements												
Element Group	BaseElements												
Model	UIVersionElements , Display , Help												
Children	Display, Help, UIVersionElements												
Instance	<pre> &lt;elLabel ID="" Type=""&gt;   &lt;UIVersionElements&gt;{1,1}&lt;/UIVersionElements&gt;   &lt;Display Show="" Span=""&gt;{1,1}&lt;/Display&gt;   &lt;Help Show=""&gt;{1,1}&lt;/Help&gt; &lt;/elLabel&gt;   </pre>												
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">QName</th><th style="width: 25%;">Type</th><th style="width: 25%;">Use</th><th style="width: 25%;"> </th></tr> </thead> <tbody> <tr> <td><b>ID</b></td><td>xs:ID</td><td>required</td><td> </td></tr> <tr> <td><b>Type</b></td><td>restriction of xs:NCName</td><td>required</td><td> </td></tr> </tbody> </table>	QName	Type	Use		<b>ID</b>	xs:ID	required		<b>Type</b>	restriction of xs:NCName	required	
QName	Type	Use											
<b>ID</b>	xs:ID	required											
<b>Type</b>	restriction of xs:NCName	required											
Source	<pre>&lt;xss:element name="elLabel" type="kw_elLabel_type" /&gt;</pre>												

### Element UIVersionElements

Namespace	No namespace				
Diagram	<pre> classDiagram     UIVersionElements "Type kw_UIVersionElements_type"     UIVersionElements --&gt; UIElement :    </pre> <p style="text-align: center;">This element is only required if the UIVersion is Custom. Otherwise all the elements are included in the view</p>				
Type	kw_UIVersionElements_type				
Properties	content: complex				
Used by	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Element</td><td>elCalcField</td></tr> <tr> <td>Complex Types</td><td>kw_elField_type, kw_elLabel_type, kw_elSelect_type</td></tr> </table>	Element	elCalcField	Complex Types	kw_elField_type, kw_elLabel_type, kw_elSelect_type
Element	elCalcField				
Complex Types	kw_elField_type, kw_elLabel_type, kw_elSelect_type				
Model	UIElement*				
Children	UIElement				
Instance	<pre> &lt;UIVersionElements&gt;   &lt;UIElement Name="" Type=""&gt;{0,unbounded}&lt;/UIElement&gt; &lt;/UIVersionElements&gt;   </pre>				
Source	<pre>&lt;xss:element name="UIVersionElements" type="kw_UIVersionElements_type" /&gt;</pre>				

### Element kw\_UIVersionElements\_type / UIElement

Namespace	No namespace
Annotations	This element is only required if the UIVersion is Custom. Otherwise all the elements are included in the view

Diagram	<pre> classDiagram     class UIElement {         @ Name         @ Type "Restriction of 'xs:NCName'"     }     note over UIElement: This element is only required if the UIVersion is Custom. Otherwise all the elements are included in the view   </pre>									
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded			
content:	complex									
minOccurs:	0									
maxOccurs:	unbounded									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td></td> <td>required</td> </tr> <tr> <td>Type</td> <td>restriction of xs:NCName</td> <td>required</td> </tr> </tbody> </table>	QName	Type	Use	Name		required	Type	restriction of xs:NCName	required
QName	Type	Use								
Name		required								
Type	restriction of xs:NCName	required								
Source	<pre> &lt;xs:element maxOccurs="unbounded" name="UIElement" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This element is only required if the UIVersion is Custom. Otherwise all the elements are included in the view&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:attribute name="Name" use="required"/&gt;     &lt;xs:attribute name="Type" use="required"&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:NCName"&gt;           &lt;xs:enumeration value="Default"/&gt;           &lt;xs:enumeration value="ReadOnly"/&gt;           &lt;xs:enumeration value="Hidden"/&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;     &lt;/xs:attribute&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;   </pre>									

## Element kw\_elLabel\_type / Display

Namespace	No namespace		
Diagram	<pre> classDiagram     class Display {         @ Show "kw_Show_type"         @ Span "xs:int"     }     class Label {         Type Extension of 'xs:string'     }     class Xform {         Type "kw_Xform_type"     }     class Format {         Type KwantuForm DataTypes     }     class Element {         Type KwantuElement type     }     class Condition     note over Display: If the @Show="Conditional" it looks to the Condition nodeset that evaluates to a boolean. If the if(Condition)=true then...   </pre>		
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> </table>	content:	complex
content:	complex		
Model	Label , Xform , Format , Element{0,1} , Condition{0,1}		
Children	Condition, Element, Format, Label, Xform		

Instance	<pre>&lt;Display Show="" Span=""&gt;     &lt;Label Show=""&gt;{1,1}&lt;/Label&gt;     &lt;Xform Appearance="" CSS=""&gt;{1,1}&lt;/Xform&gt;     &lt;Format&gt;{1,1}&lt;/Format&gt;     &lt;Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE Element&gt;         &lt;Condition Operator=""&gt;{0,1}&lt;/Condition&gt;     &lt;/Display&gt;</pre>												
Attributes	<table border="1"> <thead> <tr> <th>QName</th><th>Type</th><th>Use</th><th></th></tr> </thead> <tbody> <tr> <td>Show</td><td>kw_Show_type</td><td>required</td><td></td></tr> <tr> <td>Span</td><td>xs:int</td><td>optional</td><td></td></tr> </tbody> </table>	QName	Type	Use		Show	kw_Show_type	required		Span	xs:int	optional	
QName	Type	Use											
Show	kw_Show_type	required											
Span	xs:int	optional											
Source	<pre>&lt;xss:element name="Display"&gt;     &lt;xss:complexType&gt;         &lt;xss:sequence&gt;             &lt;xss:element ref="Label"/&gt;             &lt;xss:element name="Xform" type="kw_Xform_type"/&gt;             &lt;xss:sequence minOccurs="0"&gt;                 &lt;xss:element name="Format" type="KwantuFormDataTypes"/&gt;                 &lt;xss:element name="Element" type="KwantuElement_type" minOccurs="0"/&gt;             &lt;/xss:sequence&gt;             &lt;xss:element minOccurs="0" name="Condition"&gt;                 &lt;xss:annotation&gt;                     &lt;xss:documentation&gt;If the @Show="Conditional" it looks to the Condition nodeset that evaluates to a boolean. If the f(Condition)=true then field displays, otherwise not. Ie. to specify that the field must display only if the value of another field is "Other" then the following configuration would be applied: Condition[@Operator="Equal"] Condition/Element: {here the reference to the field that could have the value of other} Condition/Value:"Other"&lt;/xss:documentation&gt;                 &lt;/xss:annotation&gt;             &lt;xss:complexType&gt;                 &lt;xss:sequence&gt;                     &lt;xss:element name="Element" type="KwantuElement_type"/&gt;                     &lt;xss:choice&gt;                         &lt;xss:element name="Value"&gt;                             &lt;xss:complexType/&gt;                         &lt;/xss:element&gt;                         &lt;xss:element name="Element2" type="KwantuElement_type"/&gt;                     &lt;/xss:choice&gt;                 &lt;/xss:sequence&gt;                 &lt;xss:attribute name="Operator" use="required"&gt;                     &lt;xss:simpleType&gt;                         &lt;xss:restriction base="xs:Name"&gt;                             &lt;xss:enumeration value="GreaterThan"/&gt;                             &lt;xss:enumeration value="LessThan"/&gt;                             &lt;xss:enumeration value="GreaterThanOrEqual"/&gt;                             &lt;xss:enumeration value="LessThanOrEqual"/&gt;                             &lt;xss:enumeration value="Equal"/&gt;                             &lt;xss:enumeration value="NotEqual"/&gt;                         &lt;/xss:restriction&gt;                     &lt;/xss:simpleType&gt;                 &lt;/xss:attribute&gt;             &lt;/xss:complexType&gt;         &lt;/xss:element&gt;     &lt;/xss:sequence&gt;     &lt;xss:attribute ref="Show" use="required"/&gt;     &lt;xss:attribute name="Span" type="xs:int"/&gt; &lt;/xss:complexType&gt; &lt;/xss:element&gt;</pre>												

## Element Label

Namespace	No namespace
Diagram	<p>The diagram illustrates the 'xs:string' type as a built-in primitive type representing character strings in XML. It shows an 'Extension of xs:string' labeled 'Type' with an '@ Show' attribute labeled 'Type kw_Show_type'.</p> <pre> graph LR     A[Label Type Extension of 'xs:string'] --&gt; B["xs:string"]     B --&gt; C["Built-in primitive type. The string datatype represents character strings in XML."]     C --&gt; D["@ Attributes"]     D --&gt; E["@ Show Type kw_Show_type"]   </pre>
Type	extension of xs:string
Properties	content: complex
Used by	Elements elCalcField/Display, kw_elField_type/Display, kw_elLabel_type/Display, kw_elSelect_type/Display

Attributes	QName	Type	Use
	Show	kw_Show_type	required
Source	<pre>&lt;xs:element name="Label"&gt;   &lt;xs:complexType&gt;     &lt;xs:simpleContent&gt;       &lt;xs:extension base="xs:string"&gt;         &lt;xs:attribute ref="Show" use="required"/&gt;       &lt;/xs:extension&gt;     &lt;/xs:simpleContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>		

### Element kw\_elLabel\_type / Display / Xform

Namespace	No namespace									
Diagram	<pre> classDiagram     class Xform {         &lt;&lt;Type kw_Xform_type&gt;&gt;     }     class kw_Xform_type {         &lt;&lt;@ Attributes             @ Appearance                 Type Restriction of 'xs:Name'             @ CSS                 Type Restriction of 'xs:Name'         &gt;&gt;     }     Xform "1" -- "1" kw_Xform_type   </pre>									
Type	kw_Xform_type									
Properties	content: complex									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>Appearance</td> <td>restriction of xs:Name</td> <td>required</td> </tr> <tr> <td>CSS</td> <td>restriction of xs:Name</td> <td>required</td> </tr> </tbody> </table>	QName	Type	Use	Appearance	restriction of xs:Name	required	CSS	restriction of xs:Name	required
QName	Type	Use								
Appearance	restriction of xs:Name	required								
CSS	restriction of xs:Name	required								
Source	<pre>&lt;xs:element name="Xform" type="kw_Xform_type" /&gt;</pre>									

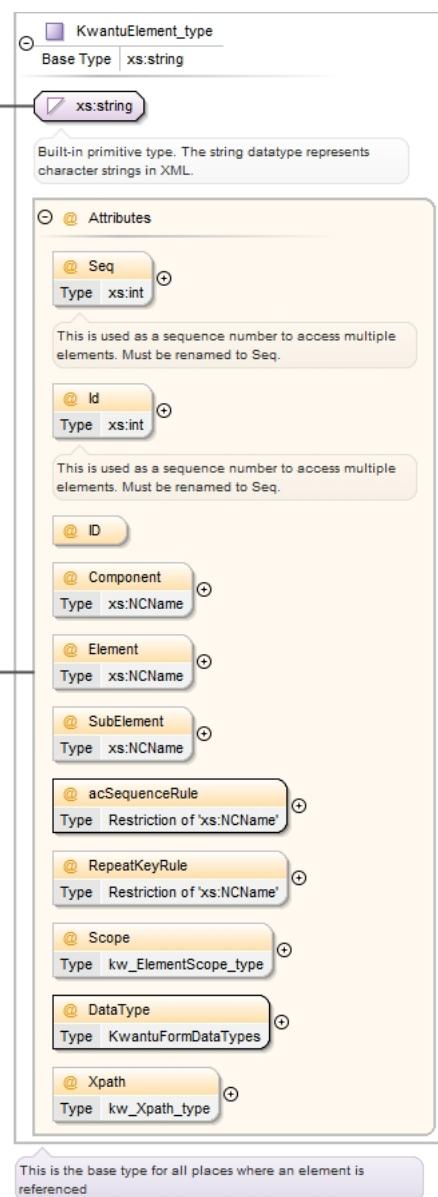
### Element kw\_elLabel\_type / Display / Format

Namespace	No namespace																								
Diagram	<pre> classDiagram     class Format {         &lt;&lt;Type KwantuFormDataTypes&gt;&gt;     }     class KwantuFormDataTypes {         &lt;&lt;&gt;&gt;     }     Format "1" -- "1" KwantuFormDataTypes   </pre>																								
Type	KwantuFormDataTypes																								
Properties	content: simple																								
Facets	<table> <tbody> <tr><td>enumeration</td><td>String</td></tr> <tr><td>enumeration</td><td>Integer</td></tr> <tr><td>enumeration</td><td>Float</td></tr> <tr><td>enumeration</td><td>Memo</td></tr> <tr><td>enumeration</td><td>Email</td></tr> <tr><td>enumeration</td><td>URI</td></tr> <tr><td>enumeration</td><td>Date</td></tr> <tr><td>enumeration</td><td>CoordinateDMS</td></tr> <tr><td>enumeration</td><td>GeoPoint</td></tr> <tr><td>enumeration</td><td>DataTypes</td></tr> <tr><td>enumeration</td><td>YesNo</td></tr> <tr><td>enumeration</td><td>Boolean</td></tr> </tbody> </table>	enumeration	String	enumeration	Integer	enumeration	Float	enumeration	Memo	enumeration	Email	enumeration	URI	enumeration	Date	enumeration	CoordinateDMS	enumeration	GeoPoint	enumeration	DataTypes	enumeration	YesNo	enumeration	Boolean
enumeration	String																								
enumeration	Integer																								
enumeration	Float																								
enumeration	Memo																								
enumeration	Email																								
enumeration	URI																								
enumeration	Date																								
enumeration	CoordinateDMS																								
enumeration	GeoPoint																								
enumeration	DataTypes																								
enumeration	YesNo																								
enumeration	Boolean																								
Source	<pre>&lt;xs:element name="Format" type="KwantuFormDataTypes" /&gt;</pre>																								

### Element kw\_elLabel\_type / Display / Element

Namespace	No namespace
-----------	--------------

Diagram



Type	KwantuElement_type																																												
Properties	content: complex minOccurs: 0																																												
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><b>Component</b></td> <td><code>xs:NCName</code></td> <td>optional</td> </tr> <tr> <td><b>DataType</b></td> <td><code>KwantuFormDataTypes</code></td> <td>required</td> </tr> <tr> <td><b>Element</b></td> <td><code>xs:NCName</code></td> <td>optional</td> </tr> <tr> <td><b>ID</b></td> <td></td> <td>optional</td> </tr> <tr> <td><b>Id</b></td> <td><code>xs:int</code></td> <td>optional</td> </tr> <tr> <td></td> <td colspan="2">This is used as a sequence number to access multiple elements. Must be renamed to <code>Seq</code>.</td><td></td></tr> <tr> <td><b>RepeatKeyRule</b></td> <td><code>restriction of xs:NCName</code></td> <td>optional</td><td></td></tr> <tr> <td><b>Scope</b></td> <td><code>kw_ElementScope_type</code></td> <td>optional</td><td></td></tr> <tr> <td><b>Seq</b></td> <td><code>xs:int</code></td> <td>optional</td><td></td></tr> <tr> <td></td> <td colspan="2">This is used as a sequence number to access multiple elements. Must be renamed to <code>Seq</code>.</td><td></td></tr> <tr> <td><b>SubElement</b></td> <td><code>xs:NCName</code></td> <td>optional</td><td></td></tr> </tbody> </table>			QName	Type	Use	<b>Component</b>	<code>xs:NCName</code>	optional	<b>DataType</b>	<code>KwantuFormDataTypes</code>	required	<b>Element</b>	<code>xs:NCName</code>	optional	<b>ID</b>		optional	<b>Id</b>	<code>xs:int</code>	optional		This is used as a sequence number to access multiple elements. Must be renamed to <code>Seq</code> .			<b>RepeatKeyRule</b>	<code>restriction of xs:NCName</code>	optional		<b>Scope</b>	<code>kw_ElementScope_type</code>	optional		<b>Seq</b>	<code>xs:int</code>	optional			This is used as a sequence number to access multiple elements. Must be renamed to <code>Seq</code> .			<b>SubElement</b>	<code>xs:NCName</code>	optional	
QName	Type	Use																																											
<b>Component</b>	<code>xs:NCName</code>	optional																																											
<b>DataType</b>	<code>KwantuFormDataTypes</code>	required																																											
<b>Element</b>	<code>xs:NCName</code>	optional																																											
<b>ID</b>		optional																																											
<b>Id</b>	<code>xs:int</code>	optional																																											
	This is used as a sequence number to access multiple elements. Must be renamed to <code>Seq</code> .																																												
<b>RepeatKeyRule</b>	<code>restriction of xs:NCName</code>	optional																																											
<b>Scope</b>	<code>kw_ElementScope_type</code>	optional																																											
<b>Seq</b>	<code>xs:int</code>	optional																																											
	This is used as a sequence number to access multiple elements. Must be renamed to <code>Seq</code> .																																												
<b>SubElement</b>	<code>xs:NCName</code>	optional																																											

	<b>QName</b>	<b>Type</b>	<b>Use</b>	
	<b>Xpath</b>	kw_Xpath_type	optional	
	<b>acSequenceRule</b>	restriction of xs:NCName	required	
Source	<xs:element name="Element" type="KwantuElement_type" minOccurs="0" />			

### Element kw\_eltLabel\_type / Display / Condition

Namespace	No namespace								
Annotations	<p>If the @Show="Conditional" it looks to the Condition nodeset that evaluates to a boolean. If the f(Condition)=true then field displays, otherwise not.</p> <p>Ie. to specify that the field must display only if the value of another field is "Other" then the following configuration would be applied:</p> <pre>Condition[@Operator="Equal"] Condition/Element: {here the reference to the field that could have the value of other} Condition/Value: "Other"</pre>								
Diagram	<pre> classDiagram     class Condition {         &lt;&lt;If the @Show="Conditional" it looks to the Condition nodeset that evaluates to a boolean. If the f(Condition)=true then...&gt;&gt;         &lt;&lt;Attributes&gt;&gt;         &lt;&lt;Operator, Type: Restriction of xs:Name&gt;&gt;         &lt;&lt;Element, Type: KwantuElement_type&gt;&gt;         &lt;&lt;Value&gt;&gt;         &lt;&lt;Element2, Type: KwantuElement_type&gt;&gt;     }   </pre>								
Properties	<p>content: complex</p> <p>minOccurs: 0</p>								
Model	Element , (Value   Element2)								
Children	Element, Element2, Value								
Instance	<pre>&lt;Condition Operator=""&gt;   &lt;Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE...   Element&gt;   &lt;Value&gt;{1,1}&lt;/Value&gt;   &lt;Element2 acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE...   Element2&gt; &lt;/Condition&gt;</pre>								
Attributes	<table border="1"> <thead> <tr> <th><b>QName</b></th> <th><b>Type</b></th> <th><b>Use</b></th> <th></th> </tr> </thead> <tbody> <tr> <td><b>Operator</b></td> <td>restriction of xs:Name</td> <td>required</td> <td></td> </tr> </tbody> </table>	<b>QName</b>	<b>Type</b>	<b>Use</b>		<b>Operator</b>	restriction of xs:Name	required	
<b>QName</b>	<b>Type</b>	<b>Use</b>							
<b>Operator</b>	restriction of xs:Name	required							
Source	<pre>&lt;xs:element minOccurs="0" name="Condition"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;If the @Show="Conditional" it looks to the Condition nodeset that evaluates to a boolean. If the f(Condition)=true then field displays, otherwise not. Ie. to specify that the field must display only if the value of another field is "Other" then the following configuration would be applied: Condition[@Operator="Equal"] Condition/Element: {here the reference to the field that could have the value of other} Condition/Value: "Other"&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="Element" type="KwantuElement_type"/&gt;       &lt;xs:choice&gt;         &lt;xs:element name="Value"&gt;           &lt;xs:complexType/&gt;         &lt;/xs:element&gt;         &lt;xs:element name="Element2" type="KwantuElement_type"/&gt;       &lt;/xs:choice&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute name="Operator" use="required"&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:Name"&gt;           &lt;xs:enumeration value="GreaterThan"/&gt;           &lt;xs:enumeration value="LessThan"/&gt;           &lt;xs:enumeration value="GreaterThanOrEqual"/&gt;           &lt;xs:enumeration value="LessThanOrEqual"/&gt;           &lt;xs:enumeration value="Equal"/&gt;           &lt;xs:enumeration value="NotEqual"/&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;     &lt;/xs:attribute&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>								

```

</xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>

```

### Element kw\_elLabel\_type / Display / Condition / Element

Namespace	No namespace																								
Diagram	<p>The diagram illustrates the structure of the KwantuElement_type. It starts with an 'Element' node (Type: KwantuElement_type) which points to a 'xs:string' node (Base Type: xs:string). The 'xs:string' node has a tooltip: 'Built-in primitive type. The string datatype represents character strings in XML.' Below this are several attributes:</p> <ul style="list-style-type: none"> <li>@ Seq (Type: xs:int): This is used as a sequence number to access multiple elements. Must be renamed to Seq.</li> <li>@ Id (Type: xs:int): This is used as a sequence number to access multiple elements. Must be renamed to Seq.</li> <li>@ ID</li> <li>@ Component (Type: xs:NCName)</li> <li>@ Element (Type: xs:NCName)</li> <li>@ SubElement (Type: xs:NCName)</li> <li>@ acSequenceRule (Type: Restriction of 'xs:NCName')</li> <li>@ RepeatKeyRule (Type: Restriction of 'xs:NCName')</li> <li>@ Scope (Type: kw_ElementScope_type)</li> <li>@ DataType (Type: KwantuFormDataTypes)</li> <li>@ Xpath (Type: kw_Xpath_type)</li> </ul> <p>A note at the bottom states: 'This is the base type for all places where an element is referenced.'</p>																								
Type	KwantuElement_type																								
Properties	content: complex																								
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>Component</td> <td>xs:NCName</td> <td>optional</td> </tr> <tr> <td>DataType</td> <td>KwantuFormDataTypes</td> <td>required</td> </tr> <tr> <td>Element</td> <td>xs:NCName</td> <td>optional</td> </tr> <tr> <td>ID</td> <td></td> <td>optional</td> </tr> <tr> <td>Id</td> <td>xs:int</td> <td>optional</td> </tr> <tr> <td></td> <td>This is used as a sequence number to access multiple elements. Must be renamed to Seq.</td> <td></td> </tr> <tr> <td>RepeatKeyRule</td> <td>restriction of xs:NCName</td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Use	Component	xs:NCName	optional	DataType	KwantuFormDataTypes	required	Element	xs:NCName	optional	ID		optional	Id	xs:int	optional		This is used as a sequence number to access multiple elements. Must be renamed to Seq.		RepeatKeyRule	restriction of xs:NCName	optional
QName	Type	Use																							
Component	xs:NCName	optional																							
DataType	KwantuFormDataTypes	required																							
Element	xs:NCName	optional																							
ID		optional																							
Id	xs:int	optional																							
	This is used as a sequence number to access multiple elements. Must be renamed to Seq.																								
RepeatKeyRule	restriction of xs:NCName	optional																							

QName	Type	Use	
<b>Scope</b>	kw_ElementScope_type	optional	
<b>Seq</b>	xs:int	optional	
	This is used as a sequence number to access multiple elements. Must be renamed to Seq.		
<b>SubElement</b>	xs:NCName	optional	
<b>Xpath</b>	kw_Xpath_type	optional	
<b>acSequenceRule</b>	restriction of xs:NCName	required	
Source	<code>&lt;xs:element name="Element" type="KwantuElement_type"/&gt;</code>		

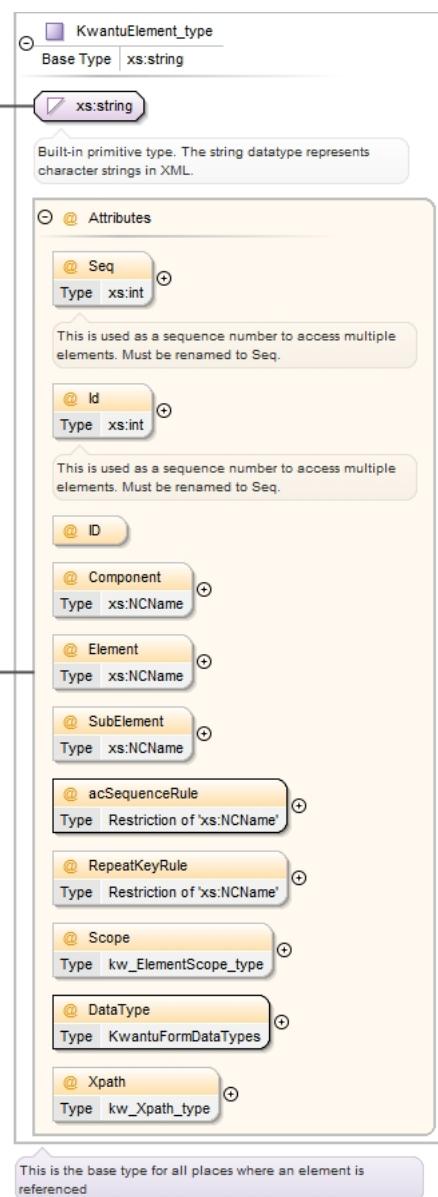
#### **Element kw\_elLabel\_type / Display / Condition / Value**

Namespace	No namespace
Diagram	
Properties	content: complex
Source	<code>&lt;xs:element name="Value"&gt;   &lt;xs:complexType/&gt; &lt;/xs:element&gt;</code>

#### **Element kw\_elLabel\_type / Display / Condition / Element2**

Namespace	No namespace
-----------	--------------

Diagram



Type	KwantuElement_type
------	--------------------

Properties	content: complex
------------	------------------

Attributes	QName	Type	Use	
	<b>Component</b>	xs:NCName	optional	
	<b>DataType</b>	KwantuFormDataTypes	required	
	<b>Element</b>	xs:NCName	optional	
	<b>ID</b>		optional	
	<b>Id</b>	xs:int	optional	
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.		
	<b>RepeatKeyRule</b>	restriction of xs:NCName	optional	
	<b>Scope</b>	kw_ElementScope_type	optional	
	<b>Seq</b>	xs:int	optional	
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.		
	<b>SubElement</b>	xs:NCName	optional	
	<b>Xpath</b>	kw_Xpath_type	optional	

	<b>QName</b>	<b>Type</b>	<b>Use</b>	
	<b>acSequenceRule</b>	restriction of xs:NCName	required	
Source	<xss:element name="Element2" type="KwantuElement_type"/>			

## Element Help

Namespace	No namespace								
Diagram									
Type	kw_Help_type								
Properties	content: complex mixed: true								
Used by	Element elCalcField Complex Types kw_elField_type, kw_elLabel_type, kw_elSelect_type								
Model									
Attributes	<table border="1"> <thead> <tr> <th><b>QName</b></th> <th><b>Type</b></th> <th><b>Use</b></th> <th></th> </tr> </thead> <tbody> <tr> <td>Show</td> <td>restriction of xs:NCName</td> <td>required</td> <td></td></tr> </tbody> </table>	<b>QName</b>	<b>Type</b>	<b>Use</b>		Show	restriction of xs:NCName	required	
<b>QName</b>	<b>Type</b>	<b>Use</b>							
Show	restriction of xs:NCName	required							
Source	<xss:element name="Help" type="kw_Help_type"/>								

## Element elField

Namespace	No namespace
Diagram	
Type	kw_elField_type
Properties	content: complex
Used by	Element Elements Element Group BaseElements

Model	UIVersionElements , Display , EntryMode , Validations , Reports , Help												
Children	Display, EntryMode, Help, Reports, UIVersionElements, Validations												
Instance	<pre>&lt;elField ID="" Type=""&gt;   &lt;UIVersionElements&gt;{1,1}&lt;/UIVersionElements&gt;   &lt;Display Show=""&gt;{1,1}&lt;/Display&gt;   &lt;EntryMode Mode=""&gt;{1,1}&lt;/EntryMode&gt;   &lt;Validations&gt;{1,1}&lt;/Validations&gt;   &lt;Reports Include=""&gt;{1,1}&lt;/Reports&gt;   &lt;Help Show=""&gt;{1,1}&lt;/Help&gt; &lt;/elField&gt;</pre>												
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><b>ID</b></td> <td>xs:ID</td> <td>required</td> </tr> <tr> <td><b>Type</b></td> <td>restriction of xs:NCName</td> <td>required</td> </tr> <tr> <td></td> <td colspan="2">If the type is select, then the datamodel will have sub-elements under the normal element. This is used when doing assign statements</td></tr> </tbody> </table>	QName	Type	Use	<b>ID</b>	xs:ID	required	<b>Type</b>	restriction of xs:NCName	required		If the type is select, then the datamodel will have sub-elements under the normal element. This is used when doing assign statements	
QName	Type	Use											
<b>ID</b>	xs:ID	required											
<b>Type</b>	restriction of xs:NCName	required											
	If the type is select, then the datamodel will have sub-elements under the normal element. This is used when doing assign statements												
Source	<code>&lt;xss:element name="elField" type="kw_elField_type" /&gt;</code>												

### Element kw\_elField\_type / Display

Namespace	No namespace						
Diagram							
Properties	content: complex						
Model	Label , Format , Xform , Condition{0,1}						
Children	Condition, Format, Label, Xform						
Instance	<pre>&lt;Display Show=""&gt;   &lt;Label Show=""&gt;{1,1}&lt;/Label&gt;   &lt;Format&gt;{1,1}&lt;/Format&gt;   &lt;Xform Appearance="" CSS=""&gt;{1,1}&lt;/Xform&gt;   &lt;Condition DataType="" Operator="" Value=""&gt;{0,1}&lt;/Condition&gt; &lt;/Display&gt;</pre>						
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>Show</td> <td>kw_Show_type</td> <td>required</td> </tr> </tbody> </table>	QName	Type	Use	Show	kw_Show_type	required
QName	Type	Use					
Show	kw_Show_type	required					
Source	<pre>&lt;xss:element name="Display"&gt;   &lt;xss:complexType&gt;     &lt;xss:sequence&gt;       &lt;xss:element ref="Label"/&gt;       &lt;xss:element name="Format" type="DisplayFormat_type" /&gt;       &lt;xss:element name="Xform"&gt;         &lt;xss:complexType&gt;           &lt;xss:attribute name="Appearance" use="required"&gt;             &lt;xss:simpleType&gt;               &lt;xss:restriction base="xs:Name"&gt;                 &lt;xss:enumeration value="Currency"/&gt;                 &lt;xss:enumeration value="Numeric"/&gt;                 &lt;xss:enumeration value="Text"/&gt;                 &lt;xss:enumeration value="DatePicker"/&gt;                 &lt;xss:enumeration value="Memo"/&gt;                 &lt;xss:enumeration value="Integer"/&gt;                 &lt;xss:enumeration value="Percentage"/&gt;                 &lt;xss:enumeration value="Output"/&gt;                 &lt;xss:enumeration value="Checkbox"/&gt;               &lt;/xss:restriction&gt;             &lt;/xss:simpleType&gt;           &lt;/xss:attribute&gt;         &lt;/xss:complexType&gt;       &lt;/xss:element&gt;     &lt;/xss:sequence&gt;   &lt;/xss:complexType&gt; &lt;/xss:element&gt;</pre>						

```

        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
<xs:attribute name="CSS">
    <xs:simpleType>
        <xs:restriction base="xs:Name">
            <xs:enumeration value="SmallField"/>
            <xs:enumeration value="MediumField"/>
            <xs:enumeration value="LargeField"/>
            <xs:enumeration value="man-label"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>
<xs:element minOccurs="0" name="Condition" type="Condition_type">
    <xs:annotation>
        <xs:documentation>If the @Show="Conditional" it looks to the Condition nodeset that evaluates to a boolean. If the f(Condition)=true then field displays, otherwise not. Ie. to specify that the field must display only if the value of another field is "Other" then the following configuration would be applied: Condition[@Operator="Equal"] Condition/Element: {here the reference to the field that could have the value of other} Condition/Value:"Other"</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
<xs:attribute ref="Show" use="required"/>
</xs:complexType>
</xs:element>

```

### **Element kw\_elField\_type / Display / Format**

Namespace	No namespace
Diagram	<pre> classDiagram     class Format {         &lt;&lt;Type&gt;&gt;     }     class DisplayFormat_type {         &lt;&lt;DisplayFormat_type&gt;&gt;     }     Format &lt; -- DisplayFormat_type   </pre>
Type	DisplayFormat_type
Properties	content: simple
Facets	enumeration      Float enumeration      Numeric enumeration      Date enumeration      String enumeration      Memo enumeration      Boolean enumeration      Integer
Source	<xs:element name="Format" type="DisplayFormat_type"/>

### **Element kw\_elField\_type / Display / Xform**

Namespace	No namespace									
Diagram	<pre> classDiagram     class Xform {         &lt;&lt;Xform&gt;&gt;     }     class Attributes {         &lt;&lt;Attributes&gt;&gt;     }     class Appearance {         &lt;&lt;Appearance&gt;&gt;         Type: Restriction of xs:Name     }     class CSS {         &lt;&lt;CSS&gt;&gt;         Type: Restriction of xs:Name     }     Xform --&gt; Attributes     Xform --&gt; Appearance     Xform --&gt; CSS   </pre>									
Properties	content: complex									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>Appearance</td> <td>restriction of xs:Name</td> <td>required</td> </tr> <tr> <td>CSS</td> <td>restriction of xs:Name</td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Use	Appearance	restriction of xs:Name	required	CSS	restriction of xs:Name	optional
QName	Type	Use								
Appearance	restriction of xs:Name	required								
CSS	restriction of xs:Name	optional								
Source	<xs:element name="Xform">     <xs:complexType>         <xs:attribute name="Appearance" use="required">             <xs:simpleType>									

```

<xs:restriction base="xs:Name">
  <xs:enumeration value="Currency" />
  <xs:enumeration value="Numeric" />
  <xs:enumeration value="Text" />
  <xs:enumeration value="DatePicker" />
  <xs:enumeration value="Memo" />
  <xs:enumeration value="Integer" />
  <xs:enumeration value="Percentage" />
  <xs:enumeration value="Output" />
  <xs:enumeration value="Checkbox" />
</xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="CSS">
  <xs:simpleType>
    <xs:restriction base="xs:Name">
      <xs:enumeration value="SmallField" />
      <xs:enumeration value="MediumField" />
      <xs:enumeration value="LargeField" />
      <xs:enumeration value="man-label" />
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>

```

## Element kw\_elField\_type / Display / Condition

Namespace	No namespace									
Annotations	<p>If the @Show="Conditional" it looks to the Condition nodeset that evaluates to a boolean. If the f(Condition)=true then field displays, otherwise not.</p> <p>Ie. to specify that the field must display only if the value of another field is "Other" then the following configuration would be applied:</p> <pre>Condition[@Operator="Equal"] Condition/Element: {here the reference to the field that could have the value of other} Condition/Value: "Other"</pre>									
Diagram	<p>The diagram illustrates the structure of the Condition_type element. It is a class with the following components:</p> <ul style="list-style-type: none"> <li><b>Attributes:</b> <ul style="list-style-type: none"> <li>Operator (Type: kwantu_comparison_types)</li> <li>Value</li> <li>DataType (Type: KwantuFormDataTypes)</li> </ul> </li> <li><b>Associations:</b> <ul style="list-style-type: none"> <li>A directed association from Condition (Type: Condition_type) to Condition_type, with a note: "If the @Show="Conditional" it looks to the Condition nodeset that evaluates to a boolean. If the f(Condition)=true then..."</li> <li>A directed association from Condition_type to Element (Type: KwantuElement_type).</li> <li>A directed association from Condition_type to ReferenceElement (Type: KwantuElement_type), with a note: "Not implemented yet".</li> </ul> </li> </ul>									
Type	Condition_type									
Properties	<p>content: complex</p> <p>minOccurs: 0</p>									
Model	Element , ReferenceElement{0,1}									
Children	Element, ReferenceElement									
Instance	<pre>&lt;Condition DataType="" Operator="" Value=""&gt;   &lt;Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE...   Element&gt;     &lt;ReferenceElement acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" ReferenceElement&gt;   &lt;/Condition&gt;</pre>									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>DataType</td> <td>KwantuFormDataTypes</td> <td>optional</td> </tr> <tr> <td>Operator</td> <td>kwantu_comparison_types</td> <td>required</td> </tr> </tbody> </table>	QName	Type	Use	DataType	KwantuFormDataTypes	optional	Operator	kwantu_comparison_types	required
QName	Type	Use								
DataType	KwantuFormDataTypes	optional								
Operator	kwantu_comparison_types	required								

	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>Value</b>		optional
Source	<pre>&lt;xss:element minOccurs="0" name="Condition" type="Condition_type"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;If the @Show="Conditional" it looks to the Condition nodeset that evaluates to a boolean. If the f(Condition)=true then field displays, otherwise not. Ie. to specify that the field must display only if the value of another field is "Other" then the following configuration would be applied: Condition[@Operator="Equal"] Condition/Element: {here the reference to the field that could have the value of other} Condition/Value:"Other" &lt;/xss:documentation&gt;   &lt;/xss:annotation&gt; &lt;/xss:element&gt;</pre>		

## Element Condition\_type / Element

Namespace	No namespace												
Diagram	<p>The diagram illustrates the <b>KwantuElement_type</b> element and its attributes. The element is defined with the type <b>KwantuElement_type</b>, which is a base type of <b>xs:string</b>. The attributes listed are:</p> <ul style="list-style-type: none"> <li><b>@ Seq</b>: Type <b>xs:int</b>. Description: This is used as a sequence number to access multiple elements. Must be renamed to Seq.</li> <li><b>@ Id</b>: Type <b>xs:int</b>. Description: This is used as a sequence number to access multiple elements. Must be renamed to Seq.</li> <li><b>@ ID</b></li> <li><b>@ Component</b>: Type <b>xs:NCName</b></li> <li><b>@ Element</b>: Type <b>xs:NCName</b></li> <li><b>@ SubElement</b>: Type <b>xs:NCName</b></li> <li><b>@ acSequenceRule</b>: Type <b>Restriction of 'xs:NCName'</b></li> <li><b>@ RepeatKeyRule</b>: Type <b>Restriction of 'xs:NCName'</b></li> <li><b>@ Scope</b>: Type <b>kw_ElementScope_type</b></li> <li><b>@ DataType</b>: Type <b>KwantuFormDataTypes</b></li> <li><b>@ Xpath</b>: Type <b>kw_Xpath_type</b></li> </ul> <p>A note at the bottom states: "This is the base type for all places where an element is referenced".</p>												
Type	<b>KwantuElement_type</b>												
Properties	content: complex												
Attributes	<table border="1"> <thead> <tr> <th><b>QName</b></th> <th><b>Type</b></th> <th><b>Use</b></th> </tr> </thead> <tbody> <tr> <td><b>Component</b></td> <td><b>xs:NCName</b></td> <td>optional</td> </tr> <tr> <td><b>DataType</b></td> <td><b>KwantuFormDataTypes</b></td> <td>required</td> </tr> <tr> <td><b>Element</b></td> <td><b>xs:NCName</b></td> <td>optional</td> </tr> </tbody> </table>	<b>QName</b>	<b>Type</b>	<b>Use</b>	<b>Component</b>	<b>xs:NCName</b>	optional	<b>DataType</b>	<b>KwantuFormDataTypes</b>	required	<b>Element</b>	<b>xs:NCName</b>	optional
<b>QName</b>	<b>Type</b>	<b>Use</b>											
<b>Component</b>	<b>xs:NCName</b>	optional											
<b>DataType</b>	<b>KwantuFormDataTypes</b>	required											
<b>Element</b>	<b>xs:NCName</b>	optional											

QName	Type	Use	
<b>ID</b>		optional	
<b>Id</b>	xs:int	optional	
	This is used as a sequence number to access multiple elements. Must be renamed to Seq.		
<b>RepeatKeyRule</b>	restriction of xs:NCName	optional	
<b>Scope</b>	kw_ElementScope_type	optional	
<b>Seq</b>	xs:int	optional	
	This is used as a sequence number to access multiple elements. Must be renamed to Seq.		
<b>SubElement</b>	xs:NCName	optional	
<b>Xpath</b>	kw_Xpath_type	optional	
<b>acSequenceRule</b>	restriction of xs:NCName	required	
<b>Source</b>	<xss:element name="Element" type="KwantuElement_type" />		

### Element Condition\_type / ReferenceElement

Namespace	No namespace
Annotations	Not implemented yet
Diagram	<pre> graph LR     KE[KwantuElement_type] --&gt; xsString[xs:string]     KE --&gt; Seq[Seq]     KE --&gt; Id[Id]     KE --&gt; Component[Component]     KE --&gt; Element[Element]     KE --&gt; SubElement[SubElement]     KE --&gt; acSequenceRule[acSequenceRule]     KE --&gt; RepeatKeyRule[RepeatKeyRule]     KE --&gt; Scope[Scope]     KE --&gt; DataType[DataType]     RE[ReferenceElement] -- "Not implemented yet" --&gt; KE   </pre> <p>The diagram illustrates the structure of the <code>KwantuElement_type</code>. It is defined as a base type for <code>xs:string</code>. The type includes several attributes:</p> <ul style="list-style-type: none"> <li><code>@Seq</code>: Type <code>xs:int</code>. Description: This is used as a sequence number to access multiple elements. Must be renamed to Seq.</li> <li><code>@Id</code>: Type <code>xs:int</code>. Description: This is used as a sequence number to access multiple elements. Must be renamed to Seq.</li> <li><code>@Component</code>: Type <code>xs:NCName</code>.</li> <li><code>@Element</code>: Type <code>xs:NCName</code>.</li> <li><code>@SubElement</code>: Type <code>xs:NCName</code>.</li> <li><code>@acSequenceRule</code>: Type <code>Restriction of 'xs:NCName'</code>.</li> <li><code>@RepeatKeyRule</code>: Type <code>Restriction of 'xs:NCName'</code>.</li> <li><code>@Scope</code>: Type <code>kw_ElementScope_type</code>.</li> <li><code>@Xpath</code>: Type <code>kw_Xpath_type</code>.</li> </ul> <p>A <code>ReferenceElement</code> node is shown pointing to <code>KwantuElement_type</code>, with the annotation "Not implemented yet".</p>

Type	KwantuElement_type		
Properties	content: complex minOccurs: 0		
Attributes			
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>Component</b>	xs:NCName	optional
	<b>DataType</b>	KwantuFormDataTypes	required
	<b>Element</b>	xs:NCName	optional
	<b>ID</b>		optional
	<b>Id</b>	xs:int	optional
	This is used as a sequence number to access multiple elements. Must be renamed to Seq.		
	<b>RepeatKeyRule</b>	restriction of xs:NCName	optional
	<b>Scope</b>	kw_ElementScope_type	optional
	<b>Seq</b>	xs:int	optional
	This is used as a sequence number to access multiple elements. Must be renamed to Seq.		
	<b>SubElement</b>	xs:NCName	optional
	<b>Xpath</b>	kw_Xpath_type	optional
	<b>acSequenceRule</b>	restriction of xs:NCName	required
Source	<pre>&lt;xss:element name="ReferenceElement" type="KwantuElement_type" minOccurs="0"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;Not implemented yet&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt; &lt;/xss:element&gt;</pre>		

## Element kw\_elField\_type / EntryMode

Namespace	No namespace		
Diagram	<p>The diagram illustrates the structure of the EntryMode element. It features a central 'EntryMode' node connected to a 'Mode' node (Type: Restriction of 'xs:NCName'). This connection is labeled with an association role 'Default'. From the 'Default' node, an arrow points to a 'Default' node (Type: Default_type). This 'Default' node is associated with an 'AssignElement' node (Type: Extension of 'xs:normalizedString'). A callout box provides the note: 'This provides for mapped fields to map the source element to the element in the current repeat. The source repeat is...'. Another arrow from the 'Default' node points to an 'AssignSubElements' node (Type: AssignSubElement_type). Finally, an arrow points from the 'Default' node to a 'ReadOnlyIf' node.</p>		
Properties	content: complex		
Model	Default{0,1} , (AssignElement{0,1}   AssignSubElements{0,1}) , ReadOnlyIf{0,1}		
Children	AssignElement, AssignSubElements, Default, ReadOnlyIf		
Instance	<pre>&lt;EntryMode Mode=""&gt;   &lt;Default&gt;{0,1}&lt;/Default&gt;   &lt;AssignElement&gt;{0,1}&lt;/AssignElement&gt;   &lt;AssignSubElements&gt;{0,1}&lt;/AssignSubElements&gt;   &lt;ReadOnlyIf&gt;{0,1}&lt;/ReadOnlyIf&gt; &lt;/EntryMode&gt;</pre>		
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>Mode</b>	restriction of xs:NCName	required
Source	<pre>&lt;xss:element name="EntryMode"&gt;   &lt;xss:complexType&gt;     &lt;xss:sequence minOccurs="0"&gt;       &lt;xss:element minOccurs="0" name="Default" type="Default_type"/&gt;</pre>		

```

<xs:choice minOccurs="0">
    <xs:element minOccurs="0" name="AssignElement">
        <xs:annotation>
            <xs:documentation>This provides for mapped fields to map the source element to the element in the current repeat. The source repeat is specified in the Component element Attributes. All the elements that you want to autoload from the source repeat are mapped here to local element ID. These are specific to the local repeat only.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:simpleContent>
                <xs:extension base="xs:normalizedString" />
            </xs:simpleContent>
        </xs:complexType>
    </xs:element>
    <xs:element name="AssignSubElements" type="AssignSubElement_type" minOccurs="0" />
</xs:choice>
<xs:element minOccurs="0" ref="ReadOnlyIf"/>
<xs:sequence>
<xs:attribute name="Mode" use="required">
    <xs:simpleType>
        <xs:restriction base="xs:NCName">
            <xs:enumeration value="Capture"/>
            <xs:enumeration value="ReadOnly"/>
            <xs:enumeration value="Hidden"/>
            <xs:enumeration value="Calculated"/>
            <xs:enumeration value="RepeatKey"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>

```

## Element kw\_elField\_type / EntryMode / Default

Namespace	No namespace				
Diagram					
Type	Default_type				
Properties	<table border="1"> <tr> <td>content:</td><td>complex</td></tr> <tr> <td>minOccurs:</td><td>0</td></tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Model	Value   (InitialValue , Calc)   ElementValue				
Children	Calc, ElementValue, InitialValue, Value				
Instance	<pre> &lt;Default&gt;     &lt;Value DataType=""&gt;{1,1}&lt;/Value&gt;     &lt;InitialValue DataType=""&gt;{1,1}&lt;/InitialValue&gt;     &lt;Calc Seq=""&gt;{1,1}&lt;/Calc&gt;     &lt;ElementValue acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" ElementValue&gt; &lt;/Default&gt; </pre>				
Source	<xs:element minOccurs="0" name="Default" type="Default_type" />				

## Element Default\_type / Value

Namespace	No namespace
-----------	--------------

Diagram	<pre> classDiagram     class Value {         &lt;&lt;Extension of 'xs:string'&gt;&gt;         @ Attributes         @ DataType         Type KwantuFormDataTypes     }     xs:string "Built-in primitive type. The string datatype represents character strings in XML."     Value &lt; -- xs:string   </pre>						
Type	extension of xs:string						
Properties	content: complex						
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">QName</th> <th style="text-align: left; padding: 2px;">Type</th> <th style="text-align: left; padding: 2px;">Use</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">@ DataType</td> <td style="padding: 2px;">KwantuFormDataTypes</td> <td style="padding: 2px;">optional</td> </tr> </tbody> </table>	QName	Type	Use	@ DataType	KwantuFormDataTypes	optional
QName	Type	Use					
@ DataType	KwantuFormDataTypes	optional					
Source	<pre> &lt;xss:element name="Value"&gt;   &lt;xss:complexType&gt;     &lt;xss:simpleContent&gt;       &lt;xss:extension base="xs:string"&gt;         &lt;xss:attribute name="DataType" type="KwantuFormDataTypes" /&gt;       &lt;/xss:extension&gt;     &lt;/xss:simpleContent&gt;   &lt;/xss:complexType&gt; &lt;/xss:element&gt;   </pre>						

### Element Default\_type / InitiatialValue

Namespace	No namespace						
Diagram	<pre> classDiagram     class initialValue {         @ Attributes         @ DataType         Type KwantuFormDataTypes     }     initialValue &lt; -- xs:complexType   </pre>						
Properties	content: complex						
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">QName</th> <th style="text-align: left; padding: 2px;">Type</th> <th style="text-align: left; padding: 2px;">Use</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">@ DataType</td> <td style="padding: 2px;">KwantuFormDataTypes</td> <td style="padding: 2px;">optional</td> </tr> </tbody> </table>	QName	Type	Use	@ DataType	KwantuFormDataTypes	optional
QName	Type	Use					
@ DataType	KwantuFormDataTypes	optional					
Source	<pre> &lt;xss:element name="InitiatialValue"&gt;   &lt;xss:complexType&gt;     &lt;xss:attribute name="DataType" type="KwantuFormDataTypes" /&gt;   &lt;/xss:complexType&gt; &lt;/xss:element&gt;   </pre>						

### Element Calc

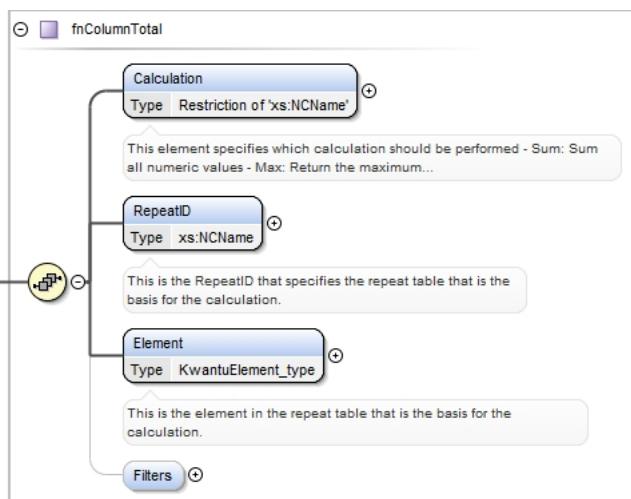
Namespace	No namespace
-----------	--------------

Diagram	<pre> classDiagram     class CalcType {         @ Attributes         Seq     }     class Calc {         Type CalcType     }     class ColumnTotal     class DateDiff     class ElementCalc     class RowTotal     class ElementsList     class CalcFinYearDates     class CalcHistory      Calc "1..1" --&gt; "1..1" CalcType :      CalcType "1..1" --&gt; "1..1" ColumnTotal :      CalcType "1..1" --&gt; "1..1" DateDiff :      CalcType "1..1" --&gt; "1..1" ElementCalc :      CalcType "1..1" --&gt; "1..1" RowTotal :      CalcType "1..1" --&gt; "1..1" ElementsList :      CalcType "1..1" --&gt; "1..1" CalcFinYearDates :      CalcType "1..1" --&gt; "1..1" CalcHistory :    </pre> <p>This can only be used in a repeat data section</p>								
Type	CalcType								
Properties	content: complex								
Used by	Element elCalcField/EntryMode/CalcField Complex Type Default_type								
Model	ColumnTotal   DateDiff   ElementCalc   RowTotal   ElementsList   CalcFinYearDates   CalcHistory								
Children	CalcFinYearDates, CalcHistory, ColumnTotal, DateDiff, ElementCalc, ElementsList, RowTotal								
Instance	<pre> &lt;Calc Seq=""&gt;   &lt;ColumnTotal&gt;{1,1}&lt;/ColumnTotal&gt;   &lt;DateDiff&gt;{1,1}&lt;/DateDiff&gt;   &lt;ElementCalc&gt;{1,1}&lt;/ElementCalc&gt;   &lt;RowTotal&gt;{1,1}&lt;/RowTotal&gt;   &lt;ElementsList&gt;{1,1}&lt;/ElementsList&gt;   &lt;CalcFinYearDates&gt;{1,1}&lt;/CalcFinYearDates&gt;   &lt;CalcHistory&gt;{1,1}&lt;/CalcHistory&gt; &lt;/Calc&gt; </pre>								
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding-right: 20px;">QName</th> <th style="text-align: left; padding-right: 20px;">Type</th> <th style="text-align: left; padding-right: 20px;">Use</th> <th></th> </tr> </thead> <tbody> <tr> <td style="padding-top: 5px;">Seq</td><td style="padding-top: 5px;">xs:integer</td><td style="padding-top: 5px;">required</td><td></td></tr> </tbody> </table>	QName	Type	Use		Seq	xs:integer	required	
QName	Type	Use							
Seq	xs:integer	required							
Source	<pre>&lt;xs:element name="Calc" type="CalcType" /&gt;</pre>								

## Element CalcType / ColumnTotal

Namespace	No namespace
-----------	--------------

Diagram



Type	fnColumnTotal
Properties	content: complex
Model	Calculation , RepeatID , Element , Filters{0,1}
Children	Calculation, Element, Filters, RepeatID
Instance	<ColumnTotal>   <Calculation>{1,1}</Calculation>   <RepeatID>{1,1}</RepeatID>   <Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE...   <Filters>{0,1}</Filters> </ColumnTotal>
Source	<xss:element name="ColumnTotal" type="fnColumnTotal"/>

### Element fnColumnTotal / Calculation

Namespace	No namespace												
Annotations	<p>This element specifies which calculation should be performed</p> <ul style="list-style-type: none"> <li>- Sum: Sum all numeric values</li> <li>- Max: Return the maximum numeric value</li> <li>- Min: Return the minimum numeric value</li> <li>- Count: Return the number of rows in the column</li> <li>- CountUnique: Return the number of Unique values in the column</li> </ul>												
Diagram	<pre> classDiagram     class Calculation {         &lt;&lt;restriction of 'xs:NCName'&gt;&gt;     }     class xsNCName {         &lt;&lt;restricts: xs:NCName&gt;&gt;     }     Calculation "0..1" --&gt; "1" xsNCName </pre>												
Type	restriction of xs:NCName												
Properties	content: simple												
Facets	<table border="1"> <tr> <td>enumeration</td> <td>Sum</td> </tr> <tr> <td>enumeration</td> <td>Max</td> </tr> <tr> <td>enumeration</td> <td>Min</td> </tr> <tr> <td>enumeration</td> <td>Count</td> </tr> <tr> <td>enumeration</td> <td>CountUnique</td> </tr> <tr> <td>enumeration</td> <td>AccumulativeSum</td> </tr> </table>	enumeration	Sum	enumeration	Max	enumeration	Min	enumeration	Count	enumeration	CountUnique	enumeration	AccumulativeSum
enumeration	Sum												
enumeration	Max												
enumeration	Min												
enumeration	Count												
enumeration	CountUnique												
enumeration	AccumulativeSum												
Source	<xss:element name="Calculation">   <xss:annotation>     <xss:documentation>This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum numeric value - Min: Return the minimum numeric value - Count: Return the number of rows in the column - CountUnique: Return the number of Unique values in the column</xss:documentation>   </xss:annotation>   <xss:simpleType>     <xss:restriction base="xs:NCName">       <xss:enumeration value="Sum"/>     </xss:restriction>   </xss:simpleType> </xss:element>												

```

<xs:enumeration value="Max" />
<xs:enumeration value="Min" />
<xs:enumeration value="Count" />
<xs:enumeration value="CountUnique" />
<xs:enumeration value="AccumulativeSum" />
</xs:restriction>
</xs:simpleType>
</xs:element>

```

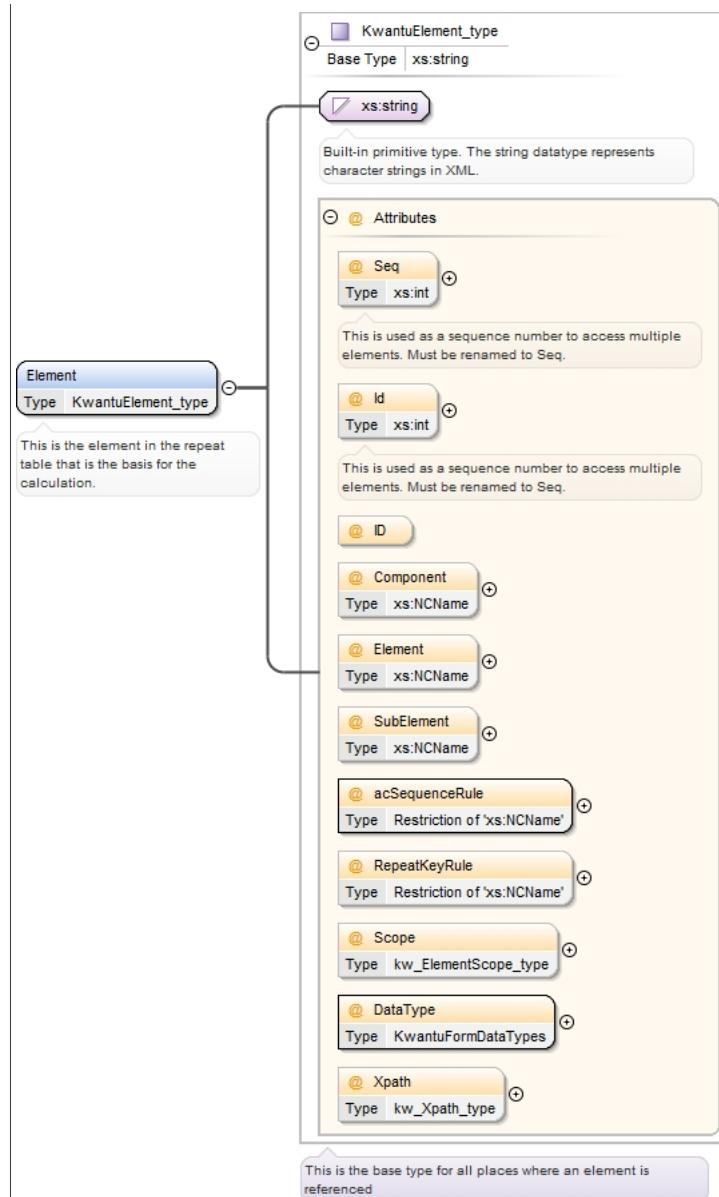
### **Element fnColumnTotal / RepeatID**

Namespace	No namespace				
Annotations	This is the RepeatID that specifies the repeat table that is the basis for the calculation.				
Diagram					
Type	xs:NCName				
Properties	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">content:</td> <td style="padding: 2px;">simple</td> </tr> <tr> <td style="padding: 2px;">maxOccurs:</td> <td style="padding: 2px;">1</td> </tr> </table>	content:	simple	maxOccurs:	1
content:	simple				
maxOccurs:	1				
Source	<pre> &lt;xs:element maxOccurs="1" name="RepeatID" type="xs:NCName"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This is the RepeatID that specifies the repeat table that is the basis for the calculation.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>				

### **Element fnColumnTotal / Element**

Namespace	No namespace
Annotations	This is the element in the repeat table that is the basis for the calculation.

Diagram



Type	KwantuElement_type																																																		
Properties	content: complex maxOccurs: 1																																																		
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> <th></th> </tr> </thead> <tbody> <tr> <td><b>Component</b></td> <td><code>xs:NCName</code></td> <td>optional</td> <td></td> </tr> <tr> <td><b>DataType</b></td> <td><code>KwantuFormDataTypes</code></td> <td>required</td> <td></td> </tr> <tr> <td><b>Element</b></td> <td><code>xs:NCName</code></td> <td>optional</td> <td></td> </tr> <tr> <td><b>ID</b></td> <td></td> <td>optional</td> <td></td> </tr> <tr> <td><b>Id</b></td> <td><code>xs:int</code></td> <td>optional</td> <td></td> </tr> <tr> <td></td> <td colspan="2">This is used as a sequence number to access multiple elements. Must be renamed to Seq.</td> <td></td> </tr> <tr> <td><b>RepeatKeyRule</b></td> <td><code>restriction of xs:NCName</code></td> <td>optional</td> <td></td> </tr> <tr> <td><b>Scope</b></td> <td><code>kw_ElementScope_type</code></td> <td>optional</td> <td></td> </tr> <tr> <td><b>Seq</b></td> <td><code>xs:int</code></td> <td>optional</td> <td></td> </tr> <tr> <td></td> <td colspan="2">This is used as a sequence number to access multiple elements. Must be renamed to Seq.</td> <td></td> </tr> <tr> <td><b>SubElement</b></td> <td><code>xs:NCName</code></td> <td>optional</td> <td></td> </tr> </tbody> </table>			QName	Type	Use		<b>Component</b>	<code>xs:NCName</code>	optional		<b>DataType</b>	<code>KwantuFormDataTypes</code>	required		<b>Element</b>	<code>xs:NCName</code>	optional		<b>ID</b>		optional		<b>Id</b>	<code>xs:int</code>	optional			This is used as a sequence number to access multiple elements. Must be renamed to Seq.			<b>RepeatKeyRule</b>	<code>restriction of xs:NCName</code>	optional		<b>Scope</b>	<code>kw_ElementScope_type</code>	optional		<b>Seq</b>	<code>xs:int</code>	optional			This is used as a sequence number to access multiple elements. Must be renamed to Seq.			<b>SubElement</b>	<code>xs:NCName</code>	optional	
QName	Type	Use																																																	
<b>Component</b>	<code>xs:NCName</code>	optional																																																	
<b>DataType</b>	<code>KwantuFormDataTypes</code>	required																																																	
<b>Element</b>	<code>xs:NCName</code>	optional																																																	
<b>ID</b>		optional																																																	
<b>Id</b>	<code>xs:int</code>	optional																																																	
	This is used as a sequence number to access multiple elements. Must be renamed to Seq.																																																		
<b>RepeatKeyRule</b>	<code>restriction of xs:NCName</code>	optional																																																	
<b>Scope</b>	<code>kw_ElementScope_type</code>	optional																																																	
<b>Seq</b>	<code>xs:int</code>	optional																																																	
	This is used as a sequence number to access multiple elements. Must be renamed to Seq.																																																		
<b>SubElement</b>	<code>xs:NCName</code>	optional																																																	

	QName	Type	Use
	Xpath	kw_Xpath_type	optional
	acSequenceRule	restriction of xs:NCName	required
Source	<pre>&lt;xs:element maxOccurs="1" name="Element" type="KwantuElement_type"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This is the element in the repeat table that is the basis for the calculation.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>		

### Element fnColumnTotal / Filters

Namespace	No namespace
Diagram	<p>This uses the same basic type as the Validation calculations</p>
Properties	content: complex minOccurs: 0
Model	Filter+
Children	Filter
Instance	<pre>&lt;Filters&gt;   &lt;Filter FilterElement="" Parameters="" Reference="" Seq="" Type=""&gt;{1,unbounded}&lt;/Filter&gt; &lt;/Filters&gt;</pre>
Source	<pre>&lt;xs:element minOccurs="0" name="Filters"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="Filter" type="Kwantu_Filter_type" maxOccurs="unbounded"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;This uses the same basic type as the Validation calculations&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>

### Element fnColumnTotal / Filters / Filter

Namespace	No namespace
Annotations	This uses the same basic type as the Validation calculations
Diagram	<p>This uses the same basic type as the Validation calculations</p> <p>If the conditional element is present, it evaluates to a boolean. If the fConditional=true then the validation rule...</p>

Type	Kwantu_Filter_type																				
Properties	content:	complex																			
	maxOccurs:	unbounded																			
Model	Parameters{0,1}   Condition{0,1}   Element																				
Children	Condition, Element, Parameters																				
Instance	<pre>&lt;Filter FilterElement="" Parameters="" Reference="" Seq="" Type=""&gt;   &lt;Parameters&gt;{0,1}&lt;/Parameters&gt;   &lt;Condition DataType="" Operator="" Value=""&gt;{0,1}&lt;/Condition&gt;   &lt;Element acSequenceRule="" Component="" DataType="" Element="" ID="" ID="" RepeatKeyRule="" Scope="" SubE&lt;/Element&gt; &lt;/Filter&gt;</pre>																				
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><b>FilterElement</b></td> <td>FilterElement_type</td> <td>optional</td> </tr> <tr> <td><b>Parameters</b></td> <td></td> <td>optional</td> </tr> <tr> <td><b>Reference</b></td> <td></td> <td>optional</td> </tr> <tr> <td><b>Seq</b></td> <td>xs:integer</td> <td>required</td> </tr> <tr> <td><b>Type</b></td> <td>kwantu_comparison_types</td> <td>required</td> </tr> </tbody> </table>	QName	Type	Use	<b>FilterElement</b>	FilterElement_type	optional	<b>Parameters</b>		optional	<b>Reference</b>		optional	<b>Seq</b>	xs:integer	required	<b>Type</b>	kwantu_comparison_types	required		
QName	Type	Use																			
<b>FilterElement</b>	FilterElement_type	optional																			
<b>Parameters</b>		optional																			
<b>Reference</b>		optional																			
<b>Seq</b>	xs:integer	required																			
<b>Type</b>	kwantu_comparison_types	required																			
Source	<pre>&lt;xss:element name="Filter" type="Kwantu_Filter_type" maxOccurs="unbounded"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;This uses the same basic type as the Validation calculations&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt; &lt;/xss:element&gt;</pre>																				

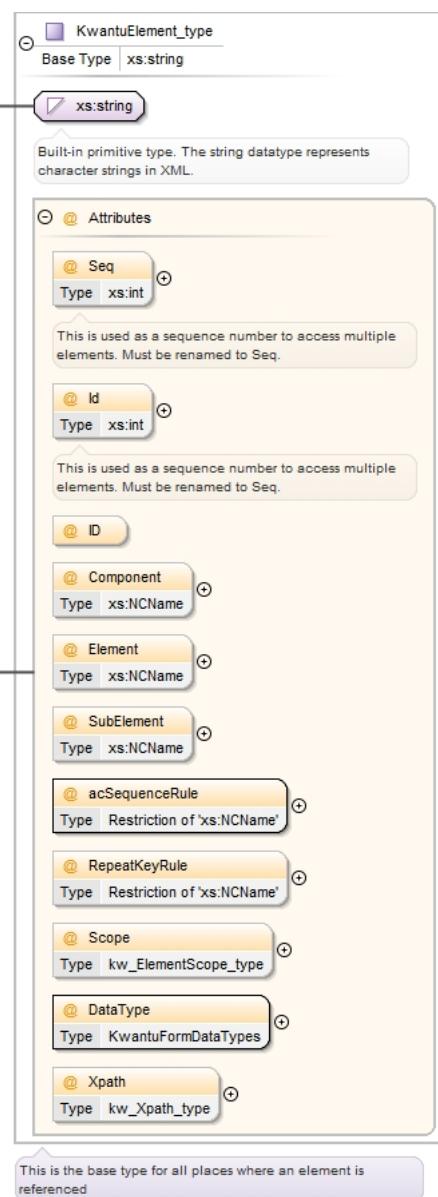
## Element Kwantu\_Filter\_type / Parameters

Namespace	No namespace				
Diagram	<pre> classDiagram     class Parameters {         &lt;&lt;Complex Type&gt;&gt;     }     class Element {         &lt;&lt;Element&gt;&gt;         &lt;&lt;Type   KwantuElement_type&gt;&gt;     }     class Constant {         &lt;&lt;Simple Type&gt;&gt;         &lt;&lt;Type   Extension of 'xs:string'&gt;&gt;     }      Parameters --&gt; Element     Parameters --&gt; Constant   </pre>				
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Model	Element   Constant				
Children	Constant, Element				
Instance	<pre>&lt;Parameters&gt;   &lt;Element acSequenceRule="" Component="" DataType="" Element="" ID="" ID="" RepeatKeyRule="" Scope="" SubE&lt;/Element&gt;   &lt;Constant DataType=""&gt;{1,1}&lt;/Constant&gt; &lt;/Parameters&gt;</pre>				
Source	<pre>&lt;xss:element minOccurs="0" name="Parameters"&gt;   &lt;xss:complexType&gt;     &lt;xss:choice&gt;       &lt;xss:element name="Element" type="KwantuElement_type"/&gt;       &lt;xss:element name="Constant"&gt;         &lt;xss:complexType&gt;           &lt;xss:simpleContent&gt;             &lt;xss:extension base="xs:string"&gt;               &lt;xss:attribute name="DataType" type="KwantuFormDataTypes"/&gt;             &lt;/xss:extension&gt;           &lt;/xss:simpleContent&gt;         &lt;/xss:complexType&gt;       &lt;/xss:element&gt;     &lt;/xss:choice&gt;   &lt;/xss:complexType&gt; &lt;/xss:element&gt;</pre>				

## Element Kwantu\_Filter\_type / Parameters / Element

Namespace	No namespace
-----------	--------------

Diagram



Type	KwantuElement_type
------	--------------------

Properties	content: complex
------------	------------------

Attributes	QName	Type	Use
	<b>Component</b>	xs:NCName	optional
	<b>DataType</b>	KwantuFormDataTypes	required
	<b>Element</b>	xs:NCName	optional
	<b>ID</b>		optional
	<b>Id</b>	xs:int	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>RepeatKeyRule</b>	restriction of xs:NCName	optional
	<b>Scope</b>	kw_ElementScope_type	optional
	<b>Seq</b>	xs:int	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>SubElement</b>	xs:NCName	optional
	<b>Xpath</b>	kw_Xpath_type	optional

	QName	Type	Use	
	acSequenceRule	restriction of xs:NCName	required	
Source	<xss:element name="Element" type="KwantuElement_type" />			

### Element Kwantu\_Filter\_type / Parameters / Constant

Namespace	No namespace						
Diagram	<pre> classDiagram     class Constant {         &lt;&lt;extension of 'xs:string'&gt;&gt;         &lt;&lt;Attributes&gt;&gt;             @Operator : kwantu_comparison_types             @Value         &lt;&lt;DataType&gt;&gt;             Type : KwantuFormDataTypes     }     class xs:string {         &lt;&lt;Built-in primitive type. The string datatype represents character strings in XML.&gt;&gt;     }     Constant &lt; -- xs:string     Constant --&gt; Element : KwantuElement_type     Constant --&gt; ReferenceElement : KwantuElement_type   </pre>						
Type	extension of xs:string						
Properties	content: complex						
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>DataType</td> <td>KwantuFormDataTypes</td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Use	DataType	KwantuFormDataTypes	optional
QName	Type	Use					
DataType	KwantuFormDataTypes	optional					
Source	<pre> &lt;xss:element name="Constant"&gt;   &lt;xss:complexType&gt;     &lt;xss:simpleContent&gt;       &lt;xss:extension base="xs:string"&gt;         &lt;xss:attribute name="DataType" type="KwantuFormDataTypes" /&gt;       &lt;/xss:extension&gt;     &lt;/xss:simpleContent&gt;   &lt;/xss:complexType&gt; &lt;/xss:element&gt;   </pre>						

### Element Kwantu\_Filter\_type / Condition

Namespace	No namespace
Annotations	<p>If the conditional element is present, it evaluates to a boolean. If the f(Conditional)=true then the validation rule applies, otherwise it does not apply.</p> <p>Ie. to specify that the field is required if the value of another field is "Other" then the following configuration would be applied:</p> <p>Validation: @Seq=1; @Type=NotNull ---&gt; this specifies that the current element value should not be null</p> <p>Condition[@Operator="Equal"]</p> <p>Condition/Element: {here the reference to the field that could have the value of other}</p> <p>Condition/Value: "Other"</p>
Diagram	<pre> classDiagram     class Condition {         &lt;&lt;Condition_type&gt;&gt;         &lt;&lt;Attributes&gt;&gt;             @Operator : kwantu_comparison_types             @Value         &lt;&lt;Element&gt;&gt;             Type : KwantuElement_type         &lt;&lt;ReferenceElement&gt;&gt;             Type : KwantuElement_type     }     class Condition_type {         &lt;&lt;If the conditional element is present, it evaluates to a boolean. If the f(Conditional)=true then the validation rule...&gt;&gt;     }     Condition &lt; -- Condition_type     Condition --&gt; Element : KwantuElement_type     Condition --&gt; ReferenceElement : KwantuElement_type   </pre>
Type	Condition_type
Properties	<p>content: complex</p> <p>minOccurs: 0</p>

Model	Element , ReferenceElement{0,1}		
Children	Element, ReferenceElement		
Instance	<pre>&lt;Condition DataType="" Operator="" Value=""&gt;   &lt;Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE...     &lt;ReferenceElement acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq...       &lt;Condition&gt;</pre>		
Attributes	<b>QName</b>	Type	Use
	<b>DataType</b>	KwantuFormDataTypes	optional
	<b>Operator</b>	kwantu_comparison_types	required
	<b>Value</b>		optional
Source	<pre>&lt;xs:element minOccurs="0" name="Condition" type="Condition_type"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;If the conditional element is present, it evaluates to a boolean. If the f(Conditional)=true then the validation rule applies, otherwise it does not apply. Ie. to specify that the field is required if the value of another field is "Other" then the following configuration would be applied: Validation: @Seq=1; @Type=NotNull ---&gt; this specifies that the current element value should not be null Condition[@Operator="Equal"] Condition/Element: {here the reference to the field that could have the value of other} Condition/Value: "Other"&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>		

### Element Kwantu\_Filter\_type / Element

Namespace	No namespace
-----------	--------------

Diagram

```

classDiagram
    class KwantuElement_type {
        xs:string
        @ Seq
        @ Id
        @ Component
        @ Element
        @ SubElement
        @ acSequenceRule
        @ RepeatKeyRule
        @ Scope
        @ DataType
        @ Xpath
    }
    class xs:string {
        <<Built-in primitive type. The string datatype represents character strings in XML.>>
    }
    class @ Seq {
        <<This is used as a sequence number to access multiple elements. Must be renamed to Seq.>>
    }
    class @ Id {
        <<This is used as a sequence number to access multiple elements. Must be renamed to Seq.>>
    }
    class @ Component {
    }
    class @ Element {
    }
    class @ SubElement {
    }
    class @ acSequenceRule {
        <<Type Restriction of 'xs:NCName'>>
    }
    class @ RepeatKeyRule {
        <<Type Restriction of 'xs:NCName'>>
    }
    class @ Scope {
        <<Type kw_ElementScope_type>>
    }
    class @ DataType {
        <<Type KwantuFormDataTypes>>
    }
    class @ Xpath {
        <<Type kw_Xpath_type>>
    }

```

This diagram illustrates the structure of the `KwantuElement_type` class and its attributes. The `KwantuElement_type` class is defined as a `xs:string` type. It contains several attributes, each with a specific type and a detailed description:

- @ Seq**: Type `xs:int`. Description: This is used as a sequence number to access multiple elements. Must be renamed to Seq.
- @ Id**: Type `xs:int`. Description: This is used as a sequence number to access multiple elements. Must be renamed to Seq.
- @ Component**: Type `xs:NCName`.
- @ Element**: Type `xs:NCName`.
- @ SubElement**: Type `xs:NCName`.
- @ acSequenceRule**: Type `Restriction of 'xs:NCName'`.
- @ RepeatKeyRule**: Type `Restriction of 'xs:NCName'`.
- @ Scope**: Type `kw_ElementScope_type`.
- @ DataType**: Type `KwantuFormDataTypes`.
- @ Xpath**: Type `kw_Xpath_type`.

A callout box at the bottom indicates that `KwantuElement_type` is the base type for all places where an element is referenced.

Type	KwantuElement_type		
Properties	content: complex		
Attributes	QName	Type	Use
	<b>Component</b>	<code>xs:NCName</code>	optional
	<b>DataType</b>	<code>KwantuFormDataTypes</code>	required
	<b>Element</b>	<code>xs:NCName</code>	optional
	<b>ID</b>		optional
	<b>Id</b>	<code>xs:int</code>	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>RepeatKeyRule</b>	<code>restriction of xs:NCName</code>	optional
	<b>Scope</b>	<code>kw_ElementScope_type</code>	optional
	<b>Seq</b>	<code>xs:int</code>	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>SubElement</b>	<code>xs:NCName</code>	optional
	<b>Xpath</b>	<code>kw_Xpath_type</code>	optional

	QName	Type	Use	
	acSequenceRule	restriction of xs:NCName	required	
Source	<xs:element name="Element" type="KwantuElement_type" />			

### Element CalcType / DateDiff

Namespace	No namespace
Diagram	
Type	fnDateDiff
Properties	content: complex
Model	Calculation , Element{2,2}
Children	Calculation, Element
Instance	<DateDiff> <Calculation>{1,1}</Calculation> <Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE...</Element> </DateDiff>
Source	<xs:element name="DateDiff" type="fnDateDiff" />

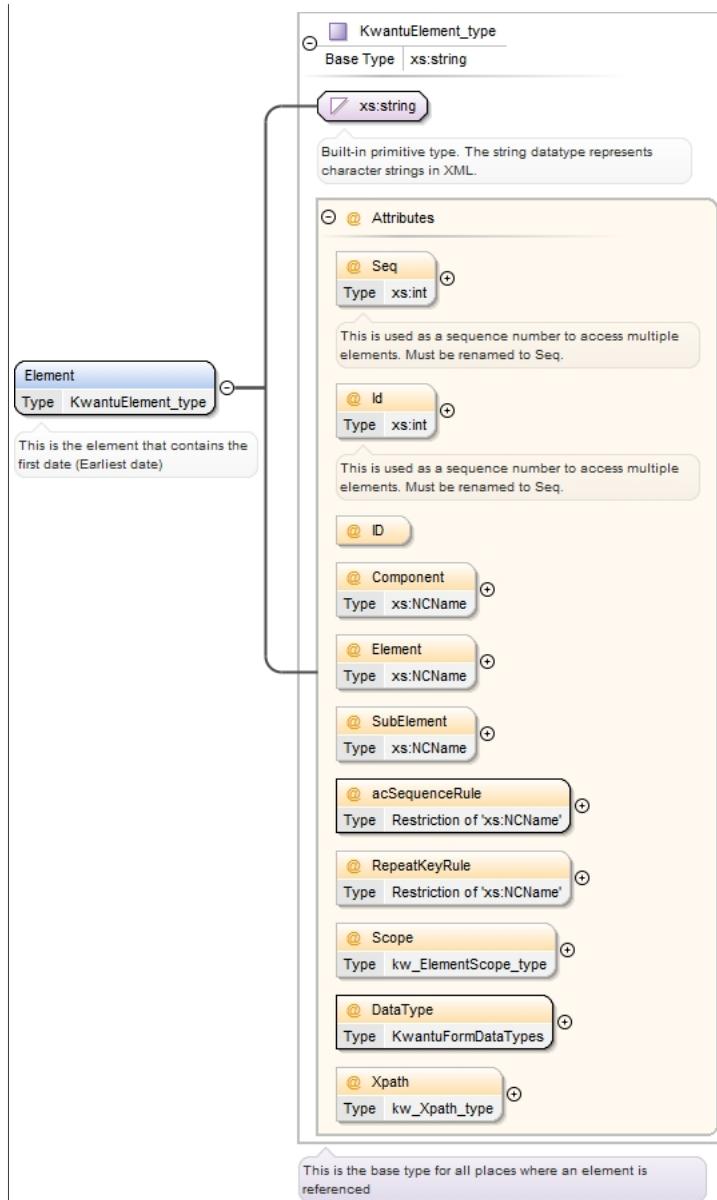
### Element fnDateDiff / Calculation

Namespace	No namespace				
Annotations	This element specifies which calculation should be performed - CalendarDays: calculates the number of calendar days between date1 and date2 - WorkDays: calculates the number of work days between date1 and date2. Work days includes Mondays to Fridays				
Diagram					
Type	restriction of xs:NCName				
Properties	content: simple				
Facets	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">CalendarDays</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">WorkDays</td> </tr> </table>	enumeration	CalendarDays	enumeration	WorkDays
enumeration	CalendarDays				
enumeration	WorkDays				
Source	<xs:element name="Calculation"> <xs:annotation> <xs:documentation>This element specifies which calculation should be performed - CalendarDays: calculates the number of calendar days between date1 and date2 - WorkDays: calculates the number of work days between date1 and date2. Work days includes Mondays to Fridays</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:NCName"> <xs:enumeration value="CalendarDays" /> <xs:enumeration value="WorkDays" /> </xs:restriction> </xs:simpleType> </xs:element>				

### Element fnDateDiff / Element

Namespace	No namespace
Annotations	This is the element that contains the first date (Earliest date)

Diagram



Type	KwantuElement_type		
Properties	content:	complex	
	minOccurs:	2	
	maxOccurs:	2	
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>Component</b>	<code>xs:NCName</code>	optional
	<b>DataType</b>	<code>KwantuFormDataTypes</code>	required
	<b>Element</b>	<code>xs:NCName</code>	optional
	<b>ID</b>		optional
	<b>Id</b>	<code>xs:int</code>	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>RepeatKeyRule</b>	restriction of <code>xs:NCName</code>	optional
	<b>Scope</b>	<code>kw_ElementScope_type</code>	optional
	<b>Seq</b>	<code>xs:int</code>	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	

	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>SubElement</b>	xs:NCName	optional
	<b>Xpath</b>	kw_Xpath_type	optional
	<b>acSequenceRule</b>	restriction of xs:NCName	required
Source	<pre>&lt;xs:element maxOccurs="2" name="Element" type="KwantuElement_type" minOccurs="2"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This is the element that contains the first date (Earliest date)&lt;/   xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>		

### Element CalcType / ElementCalc

Namespace	No namespace
Diagram	
Type	fnElementCalc
Properties	content: complex
Model	Calculation , Element{2,2}
Children	Calculation, Element
Instance	<pre>&lt;ElementCalc&gt;   &lt;Calculation&gt;{1,1}&lt;/Calculation&gt;   &lt;Element acSequenceRule="" Component="" DataType="" Element="" ID="" IDKeyRule="" Scope="" Seq="" SubE...</pre>
Source	<pre>&lt;xs:element name="ElementCalc" type="fnElementCalc"/&gt;</pre>

### Element fnElementCalc / Calculation

Namespace	No namespace								
Annotations	<p>This element specifies which calculation should be performed</p> <ul style="list-style-type: none"> <li>- CalendarDays: calculates the number of calendar days between date1 and date2</li> <li>- WorkDays: calculates the number of work days between date1 and date2. Work days includes Mondays to Fridays</li> </ul>								
Diagram									
Type	restriction of xs:NCName								
Properties	content: simple								
Facets	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">Add</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">Subtract</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">Multiply</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">Divide</td> </tr> </table>	enumeration	Add	enumeration	Subtract	enumeration	Multiply	enumeration	Divide
enumeration	Add								
enumeration	Subtract								
enumeration	Multiply								
enumeration	Divide								
Source	<pre>&lt;xs:element name="Calculation"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This element specifies which calculation should be performed - CalendarDays: calculates the number of calendar days between date1 and date2 - WorkDays: calculates the number of work days between date1 and date2. Work days includes Mondays to Fridays&lt;/xs:documentation&gt;   &lt;xs:annotation&gt;     &lt;xs:simpleType&gt;       &lt;xs:restriction base="xs:NCName"&gt;         &lt;xs:enumeration value="Add"/&gt;       &lt;/xs:restriction&gt;     &lt;/xs:simpleType&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>								

```

<xs:enumeration value="Subtract"/>
<xs:enumeration value="Multiply"/>
<xs:enumeration value="Divide"/>
</xs:restriction>
</xs:simpleType>
</xs:element>

```

## Element fnElementCalc / Element

Namespace	No namespace															
Annotations	This is the element that contains the first date (Earliest date)															
Diagram	<p>The diagram illustrates the structure of the KwantuElement_type. It starts with a box labeled "Element" (Type: KwantuElement_type) which has a note: "This is the element that contains the first date (Earliest date)". A line connects this box to a larger box labeled "KwantuElement_type". Inside this box, there is a "Base Type" section labeled "xs:string" with a note: "Built-in primitive type. The string datatype represents character strings in XML." Below this are several attributes:</p> <ul style="list-style-type: none"> <li><b>@ Seq</b>: Type xs:int. Note: "This is used as a sequence number to access multiple elements. Must be renamed to Seq."</li> <li><b>@ Id</b>: Type xs:int. Note: "This is used as a sequence number to access multiple elements. Must be renamed to Seq."</li> <li><b>@ ID</b></li> <li><b>@ Component</b>: Type xs:NCName</li> <li><b>@ Element</b>: Type xs:NCName</li> <li><b>@ SubElement</b>: Type xs:NCName</li> <li><b>@ acSequenceRule</b>: Type Restriction of 'xs:NCName'</li> <li><b>@ RepeatKeyRule</b>: Type Restriction of 'xs:NCName'</li> <li><b>@ Scope</b>: Type kw_ElementScope_type</li> <li><b>@ DataType</b>: Type KwantuFormDataTypes</li> <li><b>@ Xpath</b>: Type kw_Xpath_type</li> </ul> <p>A note at the bottom of the inner box states: "This is the base type for all places where an element is referenced".</p>															
Type	KwantuElement_type															
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>2</td> </tr> <tr> <td>maxOccurs:</td> <td>2</td> </tr> </table>	content:	complex	minOccurs:	2	maxOccurs:	2									
content:	complex															
minOccurs:	2															
maxOccurs:	2															
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><b>Component</b></td> <td>xs:NCName</td> <td>optional</td> </tr> <tr> <td><b>DataType</b></td> <td>KwantuFormDataTypes</td> <td>required</td> </tr> <tr> <td><b>Element</b></td> <td>xs:NCName</td> <td>optional</td> </tr> <tr> <td><b>ID</b></td> <td></td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Use	<b>Component</b>	xs:NCName	optional	<b>DataType</b>	KwantuFormDataTypes	required	<b>Element</b>	xs:NCName	optional	<b>ID</b>		optional
QName	Type	Use														
<b>Component</b>	xs:NCName	optional														
<b>DataType</b>	KwantuFormDataTypes	required														
<b>Element</b>	xs:NCName	optional														
<b>ID</b>		optional														

QName	Type	Use	
<b>Id</b>	xs:int	optional	
	This is used as a sequence number to access multiple elements. Must be renamed to Seq.		
<b>RepeatKeyRule</b>	restriction of xs:NCName	optional	
<b>Scope</b>	kw_ElementScope_type	optional	
<b>Seq</b>	xs:int	optional	
	This is used as a sequence number to access multiple elements. Must be renamed to Seq.		
<b>SubElement</b>	xs:NCName	optional	
<b>Xpath</b>	kw_Xpath_type	optional	
<b>acSequenceRule</b>	restriction of xs:NCName	required	
Source	<pre>&lt;xs:element maxOccurs="2" name="Element" type="KwantuElement_type" minOccurs="2"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This is the element that contains the first date (Earliest date)&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>		

### Element CalcType / RowTotal

Namespace	No namespace
Annotations	This can only be used in a repeat data section
Diagram	<p>The diagram illustrates the relationship between RowTotal and fnRowCalc. A RowTotal node (Type: fnRowCalc) is connected to a fnRowCalc node. The fnRowCalc node is associated with a Calculation node (Type: Restriction of 'xs:NCName'). A tooltip for the fnRowCalc node states: "This can only be used in a repeat data section". A tooltip for the Calculation node states: "This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum...". A tooltip for the Element node (Type: KwantuElement_type) states: "This is the element in the repeat table that is the basis for the calculation."</p>
Type	fnRowCalc
Properties	content: complex
Model	Calculation , Element+
Children	Calculation, Element
Instance	<pre>&lt;RowTotal&gt;   &lt;Calculation&gt;{1,1}&lt;/Calculation&gt;   &lt;Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE</pre>
Source	<pre>&lt;xs:element name="RowTotal" type="fnRowCalc"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This can only be used in a repeat data section&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

### Element fnRowCalc / Calculation

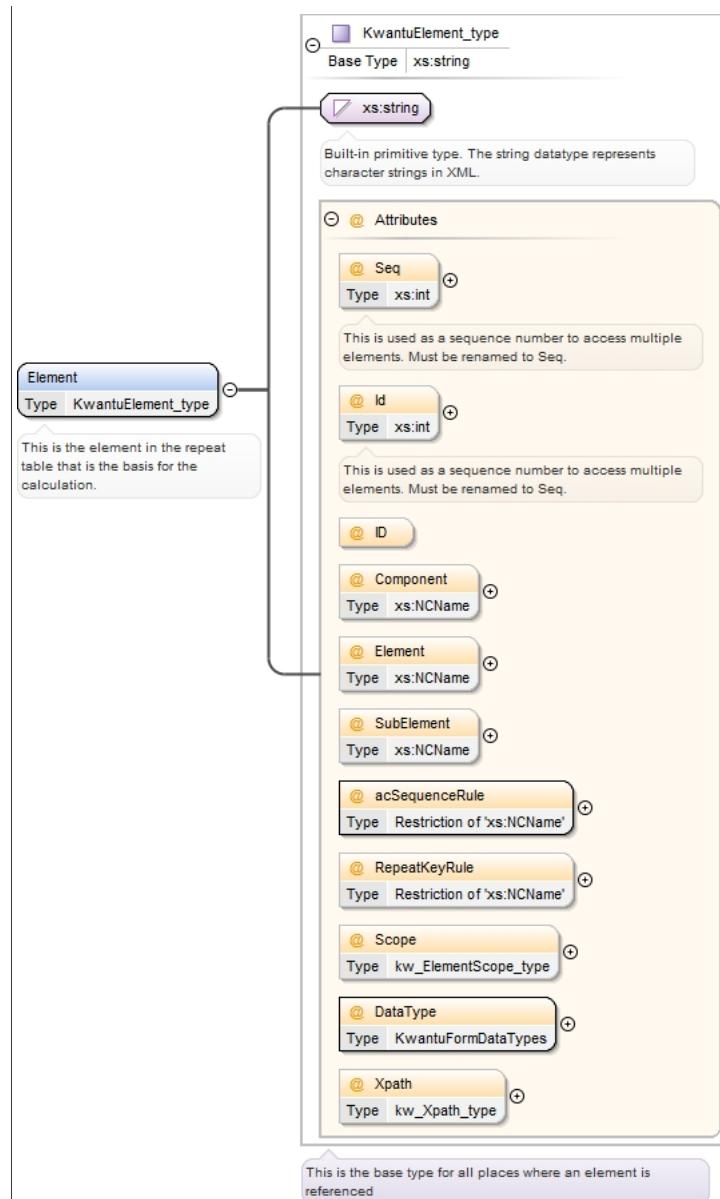
Namespace	No namespace
Annotations	<p>This element specifies which calculation should be performed</p> <ul style="list-style-type: none"> <li>- Sum: Sum all numeric values</li> <li>- Max: Return the maximum numeric value</li> <li>- Min: Return the minimum numeric value</li> <li>- Count: Return the number of rows in the column</li> <li>- CountUnique: Return the number of Unique values in the column</li> </ul>
Diagram	<p>The diagram shows a Calculation node (Type: Restriction of 'xs:NCName') connected to a restricts: xs:NCName node. A tooltip for the Calculation node states: "This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum...".</p>

Type	restriction of xs:NCName	
Properties	content: simple	
Facets	enumeration	Sum
	enumeration	Max
	enumeration	Min
	enumeration	Multiply
	enumeration	Count
	enumeration	AccumulativeSum
Source	<pre> &lt;xs:element name="Calculation"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum numeric value - Min: Return the minimum numeric value - Count: Return the number of rows in the column - CountUnique: Return the number of Unique values in the column&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="Sum"/&gt;       &lt;xs:enumeration value="Max"/&gt;       &lt;xs:enumeration value="Min"/&gt;       &lt;xs:enumeration value="Multiply"/&gt;       &lt;xs:enumeration value="Count"/&gt;       &lt;xs:enumeration value="AccumulativeSum"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:element&gt; </pre>	

### **Element fnRowCalc / Element**

Namespace	No namespace
Annotations	This is the element in the repeat table that is the basis for the calculation.

Diagram



Type	KwantuElement_type																																																			
Properties	content: complex maxOccurs: unbounded																																																			
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> <th></th> </tr> </thead> <tbody> <tr> <td>Component</td> <td>xs:NCName</td> <td>optional</td> <td></td> </tr> <tr> <td>DataType</td> <td>KwantuFormDataTypes</td> <td>required</td> <td></td> </tr> <tr> <td>Element</td> <td>xs:NCName</td> <td>optional</td> <td></td> </tr> <tr> <td>ID</td> <td></td> <td>optional</td> <td></td> </tr> <tr> <td>Id</td> <td>xs:int</td> <td>optional</td> <td></td> </tr> <tr> <td></td> <td colspan="2">This is used as a sequence number to access multiple elements. Must be renamed to <code>Seq</code>.</td> <td></td> </tr> <tr> <td>RepeatKeyRule</td> <td>restriction of xs:NCName</td> <td>optional</td> <td></td> </tr> <tr> <td>Scope</td> <td>kw_ElementScope_type</td> <td>optional</td> <td></td> </tr> <tr> <td>Seq</td> <td>xs:int</td> <td>optional</td> <td></td> </tr> <tr> <td></td> <td colspan="2">This is used as a sequence number to access multiple elements. Must be renamed to <code>Seq</code>.</td> <td></td> </tr> <tr> <td>SubElement</td> <td>xs:NCName</td> <td>optional</td> <td></td> </tr> </tbody> </table>				QName	Type	Use		Component	xs:NCName	optional		DataType	KwantuFormDataTypes	required		Element	xs:NCName	optional		ID		optional		Id	xs:int	optional			This is used as a sequence number to access multiple elements. Must be renamed to <code>Seq</code> .			RepeatKeyRule	restriction of xs:NCName	optional		Scope	kw_ElementScope_type	optional		Seq	xs:int	optional			This is used as a sequence number to access multiple elements. Must be renamed to <code>Seq</code> .			SubElement	xs:NCName	optional	
QName	Type	Use																																																		
Component	xs:NCName	optional																																																		
DataType	KwantuFormDataTypes	required																																																		
Element	xs:NCName	optional																																																		
ID		optional																																																		
Id	xs:int	optional																																																		
	This is used as a sequence number to access multiple elements. Must be renamed to <code>Seq</code> .																																																			
RepeatKeyRule	restriction of xs:NCName	optional																																																		
Scope	kw_ElementScope_type	optional																																																		
Seq	xs:int	optional																																																		
	This is used as a sequence number to access multiple elements. Must be renamed to <code>Seq</code> .																																																			
SubElement	xs:NCName	optional																																																		

	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>Xpath</b>	kw_Xpath_type	optional
	<b>acSequenceRule</b>	restriction of xs:NCName	required
Source	<pre>&lt;xs:element maxOccurs="unbounded" name="Element" type="KwantuElement_type"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This is the element in the repeat table that is the basis for the calculation.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>		

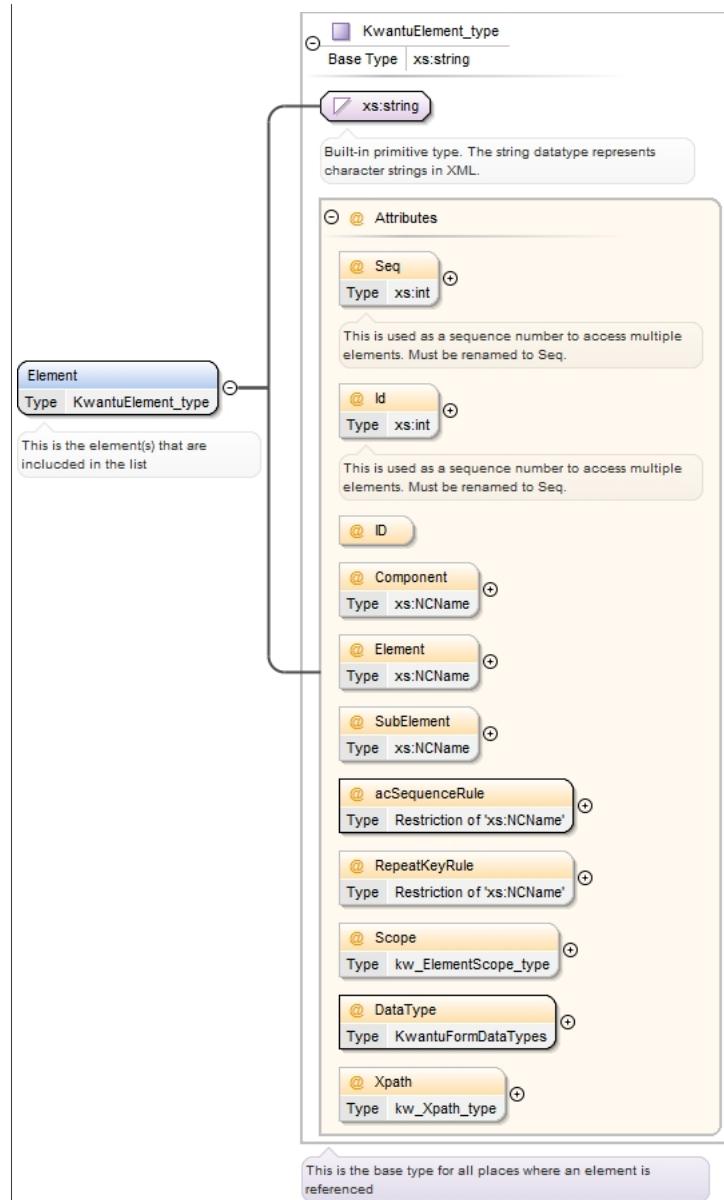
## Element CalcType / ElementsList

Namespace	No namespace
Diagram	
Type	fnElementsList
Properties	content: complex
Model	Element   (Calculation , Element+)
Children	Calculation, Element
Instance	<pre>&lt;ElementsList&gt;   &lt;Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE...   Element&gt;   &lt;Calculation&gt;{1,1}&lt;/Calculation&gt;   &lt;Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE...   Element&gt; &lt;/ElementsList&gt;</pre>
Source	<pre>&lt;xs:element name="ElementsList" type="fnElementsList"/&gt;</pre>

## Element fnElementsList / Element

Namespace	No namespace
Annotations	This is the element(s) that are included in the list

Diagram



Type	KwantuElement_type		
Properties	content: complex		
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>Component</b>	<code>xs:NCName</code>	optional
	<b>DataType</b>	<code>KwantuFormDataTypes</code>	required
	<b>Element</b>	<code>xs:NCName</code>	optional
	<b>ID</b>		optional
	<b>Id</b>	<code>xs:int</code>	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>RepeatKeyRule</b>	restriction of <code>xs:NCName</code>	optional
	<b>Scope</b>	<code>kw_ElementScope_type</code>	optional
	<b>Seq</b>	<code>xs:int</code>	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>SubElement</b>	<code>xs:NCName</code>	optional
	<b>Xpath</b>	<code>kw_Xpath_type</code>	optional

	QName	Type	Use	
	acSequenceRule	restriction of xs:NCName	required	
Source	<pre>&lt;xs:element name="Element" type="KwantuElement_type"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This is the element(s) that are included in the list&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>			

### Element fnElementsList / Calculation

Namespace	No namespace		
Annotations	<p>This element specifies which calculation should be performed</p> <p>Sum Max Min Count counts the number of non-null values Count Unique counts the unique values identified</p>		
Diagram	<p>This element specifies which calculation should be performed Sum Max Min Count counts the number of non-null values...</p>		
Type	restriction of xs:NCName		
Properties	content:	simple	
Facets	enumeration	Sum	
	enumeration	Max	
	enumeration	Min	
	enumeration	CountUnique	
Source	<pre>&lt;xs:element name="Calculation"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This element specifies which calculation should be performed Sum Max Min Count counts the number of non-null values Count Unique counts the unique values identified&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="Sum"/&gt;       &lt;xs:enumeration value="Max"/&gt;       &lt;xs:enumeration value="Min"/&gt;       &lt;xs:enumeration value="CountUnique"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:element&gt;</pre>		

### Element CalcType / CalcFinYearDates

Namespace	No namespace		
Diagram	<p>This element specifies which calculation should be performed - CalendarDays: calculates the number of calendar days...</p> <p>This is the element that contains the date that is provided to calculate the financial Years</p>		
Type	fnCalcFinYearDates		
Properties	content:	complex	
Model	Calculation , Element		
Children	Calculation, Element		

Instance	<pre>&lt;CalcFinYearDates&gt;   &lt;Calculation&gt;{1,1}&lt;/Calculation&gt;   &lt;Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE...&gt; &lt;/CalcFinYearDates&gt;</pre>
Source	<pre>&lt;xss:element name="CalcFinYearDates" type="fnCalcFinYearDates" /&gt;</pre>

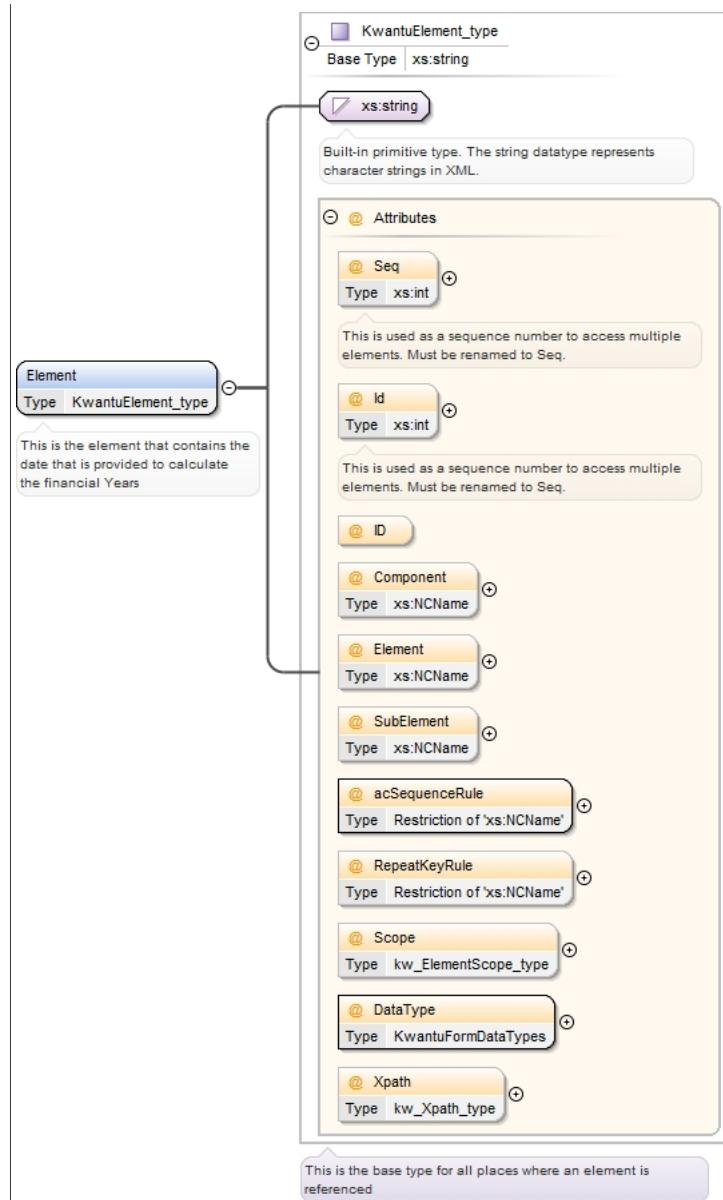
### Element fnCalcFinYearDates / Calculation

Namespace	No namespace								
Annotations	<p>This element specifies which calculation should be performed</p> <ul style="list-style-type: none"> <li>- CalendarDays: calculates the number of calendar days between date1 and date2</li> <li>- WorkDays: calculates the number of work days between date1 and date2. Work days includes Mondays to Fridays</li> </ul>								
Diagram	<pre> classDiagram     class Calculation {         &lt;&lt;Type Restriction of xs:NCName&gt;&gt;     }     Calculation --o Note: This element specifies which calculation should be performed - CalendarDays: calculates the number of calendar days...   </pre>								
Type	restriction of xs:NCName								
Properties	content: simple								
Facets	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">StartDateFY1April</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">StartDateFY1July</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">EndDateFY1April</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">EndDateFY1July</td> </tr> </table>	enumeration	StartDateFY1April	enumeration	StartDateFY1July	enumeration	EndDateFY1April	enumeration	EndDateFY1July
enumeration	StartDateFY1April								
enumeration	StartDateFY1July								
enumeration	EndDateFY1April								
enumeration	EndDateFY1July								
Source	<pre> &lt;xss:element name="Calculation"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;This element specifies which calculation should be performed - CalendarDays: calculates the number of calendar days between date1 and date2 - WorkDays: calculates the number of work days between date1 and date2. Work days includes Mondays to Fridays&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt;   &lt;xss:simpleType&gt;     &lt;xss:restriction base="xs:NCName"&gt;       &lt;xss:enumeration value="StartDateFY1April" /&gt;       &lt;xss:enumeration value="StartDateFY1July" /&gt;       &lt;xss:enumeration value="EndDateFY1April" /&gt;       &lt;xss:enumeration value="EndDateFY1July" /&gt;     &lt;/xss:restriction&gt;   &lt;/xss:simpleType&gt; &lt;/xss:element&gt;   </pre>								

### Element fnCalcFinYearDates / Element

Namespace	No namespace
Annotations	This is the element that contains the date that is provided to calculate the financial Years

Diagram



Type	KwantuElement_type		
Properties	content: complex		
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>Component</b>	<code>xs:NCName</code>	optional
	<b>DataType</b>	<code>KwantuFormDataTypes</code>	required
	<b>Element</b>	<code>xs:NCName</code>	optional
	<b>ID</b>		optional
	<b>Id</b>	<code>xs:int</code>	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>RepeatKeyRule</b>	restriction of <code>xs:NCName</code>	optional
	<b>Scope</b>	<code>kw_ElementScope_type</code>	optional
	<b>Seq</b>	<code>xs:int</code>	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>SubElement</b>	<code>xs:NCName</code>	optional
	<b>Xpath</b>	<code>kw_Xpath_type</code>	optional

	QName	Type	Use	
	acSequenceRule	restriction of xs:NCName	required	
Source	<pre>&lt;xs:element name="Element" type="KwantuElement_type"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This is the element that contains the date that is provided to calculate the financial Years&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>			

## Element CalcType / CalcHistory

Namespace	No namespace
Diagram	
Type	fnHistoryTotal
Properties	content: complex
Model	Calculation , Element , Filters{0,1}
Children	Calculation, Element, Filters
Instance	<pre>&lt;CalcHistory&gt;   &lt;Calculation&gt;{1,1}&lt;/Calculation&gt;   &lt;Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE...   &lt;Filters&gt;{0,1}&lt;/Filters&gt; &lt;/CalcHistory&gt;</pre>
Source	<pre>&lt;xs:element name="CalcHistory" type="fnHistoryTotal"/&gt;</pre>

## Element fnHistoryTotal / Calculation

Namespace	No namespace												
Annotations	<p>This element specifies which calculation should be performed</p> <ul style="list-style-type: none"> <li>- Sum: Sum all numeric values</li> <li>- Max: Return the maximum numeric value</li> <li>- Min: Return the minimum numeric value</li> <li>- Count: Return the number of rows in the column</li> <li>- CountUnique: Return the number of Unique values in the column</li> </ul>												
Diagram													
Type	restriction of xs:NCName												
Properties	content: simple												
Facets	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">Sum</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">Max</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">Min</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">Count</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">CountUnique</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">AccumulativeSum</td> </tr> </table>	enumeration	Sum	enumeration	Max	enumeration	Min	enumeration	Count	enumeration	CountUnique	enumeration	AccumulativeSum
enumeration	Sum												
enumeration	Max												
enumeration	Min												
enumeration	Count												
enumeration	CountUnique												
enumeration	AccumulativeSum												
Source	<pre>&lt;xs:element name="Calculation"&gt;   &lt;xs:annotation&gt;</pre>												

```

<xs:documentation>This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum numeric value - Min: Return the minimum numeric value - Count: Return the number of rows in the column - CountUnique: Return the number of Unique values in the column</xs:documentation>
</xs:annotation>
<xs:simpleType>
  <xs:restriction base="xs:NCName">
    <xs:enumeration value="Sum"/>
    <xs:enumeration value="Max"/>
    <xs:enumeration value="Min"/>
    <xs:enumeration value="Count"/>
    <xs:enumeration value="CountUnique"/>
    <xs:enumeration value="AccumulativeSum"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>

```

## Element fnHistoryTotal / Element

Namespace	No namespace				
Annotations	This is the element in the repeat table that is the basis for the calculation.				
Diagram	<p>The diagram illustrates the structure of the <code>KwantuElement_type</code>. It is defined as a base type for <code>xs:string</code>. The type has several attributes:</p> <ul style="list-style-type: none"> <li><code>@Seq</code>: Type <code>xs:int</code>. Description: This is used as a sequence number to access multiple elements. Must be renamed to Seq.</li> <li><code>@Id</code>: Type <code>xs:int</code>. Description: This is used as a sequence number to access multiple elements. Must be renamed to Id.</li> <li><code>@ID</code></li> <li><code>@Component</code>: Type <code>xs:NCName</code></li> <li><code>@Element</code>: Type <code>xs:NCName</code></li> <li><code>@SubElement</code>: Type <code>xs:NCName</code></li> <li><code>@acSequenceRule</code>: Type <code>Restriction of 'xs:NCName'</code></li> <li><code>@RepeatKeyRule</code>: Type <code>Restriction of 'xs:NCName'</code></li> <li><code>@Scope</code>: Type <code>kw_ElementScope_type</code></li> <li><code>@DataType</code>: Type <code>KwantuFormDataTypes</code></li> <li><code>@Xpath</code>: Type <code>kw_Xpath_type</code></li> </ul> <p>A note at the bottom states: "This is the base type for all places where an element is referenced".</p>				
Type	<code>KwantuElement_type</code>				
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	complex	maxOccurs:	1
content:	complex				
maxOccurs:	1				

Attributes	QName	Type	Use
	<b>Component</b>	xs:NCName	optional
	<b>DataType</b>	KwantuFormDataTypes	required
	<b>Element</b>	xs:NCName	optional
	<b>ID</b>		optional
	<b>Id</b>	xs:int	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>RepeatKeyRule</b>	restriction of xs:NCName	optional
	<b>Scope</b>	kw_ElementScope_type	optional
	<b>Seq</b>	xs:int	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>SubElement</b>	xs:NCName	optional
	<b>Xpath</b>	kw_Xpath_type	optional
	<b>acSequenceRule</b>	restriction of xs:NCName	required
Source	<pre>&lt;xs:element maxOccurs="1" name="Element" type="KwantuElement_type"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This is the element in the repeat table that is the basis for the calculation.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>		

## Element fnHistoryTotal / Filters

Namespace	No namespace				
Diagram	<pre> classDiagram     class Filters     class Filter     Filters "1..infinity" --&gt; Filter   </pre>				
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Model	Filter+				
Children	Filter				
Instance	<pre>&lt;Filters&gt;   &lt;Filter FilterElement="" Parameters="" Reference="" Seq="" Type=""&gt;{1,unbounded}&lt;/Filter&gt; &lt;/Filters&gt;</pre>				
Source	<pre>&lt;xs:element minOccurs="0" name="Filters"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element maxOccurs="unbounded" name="Filter"&gt;         &lt;xs:complexType&gt;           &lt;xs:sequence minOccurs="0"&gt;             &lt;xs:element minOccurs="0" name="Parameters"&gt;               &lt;xs:complexType&gt;                 &lt;xs:sequence&gt;                   &lt;xs:element name="Element" type="KwantuElement_type" minOccurs="0"/&gt;                 &lt;/xs:sequence&gt;               &lt;/xs:complexType&gt;             &lt;/xs:element&gt;             &lt;xs:element minOccurs="0" name="Condition" type="Condition_type"&gt;               &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;If the conditional element is present, it evaluates to a boolean. If the f(Conditional)=true then the validation rule applies, otherwise it does not apply. Ie. to specify that the field is required if the value of another field is "Other" then the following configuration would be applied: Validation: @Seq=1; @Type=NotNull ---&gt; this specifies that the current element value should not be null Condition[@Operator="Equal"] Condition/Element: {here the reference to the field that could have the value of other} Condition/Value:"Other"&lt;/xs:documentation&gt;               &lt;/xs:annotation&gt;             &lt;/xs:element&gt;           &lt;/xs:sequence&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre> <p>If the f(Conditional)=true then the validation rule applies, otherwise it does not apply. Ie. to specify that the field is required if the value of another field is "Other" then the following configuration would be applied: Validation: @Seq=1; @Type=NotNull ---&gt; this specifies that the current element value should not be null Condition[@Operator="Equal"] Condition/Element: {here the reference to the field that could have the value of other} Condition/Value:"Other"</p>				

```

</xs:sequence>
</xs:complexType>
</xs:element>
```

### **Element fnHistoryTotal / Filters / Filter**

Namespace	No namespace																		
Diagram																			
Properties	content: complex maxOccurs: unbounded																		
Model	Parameters{0,1} , Condition{0,1}																		
Children	Condition, Parameters																		
Instance	<pre>&lt;Filter FilterElement="" Parameters="" Reference="" Seq="" Type=""&gt;   &lt;Parameters&gt;{0,1}&lt;/Parameters&gt;   &lt;Condition DataType="" Operator="" Value=""&gt;{0,1}&lt;/Condition&gt; &lt;/Filter&gt;</pre>																		
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">QName</th> <th style="text-align: left; padding: 2px;">Type</th> <th style="text-align: left; padding: 2px;">Use</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">FilterElement</td> <td style="padding: 2px;">FilterElement_type</td> <td style="padding: 2px;">optional</td> </tr> <tr> <td style="padding: 2px;">Parameters</td> <td style="padding: 2px;"></td> <td style="padding: 2px;">optional</td> </tr> <tr> <td style="padding: 2px;">Reference</td> <td style="padding: 2px;"></td> <td style="padding: 2px;">optional</td> </tr> <tr> <td style="padding: 2px;">Seq</td> <td style="padding: 2px;">xs:integer</td> <td style="padding: 2px;">required</td> </tr> <tr> <td style="padding: 2px;">Type</td> <td style="padding: 2px;">kwantu_comparison_types</td> <td style="padding: 2px;">required</td> </tr> </tbody> </table>	QName	Type	Use	FilterElement	FilterElement_type	optional	Parameters		optional	Reference		optional	Seq	xs:integer	required	Type	kwantu_comparison_types	required
QName	Type	Use																	
FilterElement	FilterElement_type	optional																	
Parameters		optional																	
Reference		optional																	
Seq	xs:integer	required																	
Type	kwantu_comparison_types	required																	
Source	<pre>&lt;xs:element maxOccurs="unbounded" name="Filter"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence minOccurs="0"&gt;       &lt;xs:element minOccurs="0" name="Parameters"&gt;         &lt;xs:complexType&gt;           &lt;xs:sequence&gt;             &lt;xs:element name="Element" type="KwantuElement_type" minOccurs="0"/&gt;           &lt;/xs:sequence&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;       &lt;xs:element minOccurs="0" name="Condition" type="Condition_type"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;If the conditional element is present, it evaluates to a boolean. If the f(Conditional)=true then the validation rule applies, otherwise it does not apply. Ie. to specify that the field is required if the value of another field is "Other" then the following configuration would be applied: Validation: @Seq=1; @Type=NotNull ---&gt; this specifies that the current element value should not be null Condition[@Operator="Equal"] Condition/Element: {here the reference to the field that could have the value of other} Condition/Value: "Other"&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute name="Seq" type="xs:integer" use="required"/&gt;     &lt;xs:attribute name="Type" use="required" type="kwantu_comparison_types"/&gt;     &lt;xs:attribute name="Parameters"/&gt;     &lt;xs:attribute name="Reference"/&gt;     &lt;xs:attribute name="FilterElement" type="FilterElement_type"/&gt;</pre>																		

```
</xs:complexType>
</xs:element>
```

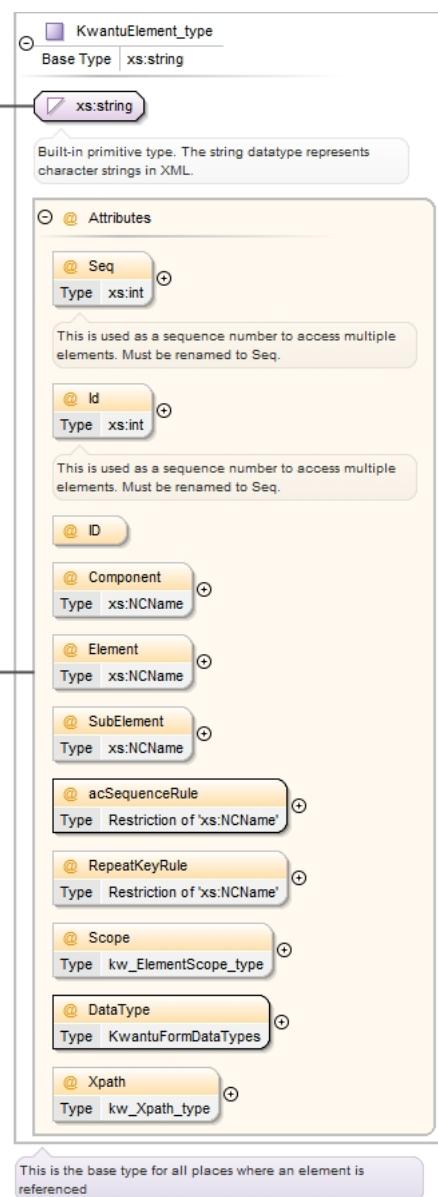
### **Element fnHistoryTotal / Filters / Filter / Parameters**

Namespace	No namespace
Diagram	
Properties	<p>content: complex</p> <p>minOccurs: 0</p>
Model	Element{0,1}
Children	Element
Instance	<pre>&lt;Parameters&gt;   &lt;Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" SubE     Element&gt;   &lt;/Parameters&gt;</pre>
Source	<pre>&lt;xs:element minOccurs="0" name="Parameters"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="Element" type="KwantuElement_type" minOccurs="0"/&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>

### **Element fnHistoryTotal / Filters / Filter / Parameters / Element**

Namespace	No namespace
-----------	--------------

Diagram



Type	KwantuElement_type																																								
Properties	content: complex minOccurs: 0																																								
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><b>Component</b></td> <td><code>xs:NCName</code></td> <td>optional</td> </tr> <tr> <td><b>DataType</b></td> <td><code>KwantuFormDataTypes</code></td> <td>required</td> </tr> <tr> <td><b>Element</b></td> <td><code>xs:NCName</code></td> <td>optional</td> </tr> <tr> <td><b>ID</b></td> <td></td> <td>optional</td> </tr> <tr> <td><b>Id</b></td> <td><code>xs:int</code></td> <td>optional</td> </tr> <tr> <td><b>Seq</b></td> <td colspan="2">This is used as a sequence number to access multiple elements. Must be renamed to <code>Seq</code>.</td><td></td></tr> <tr> <td><b>RepeatKeyRule</b></td> <td><code>restriction of xs:NCName</code></td> <td>optional</td><td></td></tr> <tr> <td><b>Scope</b></td> <td><code>kw_ElementScope_type</code></td> <td>optional</td><td></td></tr> <tr> <td><b>SubElement</b></td> <td><code>xs:NCName</code></td> <td>optional</td><td></td></tr> <tr> <td><b>Xpath</b></td> <td><code>kw_Xpath_type</code></td> <td></td><td></td></tr> </tbody> </table>			QName	Type	Use	<b>Component</b>	<code>xs:NCName</code>	optional	<b>DataType</b>	<code>KwantuFormDataTypes</code>	required	<b>Element</b>	<code>xs:NCName</code>	optional	<b>ID</b>		optional	<b>Id</b>	<code>xs:int</code>	optional	<b>Seq</b>	This is used as a sequence number to access multiple elements. Must be renamed to <code>Seq</code> .			<b>RepeatKeyRule</b>	<code>restriction of xs:NCName</code>	optional		<b>Scope</b>	<code>kw_ElementScope_type</code>	optional		<b>SubElement</b>	<code>xs:NCName</code>	optional		<b>Xpath</b>	<code>kw_Xpath_type</code>		
QName	Type	Use																																							
<b>Component</b>	<code>xs:NCName</code>	optional																																							
<b>DataType</b>	<code>KwantuFormDataTypes</code>	required																																							
<b>Element</b>	<code>xs:NCName</code>	optional																																							
<b>ID</b>		optional																																							
<b>Id</b>	<code>xs:int</code>	optional																																							
<b>Seq</b>	This is used as a sequence number to access multiple elements. Must be renamed to <code>Seq</code> .																																								
<b>RepeatKeyRule</b>	<code>restriction of xs:NCName</code>	optional																																							
<b>Scope</b>	<code>kw_ElementScope_type</code>	optional																																							
<b>SubElement</b>	<code>xs:NCName</code>	optional																																							
<b>Xpath</b>	<code>kw_Xpath_type</code>																																								

	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>Xpath</b>	kw_Xpath_type	optional
	<b>acSequenceRule</b>	restriction of xs:NCName	required
Source	<xss:element name="Element" type="KwantuElement_type" minOccurs="0" />		

### Element fnHistoryTotal / Filters / Filter / Condition

Namespace	No namespace												
Annotations	<p>If the conditional element is present, it evaluates to a boolean. If the f(Conditional)=true then the validation rule applies, otherwise it does not apply.</p> <p>Ie. to specify that the field is required if the value of another field is "Other" then the following configuration would be applied:</p> <p>Validation: @Seq=1; @Type=NotNull ---&gt; this specifies that the current element value should not be null</p> <p>Condition[@Operator="Equal"]</p> <p>Condition/Element: {here the reference to the field that could have the value of other}</p> <p>Condition/Value:"Other"</p>												
Diagram	<pre> classDiagram     class Condition_type {         @ Attributes         @ Operator Type kwantu_comparison_types         @ Value         @ DataType Type KwantuFormDataTypes     }     class Condition {         Type Condition_type     }     Condition "1" -- "0..1" Condition_type     Condition_type "1" -- "1" Element Type KwantuElement_type     Condition_type "1" -- "1" ReferenceElement Type KwantuElement_type     Note over Condition_type: If the conditional element is present, it evaluates to a boolean. If the f(Conditional)=true then the validation rule...   </pre>												
Type	Condition_type												
Properties	<p>content: complex</p> <p>minOccurs: 0</p>												
Model	Element , ReferenceElement{0,1}												
Children	Element, ReferenceElement												
Instance	<pre> &lt;Condition DataType="" Operator="" Value=""&gt;   &lt;Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE...   Element&gt;   &lt;ReferenceElement acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" Se...   ReferenceElement&gt; &lt;/Condition&gt;   </pre>												
Attributes	<table border="1"> <thead> <tr> <th><b>QName</b></th> <th><b>Type</b></th> <th><b>Use</b></th> </tr> </thead> <tbody> <tr> <td><b>DataType</b></td><td>KwantuFormDataTypes</td><td>optional</td></tr> <tr> <td><b>Operator</b></td><td>kwantu_comparison_types</td><td>required</td></tr> <tr> <td><b>Value</b></td><td></td><td>optional</td></tr> </tbody> </table>	<b>QName</b>	<b>Type</b>	<b>Use</b>	<b>DataType</b>	KwantuFormDataTypes	optional	<b>Operator</b>	kwantu_comparison_types	required	<b>Value</b>		optional
<b>QName</b>	<b>Type</b>	<b>Use</b>											
<b>DataType</b>	KwantuFormDataTypes	optional											
<b>Operator</b>	kwantu_comparison_types	required											
<b>Value</b>		optional											
Source	<pre> &lt;xss:element minOccurs="0" name="Condition" type="Condition_type"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;If the conditional element is present, it evaluates to a boolean. If the f(Conditional)=true then the validation rule applies, otherwise it does not apply. Ie. to specify that the field is required if the value of another field is "Other" then the following configuration would be applied: Validation: @Seq=1; @Type=NotNull ---&gt; this specifies that the current element value should not be null Condition[@Operator="Equal"] Condition/Element: {here the reference to the field that could have the value of other} Condition/Value:"Other"&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt; &lt;/xss:element&gt;   </pre>												

## Element Default\_type / ElementValue

Namespace	No namespace																														
Annotations	This option returns a value from a different element as the default value																														
Diagram	<pre> classDiagram     class KwantuElement_type {         xs:string     }     class ElementValue {         Type KwantuElement_type     }     KwantuElement_type &lt; -- ElementValue     ElementValue &lt; -- xs:string     ElementValue &lt; -- @ Seq : xs:int     ElementValue &lt; -- @ Id : xs:int     ElementValue &lt; -- @ ID : ID     ElementValue &lt; -- @ Component : xs:NCName     ElementValue &lt; -- @ Element : xs:NCName     ElementValue &lt; -- @ SubElement : xs:NCName     ElementValue &lt; -- @ acSequenceRule : Restriction of 'xs:NCName'     ElementValue &lt; -- @ RepeatKeyRule : Restriction of 'xs:NCName'     ElementValue &lt; -- @ Scope : kw_ElementScope_type     ElementValue &lt; -- @ DataType : KwantuFormDataTypes     ElementValue &lt; -- @ Xpath : kw_Xpath_type   </pre> <p>This is the base type for all places where an element is referenced</p>																														
Type	KwantuElement_type																														
Properties	content: complex																														
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>Component</td> <td>xs:NCName</td> <td>optional</td> </tr> <tr> <td>DataType</td> <td>KwantuFormDataTypes</td> <td>required</td> </tr> <tr> <td>Element</td> <td>xs:NCName</td> <td>optional</td> </tr> <tr> <td>ID</td> <td></td> <td>optional</td> </tr> <tr> <td>Id</td> <td>xs:int</td> <td>optional</td> </tr> <tr> <td></td> <td colspan="2">This is used as a sequence number to access multiple elements. Must be renamed to Seq.</td></tr> <tr> <td>RepeatKeyRule</td> <td>restriction of xs:NCName</td> <td>optional</td> </tr> <tr> <td>Scope</td> <td>kw_ElementScope_type</td> <td>optional</td> </tr> <tr> <td>Seq</td> <td>xs:int</td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Use	Component	xs:NCName	optional	DataType	KwantuFormDataTypes	required	Element	xs:NCName	optional	ID		optional	Id	xs:int	optional		This is used as a sequence number to access multiple elements. Must be renamed to Seq.		RepeatKeyRule	restriction of xs:NCName	optional	Scope	kw_ElementScope_type	optional	Seq	xs:int	optional
QName	Type	Use																													
Component	xs:NCName	optional																													
DataType	KwantuFormDataTypes	required																													
Element	xs:NCName	optional																													
ID		optional																													
Id	xs:int	optional																													
	This is used as a sequence number to access multiple elements. Must be renamed to Seq.																														
RepeatKeyRule	restriction of xs:NCName	optional																													
Scope	kw_ElementScope_type	optional																													
Seq	xs:int	optional																													

	QName	Type	Use	
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.		
	<b>SubElement</b>	xs:NCName	optional	
	<b>Xpath</b>	kw_Xpath_type	optional	
	<b>acSequenceRule</b>	restriction of xs:NCName	required	
Source	<pre>&lt;xs:element name="ElementValue" type="KwantuElement_type"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This option returns a value from a different element as the default value&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>			

### Element kw\_elField\_type / EntryMode / AssignElement

Namespace	No namespace				
Annotations	<p>This provides for mapped fields to map the source element to the element in the current repeat. The source repeat is specified in the Component element Attributes.</p> <p>All the elements that you want to autoload from the source repeat are mapped here to local element ID. These are specific to the local repeat only.</p>				
Diagram	<p>This provides for mapped fields to map the source element to the element in the current repeat. The source repeat is...</p>				
Type	extension of xs:normalizedString				
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Source	<pre>&lt;xs:element minOccurs="0" name="AssignElement"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This provides for mapped fields to map the source element to the element in the current repeat. The source repeat is specified in the Component element Attributes. All the elements that you want to autoload from the source repeat are mapped here to local element ID. These are specific to the local repeat only.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:simpleContent&gt;       &lt;xs:extension base="xs:normalizedString" /&gt;     &lt;/xs:simpleContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>				

### Element kw\_elField\_type / EntryMode / AssignSubElements

Namespace	No namespace		
Diagram	<p>Here you assign one or more values from the taxonomy to element IDs associated with the Select Element ID</p> <p>Here you assign one or more values from the taxonomy to element IDs associated with the Select Element ID</p> <p>Here you assign one or more values from the taxonomy to element IDs associated with the Select Element ID</p>		
Type	AssignSubElement_type		
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> </table>	content:	complex
content:	complex		

	minOccurs:	0
Model	AssignCode , AssignName{0,1} , AssignSubElement*	
Children	AssignCode, AssignName, AssignSubElement	
Instance	<pre>&lt;AssignSubElements&gt;   &lt;AssignCode DataType="" SubElementID="" {1,1}&gt;&lt;/AssignCode&gt;   &lt;AssignName DataType="" SubElementID="" {0,1}&gt;&lt;/AssignName&gt;   &lt;AssignSubElement Action="" DataType="" SubElementID="" {0,unbounded}&gt;&lt;/AssignSubElement&gt; &lt;/AssignSubElements&gt;</pre>	
Source	<pre>&lt;xss:element name="AssignSubElements" type="AssignSubElement_type" minOccurs="0"/&gt;</pre>	

### Element AssignSubElement\_type / AssignCode

Namespace	No namespace											
Annotations	Here you assign one or more values from the taxonomy to element IDs associated with the Select Element ID											
Diagram	<pre> classDiagram     class AssignCode {         &lt;&lt;Extension of xs:normalizedString&gt;&gt;         &lt;&lt;Here you assign one or more values from the taxonomy to element IDs associated with the Select Element ID&gt;&gt;         @Attributes             SubElementID : xs:NCName             DataType : KwantuFormDataTypes     } </pre>											
Type	extension of xs:normalizedString											
Properties	content: complex											
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>SubElementID</td> <td>xs:NCName</td> <td>required</td> </tr> <tr> <td>DataType</td> <td>KwantuFormDataTypes</td> <td>required</td> </tr> </tbody> </table>			QName	Type	Use	SubElementID	xs:NCName	required	DataType	KwantuFormDataTypes	required
QName	Type	Use										
SubElementID	xs:NCName	required										
DataType	KwantuFormDataTypes	required										
Source	<pre> &lt;xss:element name="AssignCode"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;Here you assign one or more values from the taxonomy to element IDs associated with the Select Element ID&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt;   &lt;xss:complexType&gt;     &lt;xss:simpleContent&gt;       &lt;xss:extension base="xs:normalizedString"&gt;         &lt;xss:attribute name="SubElementID" type="xs:NCName" use="required"/&gt;         &lt;xss:attribute name="DataType" type="KwantuFormDataTypes" use="required"/&gt;       &lt;/xss:extension&gt;     &lt;/xss:simpleContent&gt;   &lt;/xss:complexType&gt; &lt;/xss:element&gt; </pre>											

### Element AssignSubElement\_type / AssignName

Namespace	No namespace		
Annotations	Here you assign one or more values from the taxonomy to element IDs associated with the Select Element ID		
Diagram	<pre> classDiagram     class AssignName {         &lt;&lt;Extension of xs:normalizedString&gt;&gt;         &lt;&lt;Here you assign one or more values from the taxonomy to element IDs associated with the Select Element ID&gt;&gt;         @Attributes             SubElementID : xs:NCName             DataType : KwantuFormDataTypes     } </pre>		

Type	extension of xs:normalizedString		
Properties	content: complex minOccurs: 0		
Attributes	<b>QName</b> <b>Type</b> <b>Use</b> <b>DataType</b> KwantuFormDataTypes      required <b>SubElementID</b> xs:NCName      required		
Source	<pre>&lt;xs:element name="AssignName" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Here you assign one or more values from the taxonomy to element IDs associated with the Select Element ID&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:simpleContent&gt;       &lt;xs:extension base="xs:normalizedString"&gt;         &lt;xs:attribute name="SubElementID" type="xs:NCName" use="required"/&gt;         &lt;xs:attribute name="DataType" type="KwantuFormDataTypes" use="required"/&gt;       &lt;/xs:extension&gt;     &lt;/xs:simpleContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>		

## Element AssignSubElement\_type / AssignSubElement

Namespace	No namespace		
Annotations	Here you assign one or more values from the taxonomy to element IDs associated with the Select Element ID		
Diagram	<p>The diagram illustrates the structure of the <code>AssignSubElement</code> element. It is defined as an extension of <code>xs:normalizedString</code>. The element has attributes: <code>Action</code> (restriction of <code>xs:NCName</code>, with a note that <code>Action="Default"</code> means assignment only if element is null), <code>SubElementID</code> (type <code>xs:NCName</code>), and <code>DataType</code> (type <code>KwantuFormDataTypes</code>). A callout box provides a detailed explanation of the <code>xs:normalizedString</code> type.</p>		
Type	extension of xs:normalizedString		
Properties	content: complex minOccurs: 0 maxOccurs: unbounded		
Attributes	<b>QName</b> <b>Type</b> <b>Use</b> <b>Action</b> restriction of xs:NCName      optional (Note: Action="Default" means that the assignment only takes place if the element is currently null.) <b>DataType</b> KwantuFormDataTypes      required <b>SubElementID</b> xs:NCName      required		
Source	<pre>&lt;xs:element maxOccurs="unbounded" name="AssignSubElement" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Here you assign one or more values from the taxonomy to element IDs associated with the Select Element ID&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:simpleContent&gt;       &lt;xs:extension base="xs:normalizedString"&gt;         &lt;xs:attribute name="Action"&gt;           &lt;xs:annotation&gt;             &lt;xs:documentation&gt;Action="Default" means that the assignment only takes place if the element is currently null.&lt;/xs:documentation&gt;           &lt;/xs:annotation&gt;         &lt;/xs:attribute&gt;       &lt;/xs:extension&gt;     &lt;/xs:simpleContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>		

```

</xs:annotation>
<xs:simpleType>
    <xs:restriction base="xs:NCName">
        <xs:enumeration value="Default"/>
    </xs:restriction>
</xs:simpleType>
<xs:attribute name="SubElementID" type="xs:NCName" use="required"/>
<xs:attribute name="DataType" type="KwantuFormDataTypes" use="required"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>

```

## Element ReadOnlyIf

Namespace	No namespace
Diagram	<p>If the conditional element is present, it evaluates to a boolean. If the f(Conditional)=true then the validation rule...</p>
Properties	content: complex
Used by	Elements kw_elField_type/EntryMode, kw_elSelect_type/EntryMode
Model	Condition+
Children	Condition
Instance	<pre>&lt;ReadOnlyIf&gt;     &lt;Condition DataType="" Operator="" Value=""&gt;{1,unbounded}&lt;/Condition&gt; &lt;/ReadOnlyIf&gt;</pre>
Source	<pre>&lt;xs:element name="ReadOnlyIf"&gt;     &lt;xs:complexType&gt;         &lt;xs:sequence&gt;             &lt;xs:element name="Condition" type="Condition_type" maxOccurs="unbounded"&gt;                 &lt;xs:annotation&gt;                     &lt;xs:documentation&gt;If the conditional element is present, it evaluates to a boolean. If the f(Conditional)=true then the validation rule applies, otherwise it does not apply. Ie. to specify that the field is required if the value of another field is "Other" then the following configuration would be applied: Validation: @Seq=1; @Type=NotNull ---&gt; this specifies that the current element value should not be null Condition[@Operator="Equal"] Condition/Element: {here the reference to the field that could have the value of other} Condition/Value:"Other"&lt;/xs:documentation&gt;                 &lt;/xs:annotation&gt;             &lt;/xs:element&gt;         &lt;/xs:sequence&gt;     &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>

## Element ReadOnlyIf / Condition

Namespace	No namespace
Annotations	<p>If the conditional element is present, it evaluates to a boolean. If the f(Conditional)=true then the validation rule applies, otherwise it does not apply.</p> <p>Ie. to specify that the field is required if the value of another field is "Other" then the following configuration would be applied:</p> <p>Validation: @Seq=1; @Type=NotNull ---&gt; this specifies that the current element value should not be null</p> <p>Condition[@Operator="Equal"]</p> <p>Condition/Element: {here the reference to the field that could have the value of other}</p> <p>Condition/Value:"Other"</p>

Diagram	<pre> classDiagram     class Condition_type {         @ Attributes         @ Operator         @ Value         @ DataType     }     class Condition {         Type Condition_type     }     Condition "1" --&gt; Condition_type :      Condition_type "*" --&gt; Element :      Condition_type "*" --&gt; ReferenceElement :      Note over Condition: If the conditional element is present, it evaluates to a boolean. If the f(Conditional)=true then the validation rule...     Note over Condition_type: Not implemented yet   </pre>												
Type	Condition_type												
Properties	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">content:</td><td style="padding: 2px;">complex</td></tr> <tr> <td style="padding: 2px;">maxOccurs:</td><td style="padding: 2px;">unbounded</td></tr> </table>	content:	complex	maxOccurs:	unbounded								
content:	complex												
maxOccurs:	unbounded												
Model	Element , ReferenceElement{0,1}												
Children	Element, ReferenceElement												
Instance	<Condition DataType="" Operator="" Value="">   <Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE...   Element>   <ReferenceElement acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" Se...   ReferenceElement> </Condition>												
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="padding: 2px;">QName</th><th style="padding: 2px;">Type</th><th style="padding: 2px;">Use</th></tr> </thead> <tbody> <tr> <td style="padding: 2px;"><b>DataType</b></td><td style="padding: 2px;">KwantuFormDataTypes</td><td style="padding: 2px;">optional</td></tr> <tr> <td style="padding: 2px;"><b>Operator</b></td><td style="padding: 2px;">kwantu_comparison_types</td><td style="padding: 2px;">required</td></tr> <tr> <td style="padding: 2px;"><b>Value</b></td><td style="padding: 2px;"></td><td style="padding: 2px;">optional</td></tr> </tbody> </table>	QName	Type	Use	<b>DataType</b>	KwantuFormDataTypes	optional	<b>Operator</b>	kwantu_comparison_types	required	<b>Value</b>		optional
QName	Type	Use											
<b>DataType</b>	KwantuFormDataTypes	optional											
<b>Operator</b>	kwantu_comparison_types	required											
<b>Value</b>		optional											
Source	<pre> &lt;xs:element name="Condition" type="Condition_type" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;If the conditional element is present, it evaluates to a boolean. If the f(Conditional)=true then the validation rule applies, otherwise it does not apply. Ie. to specify that the field is required if the value of another field is "Other" then the following configuration would be applied: Validation: @Seq=1; @Type=NotNull ---&gt; this specifies that the current element value should not be null Condition[@Operator="Equal"] Condition/Element: {here the reference to the field that could have the value of other} Condition/Value:"Other"&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>												

## Element Validations

Namespace	No namespace		
Diagram	<pre> classDiagram     class kw_Validations_type {         @ Attributes         @ Required         @ DataType         0..∞ Validation         @ ErrorMessage     }     class Validations {         Type kw_Validations_type     }     Validations "1" --&gt; kw_Validations_type :    </pre>		
Type	kw_Validations_type		
Properties	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">content:</td><td style="padding: 2px;">complex</td></tr> </table>	content:	complex
content:	complex		

Used by	Element elCalcField Complex Types kw_elField_type, kw_elSelect_type
Model	Required , DataType , Validation* , ErrorMessage
Children	DataType, ErrorMessage, Required, Validation
Instance	<Validations> <Required>{1,1}</Required> <DataType>{1,1}</DataType> <Validation FilterElement="" Parameters="" Reference="" Seq="" Type="">{0,unbounded}</Validation> <ErrorMessage>{1,1}</ErrorMessage> </Validations>
Source	<xss:element name="Validations" type="kw_Validations_type"/>

### Element kw\_Validations\_type / Required

Namespace	No namespace
Diagram	
Type	xs:boolean
Properties	content: simple
Source	<xss:element name="Required" type="xs:boolean"/>

### Element DataType

Namespace	No namespace
Diagram	
Type	KwantuFormDataTypes
Properties	content: simple
Facets	enumeration String enumeration Integer enumeration Float enumeration Memo enumeration Email enumeration URI enumeration Date enumeration CoordinateDMS enumeration GeoPoint enumeration DataTypes enumeration YesNo enumeration Boolean
Used by	Complex Type kw_Validations_type
Source	<xss:element name="DataType" type="KwantuFormDataTypes"/>

### Element Validation

Namespace	No namespace
-----------	--------------

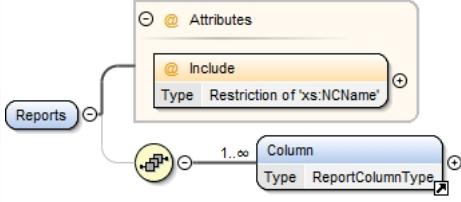
Diagram																			
Type	Kwantu_Filter_type																		
Properties	content: complex																		
Used by	Complex Type kw_Validations_type																		
Model	Parameters{0,1}   Condition{0,1}   Element																		
Children	Condition, Element, Parameters																		
Instance	<pre>&lt;Validation FilterElement="" Parameters="" References="" Seq="" Type=""&gt;   &lt;Parameters&gt;{0,1}&lt;/Parameters&gt;   &lt;Condition DataType="" Operator="" Value=""&gt;{0,1}&lt;/Condition&gt;   &lt;Element acSequenceRule="" Component="" DataType="" Element="" ID="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE&lt;/Element&gt; &lt;/Validation&gt;</pre>																		
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">QName</th> <th style="text-align: left; padding: 2px;">Type</th> <th style="text-align: left; padding: 2px;">Use</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">FilterElement</td><td style="padding: 2px;">FilterElement_type</td><td style="padding: 2px;">optional</td></tr> <tr> <td style="padding: 2px;">Parameters</td><td style="padding: 2px;"></td><td style="padding: 2px;">optional</td></tr> <tr> <td style="padding: 2px;">Reference</td><td style="padding: 2px;"></td><td style="padding: 2px;">optional</td></tr> <tr> <td style="padding: 2px;">Seq</td><td style="padding: 2px;">xs:integer</td><td style="padding: 2px;">required</td></tr> <tr> <td style="padding: 2px;">Type</td><td style="padding: 2px;">kwantu_comparison_types</td><td style="padding: 2px;">required</td></tr> </tbody> </table>	QName	Type	Use	FilterElement	FilterElement_type	optional	Parameters		optional	Reference		optional	Seq	xs:integer	required	Type	kwantu_comparison_types	required
QName	Type	Use																	
FilterElement	FilterElement_type	optional																	
Parameters		optional																	
Reference		optional																	
Seq	xs:integer	required																	
Type	kwantu_comparison_types	required																	
Source	<pre>&lt;xss:element name="Validation" type="Kwantu_Filter_type" /&gt;</pre>																		

### Element kw\_Validations\_type / ErrorMessage

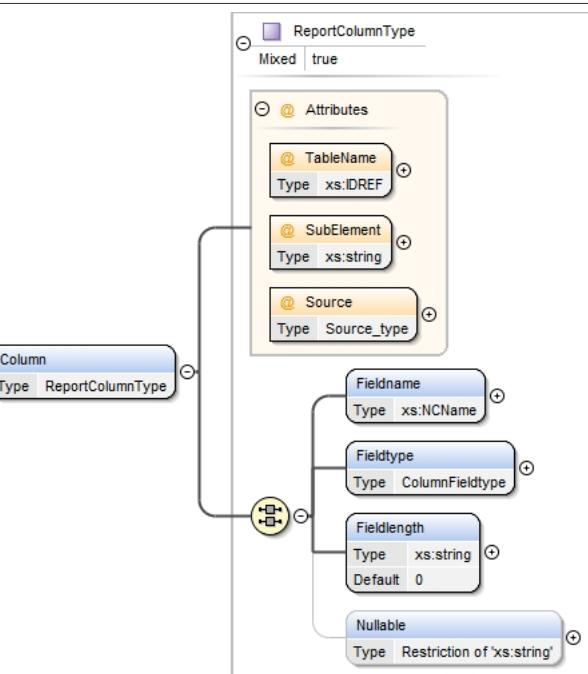
Namespace	No namespace
Diagram	
Type	xs:string
Properties	content: simple
Source	<pre>&lt;xss:element name="ErrorMessage" type="xs:string" /&gt;</pre>

### Element Reports

Namespace	No namespace
-----------	--------------

Diagram									
Properties	content: complex								
Used by	Element elCalcField Complex Types kw_elField_type, kw_elSelect_type								
Model	Column+								
Children	Column								
Instance	<pre>&lt;Reports Include=""&gt;   &lt;Column Source="" SubElement="" TableName=""&gt;{1,unbounded}&lt;/Column&gt; &lt;/Reports&gt;</pre>								
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> <th></th> </tr> </thead> <tbody> <tr> <td>Include</td> <td>restriction of xs:NCName</td> <td>required</td> <td></td> </tr> </tbody> </table>	QName	Type	Use		Include	restriction of xs:NCName	required	
QName	Type	Use							
Include	restriction of xs:NCName	required							
Source	<pre>&lt;xss:element name="Reports"&gt;   &lt;xss:complexType&gt;     &lt;xss:sequence minOccurs="0"&gt;       &lt;xss:element maxOccurs="unbounded" ref="Column"/&gt;     &lt;/xss:sequence&gt;     &lt;xss:attribute name="Include" use="required"&gt;       &lt;xss:simpleType&gt;         &lt;xss:restriction base="xs:NCName"&gt;           &lt;xss:enumeration value="Yes"/&gt;           &lt;xss:enumeration value="No"/&gt;         &lt;/xss:restriction&gt;       &lt;/xss:simpleType&gt;     &lt;/xss:attribute&gt;   &lt;/xss:complexType&gt; &lt;/xss:element&gt;</pre>								

## Element Column

Namespace	No namespace
Diagram	
Type	ReportColumnType
Properties	content: complex mixed: true

Used by	Element Reports																
Model	ALL(Fieldname Fieldtype Fieldlength Nullable{0,1})																
Children	Fieldlength, Fieldname, Fieldtype, Nullable																
Instance	<pre>&lt;Column Source="" SubElement="" TableName=""&gt;   &lt;Fieldname&gt;{1,1}&lt;/Fieldname&gt;   &lt;Fieldtype&gt;{1,1}&lt;/Fieldtype&gt;   &lt;Fieldlength&gt;{1,1}&lt;/Fieldlength&gt;   &lt;Nullable&gt;{0,1}&lt;/Nullable&gt; &lt;/Column&gt;</pre>																
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding-right: 10px;">QName</th> <th style="text-align: left; padding-right: 10px;">Type</th> <th style="text-align: left; padding-right: 10px;">Use</th> <th></th> </tr> </thead> <tbody> <tr> <td style="padding-right: 10px;"><b>Source</b></td><td style="padding-right: 10px;">Source_type</td><td style="padding-right: 10px;">required</td><td></td></tr> <tr> <td style="padding-right: 10px;"><b>SubElement</b></td><td style="padding-right: 10px;">xs:string</td><td style="padding-right: 10px;">required</td><td></td></tr> <tr> <td style="padding-right: 10px;"><b>TableName</b></td><td style="padding-right: 10px;">xs:IDREF</td><td style="padding-right: 10px;">required</td><td></td></tr> </tbody> </table>	QName	Type	Use		<b>Source</b>	Source_type	required		<b>SubElement</b>	xs:string	required		<b>TableName</b>	xs:IDREF	required	
QName	Type	Use															
<b>Source</b>	Source_type	required															
<b>SubElement</b>	xs:string	required															
<b>TableName</b>	xs:IDREF	required															
Source	<code>&lt;xss:element name="Column" type="ReportColumnType" /&gt;</code>																

### Element ReportColumnType / Fieldname

Namespace	No namespace
Diagram	<p>The diagram shows the <code>Fieldname</code> element as a complex type. It has a single attribute, <code>Type</code>, which is of type <code>xs:string</code>. A callout box indicates that <code>xs:string</code> is a built-in primitive type representing character strings in XML.</p>
Type	xs:NCName
Properties	content: simple
Source	<code>&lt;xss:element name="Fieldname" type="xs:NCName" /&gt;</code>

### Element ReportColumnType / Fieldtype

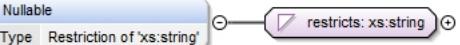
Namespace	No namespace																				
Diagram	<p>The diagram shows the <code>Fieldtype</code> element as a complex type. It has a single attribute, <code>Type</code>, which is of type <code>ColumnFieldtype</code>.</p>																				
Type	ColumnFieldtype																				
Properties	content: simple																				
Facets	<table border="0" style="width: 100%;"> <tr> <td style="vertical-align: top; width: 15%;">enumeration</td> <td>Varchar</td> </tr> <tr> <td>enumeration</td> <td>Char</td> </tr> <tr> <td>enumeration</td> <td>Date</td> </tr> <tr> <td>enumeration</td> <td>Time</td> </tr> <tr> <td>enumeration</td> <td>DateTime</td> </tr> <tr> <td>enumeration</td> <td>Numeric</td> </tr> <tr> <td>enumeration</td> <td>Boolean</td> </tr> <tr> <td>enumeration</td> <td>Int</td> </tr> <tr> <td>enumeration</td> <td>Decimal</td> </tr> <tr> <td>enumeration</td> <td>Mediumblob</td> </tr> </table>	enumeration	Varchar	enumeration	Char	enumeration	Date	enumeration	Time	enumeration	DateTime	enumeration	Numeric	enumeration	Boolean	enumeration	Int	enumeration	Decimal	enumeration	Mediumblob
enumeration	Varchar																				
enumeration	Char																				
enumeration	Date																				
enumeration	Time																				
enumeration	DateTime																				
enumeration	Numeric																				
enumeration	Boolean																				
enumeration	Int																				
enumeration	Decimal																				
enumeration	Mediumblob																				
Source	<code>&lt;xss:element name="Fieldtype" type="ColumnFieldtype" /&gt;</code>																				

### Element ReportColumnType / Fieldlength

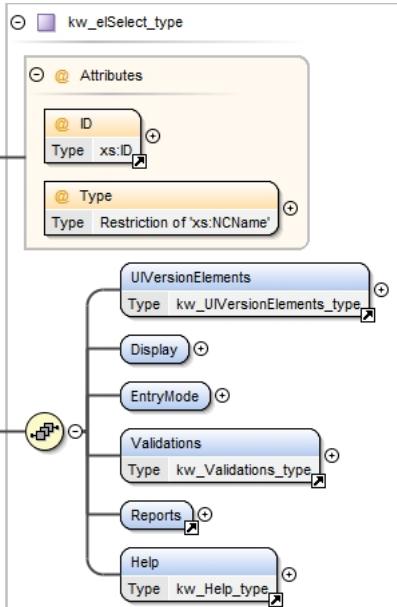
Namespace	No namespace
Diagram	<p>The diagram shows the <code>Fieldlength</code> element as a complex type. It has two attributes: <code>Type</code> (of type <code>xs:string</code>) and <code>Default</code> (with a value of <code>0</code>). A callout box indicates that <code>xs:string</code> is a built-in primitive type representing character strings in XML.</p>

Type	xs:string
Properties	content: simple default: 0
Source	<xs:element default="0" name="Fieldlength" type="xs:string"/>

### Element ReportColumnType / Nullable

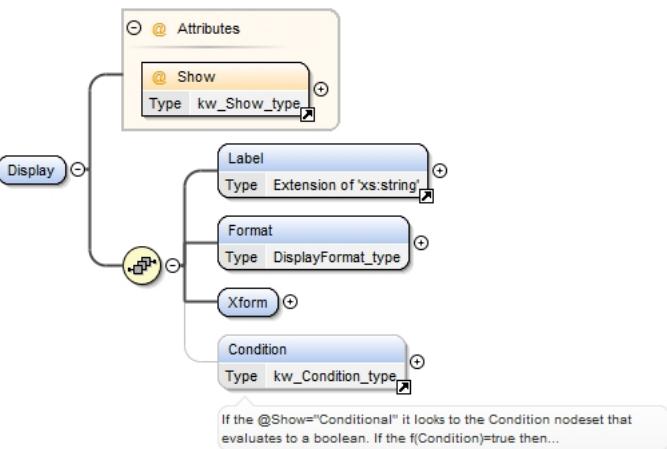
Namespace	No namespace
Diagram	
Type	restriction of xs:string
Properties	content: simple minOccurs: 0
Facets	enumeration Not Null enumeration Nullable
Source	<pre> &lt;xs:element minOccurs="0" name="Nullable"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:string"&gt;       &lt;xs:enumeration value="Not Null"/&gt;       &lt;xs:enumeration value="Nullable"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:element&gt; </pre>

### Element elSelect

Namespace	No namespace
Diagram	
Type	kw_elSelect_type
Properties	content: complex
Used by	Elements Elements, elSelectGroup/SelectGroupKey Element Group BaseElements
Model	UIVersionElements , Display , EntryMode , Validations , Reports , Help
Children	Display, EntryMode, Help, Reports, UIVersionElements, Validations
Instance	<pre> &lt;elSelect ID="" Type=""&gt;   &lt;UIVersionElements&gt;{1,1}&lt;/UIVersionElements&gt;   &lt;Display Show=""&gt;{1,1}&lt;/Display&gt; </pre>

	<pre>&lt;EntryMode Mode="" Source=""&gt;{1,1}&lt;/EntryMode&gt; &lt;Validations&gt;{1,1}&lt;/Validations&gt; &lt;Reports Include=""&gt;{1,1}&lt;/Reports&gt; &lt;Help Show=""&gt;{1,1}&lt;/Help&gt; &lt;/elSelect&gt;</pre>		
Attributes	<b>QName</b> <b>ID</b> <b>Type</b>	<b>Type</b> xs:ID restriction of xs:NCName	<b>Use</b> required required
Source	<code>&lt;xss:element name="elSelect" type="kw_elSelect_type" /&gt;</code>		

## Element kw\_elSelect\_type / Display

Namespace	No namespace		
Diagram			
Properties	content: complex		
Model	Label , Format , Xform , Condition{0,1}		
Children	Condition, Format, Label, Xform		
Instance	<pre>&lt;Display Show=""&gt;   &lt;Label Show=""&gt;{1,1}&lt;/Label&gt;   &lt;Format&gt;{1,1}&lt;/Format&gt;   &lt;Xform Appearance="" CSS=""&gt;{1,1}&lt;/Xform&gt;   &lt;Condition Operator=""&gt;{0,1}&lt;/Condition&gt; &lt;/Display&gt;</pre>		
Attributes	<b>QName</b> <b>Show</b>	<b>Type</b> kw_Show_type	<b>Use</b> required
Source	<code>&lt;xss:element name="Display"&gt;   &lt;xss:complexType&gt;     &lt;xss:sequence&gt;       &lt;xss:element ref="Label"/&gt;       &lt;xss:element name="Format" type="DisplayFormat_type" /&gt;       &lt;xss:element name="Xform"&gt;         &lt;xss:complexType&gt;           &lt;xss:attribute name="Appearance" use="required"&gt;             &lt;xss:simpleType&gt;               &lt;xss:restriction base="xs:Name"&gt;                 &lt;xss:enumeration value="SingleSelect"/&gt;                 &lt;xss:enumeration value="MultiSelect"/&gt;                 &lt;xss:enumeration value="RadioSelect"/&gt;                 &lt;xss:enumeration value="CheckBox"/&gt;                 &lt;xss:enumeration value="Output"/&gt;                 &lt;xss:enumeration value="Search"/&gt;                 &lt;xss:enumeration value="Button"/&gt;               &lt;/xss:restriction&gt;             &lt;/xss:simpleType&gt;           &lt;/xss:attribute&gt;           &lt;xss:attribute name="CSS"&gt;             &lt;xss:simpleType&gt;               &lt;xss:restriction base="xs:Name"&gt;                 &lt;xss:enumeration value="MediumSelect"/&gt;                 &lt;xss:enumeration value="LargeSelect"/&gt;                 &lt;xss:enumeration value="man-label"/&gt;                 &lt;xss:enumeration value="Kwantu-Standard-Label"/&gt;               &lt;/xss:restriction&gt;             &lt;/xss:simpleType&gt;           &lt;/xss:attribute&gt;         &lt;/xss:complexType&gt;       &lt;/xss:element&gt;     &lt;/xss:sequence&gt;   &lt;/xss:complexType&gt; &lt;/xss:element&gt;</code>		

```

        </xs:simpleType>
    </xs:attribute>
    </xs:complexType>
</xs:element>
<xs:element minOccurs="0" ref="Condition"/>
</xs:sequence>
<xs:attribute ref="Show" use="required"/>
</xs:complexType>
</xs:elements>

```

### Element kw\_elSelect\_type / Display / Format

Namespace	No namespace														
Diagram	<pre> classDiagram     class Format {         &lt;&lt;Type&gt;&gt;     }     class DisplayFormat_type {         &lt;&lt;DisplayFormat_type&gt;&gt;     }     Format "1" -- "1" DisplayFormat_type </pre>														
Type	DisplayFormat_type														
Properties	content: simple														
Facets	<table border="1"> <tr><td>enumeration</td><td>Float</td></tr> <tr><td>enumeration</td><td>Numeric</td></tr> <tr><td>enumeration</td><td>Date</td></tr> <tr><td>enumeration</td><td>String</td></tr> <tr><td>enumeration</td><td>Memo</td></tr> <tr><td>enumeration</td><td>Boolean</td></tr> <tr><td>enumeration</td><td>Integer</td></tr> </table>	enumeration	Float	enumeration	Numeric	enumeration	Date	enumeration	String	enumeration	Memo	enumeration	Boolean	enumeration	Integer
enumeration	Float														
enumeration	Numeric														
enumeration	Date														
enumeration	String														
enumeration	Memo														
enumeration	Boolean														
enumeration	Integer														
Source	<pre>&lt;xs:element name="Format" type="DisplayFormat_type"/&gt;</pre>														

### Element kw\_elSelect\_type / Display / Xform

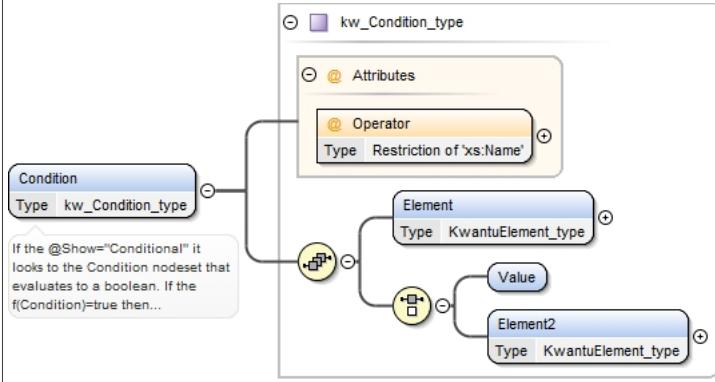
Namespace	No namespace									
Diagram	<pre> classDiagram     class Xform     class Attributes {         &lt;&lt;Attributes&gt;&gt;         class Appearance {             &lt;&lt;Appearance&gt;&gt;             Type Restriction of 'xs:Name'         }         class CSS {             &lt;&lt;CSS&gt;&gt;             Type Restriction of 'xs:Name'         }     }     Xform --&gt; Attributes </pre>									
Properties	content: complex									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>Appearance</td> <td>restriction of xs:Name</td> <td>required</td> </tr> <tr> <td>CSS</td> <td>restriction of xs:Name</td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Use	Appearance	restriction of xs:Name	required	CSS	restriction of xs:Name	optional
QName	Type	Use								
Appearance	restriction of xs:Name	required								
CSS	restriction of xs:Name	optional								
Source	<pre>&lt;xs:element name="Xform"&gt; &lt;xs:complexType&gt; &lt;xs:attribute name="Appearance" use="required"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:Name"&gt; &lt;xs:enumeration value="SingleSelect"/&gt; &lt;xs:enumeration value="MultiSelect"/&gt; &lt;xs:enumeration value="RadioSelect"/&gt; &lt;xs:enumeration value="CheckBox"/&gt; &lt;xs:enumeration value="Output"/&gt; &lt;xs:enumeration value="Search"/&gt; &lt;xs:enumeration value="Button"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; &lt;xs:attribute name="CSS"&gt; &lt;xs:simpleType&gt; &lt;xs:restriction base="xs:Name"&gt; &lt;xs:enumeration value="MediumSelect"/&gt; &lt;xs:enumeration value="LargeSelect"/&gt; &lt;xs:enumeration value="man-label"/&gt; &lt;xs:enumeration value="Kwantu-Standard-Label"/&gt; &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; </pre>									

```

</xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>

```

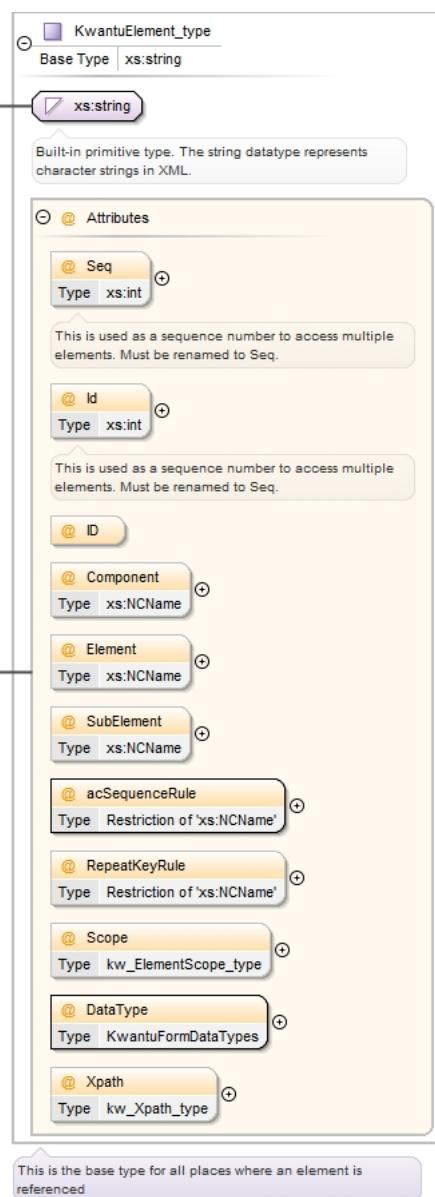
## Element Condition

Namespace	No namespace								
Annotations	<p>If the @Show="Conditional" it looks to the Condition nodeset that evaluates to a boolean. If the f(Condition)=true then field displays, otherwise not.</p> <p>Ie. to specify that the field must display only if the value of another field is "Other" then the following configuration would be applied:</p> <pre>Condition[@Operator="Equal"] Condition/Element: {here the reference to the field that could have the value of other} Condition/Value:"Other"</pre>								
Diagram									
Type	kw_Condition_type								
Properties	content: complex								
Used by	Elements elCalcField/Display, kw_elSelect_type/Display								
Model	Element , (Value   Element2)								
Children	Element, Element2, Value								
Instance	<pre>&lt;Condition Operator=""&gt;   &lt;Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE...   Element&gt;     &lt;Value&gt;{1,1}&lt;/Value&gt;     &lt;Element2 acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE...   Element2&gt; &lt;/Condition&gt;</pre>								
Attributes	<table border="1"> <tr> <td><b>QName</b></td><td><b>Type</b></td><td><b>Use</b></td><td></td></tr> <tr> <td><b>Operator</b></td><td>restriction of xs:Name</td><td>required</td><td></td></tr> </table>	<b>QName</b>	<b>Type</b>	<b>Use</b>		<b>Operator</b>	restriction of xs:Name	required	
<b>QName</b>	<b>Type</b>	<b>Use</b>							
<b>Operator</b>	restriction of xs:Name	required							
Source	<pre>&lt;xs:element name="Condition" type="kw_Condition_type"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;If the @Show="Conditional" it looks to the Condition nodeset that evaluates to a boolean. If the f(Condition)=true then field displays, otherwise not. Ie. to specify that the field must display only if the value of another field is "Other" then the following configuration would be applied: Condition[@Operator="Equal"] Condition/Element: {here the reference to the field that could have the value of other} Condition/Value:"Other"&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>								

## Element kw\_Condition\_type / Element

Namespace	No namespace
-----------	--------------

Diagram



Type	KwantuElement_type		
Properties	content: complex		
Attributes	QName	Type	Use
	<b>Component</b>	xs:NCName	optional
	<b>DataType</b>	KwantuFormDataTypes	required
	<b>Element</b>	xs:NCName	optional
	<b>ID</b>		optional
	<b>Id</b>	xs:int	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>RepeatKeyRule</b>	restriction of xs:NCName	optional
	<b>Scope</b>	kw_ElementScope_type	optional
	<b>Seq</b>	xs:int	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>SubElement</b>	xs:NCName	optional
	<b>Xpath</b>	kw_Xpath_type	optional

	QName	Type	Use	
	acSequenceRule	restriction of xs:NCName	required	
Source	<xs:element name="Element" type="KwantuElement_type" />			

### Element kw\_Condition\_type / Value

Namespace	No namespace
Diagram	(Value)
Properties	content: complex
Source	<xs:element name="Value"> <xs:complexType/> </xs:element>

### Element kw\_Condition\_type / Element2

Namespace	No namespace
Diagram	<p>The diagram illustrates the structure of the KwantuElement_type. It starts with a box labeled "Element2" (Type KwantuElement_type). A line connects this box to a larger box labeled "KwantuElement_type". Inside the "KwantuElement_type" box, there are several properties listed in a vertical stack:</p> <ul style="list-style-type: none"> <li>xs:string: A built-in primitive type representing character strings in XML.</li> <li>@ Seq: A sequence number to access multiple elements. Must be renamed to Seq.</li> <li>@ Id: A sequence number to access multiple elements. Must be renamed to Seq.</li> <li>@ ID: A unique identifier for an element.</li> <li>@ Component: A component identifier.</li> <li>@ Element: An element identifier.</li> <li>@ SubElement: A sub-element identifier.</li> <li>@ acSequenceRule: A sequence rule restriction.</li> <li>@ RepeatKeyRule: A repeat key rule restriction.</li> <li>@ Scope: A scope identifier.</li> <li>@ DataType: A data type identifier.</li> <li>@ Xpath: An XPath identifier.</li> </ul> <p>A note at the bottom of the "KwantuElement_type" box states: "This is the base type for all places where an element is referenced".</p>
Type	KwantuElement_type
Properties	content: complex

Attributes	QName	Type	Use
	<b>Component</b>	xs:NCName	optional
	<b>DataType</b>	KwantuFormDataTypes	required
	<b>Element</b>	xs:NCName	optional
	<b>ID</b>		optional
	<b>Id</b>	xs:int	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>RepeatKeyRule</b>	restriction of xs:NCName	optional
	<b>Scope</b>	kw_ElementScope_type	optional
	<b>Seq</b>	xs:int	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>SubElement</b>	xs:NCName	optional
	<b>Xpath</b>	kw_Xpath_type	optional
	<b>acSequenceRule</b>	restriction of xs:NCName	required
Source	<xs:element name="Element2" type="KwantuElement_type"/>		

### Element kw\_elSelect\_type / EntryMode

Namespace	No namespace									
Diagram										
Properties	content: complex									
Model	Select , AssignSubElements , Assigns{0,1} , Default{0,1} , ReadOnlyIf{0,1}									
Children	AssignSubElements, Assigns, Default, ReadOnlyIf, Select									
Instance	<pre>&lt;EntryMode Mode="" Source=""&gt;   &lt;Select&gt;{1,1}&lt;/Select&gt;   &lt;AssignSubElements&gt;{1,1}&lt;/AssignSubElements&gt;   &lt;Assigns&gt;{0,1}&lt;/Assigns&gt;   &lt;Default&gt;{0,1}&lt;/Default&gt;   &lt;ReadOnlyIf&gt;{0,1}&lt;/ReadOnlyIf&gt; &lt;/EntryMode&gt;</pre>									
Attributes	<table border="1"> <thead> <tr> <th>QName</th><th>Type</th><th>Use</th></tr> </thead> <tbody> <tr> <td><b>Mode</b></td><td>restriction of xs:NCName</td><td>required</td></tr> <tr> <td></td><td colspan="2">The mode defines how the data elements are entered into the field. Capture - User takes an action ReadOnly - User does not take action but the field can only be viewed Hidden - User cannot see the values</td></tr> </tbody> </table>	QName	Type	Use	<b>Mode</b>	restriction of xs:NCName	required		The mode defines how the data elements are entered into the field. Capture - User takes an action ReadOnly - User does not take action but the field can only be viewed Hidden - User cannot see the values	
QName	Type	Use								
<b>Mode</b>	restriction of xs:NCName	required								
	The mode defines how the data elements are entered into the field. Capture - User takes an action ReadOnly - User does not take action but the field can only be viewed Hidden - User cannot see the values									

QName	Type	Use	
	Calculated - Value is calculated RepeatKey - The value is picked up from the way that the repeat key is specified.		
Source	Source_type	optional	
Source	<pre> &lt;xs:element name="EntryMode"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="Select"&gt;         &lt;xs:complexType&gt;           &lt;xs:sequence&gt;             &lt;xs:choice minOccurs="0"&gt;               &lt;xs:element ref="Codelist"/&gt;               &lt;xs:element name="SharedData"&gt;                 &lt;xs:complexType&gt;                   &lt;xs:sequence&gt;                     &lt;xs:element name="Collection" type="xs:normalizedString"/&gt;                     &lt;xs:element name="TopElement" type="xs:normalizedString"/&gt;                     &lt;xs:element maxOccurs="unbounded" name="KeyValue" type="xs:normalizedString"&gt;                       &lt;xs:annotation&gt;                         &lt;xs:documentation&gt;Key Value is the Unique identifier for the block of data. This holds the path to the key value. If there are more than one then they are assigned multiple times&lt;/xs:documentation&gt;                       &lt;/xs:annotation&gt;                     &lt;/xs:element&gt;                   &lt;/xs:sequence&gt;                 &lt;/xs:complexType&gt;                 &lt;xs:element minOccurs="0" name="Filter"&gt;                   &lt;xs:complexType&gt;                     &lt;xs:sequence&gt;                       &lt;xs:element name="FilterElement" type="xs:normalizedString"&gt;                         &lt;xs:annotation&gt;                           &lt;xs:documentation&gt;This holds the path from the TopElement from which the data will be searched&lt;/xs:documentation&gt;                         &lt;/xs:annotation&gt;                       &lt;/xs:element&gt;                     &lt;/xs:sequence&gt;                   &lt;/xs:complexType&gt;                   &lt;xs:element name="FilterValue"&gt;                     &lt;xs:complexType&gt;                       &lt;xs:choice&gt;                         &lt;xs:element name="Element" type="KwantuElement_type"/&gt;                         &lt;xs:element name="Value" type="xs:Name"/&gt;                       &lt;/xs:choice&gt;                     &lt;/xs:complexType&gt;                     &lt;/xs:element&gt;                   &lt;/xs:sequence&gt;                 &lt;/xs:complexType&gt;                 &lt;/xs:element&gt;               &lt;/xs:sequence&gt;             &lt;/xs:choice&gt;           &lt;/xs:complexType&gt;         &lt;/xs:element&gt;       &lt;/xs:sequence&gt;     &lt;/xs:complexType&gt;   &lt;/xs:element&gt; &lt;xs:element name="Rest" type="RestType"/&gt; &lt;xs:element name="Taxonomy"&gt;   &lt;xs:complexType&gt;     &lt;xs:simpleContent&gt;       &lt;xs:extension base="xs:NCName"&gt;         &lt;xs:attribute name="Type"&gt;           &lt;xs:simpleType&gt;             &lt;xs:restriction base="xs:NCName"&gt;               &lt;xs:enumeration value="XSL"/&gt;             &lt;/xs:restriction&gt;           &lt;/xs:simpleType&gt;         &lt;/xs:attribute&gt;       &lt;/xs:extension&gt;     &lt;/xs:simpleContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:choice&gt; &lt;xs:element minOccurs="0" name="Refresh"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element maxOccurs="unbounded" name="ElementID" type="xs:IDREF"/&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;xs:element name="AssignSubElements" type="AssignSubElement_type"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This Assign only assigns to this element node&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>		

```

<xs:element name="Assigns" type="Assigns_type" minOccurs="0">
    <xs:annotation>
        <xs:documentation>This assigns value to elements that are not the current element</xs:documentation>
    </xs:annotation>
    </xs:element>
<xs:element minOccurs="0" name="Default" type="Default_type"/>
<xs:element minOccurs="0" ref="ReadOnlyIf"/>
</xs:sequence>
<xs:attribute name="Mode" use="required">
    <xs:annotation id="Select.EntryMode.Mode">
        <xs:documentation>The mode defines how the data elements are entered into the field. Capture
        - User takes an action ReadOnly - User does not take action but the field can only be viewed Hidden
        - User cannot see the values Calculated - Value is calculated RepeatKey - The value is picked up
        from the way that the repeat key is specified.</xs:documentation>
    </xs:annotation>
<xs:simpleType>
    <xs:restriction base="xs:NCName">
        <xs:enumeration value="Capture"/>
        <xs:enumeration value="ReadOnly"/>
        <xs:enumeration value="RepeatKey"/>
    </xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="Source" type="Source_type"/>
</xs:complexType>
</xs:element>

```

### Element kw\_elSelect\_type / EntryMode / Select

Namespace	No namespace
Diagram	<pre> graph LR     Select((Select)) --&gt; Codelist((Codelist))     Select --&gt; SharedData((SharedData))     Select --&gt; Rest((Rest))     Select --&gt; Refresh((Refresh))     Codelist --&gt; Taxonomy[("Taxonomy Type Extension of 'xs:NCName'")]     SharedData --&gt; Taxonomy     Rest --&gt; Taxonomy     </pre>
Properties	content: complex
Model	(Codelist   SharedData   Rest   Taxonomy) , Refresh{0,1}
Children	Codelist, Refresh, Rest, SharedData, Taxonomy
Instance	<pre> &lt;Select&gt;     &lt;Codelist SaveCode=""&gt;{1,1}&lt;/Codelist&gt;     &lt;SharedData&gt;{1,1}&lt;/SharedData&gt;     &lt;Rest ID=""&gt;{1,1}&lt;/Rest&gt;     &lt;Taxonomy Type=""&gt;{1,1}&lt;/Taxonomy&gt;     &lt;Refresh&gt;{0,1}&lt;/Refresh&gt; &lt;/Select&gt; </pre>
Source	<pre> &lt;xs:element name="Select"&gt;     &lt;xs:complexType&gt;         &lt;xs:sequence&gt;             &lt;xs:choice minOccurs="0"&gt;                 &lt;xs:element ref="Codelist"/&gt;                 &lt;xs:element name="SharedData"&gt;                     &lt;xs:complexType&gt;                         &lt;xs:sequence&gt;                             &lt;xs:element name="Collection" type="xs:normalizedString"/&gt;                             &lt;xs:element name="TopElement" type="xs:normalizedString"/&gt;                             &lt;xs:element maxOccurs="unbounded" name="KeyValue" type="xs:normalizedString"&gt;                                 &lt;xs:annotation&gt;                                     &lt;xs:documentation&gt;Key Value is the Unique identifier for the block of data. This                                     holds the path to the key value. If there are more than one then they are assigned multiple times&lt;/xs:documentation&gt;                                 &lt;/xs:annotation&gt;                             &lt;/xs:element&gt;                         &lt;/xs:sequence&gt;                     &lt;/xs:complexType&gt;                 &lt;/xs:element&gt;                 &lt;xs:element minOccurs="0" name="Filter"&gt;                     &lt;xs:complexType&gt;                         &lt;xs:sequence&gt;                             &lt;xs:element name="FilterElement" type="xs:normalizedString"&gt;                                 &lt;xs:annotation&gt;                                     &lt;xs:documentation&gt;This holds the path from the TopElement from which the                                     data will be searched&lt;/xs:documentation&gt;                                 &lt;/xs:annotation&gt;                             &lt;/xs:element&gt;                         &lt;/xs:sequence&gt;                     &lt;/xs:complexType&gt;                 &lt;/xs:element&gt;             &lt;/xs:choice&gt;         &lt;/xs:sequence&gt;     &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>

```

        </xs:annotation>
    </xs:element>
<xs:element name="FilterValue">
    <xs:complexType>
        <xs:choice>
            <xs:element name="Element" type="KwantuElement_type"/>
            <xs:element name="Value" type="xs:Name"/>
        </xs:choice>
        </xs:complexType>
    </xs:element>
    <xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Rest" type="RestType"/>
<xs:element name="Taxonomy">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension bases="xs:NCName">
                <xs:attribute name="Type">
                    <xs:simpleType>
                        <xs:restriction base="xs:NCName">
                            <xs:enumeration value="XSL"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:attribute>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
</xs:choice>
<xs:element minOccurs="0" name="Refresh">
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded" name="ElementID" type="xs:IDREF"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

```

## Element Codelist

Namespace	No namespace								
Diagram	<pre> classDiagram     class Codelist     class Attributes {         @ SaveCode         Type Restriction of 'xs:NCName'     }     class Taxonomy {         Type Extension of 'xs:NCName'     }     Codelist "1" -- "1" Attributes     Codelist "1" -- "1" Taxonomy </pre>								
Properties	content:	complex							
Used by	Element	kw_elSelect_type/EntryMode>Select							
Model	Taxonomy								
Children	Taxonomy								
Instance	<pre> &lt;Codelist SaveCode=""&gt;     &lt;Taxonomy Type=""&gt;{1,1}&lt;/Taxonomy&gt; &lt;/Codelist&gt; </pre>								
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>SaveCode</td> <td>restriction of xs:NCName</td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Use	SaveCode	restriction of xs:NCName	optional		
QName	Type	Use							
SaveCode	restriction of xs:NCName	optional							
Source	<pre> &lt;xs:element name="Codelist"&gt;     &lt;xs:complexType&gt;         &lt;xs:sequence&gt;             &lt;xs:element name="Taxonomy"&gt;                 &lt;xs:complexType&gt;                     &lt;xs:simpleContent&gt;                         &lt;xs:extension base="xs:NCName"&gt;                             &lt;xs:attribute name="Type"&gt;                                 &lt;xs:simpleType&gt; </pre>								

```

<xs:restriction base="xs:NCName">
  <xs:enumeration value="XSL" />
  <xs:enumeration value="Rest" />
  <xs:enumeration value="Default" />
</xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="SaveCode">
  <xs:simpleType>
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="Both" />
      <xs:enumeration value="Default" />
      <xs:enumeration value="Name" />
      <xs:enumeration value="Code" />
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>

```

## Element **Codelist** / **Taxonomy**

Namespace	No namespace						
Diagram	<pre> classDiagram     class xsNCName {         &lt;&lt;Built-in derived type. NCName represents XML "non-colonized" Names. The base type of NCName is Name.&gt;&gt;         &lt;&lt;Attributes&gt;&gt;         &lt;&lt;Type   Restriction of 'xs:NCName'&gt;&gt;     }     class Taxonomy {         &lt;&lt;Extension of 'xs:NCName'&gt;&gt;     }     Taxonomy &lt; -- xsNCName   </pre>						
Type	extension of xs:NCName						
Properties	content: complex						
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>Type</td> <td>restriction of xs:NCName</td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Use	Type	restriction of xs:NCName	optional
QName	Type	Use					
Type	restriction of xs:NCName	optional					
Source	<pre> &lt;xs:element name="Taxonomy"&gt;   &lt;xs:complexType&gt;     &lt;xs:simpleContent&gt;       &lt;xs:extension base="xs:NCName"&gt;         &lt;xs:attribute name="Type"&gt;           &lt;xs:simpleType&gt;             &lt;xs:restriction base="xs:NCName"&gt;               &lt;xs:enumeration value="XSL" /&gt;               &lt;xs:enumeration value="Rest" /&gt;               &lt;xs:enumeration value="Default" /&gt;             &lt;/xs:restriction&gt;           &lt;/xs:simpleType&gt;         &lt;/xs:attribute&gt;       &lt;/xs:extension&gt;     &lt;/xs:simpleContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>						

## Element **kw\_elSelect\_type** / **EntryMode** / **Select** / **SharedData**

Namespace	No namespace
-----------	--------------

Diagram	<pre> classDiagram     class SharedData     class Collection {         &lt;&lt;Collection&gt;&gt;         &lt;&lt;xs:normalizedString&gt;&gt;     }     class TopElement {         &lt;&lt;TopElement&gt;&gt;         &lt;&lt;xs:normalizedString&gt;&gt;     }     class KeyValue {         &lt;&lt;KeyValue&gt;&gt;         &lt;&lt;xs:normalizedString&gt;&gt;         &lt;&lt;1..oo&gt;&gt;     }     class Filter {         &lt;&lt;Filter&gt;&gt;     }      SharedData --&gt; Collection     SharedData --&gt; TopElement     SharedData --&gt; KeyValue     SharedData --&gt; Filter   </pre>
Properties	content: complex
Model	Collection , TopElement , KeyValue+ , Filter{0,1}
Children	Collection, Filter, KeyValue, TopElement
Instance	<pre> &lt;SharedData&gt;   &lt;Collection&gt;{1,1}&lt;/Collection&gt;   &lt;TopElement&gt;{1,1}&lt;/TopElement&gt;   &lt;KeyValue&gt;{1,unbounded}&lt;/KeyValue&gt;   &lt;Filter&gt;{0,1}&lt;/Filter&gt; &lt;/SharedData&gt;   </pre>
Source	<pre> &lt;xs:element name="SharedData"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="Collection" type="xs:normalizedString"/&gt;       &lt;xs:element name="TopElement" type="xs:normalizedString"/&gt;       &lt;xs:element maxOccurs="unbounded" name="KeyValue" type="xs:normalizedString"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;Key Value is the Unique identifier for the block of data. This holds the path to the key value. If there are more than one then they are assigned multiple times&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element minOccurs="0" name="Filter"&gt;         &lt;xs:complexType&gt;           &lt;xs:sequence&gt;             &lt;xs:element name="FilterElement" type="xs:normalizedString"&gt;               &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;This holds the path from the TopElement from which the data will be searched&lt;/xs:documentation&gt;               &lt;/xs:annotation&gt;             &lt;/xs:element&gt;             &lt;xs:element name="FilterValue"&gt;               &lt;xs:complexType&gt;                 &lt;xs:choice&gt;                   &lt;xs:element name="Element" type="KwantuElement_type"/&gt;                   &lt;xs:element name="Value" type="xs:Name"/&gt;                 &lt;/xs:choice&gt;               &lt;/xs:complexType&gt;             &lt;/xs:element&gt;           &lt;/xs:sequence&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;   </pre>

#### Element kw\_elSelect\_type / EntryMode / Select / SharedData / Collection

Namespace	No namespace
Diagram	<pre> classDiagram     class Collection {         &lt;&lt;Collection&gt;&gt;         &lt;&lt;xs:normalizedString&gt;&gt;     }   </pre>
Type	xs:normalizedString
Properties	content: simple
Source	<pre> &lt;xs:element name="Collection" type="xs:normalizedString"/&gt;   </pre>

#### Element kw\_elSelect\_type / EntryMode / Select / SharedData / TopElement

Namespace	No namespace
-----------	--------------

Diagram	
Type	xs:normalizedString
Properties	content: simple
Source	<xs:element name="TopElement" type="xs:normalizedString"/>

### **Element kw\_elSelect\_type / EntryMode / Select / SharedData / KeyValue**

Namespace	No namespace
Annotations	Key Value is the Unique identifier for the block of data. This holds the path to the key value. If there are more than one then they are assigned multiple times
Diagram	
Type	xs:normalizedString
Properties	content: simple maxOccurs: unbounded
Source	<pre>&lt;xs:element maxOccurs="unbounded" name="KeyValue" type="xs:normalizedString"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Key Value is the Unique identifier for the block of data. This holds the path to the key value. If there are more than one then they are assigned multiple times&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

### **Element kw\_elSelect\_type / EntryMode / Select / SharedData / Filter**

Namespace	No namespace
Diagram	
Properties	content: complex minOccurs: 0
Model	FilterElement , FilterValue
Children	FilterElement, FilterValue
Instance	<pre>&lt;Filter&gt;   &lt;FilterElement&gt;{1,1}&lt;/FilterElement&gt;   &lt;FilterValue&gt;{1,1}&lt;/FilterValue&gt; &lt;/Filter&gt;</pre>
Source	<pre>&lt;xs:element minOccurs="0" name="Filter"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="FilterElement" type="xs:normalizedString"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;This holds the path from the TopElement from which the data will be searched&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="FilterValue"&gt;         &lt;xs:complexType&gt;           &lt;xs:choice&gt;             &lt;xs:element name="Element" type="KwantuElement_type"/&gt;             &lt;xs:element name="Value" type="xs:Name"/&gt;           &lt;/xs:choice&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>

```

</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

```

### **Element kw\_elSelect\_type / EntryMode / Select / SharedData / Filter / FilterElement**

Namespace	No namespace
Annotations	This holds the path from the TopElement from which the data will be searched
Diagram	<p>This holds the path from the TopElement from which the data will be searched</p> <p>Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...</p>
Type	xs:normalizedString
Properties	content: simple
Source	<pre> &lt;xs:element name="FilterElement" type="xs:normalizedString"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This holds the path from the TopElement from which the data will be searched&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>

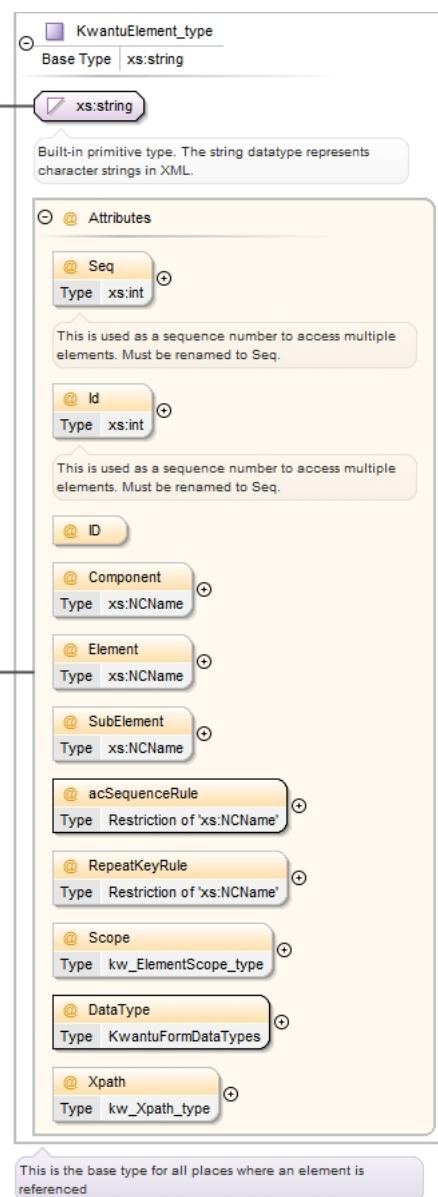
### **Element kw\_elSelect\_type / EntryMode / Select / SharedData / Filter / FilterValue**

Namespace	No namespace
Diagram	
Properties	content: complex
Model	Element   Value
Children	Element, Value
Instance	<pre> &lt;FilterValue&gt;   &lt;Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE...   &lt;Value&gt;{1,1}&lt;/Value&gt; &lt;/FilterValue&gt; </pre>
Source	<pre> &lt;xs:element name="FilterValue"&gt;   &lt;xs:complexType&gt;     &lt;xs:choice&gt;       &lt;xs:element name="Element" type="KwantuElement_type"/&gt;       &lt;xs:element name="Value" type="xs:Name"/&gt;     &lt;/xs:choice&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>

### **Element kw\_elSelect\_type / EntryMode / Select / SharedData / Filter / FilterValue / Element**

Namespace	No namespace
-----------	--------------

Diagram



Type	KwantuElement_type		
Properties	content: complex		
Attributes	<b>QName</b>	Type	Use
	<b>Component</b>	<code>xs:NCName</code>	optional
	<b>DataType</b>	<code>KwantuFormDataTypes</code>	required
	<b>Element</b>	<code>xs:NCName</code>	optional
	<b>ID</b>		optional
	<b>Id</b>	<code>xs:int</code>	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>RepeatKeyRule</b>	restriction of <code>xs:NCName</code>	optional
	<b>Scope</b>	<code>kw_ElementScope_type</code>	optional
	<b>Seq</b>	<code>xs:int</code>	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>SubElement</b>	<code>xs:NCName</code>	optional
	<b>Xpath</b>	<code>kw_Xpath_type</code>	optional

	QName	Type	Use	
	acSequenceRule	restriction of xs:NCName	required	
Source	<xs:element name="Element" type="KwantElement_type" />			

**Element kw\_elSelect\_type / EntryMode / Select / SharedData / Filter / FilterValue / Value**

Namespace	No namespace
Diagram	
Type	xs:Name
Properties	content: simple
Source	<xs:element name="Value" type="xs:Name" />

**Element kw\_elSelect\_type / EntryMode / Select / Rest**

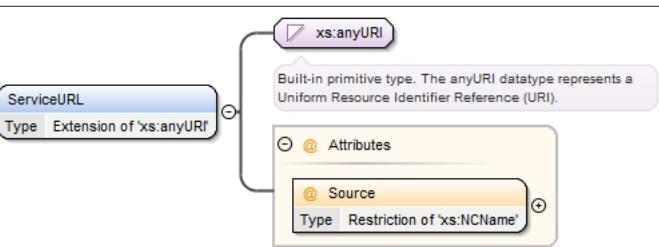
Namespace	No namespace						
Diagram							
Type	RestType						
Properties	content: complex						
Model	HostID , ServiceURL , Parameters , Return , IfError						
Children	HostID, IfError, Parameters, Return, ServiceURL						
Instance	<pre>&lt;Rest ID=""&gt; &lt;HostID&gt;{1,1}&lt;/HostID&gt; &lt;ServiceURL Source=""&gt;{1,1}&lt;/ServiceURL&gt; &lt;Parameters&gt;{1,1}&lt;/Parameters&gt; &lt;Return&gt;{1,1}&lt;/Return&gt; &lt;IfError&gt;{1,1}&lt;/IfError&gt; &lt;/Rest&gt;</pre>						
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>ID</td> <td></td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Use	ID		optional
QName	Type	Use					
ID		optional					
Source	<xs:element name="Rest" type="RestType" />						

**Element RestType / HostID**

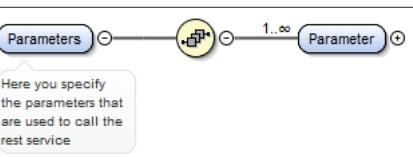
Namespace	No namespace
-----------	--------------

Diagram	
Type	xs:NCName
Properties	content: simple
Source	<code>&lt;xss:element name="HostID" type="xs:NCName" /&gt;</code>

### Element RestType / ServiceURL

Namespace	No namespace						
Diagram							
Type	extension of xs:anyURI						
Properties	content: complex						
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">QName</th> <th style="width: 30%;">Type</th> <th style="width: 30%;">Use</th> </tr> </thead> <tbody> <tr> <td>Source</td> <td>restriction of xs:NCName</td> <td>required</td> </tr> </tbody> </table>	QName	Type	Use	Source	restriction of xs:NCName	required
QName	Type	Use					
Source	restriction of xs:NCName	required					
Source	<pre> &lt;xss:element name="ServiceURL"&gt;   &lt;xss:complexType&gt;     &lt;xss:simpleContent&gt;       &lt;xss:extension base="xs:anyURI"&gt;         &lt;xss:attribute name="Source" use="required"&gt;           &lt;xss:simpleType&gt;             &lt;xss:restriction base="xs:NCName"&gt;               &lt;xss:enumeration value="ProcessVariable"/&gt;               &lt;xss:enumeration value="SubProcessVariable"/&gt;               &lt;xss:enumeration value="ElementID"/&gt;               &lt;xss:enumeration value="ProcessObject"/&gt;               &lt;xss:enumeration value="SubProcessObject"/&gt;               &lt;xss:enumeration value="ServiceURL"/&gt;             &lt;/xss:restriction&gt;           &lt;/xss:simpleType&gt;         &lt;/xss:attribute&gt;       &lt;/xss:extension&gt;     &lt;/xss:simpleContent&gt;   &lt;/xss:complexType&gt; &lt;/xss:element&gt; </pre>						

### Element RestType / Parameters

Namespace	No namespace
Annotations	Here you specify the parameters that are used to call the rest service
Diagram	
Properties	content: complex
Model	Parameter+
Children	Parameter
Instance	<code>&lt;Parameters&gt;   &lt;Parameter Format="" Name="" Seq=""&gt;{1,unbounded}&lt;/Parameter&gt; &lt;/Parameters&gt;</code>
Source	<code>&lt;xss:element name="Parameters"&gt;   &lt;xss:annotation&gt;</code>

```

<xs:documentation>Here you specify the parameters that are used to call the rest service</xs:documentation>
</xs:annotation>
<xs:complexType>
  <xs:sequence>
    <xs:element maxOccurs="unbounded" name="Parameter">
      <xs:complexType>
        <xs:choice>
          <xs:element name="Element" type="KwantuElement_type"/>
          <xs:element name="Constant" type="xs:string"/>
        </xs:choice>
        <xs:attribute name="Seq"/>
        <xs:attribute name="Name" type="xs:NCName"/>
        <xs:attribute name="Format"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>

```

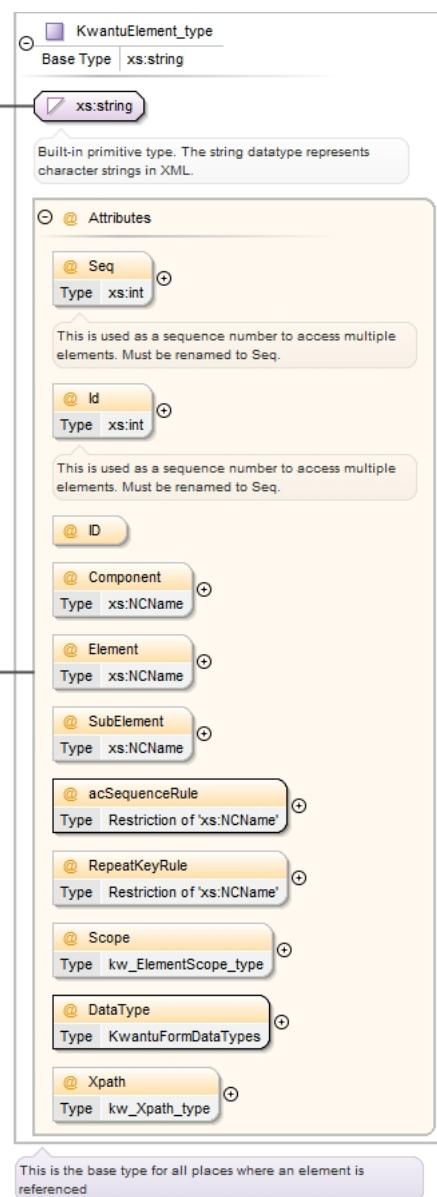
### Element RestType / Parameters / Parameter

Namespace	No namespace												
Diagram	<pre> classDiagram     class Parameter {         @ Seq         @ Name "Type xs:NCName"         @ Format         + Element "Type KwantuElement_type"         + Constant "Type xs:string"     } </pre>												
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	maxOccurs:	unbounded								
content:	complex												
maxOccurs:	unbounded												
Model	Element   Constant												
Children	Constant, Element												
Instance	<pre> &lt;Parameter Format="" Name="" Seq=""&gt;   &lt;Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" SubE   Element&gt;     &lt;Constant&gt;{1,1}&lt;/Constant&gt;   &lt;/Parameter&gt; </pre>												
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>Format</td> <td></td> <td>optional</td> </tr> <tr> <td>Name</td> <td>xs:NCName</td> <td>optional</td> </tr> <tr> <td>Seq</td> <td></td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Use	Format		optional	Name	xs:NCName	optional	Seq		optional
QName	Type	Use											
Format		optional											
Name	xs:NCName	optional											
Seq		optional											
Source	<pre> &lt;xs:element maxOccurs="unbounded" name="Parameter"&gt;   &lt;xs:complexType&gt;     &lt;xs:choice&gt;       &lt;xs:element name="Element" type="KwantuElement_type"/&gt;       &lt;xs:element name="Constant" type="xs:string"/&gt;     &lt;/xs:choice&gt;     &lt;xs:attribute name="Seq"/&gt;     &lt;xs:attribute name="Name" type="xs:NCName"/&gt;     &lt;xs:attribute name="Format"/&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>												

### Element RestType / Parameters / Parameter / Element

Namespace	No namespace
-----------	--------------

Diagram



Type	KwantuElement_type		
Properties	content: complex		
Attributes	QName	Type	Use
	<b>Component</b>	xs:NCName	optional
	<b>DataType</b>	KwantuFormDataTypes	required
	<b>Element</b>	xs:NCName	optional
	<b>ID</b>		optional
	<b>Id</b>	xs:int	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>RepeatKeyRule</b>	restriction of xs:NCName	optional
	<b>Scope</b>	kw_ElementScope_type	optional
	<b>Seq</b>	xs:int	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>SubElement</b>	xs:NCName	optional
	<b>Xpath</b>	kw_Xpath_type	optional

	QName	Type	Use	
	acSequenceRule	restriction of xs:NCName	required	
Source	<xs:element name="Element" type="KwantuElement_type" />			

### Element RestType / Parameters / Parameter / Constant

Namespace	No namespace
Diagram	
Type	xs:string
Properties	content: simple
Source	<xs:element name="Constant" type="xs:string" />

### Element RestType / Return

Namespace	No namespace
Annotations	Here you specify the return elements and what to do with them.
Diagram	
Properties	content: complex
Model	Result , Status+ , Assign*
Children	Assign, Result, Status
Instance	<pre> &lt;Return&gt;   &lt;Result&gt;{1,1}&lt;/Result&gt;   &lt;Status Type=""&gt;{1,unbounded}&lt;/Status&gt;   &lt;Assign AssignTo="" AssignToID="" Type=""&gt;{0,unbounded}&lt;/Assign&gt; &lt;/Return&gt; </pre>
Source	<pre> &lt;xs:element name="Return"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Here you specify the return elements and what to do with them.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="Result" type="xs:string"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;This is the xpath to the root of the result set that you are dealing with&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element maxOccurs="unbounded" name="Status"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;This elements contains any status aspects that you want to deal with&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;Here we define the xpath relative to the result that contains the error code, and it compares to the value specified to determine if it is an error or not&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>

```

</xs:annotation>
<xs:complexType>
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="Value"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="StatusMessage" type="xs:string"/>
</xs:sequence>
<xs:attribute name="Type">
  <xs:annotation>
    <xs:documentation>If the type="Error" then this a potential error status If the type="Success" then this a not an error</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="Error"/>
      <xs:enumeration value="Success"/>
    </xs:restriction>
  </xs:simpleType>
  </xs:attribute>
</xs:complexType>
</xs:element>
<xs:element maxOccurs="unbounded" name="Assign" minOccurs="0">
  <xs:annotation>
    <xs:documentation>If you need to make any assignments of data based on the rest service, then you can do it here.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:anyURI">
        <xs:attribute name="AssignTo" use="required">
          <xs:simpleType>
            <xs:restriction base="xs:NCName">
              <xs:enumeration value="Variable"/>
              <xs:enumeration value="Element"/>
              <xs:enumeration value="Object"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="AssignToID" use="required"/>
        <xs:attribute name="Type" use="required">
          <xs:simpleType>
            <xs:restriction base="xs:NCName">
              <xs:enumeration value="XPath"/>
              <xs:enumeration value="JSON"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

```

## Element RestType / Return / Result

Namespace	No namespace
Annotations	This is the xpath to the root of the result set that you are dealing with
Diagram	<pre> classDiagram     class Result {         Type xs:string     }     xs:string     Result "1" -- "0..1" xs:string     </pre>
Type	xs:string
Properties	content: simple
Source	<pre> &lt;xs:element name="Result" type="xs:string"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This is the xpath to the root of the result set that you are dealing with&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>

## Element RestType / Return / Status

Namespace	No namespace											
Annotations	This elements contains any status aspects that you want to deal with											
Diagram	<pre> classDiagram     class Status {         &lt;&lt;This elements contains any status aspects that you want to deal with&gt;&gt;         attribute Type         attribute Value     }     class Type {         &lt;&lt;Restriction of xs:NCName&gt;&gt;         &lt;&lt;If the type="Error" then this a potential error status If the type="Success" then this a not an error&gt;&gt;     }     class Value {         &lt;&lt;Extension of xs:string&gt;&gt;         &lt;&lt;Here we define the xpath relative to the result that contains the error code, and it compares to the value specified to...&gt;&gt;     } </pre>											
Properties	<p>content: complex</p> <p>maxOccurs: unbounded</p>											
Model	StatusCode , StatusMessage											
Children	StatusCode, StatusMessage											
Instance	<pre> &lt;Status Type=""&gt;     &lt;StatusCode Value=""&gt;{1,1}&lt;/StatusCode&gt;     &lt;StatusMessage&gt;{1,1}&lt;/StatusMessage&gt; &lt;/Status&gt; </pre>											
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>Type</td> <td>restriction of xs:NCName</td> <td>optional</td> </tr> <tr> <td></td> <td>If the type="Error" then this a potential error status If the type="Success" then this a not an error</td> <td></td> </tr> </tbody> </table>			QName	Type	Use	Type	restriction of xs:NCName	optional		If the type="Error" then this a potential error status If the type="Success" then this a not an error	
QName	Type	Use										
Type	restriction of xs:NCName	optional										
	If the type="Error" then this a potential error status If the type="Success" then this a not an error											
Source	<pre> &lt;xs:element maxOccurs="unbounded" name="Status"&gt;     &lt;xs:annotation&gt;         &lt;xs:documentation&gt;This elements contains any status aspects that you want to deal with&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;     &lt;xs:complexType&gt;         &lt;xs:sequence&gt;             &lt;xs:element name="StatusCode"&gt;                 &lt;xs:annotation&gt;                     &lt;xs:documentation&gt;Here we define the xpath relative to the result that contains the error code, and it compares to the value specified to determine if it is an error or not&lt;/xs:documentation&gt;                 &lt;/xs:annotation&gt;                 &lt;xs:complexType&gt;                     &lt;xs:simpleContent&gt;                         &lt;xs:extension base="xs:string"&gt;                             &lt;xs:attribute name="Value"/&gt;                         &lt;/xs:extension&gt;                     &lt;/xs:simpleContent&gt;                 &lt;/xs:complexType&gt;             &lt;/xs:element&gt;             &lt;xs:element name="StatusMessage" type="xs:string"/&gt;         &lt;/xs:sequence&gt;         &lt;xs:attribute name="Type"&gt;             &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;If the type="Error" then this a potential error status If the type="Success" then this a not an error&lt;/xs:documentation&gt;             &lt;/xs:annotation&gt;             &lt;xs:simpleType&gt;                 &lt;xs:restriction base="xs:NCName"&gt;                     &lt;xs:enumeration value="Error"/&gt;                     &lt;xs:enumeration value="Success"/&gt;                 &lt;/xs:restriction&gt;             &lt;/xs:simpleType&gt;         &lt;/xs:attribute&gt;     &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>											

### **Element RestType / Return / Status / StatusCode**

Namespace	No namespace								
Annotations	Here we define the xpath relative to the result that contains the error code, and it compares to the value specified to determine if it is an error or not								
Diagram	<p>Diagram illustrating the definition of the StatusCode element:</p> <ul style="list-style-type: none"> <li><b>StatusCode</b> (Type: Extension of 'xs:string')</li> <li><b>xs:string</b> (Built-in primitive type. The string datatype represents character strings in XML.)</li> <li><b>Attributes</b> (including <b>@ Value</b>)</li> </ul> <p>Annotations:</p> <ul style="list-style-type: none"> <li>Here we define the xpath relative to the result that contains the error code, and it compares to the value specified to...</li> </ul>								
Type	extension of xs:string								
Properties	content: complex								
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><b>Value</b></td> <td></td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Use	<b>Value</b>		optional		
QName	Type	Use							
<b>Value</b>		optional							
Source	<pre>&lt;xs:element name="StatusCode"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Here we define the xpath relative to the result that contains the error code, and it compares to the value specified to determine if it is an error or not&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:simpleContent&gt;       &lt;xs:extension base="xs:string"&gt;         &lt;xs:attribute name="Value" /&gt;       &lt;/xs:extension&gt;     &lt;/xs:simpleContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>								

### **Element RestType / Return / Status / StatusMessage**

Namespace	No namespace		
Diagram	<p>Diagram illustrating the definition of the StatusMessage element:</p> <ul style="list-style-type: none"> <li><b>StatusMessage</b> (Type: xs:string)</li> <li><b>xs:string</b> (Built-in primitive type. The string datatype represents character strings in XML.)</li> </ul>		
Type	xs:string		
Properties	content: simple		
Source	<pre>&lt;xs:element name="StatusMessage" type="xs:string"/&gt;</pre>		

### **Element RestType / Return / Assign**

Namespace	No namespace						
Annotations	If you need to make any assignments of data based on the rest service, then you can do it here.						
Diagram	<p>Diagram illustrating the definition of the Assign element:</p> <ul style="list-style-type: none"> <li><b>Assign</b> (Type: Extension of 'xs:anyURI')</li> <li><b>xs:anyURI</b> (Built-in primitive type. The anyURI datatype represents a Uniform Resource Identifier Reference (URI).)</li> <li><b>Attributes</b> (including <b>@ AssignTo</b> and <b>@ AssignToID</b>)</li> <li><b>AssignTo</b> (Type: Restriction of 'xs:NCName')</li> <li><b>AssignToID</b></li> <li><b>Type</b> (Type: Restriction of 'xs:NCName')</li> </ul> <p>Annotations:</p> <ul style="list-style-type: none"> <li>If you need to make any assignments of data based on the rest service, then you can do it here.</li> </ul>						
Type	extension of xs:anyURI						
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>			content:	complex	minOccurs:	0
content:	complex						
minOccurs:	0						

	maxOccurs:	unbounded	
Attributes	QName	Type	Use
	AssignTo	restriction of xs:NCName	required
	AssignToID		required
	Type	restriction of xs:NCName	required
Source	<pre>&lt;xs:element maxOccurs="unbounded" name="Assign" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;If you need to make any assignments of data based on the rest service, then you can do it here.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:simpleContent&gt;       &lt;xs:extension base="xs:anyURI"&gt;         &lt;xs:attribute name="AssignTo" use="required"&gt;           &lt;xs:simpleType&gt;             &lt;xs:restriction base="xs:NCName"&gt;               &lt;xs:enumeration value="Variable"/&gt;               &lt;xs:enumeration value="Element"/&gt;               &lt;xs:enumeration value="Object"/&gt;             &lt;/xs:restriction&gt;           &lt;/xs:simpleType&gt;         &lt;/xs:attribute&gt;         &lt;xs:attribute name="AssignToID" use="required"/&gt;         &lt;xs:attribute name="Type" use="required"&gt;           &lt;xs:simpleType&gt;             &lt;xs:restriction base="xs:NCName"&gt;               &lt;xs:enumeration value="XPath"/&gt;               &lt;xs:enumeration value="JSON"/&gt;             &lt;/xs:restriction&gt;           &lt;/xs:simpleType&gt;         &lt;/xs:attribute&gt;       &lt;/xs:extension&gt;     &lt;/xs:simpleContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>		

## Element RestType / IfError

Namespace	No namespace								
Annotations	Here you specify what to do if the rest service returned an error. Ignore - means that you continue Abort - means that you abort the operation Warning - issue a warning to the user and then continue Retry - retry a number of times specified, and then abort								
Diagram	<pre> classDiagram     class IfError {         &lt;&lt;Type Restriction of xs:NCName&gt;&gt;     }     IfError --o RestrictionOfXsNCName : restricts: xs:NCName   </pre> <p>Here you specify what to do if the rest service returned an error. Ignore - means that you continue Abort - means that...</p>								
Type	restriction of xs:NCName								
Properties	content: simple								
Facets	<table border="1"> <tr><td>enumeration</td><td>Ignore</td></tr> <tr><td>enumeration</td><td>Abort</td></tr> <tr><td>enumeration</td><td>Warning</td></tr> <tr><td>enumeration</td><td>Retry</td></tr> </table>	enumeration	Ignore	enumeration	Abort	enumeration	Warning	enumeration	Retry
enumeration	Ignore								
enumeration	Abort								
enumeration	Warning								
enumeration	Retry								
Source	<pre>&lt;xs:element name="IfError"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Here you specify what to do if the rest service returned an error. Ignore - means that you continue Abort - means that you abort the operation Warning - issue a warning to the user and then continue Retry - retry a number of times specified, and then abort&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="Ignore"/&gt;       &lt;xs:enumeration value="Abort"/&gt;       &lt;xs:enumeration value="Warning"/&gt;       &lt;xs:enumeration value="Retry"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:element&gt;</pre>								

| </xs:element> |

### Element kw\_elSelect\_type / EntryMode / Select / Taxonomy

Namespace	No namespace								
Diagram	<pre> classDiagram     class Taxonomy {         &lt;&lt;Extension of xs:NCName&gt;&gt;     }     class xsNCName {         &lt;&lt;Built-in derived type. NCName represents XML "non-colonized" Names. The base type of NCName is Name.&gt;&gt;     }     Taxonomy &lt; -- xsNCName     class Attributes {         &lt;&lt;Attributes&gt;&gt;     }     class Type {         &lt;&lt;Type&gt;&gt;         &lt;&lt;Restriction of xs:NCName&gt;&gt;     }     Type &lt; -- Attributes   </pre>								
Type	extension of xs:NCName								
Properties	content: complex								
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">QName</th> <th style="text-align: left; padding: 2px;">Type</th> <th style="text-align: left; padding: 2px;">Use</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">Type</td> <td style="padding: 2px;">restriction of xs:NCName</td> <td style="padding: 2px;">optional</td> </tr> </tbody> </table>			QName	Type	Use	Type	restriction of xs:NCName	optional
QName	Type	Use							
Type	restriction of xs:NCName	optional							
Source	<pre> &lt;xs:element name="Taxonomy"&gt;   &lt;xs:complexType&gt;     &lt;xs:simpleContent&gt;       &lt;xs:extension base="xs:NCName"&gt;         &lt;xs:attribute name="Type"&gt;           &lt;xs:simpleType&gt;             &lt;xs:restriction base="xs:NCName"&gt;               &lt;xs:enumeration value="XSL"/&gt;             &lt;/xs:restriction&gt;           &lt;/xs:simpleType&gt;         &lt;/xs:attribute&gt;       &lt;/xs:extension&gt;     &lt;/xs:simpleContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;   </pre>								

### Element kw\_elSelect\_type / EntryMode / Select / Refresh

Namespace	No namespace		
Diagram	<pre> sequenceDiagram     participant Refresh     participant ElementID     Refresh-&gt;&gt;ElementID:      activate ElementID     ...      deactivate ElementID   </pre>		
Properties	content: complex minOccurs: 0		
Model	ElementID+		
Children	ElementID		
Instance	<pre> &lt;Refresh&gt;   &lt;ElementID&gt;{1,unbounded}&lt;/ElementID&gt; &lt;/Refresh&gt;   </pre>		
Source	<pre> &lt;xs:element minOccurs="0" name="Refresh"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element maxOccurs="unbounded" name="ElementID" type="xs:IDREF" /&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;   </pre>		

### Element kw\_elSelect\_type / EntryMode / Select / Refresh / ElementID

Namespace	No namespace		
Diagram	<pre> sequenceDiagram     participant ElementID     participant xsIDREF     ElementID-&gt;&gt;xsIDREF:      activate xsIDREF     ...      deactivate xsIDREF   </pre>		
Type	xs:IDREF		
Properties	content: simple		

	maxOccurs:	unbounded
Source	<xs:element maxOccurs="unbounded" name="ElementID" type="xs:IDREF"/>	

### Element kw\_elSelect\_type / EntryMode / AssignSubElements

Namespace	No namespace
Annotations	This Assign only assigns to this element node
Diagram	<pre> graph LR     A[AssignSubElements&lt;br&gt;Type: AssignSubElement_type] --&gt; B(( ))     B --&gt; C[AssignCode&lt;br/&gt;Type: Extension of xs:normalizedString]     B --&gt; D[AssignName&lt;br/&gt;Type: Extension of xs:normalizedString]     B --&gt; E[AssignSubElement&lt;br/&gt;Type: Extension of xs:normalizedString]   </pre> <p>The diagram illustrates the <code>AssignSubElements</code> type. It consists of a central yellow square node with four outgoing arrows pointing to three rounded rectangular boxes: <code>AssignCode</code>, <code>AssignName</code>, and <code>AssignSubElement</code>. Each box contains its name and type information. Below each box is a callout bubble with the text: "Here you assign one or more values from the taxonomy to element IDs associated with the Select Element ID".</p>
Type	AssignSubElement_type
Properties	content: complex
Model	AssignCode , AssignName{0,1} , AssignSubElement*
Children	AssignCode, AssignName, AssignSubElement
Instance	<pre> &lt;AssignSubElements&gt;   &lt;AssignCode DataTypes="" SubElementID=""&gt;{1,1}&lt;/AssignCode&gt;   &lt;AssignName DataTypes="" SubElementID=""&gt;{0,1}&lt;/AssignName&gt;   &lt;AssignSubElement Action="" DataType="" SubElementID=""&gt;{0,unbounded}&lt;/AssignSubElement&gt; &lt;/AssignSubElements&gt;   </pre>
Source	<pre> &lt;xs:element name="AssignSubElements" type="AssignSubElement_type"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This Assign only assigns to this element node&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;   </pre>

### Element kw\_elSelect\_type / EntryMode / Assigns

Namespace	No namespace				
Annotations	This assigns value to elements that are not the current element				
Diagram	<pre> graph LR     A[Assigns&lt;br&gt;Type: Assigns_type] --&gt; B(( ))     B --&gt; C[Assign&lt;br/&gt;Type: Extension of xs:normalizedString]   </pre> <p>The diagram illustrates the <code>Assigns</code> type. It consists of a central yellow square node with two outgoing arrows pointing to two rounded rectangular boxes: <code>Assign</code>. The <code>Assign</code> box contains its name and type information. Below the <code>Assign</code> box is a callout bubble with the text: "Here you assign one or more values from the taxonomy to element IDs associated with the Select Element ID".</p>				
Type	Assigns_type				
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Model	Assign*				
Children	Assign				
Instance	<pre> &lt;Assigns&gt;   &lt;Assign Action="" ElementID="" RepeatKeyValue="" Scope="" SubElementID=""&gt;{0,unbounded}&lt;/Assign&gt; &lt;/Assigns&gt;   </pre>				
Source	<pre> &lt;xs:element name="Assigns" type="Assigns_type" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This assigns value to elements that are not the current element&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;   </pre>				

## Element Assigns\_type / Assign

Namespace	No namespace																					
Annotations	Here you assign one or more values from the taxonomy to element IDs associated with the Select Element ID																					
Diagram	<pre> graph LR     Assign[Assign Type Extension of 'xs:normalizedString'] --&gt; xs[xs:normalizedString]     subgraph Attributes         Action["@ Action Type Restriction of 'xs:NCName'"]         Scope["@ Scope Type Restriction of 'xs:NCName'"]         ElementID["@ ElementID Type xs:NCName"]         SubElementID["@ SubElementID Type xs:NCName"]         RepeatKeyValue["@ RepeatKeyValue"]     end     Note["Action='Default' means that the assignment only takes place if the element is currently null."]     </pre>																					
Type	extension of xs:normalizedString																					
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded															
content:	complex																					
minOccurs:	0																					
maxOccurs:	unbounded																					
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>Action</td> <td>restriction of xs:NCName</td> <td>optional</td> </tr> <tr> <td></td> <td>Action="Default" means that the assignment only takes place if the element is currently null.</td> <td></td> </tr> <tr> <td>ElementID</td> <td>xs:NCName</td> <td>required</td> </tr> <tr> <td>RepeatKeyValue</td> <td></td> <td>optional</td> </tr> <tr> <td>Scope</td> <td>restriction of xs:NCName</td> <td>required</td> </tr> <tr> <td>SubElementID</td> <td>xs:NCName</td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Use	Action	restriction of xs:NCName	optional		Action="Default" means that the assignment only takes place if the element is currently null.		ElementID	xs:NCName	required	RepeatKeyValue		optional	Scope	restriction of xs:NCName	required	SubElementID	xs:NCName	optional
QName	Type	Use																				
Action	restriction of xs:NCName	optional																				
	Action="Default" means that the assignment only takes place if the element is currently null.																					
ElementID	xs:NCName	required																				
RepeatKeyValue		optional																				
Scope	restriction of xs:NCName	required																				
SubElementID	xs:NCName	optional																				
Source	<pre> &lt;xs:element maxOccurs="unbounded" name="Assign" minOccurs="0"&gt;     &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Here you assign one or more values from the taxonomy to element IDs associated with the Select Element ID&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;     &lt;xs:complexType&gt;         &lt;xs:simpleContent&gt;             &lt;xs:extension base="xs:normalizedString"&gt;                 &lt;xs:attribute name="Action"&gt;                     &lt;xs:annotation&gt;                         &lt;xs:documentation&gt;Action="Default" means that the assignment only takes place if the element is currently null.&lt;/xs:documentation&gt;                     &lt;/xs:annotation&gt;                     &lt;xs:simpleType&gt;                         &lt;xs:restriction base="xs:NCName"&gt;                             &lt;xs:enumeration value="Default"/&gt;                         &lt;/xs:restriction&gt;                     &lt;/xs:simpleType&gt;                 &lt;/xs:attribute&gt;                 &lt;xs:attribute name="Scope" use="required"&gt;                     &lt;xs:simpleType&gt;                         &lt;xs:restriction base="xs:NCName"&gt;                             &lt;xs:enumeration value="SubElement"/&gt;                             &lt;xs:enumeration value="Element"/&gt;                             &lt;xs:enumeration value="Repeat"/&gt;                             &lt;xs:enumeration value="Variable"/&gt;                             &lt;xs:enumeration value="Registry"/&gt;                         &lt;/xs:restriction&gt;                     &lt;/xs:simpleType&gt;                 &lt;/xs:attribute&gt;             &lt;/xs:extension&gt;         &lt;/xs:simpleContent&gt;     &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>																					

```

<xs:attribute name="ElementID" type="xs:NCName" use="required"/>
<xs:attribute name="SubElementID" type="xs:NCName" />
<xs:attribute name="RepeatKeyValue" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>

```

### **Element kw\_elSelect\_type / EntryMode / Default**

Namespace	No namespace				
Diagram					
Type	Default_type				
Properties	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">content:</td><td style="padding: 2px;">complex</td></tr> <tr> <td style="padding: 2px;">minOccurs:</td><td style="padding: 2px;">0</td></tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Model	Value   (InitiatialValue , Calc)   ElementValue				
Children	Calc, ElementValue, InitiatialValue, Value				
Instance	<pre> &lt;Default&gt;   &lt;Value DataType=""&gt;{1,1}&lt;/Value&gt;   &lt;InitiatialValue DataType=""&gt;{1,1}&lt;/InitiatialValue&gt;   &lt;Calc Seq=""&gt;{1,1}&lt;/Calc&gt;   &lt;ElementValue acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" ElementValue&gt; &lt;/Default&gt; </pre>				
Source	<xs:element minOccurs="0" name="Default" type="Default_type"/>				

### **Element Repeat**

Namespace	No namespace
Annotations	Root element for the structure that defines the repeat or table structure

Diagram										
Type	kw_Repeat_type									
Properties	content: complex									
Used by	Element Group BaseElements Element Elements									
Model	RepeatDef , RepeatHeader , RepeatData , RepeatSummary{0,1}									
Children	RepeatData, RepeatDef, RepeatHeader, RepeatSummary									
Instance	<pre>&lt;Repeat ID="" Type=""&gt;   &lt;RepeatDef&gt;{1,1}&lt;/RepeatDef&gt;   &lt;RepeatHeader GroupName="" ID=""&gt;{1,1}&lt;/RepeatHeader&gt;   &lt;RepeatData GroupName="" ID=""&gt;{1,1}&lt;/RepeatData&gt;   &lt;RepeatSummary GroupName="" ID=""&gt;{0,1}&lt;/RepeatSummary&gt; &lt;/Repeat&gt;</pre>									
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>QName</th><th>Type</th><th>Use</th></tr> </thead> <tbody> <tr> <td><b>ID</b></td><td>xs:ID</td><td>required</td></tr> <tr> <td><b>Type</b></td><td>xs:NCName</td><td>required</td></tr> </tbody> </table>	QName	Type	Use	<b>ID</b>	xs:ID	required	<b>Type</b>	xs:NCName	required
QName	Type	Use								
<b>ID</b>	xs:ID	required								
<b>Type</b>	xs:NCName	required								
Source	<pre>&lt;xss:element name="Repeat" type="kw_Repeat_type"&gt;   &lt;xss:annotation id="Repeat"&gt;     &lt;xss:documentation&gt;Root element for the structure that defines the repeat or table structure&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt; &lt;/xss:element&gt;</pre>									

## Element RepeatDef

Namespace	No namespace
Annotations	The RepeatDef defines the overall aspects of the repeat table such as how rows are added and what are the key elements

Diagram	<pre> classDiagram     class RepeatDef {         &lt;&lt;The RepeatDef defines the overall aspects of the repeat table such as how rows are added and what are the key elements&gt;&gt;     }     class AddRows     class HeaderRow     class SummaryRow     class Columns     class KeyValue {         &lt;&lt;Type Extension of 'xs:NCName'&gt;&gt;     }      RepeatDef "1..1" -- "0..1" AddRows :      RepeatDef "1..1" -- "0..1" HeaderRow :      RepeatDef "1..1" -- "0..1" SummaryRow :      RepeatDef "1..1" -- "0..1" Columns :      RepeatDef "1..1" -- "0..1" KeyValue :    </pre>
Properties	content: complex
Used by	Complex Type kw_Repeat_type
Model	AddRows , HeaderRow , SummaryRow , Columns , KeyValue
Children	AddRows, Columns, HeaderRow, KeyValue, SummaryRow
Instance	<pre> &lt;RepeatDef&gt;   &lt;AddRows&gt;{1,1}&lt;/AddRows&gt;   &lt;HeaderRow LabelsFromData="" Show=""&gt;{1,1}&lt;/HeaderRow&gt;   &lt;SummaryRow Show=""&gt;{1,1}&lt;/SummaryRow&gt;   &lt;Columns NumberColumns=""&gt;{1,1}&lt;/Columns&gt;   &lt;KeyValue&gt;{1,1}&lt;/KeyValue&gt; &lt;/RepeatDef&gt;   </pre>
Source	<pre> &lt;xs:element name="RepeatDef"&gt;   &lt;xs:annotation id="Repeat.RepeatDef"&gt;     &lt;xs:documentation&gt;The RepeatDef defines the overall aspects of the repeat table such as how rows are added and what are the key elements&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="AddRows"&gt;         &lt;xs:annotation id="AddRows"&gt;           &lt;xs:documentation&gt;AddRows defines how the Repeat determines how many and which rows are added to the repeat table.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;         &lt;xs:complexType&gt;           &lt;xs:choice&gt;             &lt;xs:element name="UserAdd"&gt;               &lt;xs:annotation id="AddRows.UserAdd"&gt;                 &lt;xs:documentation&gt;If the UserAdd element is chosen, then the rows for the repeat table are added by the user selecting or adding them. The ways in which that can happen is defined in the child elements&lt;/xs:documentation&gt;               &lt;/xs:annotation&gt;               &lt;xs:complexType&gt;                 &lt;xs:sequence&gt;                   &lt;xs:element ref="Add"/&gt;                   &lt;xs:element ref="Delete"/&gt;                   &lt;xs:element name="MaxIterations" type="xs:short"/&gt;                   &lt;xs:element name="MinIterations" type="xs:short"/&gt;                   &lt;xs:element name="AddLinesMode"&gt;                     &lt;xs:simpleType&gt;                       &lt;xs:restriction base="xs:Name"&gt;                         &lt;xs:enumeration value="BlankLines"/&gt;                         &lt;xs:enumeration value="UniqueLookup"/&gt;                         &lt;xs:enumeration value="SingleLookup"/&gt;                         &lt;xs:enumeration value="MultipleLookup"/&gt;                         &lt;xs:enumeration value="MultipleUniqueLookup"/&gt;                       &lt;/xs:restriction&gt;                     &lt;/xs:simpleType&gt;                   &lt;/xs:element&gt;                 &lt;/xs:sequence&gt;               &lt;/xs:complexType&gt;             &lt;/xs:element&gt;             &lt;xs:element name="AutoLoad"&gt;               &lt;xs:annotation id="AutoLoad"&gt;                 &lt;xs:documentation&gt;If the Autoload element is chosen, then the rows for the repeat table are added programmatically, and not by user choice. The ways in which that can happen is defined&lt;/xs:documentation&gt;               &lt;/xs:annotation&gt;             &lt;/xs:element&gt;           &lt;/xs:choice&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; </pre>

```

</xs:annotation>
<xs:complexType>
  <xs:choice>
    <xs:element name="BlankLines" type="xs:short">
      <xs:annotation id="Addrows.BlankLines">
        <xs:documentation>Provide a number of blank rows as specified by the maxrows attribute</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Codelist">
      <xs:annotation id="AddRows.Codelist">
        <xs:documentation>Use the codelist specified by the KeyValue element to generate the rows.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Taxonomy" type="xs:NCName"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Periods">
    <xs:annotation id="AddRows.Periods">
      <xs:documentation>Adds rows according the number of periods between the start and end date, one for each period (or part of period for the first and last ones)</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="StartDate">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Element" type="KwantuElement_type"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="EndDate">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Element" type="KwantuElement_type"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="PeriodType" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:Name">
          <xs:enumeration value="Month"/>
          <xs:enumeration value="Quarter"/>
          <xs:enumeration value="YearCalendar"/>
          <xs:enumeration value="YearAprilToMay"/>
          <xs:enumeration value="YearJulyToJune"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
  </xs:element>
  <xs:element name="Component">
    <xs:annotation id="AddRows.Component">
      <xs:documentation>Adds blank rows depending on the values in a repeat structure in another component. The value of the component specifies the ID of the repeat structure</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence minOccurs="0">
        <xs:element maxOccurs="unbounded" name="Filter">
          <xs:annotation>
            <xs:documentation>The filter value(s) will compare an element (and subelement if specified) from the repeatID in the source component and filter them to the value of the Value or the Source element specified</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:complexType>
          <xs:choice>
            <xs:element name="Value"/>
            <xs:element name="ElementID" type="KwantuElement_type"/>
            <xs:element name="ElementCalc" type="fnElementCalc"/>
            <xs:element name="ElementsList" type="fnElementsList"/>
          </xs:choice>
          <xs:attribute name="ElementID"/>
          <xs:attribute name="SubElement"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="ComponentName" type="xs:NCName" use="required">
  
```

```

<xs:annotation id="AddRows.Component.ComponentName">
    <xs:documentation>Name of the component from which the rows will be
retrieved.</xs:documentation>
    </xs:annotation>
    </xs:attribute>
    <xs:attribute name="RepeatID" type="xs:NCName" use="required">
        <xs:annotation id="AddRows.Component.RepeatID">
            <xs:documentation>Name of the RepeatID from which the rows will be
retrieved.</xs:documentation>
        </xs:annotation>
        </xs:attribute>
        </xs:complexType>
        </xs:element>
        </xs:choice>
        </xs:complexType>
        </xs:element>
    </xs:choice>
    </xs:complexType>
</xs:element>
<xs:element name="HeaderRow">
    <xs:annotation id="RepeatDef.HeaderRow">
        <xs:documentation>This element specifies how to generate the labels, and whether to show
the header row or not.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:attribute ref="Show"/>
        <xs:attribute name="LabelsFromData">
            <xs:annotation id="HeaderRow.LabelsFromData">
                <xs:documentation>If Yes, then use the Label components from the first elements in the
columns for the data section to determine the labels</xs:documentation>
            </xs:annotation>
            <xs:simpleType>
                <xs:restriction base="xs:NCName">
                    <xs:enumeration value="Yes"/>
                    <xs:enumeration value="No"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        </xs:complexType>
    </xs:element>
<xs:element ref="SummaryRow"/>
<xs:element name="Columns">
    <xs:complexType>
        <xs:sequence maxOccurs="unbounded">
            <xs:element name="Column">
                <xs:complexType>
                    <xs:attribute ref="Seq">
                        <xs:annotation id="Repeat.RepeatDef.Columns.Column.Seq">
                            <xs:documentation>The number of the column</xs:documentation>
                        </xs:annotation>
                    </xs:attribute>
                    <xs:attribute name="Width">
                        <xs:annotation id="Repeat.RepeatDef.Columns.Column.Width">
                            <xs:documentation>Defines the width of the column</xs:documentation>
                        </xs:annotation>
                    </xs:attribute>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="NumberColumns" type="xs:integer">
            <xs:annotation id="Repeat.RepeatDef.Columns.NumberColumns">
                <xs:documentation>Defines the number of columns that should be generated for the
repeat structure. These are Visible columns. Each column has a header row, a data area and a
summary area. In each of these areas there may be one or more elements. The Number of columns in
the header, data and summary bands need not be equal, but a means must be devised to specify the
span.</xs:documentation>
            </xs:annotation>
        </xs:attribute>
        </xs:complexType>
    </xs:element>
    <xs:element ref="KeyValue"/>
</xs:sequence>
</xs:complexType>
</xs:element>

```

## Element RepeatDef / AddRows

Namespace	No namespace
Annotations	AddRows defines how the Repeat determines how many and which rows are added to the repeat table.

Diagram	<pre> graph LR     AddRows((AddRows)) --&gt; UserAddOrAutoLoad{ }     UserAddOrAutoLoad --&gt; UserAdd((UserAdd))     UserAddOrAutoLoad --&gt; AutoLoad((AutoLoad))     UserAdd --- docUserAdd[If the UserAdd element is chosen, then the rows for the repeat table are added by the user selecting or adding them....]     AutoLoad --- docAutoLoad[If the Autoload element is chosen, then the rows for the repeat table are added programmatically, and not by user...]   </pre>
Properties	content: complex
Model	UserAdd   AutoLoad
Children	AutoLoad, UserAdd
Instance	<pre> &lt;AddRows&gt;   &lt;UserAdd&gt;{1,1}&lt;/UserAdd&gt;   &lt;AutoLoad&gt;{1,1}&lt;/AutoLoad&gt; &lt;/AddRows&gt;   </pre>
Source	<pre> &lt;xs:element name="AddRows"&gt;   &lt;xs:annotation id="AddRows"&gt;     &lt;xs:documentation&gt;AddRows defines how the Repeat determines how many and which rows are added to the repeat table.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:choice&gt;       &lt;xs:element name="UserAdd"&gt;         &lt;xs:annotation id="AddRows.UserAdd"&gt;           &lt;xs:documentation&gt;If the UserAdd element is chosen, then the rows for the repeat table are added by the user selecting or adding them. The ways in which that can happen is defined in the child elements&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;         &lt;xs:complexType&gt;           &lt;xs:sequence&gt;             &lt;xs:element ref="Add"/&gt;             &lt;xs:element ref="Delete"/&gt;             &lt;xs:element name="MaxIterations" type="xs:short"/&gt;             &lt;xs:element name="MinIterations" type="xs:short"/&gt;             &lt;xs:element name="AddLinesMode"&gt;               &lt;xs:simpleType&gt;                 &lt;xs:restriction base="xs:Name"&gt;                   &lt;xs:enumeration value="BlankLines"/&gt;                   &lt;xs:enumeration value="UniqueLookup"/&gt;                   &lt;xs:enumeration value="SingleLookup"/&gt;                   &lt;xs:enumeration value="MultipleLookup"/&gt;                   &lt;xs:enumeration value="MultipleUniqueLookup"/&gt;                 &lt;/xs:restriction&gt;               &lt;/xs:simpleType&gt;             &lt;/xs:element&gt;           &lt;/xs:sequence&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;       &lt;xs:element name="AutoLoad"&gt;         &lt;xs:annotation id="AutoLoad"&gt;           &lt;xs:documentation&gt;If the Autoload element is chosen, then the rows for the repeat table are added programmatically, and not by user choice. The ways in which that can happen is defined&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;         &lt;xs:complexType&gt;           &lt;xs:choice&gt;             &lt;xs:element name="BlankLines" type="xs:short"&gt;               &lt;xs:annotation id="AddRows.BlankLines"&gt;                 &lt;xs:documentation&gt;Provide a number of blank rows as specified by the maxrows attribute&lt;/xs:documentation&gt;               &lt;/xs:annotation&gt;             &lt;/xs:element&gt;             &lt;xs:element name="Codelist"&gt;               &lt;xs:annotation id="AddRows.Codelist"&gt;                 &lt;xs:documentation&gt;Use the codelist specified by the KeyValue element to generate the rows.&lt;/xs:documentation&gt;               &lt;/xs:annotation&gt;             &lt;/xs:element&gt;           &lt;/xs:choice&gt;           &lt;xs:element name="Periods"&gt;             &lt;xs:annotation id="AddRows.Periods"&gt;               &lt;xs:documentation&gt;&lt;/xs:documentation&gt;             &lt;/xs:annotation&gt;           &lt;/xs:element&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:choice&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;   </pre>

```

<xs:documentation>Adds rows according the number of periods between the start and
end date, one for each period (or part of period for the first and last ones)</xs:documentation>
</xs:annotation>
<xs:complexType>
  <xs:sequence>
    <xs:element name="StartDate">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Element" type="KwantuElement_type"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="EndDate">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Element" type="KwantuElement_type"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="PeriodType" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:Name">
        <xs:enumeration value="Month"/>
        <xs:enumeration value="Quarter"/>
        <xs:enumeration value="YearCalendar"/>
        <xs:enumeration value="YearAprilToMay"/>
        <xs:enumeration value="YearJulyToJune"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
</xs:element>
<xs:element name="Component">
  <xs:annotation id="AddRows.Component">
    <xs:documentation>Adds blank rows depending on the values in a repeat structure
in another component. The value of the component specifies the ID of the repeat structure</
xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence minOccurs="0">
        <xs:element maxOccurs="unbounded" name="Filter">
          <xs:annotation>
            <xs:documentation>The filter value(s) will complare an element (and subelement
if specified) from the repeatID in the source component and filter them to the value of the Value
or the Source element specified</xs:documentation>
          </xs:annotation>
          <xs:complexType>
            <xs:choice>
              <xs:element name="Value"/>
              <xs:element name="ElementID" type="KwantuElement_type"/>
              <xs:element name="ElementCalc" type="fnElementCalc"/>
              <xs:element name="ElementsList" type="fnElementsList"/>
            </xs:choice>
            <xs:attribute name="ElementID"/>
            <xs:attribute name="SubElement"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="ComponentName" type="xs:NCName" use="required">
        <xs:annotation id="AddRows.Component.ComponentName">
          <xs:documentation>Name of the component from which the rows will be retrieved.</
xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="RepeatID" type="xs:NCName" use="required">
        <xs:annotation id="AddRows.Component.RepeatID">
          <xs:documentation>Name of the RepeatID from which the rows will be retrieved.</
xs:documentation>
        </xs:annotation>
      </xs:attribute>
    </xs:complexType>
  </xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>

```

### Element RepeatDef / AddRows / UserAdd

Namespace	No namespace
Annotations	If the UserAdd element is chosen, then the rows for the repeat table are added by the user selecting or adding them. The ways in which that can happen is defined in the child elements
Diagram	<pre> classDiagram     class UserAdd {         &lt;&lt;If the UserAdd element is chosen, then the rows for the repeat table are added by the user selecting or adding them....&gt;&gt;         Add         Delete         MaxIterations         MinIterations         AddLinesMode     }     Add &lt; -- UserAdd     Delete &lt; -- UserAdd     MaxIterations &lt; -- UserAdd     MinIterations &lt; -- UserAdd     AddLinesMode &lt; -- UserAdd   </pre>
Properties	content: complex
Model	Add , Delete , MaxIterations , MinIterations , AddLinesMode
Children	Add, AddLinesMode, Delete, MaxIterations, MinIterations
Instance	<pre> &lt;UserAdd&gt;   &lt;Add&gt;{1,1}&lt;/Add&gt;   &lt;Delete&gt;{1,1}&lt;/Delete&gt;   &lt;MaxIterations&gt;{1,1}&lt;/MaxIterations&gt;   &lt;MinIterations&gt;{1,1}&lt;/MinIterations&gt;   &lt;AddLinesMode&gt;{1,1}&lt;/AddLinesMode&gt; &lt;/UserAdd&gt;   </pre>
Source	<pre> &lt;xs:element name="UserAdd"&gt;   &lt;xs:annotation id="AddRows.UserAdd"&gt;     &lt;xs:documentation&gt;If the UserAdd element is chosen, then the rows for the repeat table are added by the user selecting or adding them. The ways in which that can happen is defined in the child elements&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element ref="Add"/&gt;       &lt;xs:element ref="Delete"/&gt;       &lt;xs:element name="MaxIterations" type="xs:short"/&gt;       &lt;xs:element name="MinIterations" type="xs:short"/&gt;       &lt;xs:element name="AddLinesMode"&gt;         &lt;xs:simpleType&gt;           &lt;xs:restriction base="xs:Name"&gt;             &lt;xs:enumeration value="BlankLines"/&gt;             &lt;xs:enumeration value="UniqueLookup"/&gt;             &lt;xs:enumeration value="SingleLookup"/&gt;             &lt;xs:enumeration value="MultipleLookup"/&gt;             &lt;xs:enumeration value="MultipleUniqueLookup"/&gt;           &lt;/xs:restriction&gt;         &lt;/xs:simpleType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;   </pre>

### Element Add

Namespace	No namespace
Diagram	<pre> classDiagram     class Add {         &lt;&lt;Serves as the root of the type definition hierarchy for any schema. It has the unique characteristic that it can...&gt;&gt;         xs:anyType         Attributes         #any     }     xs:anyType &lt; -- Add     Attributes &lt; -- Add     #any &lt; -- Add   </pre>

Used by	Element	RepeatDef/AddRows/UserAdd
Source		<xs:element name="Add" type="xs:anyType" />

### Element Delete

Namespace	No namespace
Diagram	<p>Serves as the root of the type definition hierarchy for any schema. It has the unique characteristic that it can...</p>
Used by	Element RepeatDef/AddRows/UserAdd
Source	<xs:element name="Delete" type="xs:anyType" />

### Element RepeatDef / AddRows / UserAdd / MaxIterations

Namespace	No namespace
Diagram	<p>Built-in derived type. The short datatype is derived from int by setting the value of maxInclusive to be 32767 and...</p>
Type	xs:short
Properties	content: simple

Source <xs:element name="MaxIterations" type="xs:short" />

### Element RepeatDef / AddRows / UserAdd / MinIterations

Namespace	No namespace
Diagram	<p>Built-in derived type. The short datatype is derived from int by setting the value of maxInclusive to be 32767 and...</p>
Type	xs:short
Properties	content: simple

Source <xs:element name="MinIterations" type="xs:short" />

### Element RepeatDef / AddRows / UserAdd / AddLinesMode

Namespace	No namespace						
Diagram							
Type	restriction of xs:Name						
Properties	content: simple						
Facets	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">BlankLines</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">UniqueLookup</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">SingleLookup</td> </tr> </table>	enumeration	BlankLines	enumeration	UniqueLookup	enumeration	SingleLookup
enumeration	BlankLines						
enumeration	UniqueLookup						
enumeration	SingleLookup						

	enumeration	MultipleLookup
	enumeration	MultipleUniqueLookup
Source		<pre>&lt;xs:element name="AddLinesMode"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:Name"&gt;       &lt;xs:enumeration values="BlankLines"/&gt;       &lt;xs:enumeration value="UniqueLookup"/&gt;       &lt;xs:enumeration value="SingleLookup"/&gt;       &lt;xs:enumeration value="MultipleLookup"/&gt;       &lt;xs:enumeration value="MultipleUniqueLookup"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:element&gt;</pre>

### Element RepeatDef / AddRows / AutoLoad

Namespace	No namespace
Annotations	If the Autoload element is chosen, then the rows for the repeat table are added programmatically, and not by user choice. The ways in which that can happen is defined
Diagram	<p>The diagram illustrates the relationships between the <b>AutoLoad</b>, <b>BlankLines</b>, <b>Codelist</b>, <b>Periods</b>, and <b>Component</b> elements. The <b>AutoLoad</b> element is connected to four other elements: <b>BlankLines</b>, <b>Codelist</b>, <b>Periods</b>, and <b>Component</b>. A callout box next to the <b>AutoLoad</b> element states: "If the Autoload element is chosen, then the rows for the repeat table are added programmatically, and not by user...". Another callout box next to the <b>AutoLoad</b> element indicates that it is a choice element. The <b>BlankLines</b> element is associated with a note: "Provide a number of blank rows as specified by the maxrows attribute". The <b>Codelist</b> element is associated with a note: "Use the codelist specified by the KeyValue element to generate the rows.". The <b>Periods</b> element is associated with a note: "Adds rows according the number of periods between the start and end date, one for each period (or part of period for...).". The <b>Component</b> element is associated with a note: "Adds blank rows depending on the values in a repeat structure in another component. The value of the component...".</p>
Properties	content: complex
Model	BlankLines   Codelist   Periods   Component
Children	BlankLines, Codelist, Component, Periods
Instance	<pre>&lt;AutoLoad&gt;   &lt;BlankLines&gt;{1,1}&lt;/BlankLines&gt;   &lt;Codelist&gt;{1,1}&lt;/Codelist&gt;   &lt;Periods PeriodType=""&gt;{1,1}&lt;/Periods&gt;   &lt;Component ComponentName="" RepeatID=""&gt;{1,1}&lt;/Component&gt; &lt;/AutoLoad&gt;</pre>
Source	<pre>&lt;xs:element name="AutoLoad"&gt;   &lt;xs:annotation id="AutoLoad"&gt;     &lt;xs:documentation&gt;If the Autoload element is chosen, then the rows for the repeat table are added programmatically, and not by user choice. The ways in which that can happen is defined&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:choice&gt;       &lt;xs:element name="BlankLines" type="xs:short"&gt;         &lt;xs:annotation id="Addrows.BlankLines"&gt;           &lt;xs:documentation&gt;Provide a number of blank rows as specified by the maxrows attribute&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="Codelist"&gt;         &lt;xs:annotation id="AddRows.Codelist"&gt;           &lt;xs:documentation&gt;Use the codelist specified by the KeyValue element to generate the rows.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:sequence&gt;         &lt;xs:element name="Taxonomy" type="xs:NCName"/&gt;       &lt;/xs:sequence&gt;     &lt;/xs:choice&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>

```

<xs:element name="Periods">
    <xs:annotation id="AddRows.Periods">
        <xs:documentation>Adds rows according the number of periods between the start and end date, one for each period (or part of period for the first and last ones)</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="StartDate">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Element" type="KwantuElement_type"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="EndDate">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Element" type="KwantuElement_type"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="PeriodType" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:Name">
                    <xs:enumeration value="Month"/>
                    <xs:enumeration value="Quarter"/>
                    <xs:enumeration value="YearCalendar"/>
                    <xs:enumeration value="YearAprilToMay"/>
                    <xs:enumeration value="YearJulyToJune"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>
</xs:element>
<xs:element name="Component">
    <xs:annotation id="AddRows.Component">
        <xs:documentation>Adds blank rows depending on the values in a repeat structure in another component. The value of the component specifies the ID of the repeat structure</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence minOccurs="0">
            <xs:element maxOccurs="unbounded" name="Filter">
                <xs:annotation>
                    <xs:documentation>The filter value(s) will compare an element (and subelement if specified) from the repeatID in the source component and filter them to the value of the Value or the Source element specified</xs:documentation>
                </xs:annotation>
                <xs:complexType>
                    <xs:choice>
                        <xs:element name="Value"/>
                        <xs:element name="ElementID" type="KwantuElement_type"/>
                        <xs:element name="ElementCalc" type="fnElementCalc"/>
                        <xs:element name="ElementsList" type="fnElementsList"/>
                    </xs:choice>
                    <xs:attribute name="ElementID"/>
                    <xs:attribute name="SubElement"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="ComponentName" type="xs:NCName" use="required">
            <xs:annotation id="AddRows.Component.ComponentName">
                <xs:documentation>Name of the component from which the rows will be retrieved.</xs:documentation>
            </xs:annotation>
        </xs:attribute>
        <xs:attribute name="RepeatID" type="xs:NCName" use="required">
            <xs:annotation id="AddRows.Component.RepeatID">
                <xs:documentation>Name of the RepeatID from which the rows will be retrieved.</xs:documentation>
            </xs:annotation>
        </xs:attribute>
    </xs:complexType>
</xs:element>
</xs:elements>

```

## **Element RepeatDef / AddRows / AutoLoad / BlankLines**

Namespace	No namespace
-----------	--------------

Annotations	Provide a number of blank rows as specified by the maxrows attribute
Diagram	<p>BlankLines (Type xs:short) is connected to xs:short. A callout box states: "Provide a number of blank rows as specified by the maxrows attribute". Another callout box states: "Built-in derived type. The short datatype is derived from int by setting the value of maxInclusive to be 32767 and...".</p>
Type	xs:short
Properties	content: simple
Source	<pre>&lt;xs:element name="BlankLines" type="xs:short"&gt;   &lt;xs:annotation id="AddRows.BlankLines"&gt;     &lt;xs:documentation&gt;Provide a number of blank rows as specified by the maxrows attribute&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

### Element RepeatDef / AddRows / AutoLoad / Codelist

Namespace	No namespace
Annotations	Use the codelist specified by the KeyValue element to generate the rows.
Diagram	<p>Codelist is connected to Taxonomy (Type xs:NCName). A callout box states: "Use the codelist specified by the KeyValue element to generate the rows."</p>
Properties	content: complex
Model	Taxonomy
Children	Taxonomy
Instance	<pre>&lt;Codelist&gt;   &lt;Taxonomy&gt;{1,1}&lt;/Taxonomy&gt; &lt;/Codelist&gt;</pre>
Source	<pre>&lt;xs:element name="Codelist"&gt;   &lt;xs:annotation id="AddRows.Codelist"&gt;     &lt;xs:documentation&gt;Use the codelist specified by the KeyValue element to generate the rows.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="Taxonomy" type="xs:NCName" /&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>

### Element RepeatDef / AddRows / AutoLoad / Codelist / Taxonomy

Namespace	No namespace
Diagram	<p>Taxonomy (Type xs:NCName) is connected to xs:NCName. A callout box states: "Built-in derived type. NCName represents XML 'non-colonized' Names. The base type of NCName is Name."</p>
Type	xs:NCName
Properties	content: simple
Source	<pre>&lt;xs:element name="Taxonomy" type="xs:NCName" /&gt;</pre>

### Element RepeatDef / AddRows / AutoLoad / Periods

Namespace	No namespace
Annotations	Adds rows according the number of periods between the start and end date, one for each period (or part of period for the first and last ones)

Diagram	<pre> classDiagram     class Periods {         &lt;&lt;@ PeriodType         Type Restriction of 'xs:Name'&gt;&gt;         StartDate         EndDate     }     Periods &lt; -- Periods     Periods --&gt; PeriodType     Periods --&gt; StartDate     Periods --&gt; EndDate     note over Periods: Adds rows according the number of periods between the start and end date, one for each period (or part of period for...)   </pre>						
Properties	content: complex						
Model	StartDate , EndDate						
Children	EndDate, StartDate						
Instance	<pre> &lt;Periods PeriodType=""&gt;   &lt;StartDate&gt;{1,1}&lt;/StartDate&gt;   &lt;EndDate&gt;{1,1}&lt;/EndDate&gt; &lt;/Periods&gt;   </pre>						
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">QName</th> <th style="text-align: left; padding: 2px;">Type</th> <th style="text-align: left; padding: 2px;">Use</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">PeriodType</td> <td style="padding: 2px;">restriction of xs:Name</td> <td style="padding: 2px;">required</td> </tr> </tbody> </table>	QName	Type	Use	PeriodType	restriction of xs:Name	required
QName	Type	Use					
PeriodType	restriction of xs:Name	required					
Source	<pre> &lt;xss:element name="Periods"&gt;   &lt;xss:annotation id="AddRows.Periods"&gt;     &lt;xss:documentation&gt;Adds rows according the number of periods between the start and end date, one for each period (or part of period for the first and last ones)&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt;   &lt;xss:complexType&gt;     &lt;xss:sequence&gt;       &lt;xss:element name="StartDate"&gt;         &lt;xss:complexType&gt;           &lt;xss:sequence&gt;             &lt;xss:element name="Element" type="KwantuElement_type"/&gt;           &lt;/xss:sequence&gt;         &lt;/xss:complexType&gt;       &lt;/xss:element&gt;       &lt;xss:element name="EndDate"&gt;         &lt;xss:complexType&gt;           &lt;xss:sequence&gt;             &lt;xss:element name="Element" type="KwantuElement_type"/&gt;           &lt;/xss:sequence&gt;         &lt;/xss:complexType&gt;       &lt;/xss:element&gt;     &lt;/xss:sequence&gt;     &lt;xss:attribute name="PeriodType" use="required"&gt;       &lt;xss:simpleType&gt;         &lt;xss:restriction base="xs:Name"&gt;           &lt;xss:enumeration value="Month"/&gt;           &lt;xss:enumeration value="Quarter"/&gt;           &lt;xss:enumeration value="YearCalendar"/&gt;           &lt;xss:enumeration value="YearAprilToMay"/&gt;           &lt;xss:enumeration value="YearJulyToJune"/&gt;         &lt;/xss:restriction&gt;       &lt;/xss:simpleType&gt;     &lt;/xss:attribute&gt;   &lt;/xss:complexType&gt; &lt;/xss:element&gt;   </pre>						

### Element RepeatDef / AddRows / AutoLoad / Periods / StartDate

Namespace	No namespace
Diagram	<pre> classDiagram     class StartDate {         &lt;&lt;@ PeriodType         Type Restriction of 'xs:Name'&gt;&gt;     }     StartDate &lt; -- StartDate     StartDate --&gt; Element     note over StartDate: Adds rows according the number of periods between the start and end date, one for each period (or part of period for...)   </pre>
Properties	content: complex
Model	Element
Children	Element
Instance	<pre> &lt;StartDate&gt;   </pre>

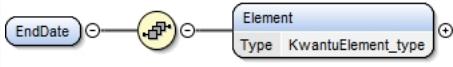
	<pre>&lt;Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE...&gt; &lt;/StartElement&gt;</pre>
Source	<pre>&lt;x:element name="StartDate"&gt;   &lt;x:complexType&gt;     &lt;x:sequence&gt;       &lt;x:element name="Element" type="KwantuElement_type"/&gt;     &lt;/x:sequence&gt;   &lt;/x:complexType&gt; &lt;/x:element&gt;</pre>

### Element RepeatDef / AddRows / AutoLoad / Periods / StartDate / Element

Namespace	No namespace															
Diagram	<p>The diagram illustrates the structure of the <code>KwantuElement_type</code>. It inherits from the <code>xs:string</code> base type. It includes attributes for <code>Seq</code> (xs:int), <code>Id</code> (xs:int), <code>ID</code> (xs:NCName), <code>Component</code> (xs:NCName), <code>Element</code> (xs:NCName), <code>SubElement</code> (xs:NCName), <code>acSequenceRule</code> (Restriction of xs:NCName), <code>RepeatKeyRule</code> (Restriction of xs:NCName), <code>Scope</code> (kw_ElementScope_type), <code>(DataType)</code> (KwantuFormDataTypes), and <code>Xpath</code> (kw_Xpath_type). A note at the bottom states: "This is the base type for all places where an element is referenced".</p>															
Type	<code>KwantuElement_type</code>															
Properties	content: complex															
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><b>Component</b></td> <td>xs:NCName</td> <td>optional</td> </tr> <tr> <td><b>DataType</b></td> <td>KwantuFormDataTypes</td> <td>required</td> </tr> <tr> <td><b>Element</b></td> <td>xs:NCName</td> <td>optional</td> </tr> <tr> <td><b>ID</b></td> <td></td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Use	<b>Component</b>	xs:NCName	optional	<b>DataType</b>	KwantuFormDataTypes	required	<b>Element</b>	xs:NCName	optional	<b>ID</b>		optional
QName	Type	Use														
<b>Component</b>	xs:NCName	optional														
<b>DataType</b>	KwantuFormDataTypes	required														
<b>Element</b>	xs:NCName	optional														
<b>ID</b>		optional														

QName	Type	Use	
<b>Id</b>	xs:int	optional	
	This is used as a sequence number to access multiple elements. Must be renamed to Seq.		
<b>RepeatKeyRule</b>	restriction of xs:NCName	optional	
<b>Scope</b>	kw_ElementScope_type	optional	
<b>Seq</b>	xs:int	optional	
	This is used as a sequence number to access multiple elements. Must be renamed to Seq.		
<b>SubElement</b>	xs:NCName	optional	
<b>Xpath</b>	kw_Xpath_type	optional	
<b>acSequenceRule</b>	restriction of xs:NCName	required	
Source	<code>&lt;xs:element name="Element" type="KwantuElement_type" /&gt;</code>		

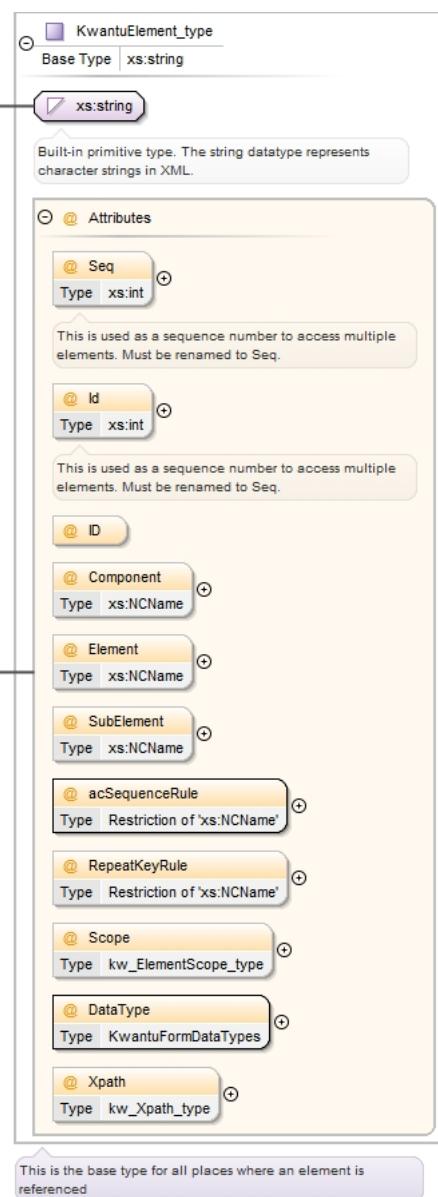
#### **Element RepeatDef / AddRows / AutoLoad / Periods / EndDate**

Namespace	No namespace
Diagram	
Properties	content: complex
Model	Element
Children	Element
Instance	<code>&lt;EndDate&gt;   &lt;Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE   &lt;/EndElement&gt;</code>
Source	<code>&lt;xs:element name="EndDate"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="Element" type="KwantuElement_type"/&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</code>

#### **Element RepeatDef / AddRows / AutoLoad / Periods / EndDate / Element**

Namespace	No namespace
-----------	--------------

Diagram



Type	KwantuElement_type
------	--------------------

Properties	content:	complex
------------	----------	---------

Attributes	QName	Type	Use
	<b>Component</b>	xs:NCName	optional
	<b>DataType</b>	KwantuFormDataTypes	required
	<b>Element</b>	xs:NCName	optional
	<b>ID</b>		optional
	<b>Id</b>	xs:int	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>RepeatKeyRule</b>	restriction of xs:NCName	optional
	<b>Scope</b>	kw_ElementScope_type	optional
	<b>Seq</b>	xs:int	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>SubElement</b>	xs:NCName	optional
	<b>Xpath</b>	kw_Xpath_type	optional

	<b>QName</b>	<b>Type</b>	<b>Use</b>	
	<b>acSequenceRule</b>	restriction of xs:NCName	required	
Source	<xss:element name="Element" type="KwantuElement_type" />			

## Element RepeatDef / AddRows / AutoLoad / Component

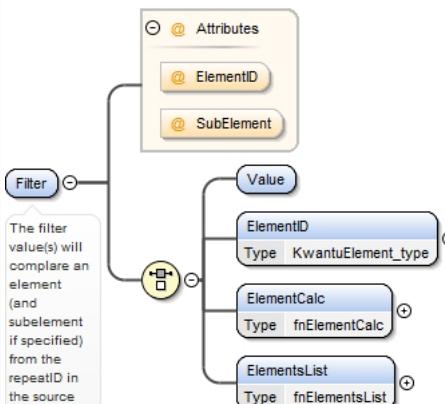
Namespace	No namespace																							
Annotations	Adds blank rows depending on the values in a repeat structure in another component. The value of the component specifies the ID of the repeat structure																							
Diagram																								
Properties	content: complex																							
Model	Filter+																							
Children	Filter																							
Instance	<Component ComponentName="" RepeatID="">     <Filter ElementID="" SubElement="">{1,unbounded}</Filter> </Component>																							
Attributes	<table border="1"> <thead> <tr> <th><b>QName</b></th> <th><b>Type</b></th> <th><b>Use</b></th> <th></th> </tr> </thead> <tbody> <tr> <td><b>ComponentName</b></td> <td>xs:NCName</td> <td>required</td> <td></td> </tr> <tr> <td></td> <td colspan="3">Name of the component from which the rows will be retrieved.</td></tr> <tr> <td><b>RepeatID</b></td> <td>xs:NCName</td> <td>required</td> <td></td> </tr> <tr> <td></td> <td colspan="3">Name of the RepeatID from which the rows will be retrieved.</td></tr> </tbody> </table>				<b>QName</b>	<b>Type</b>	<b>Use</b>		<b>ComponentName</b>	xs:NCName	required			Name of the component from which the rows will be retrieved.			<b>RepeatID</b>	xs:NCName	required			Name of the RepeatID from which the rows will be retrieved.		
<b>QName</b>	<b>Type</b>	<b>Use</b>																						
<b>ComponentName</b>	xs:NCName	required																						
	Name of the component from which the rows will be retrieved.																							
<b>RepeatID</b>	xs:NCName	required																						
	Name of the RepeatID from which the rows will be retrieved.																							
Source	<xss:element name="Component">     <xss:annotation id="AddRows.Component">         <xss:documentation>Adds blank rows depending on the values in a repeat structure in another component. The value of the component specifies the ID of the repeat structure</xss:documentation>     </xss:annotation>     <xss:complexType>         <xss:sequence minOccurs="0">             <xss:element maxOccurs="unbounded" name="Filter">                 <xss:annotation>                     <xss:documentation>The filter value(s) will compare an element (and subelement if specified) from the repeatID in the source component and filter them to the value of the Value or the Source element specified</xss:documentation>                 </xss:annotation>                 <xss:complexType>                     <xss:choice>                         <xss:element name="Value"/>                         <xss:element name="ElementID" type="KwantuElement_type"/>                         <xss:element name="ElementCalc" type="fnElementCalc"/>                         <xss:element name="ElementsList" type="fnElementsList"/>                     </xss:choice>                     <xss:attribute name="ElementID" />                     <xss:attribute name="SubElement" />                 </xss:complexType>             </xss:element>         </xss:sequence>         <xss:attribute name="ComponentName" type="xs:NCName" use="required">             <xss:annotation id="AddRows.Component.ComponentName">                 <xss:documentation>Name of the component from which the rows will be retrieved.</xss:documentation>             </xss:annotation>         </xss:attribute>         <xss:attribute name="RepeatID" type="xs:NCName" use="required">     </xss:complexType> </xss:element>																							

```

<xs:annotation id="AddRows.Component.RepeatID">
    <xs:documentation>Name of the RepeatID from which the rows will be retrieved.</
xs:documentation>
</xs:annotation>
</xs:attribute>
</xs:complexType>
</xs:element>

```

### **Element RepeatDef / AddRows / AutoLoad / Component / Filter**

Namespace	No namespace											
Annotations	The filter value(s) will compare an element (and subelement if specified) from the repeatID in the source component and filter them to the value or the Source element specified											
Diagram	 <p>The filter value(s) will compare an element (and subelement if specified) from the repeatID in the source component...</p>											
Properties	content: complex maxOccurs: unbounded											
Model	Value   ElementID   ElementCalc   ElementsList											
Children	ElementCalc, ElementID, ElementsList, Value											
Instance	<Filter ElementID="" SubElement="">     <Value>{1,1}</Value>     <ElementID acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubElementID>         <ElementCalc>{1,1}</ElementCalc>         <ElementsList>{1,1}</ElementsList>     </ElementID> </Filter>											
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">QName</th> <th style="text-align: left; padding: 2px;">Type</th> <th style="text-align: left; padding: 2px;">Use</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">ElementID</td> <td style="padding: 2px;"></td> <td style="padding: 2px;">optional</td> </tr> <tr> <td style="padding: 2px;">SubElement</td> <td style="padding: 2px;"></td> <td style="padding: 2px;">optional</td> </tr> </tbody> </table>			QName	Type	Use	ElementID		optional	SubElement		optional
QName	Type	Use										
ElementID		optional										
SubElement		optional										
Source	<xs:element maxOccurs="unbounded" name="Filter">     <xs:annotation>         <xs:documentation>The filter value(s) will compare an element (and subelement if specified) from the repeatID in the source component and filter them to the value or the Source element specified</xs:documentation>     </xs:annotation>     <xs:complexType>         <xs:choice>             <xs:element name="Value"/>             <xs:element name="ElementID" type="KwantuElement_type"/>             <xs:element name="ElementCalc" type="fnElementCalc"/>             <xs:element name="ElementsList" type="fnElementsList"/>         </xs:choice>         <xs:attribute name="ElementID"/>         <xs:attribute name="SubElement"/>     </xs:complexType> </xs:element>											

### **Element RepeatDef / AddRows / AutoLoad / Component / Filter / Value**

Namespace	No namespace		
Diagram			
Source	<xs:element name="Value"/>		

**Element RepeatDef / AddRows / AutoLoad / Component / Filter / ElementID**

Namespace	No namespace																														
Diagram	<pre> classDiagram     class KwantuElement_type {         xs:string         @ Seq {xs:int}         @ Id {xs:int}         @ ID         @ Component {xs:NCName}         @ Element {xs:NCName}         @ SubElement {xs:NCName}         @ acSequenceRule {Restriction of 'xs:NCName'}         @ RepeatKeyRule {Restriction of 'xs:NCName'}         @ Scope {kw_ElementScope_type}         @ DataType {KwantuFormDataTypes}         @ Xpath {kw_Xpath_type}     }     class ElementID {         Type KwantuElement_type     }     ElementID &lt; -- KwantuElement_type   </pre> <p>This is the base type for all places where an element is referenced</p>																														
Type	KwantuElement_type																														
Properties	content: complex																														
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>Component</td> <td>xs:NCName</td> <td>optional</td> </tr> <tr> <td>DataType</td> <td>KwantuFormDataTypes</td> <td>required</td> </tr> <tr> <td>Element</td> <td>xs:NCName</td> <td>optional</td> </tr> <tr> <td>ID</td> <td></td> <td>optional</td> </tr> <tr> <td>Id</td> <td>xs:int</td> <td>optional</td> </tr> <tr> <td>Seq</td> <td></td> <td>This is used as a sequence number to access multiple elements. Must be renamed to Seq.</td> </tr> <tr> <td>RepeatKeyRule</td> <td>restriction of xs:NCName</td> <td>optional</td> </tr> <tr> <td>Scope</td> <td>kw_ElementScope_type</td> <td>optional</td> </tr> <tr> <td>SubElement</td> <td></td> <td>This is used as a sequence number to access multiple elements. Must be renamed to SubElement.</td> </tr> </tbody> </table>	QName	Type	Use	Component	xs:NCName	optional	DataType	KwantuFormDataTypes	required	Element	xs:NCName	optional	ID		optional	Id	xs:int	optional	Seq		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	RepeatKeyRule	restriction of xs:NCName	optional	Scope	kw_ElementScope_type	optional	SubElement		This is used as a sequence number to access multiple elements. Must be renamed to SubElement.
QName	Type	Use																													
Component	xs:NCName	optional																													
DataType	KwantuFormDataTypes	required																													
Element	xs:NCName	optional																													
ID		optional																													
Id	xs:int	optional																													
Seq		This is used as a sequence number to access multiple elements. Must be renamed to Seq.																													
RepeatKeyRule	restriction of xs:NCName	optional																													
Scope	kw_ElementScope_type	optional																													
SubElement		This is used as a sequence number to access multiple elements. Must be renamed to SubElement.																													

QName	Type	Use
<b>SubElement</b>	xs:NCName	optional
<b>Xpath</b>	kw_Xpath_type	optional
<b>acSequenceRule</b>	restriction of xs:NCName	required
Source	<xs:element name="ElementID" type="KwantuElement_type"/>	

### Element RepeatDef / AddRows / AutoLoad / Component / Filter / ElementCalc

Namespace	No namespace
Diagram	
Type	fnElementCalc
Properties	content: complex
Model	Calculation , Element{2,2}
Children	Calculation, Element
Instance	<pre>&lt;ElementCalc&gt;   &lt;Calculation&gt;{1,1}&lt;/Calculation&gt;   &lt;Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE&lt;/Element&gt; &lt;/ElementCalc&gt;</pre>
Source	<xs:element name="ElementCalc" type="fnElementCalc"/>

### Element RepeatDef / AddRows / AutoLoad / Component / Filter / ElementsList

Namespace	No namespace
Diagram	
Type	fnElementsList
Properties	content: complex
Model	Element   (Calculation , Element+)
Children	Calculation, Element
Instance	<pre>&lt;ElementsList&gt;   &lt;Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE&lt;/Element&gt;   &lt;Calculation&gt;{1,1}&lt;/Calculation&gt;   &lt;Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE&lt;/Element&gt; &lt;/ElementsList&gt;</pre>
Source	<xs:element name="ElementsList" type="fnElementsList"/>

## Element RepeatDef / HeaderRow

Namespace	No namespace																		
Annotations	This element specifies how to generate the labels, and whether to show the header row or not.																		
Diagram	<p>The diagram shows the <code>HeaderRow</code> element with two attributes: <code>Show</code> (Type: <code>kw_Show_type</code>) and <code>LabelsFromData</code> (Type: <code>Restriction of 'xs:NCName'</code>). A callout box indicates that if <code>LabelsFromData</code> is Yes, it uses the first elements in the columns for the data section to determine the labels.</p>																		
Properties	content: complex																		
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> <th></th> </tr> </thead> <tbody> <tr> <td><code>LabelsFromData</code></td> <td>restriction of <code>xs:NCName</code></td> <td>optional</td> <td></td> </tr> <tr> <td></td> <td colspan="3">If Yes, then use the Label components from the first elements in the columns for the data section to determine the labels</td></tr> <tr> <td><code>Show</code></td> <td><code>kw_Show_type</code></td> <td>optional</td> <td></td> </tr> </tbody> </table>	QName	Type	Use		<code>LabelsFromData</code>	restriction of <code>xs:NCName</code>	optional			If Yes, then use the Label components from the first elements in the columns for the data section to determine the labels			<code>Show</code>	<code>kw_Show_type</code>	optional			
QName	Type	Use																	
<code>LabelsFromData</code>	restriction of <code>xs:NCName</code>	optional																	
	If Yes, then use the Label components from the first elements in the columns for the data section to determine the labels																		
<code>Show</code>	<code>kw_Show_type</code>	optional																	
Source	<pre> &lt;xs:element name="HeaderRow"&gt;   &lt;xs:annotation id="RepeatDef.HeaderRow"&gt;     &lt;xs:documentation&gt;This element specifies how to generate the labels, and whether to show the header row or not.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:attribute ref="Show" /&gt;     &lt;xs:attribute name="LabelsFromData"&gt;       &lt;xs:annotation id="HeaderRow.LabelsFromData"&gt;         &lt;xs:documentation&gt;If Yes, then use the Label components from the first elements in the columns for the data section to determine the labels&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:NCName"&gt;           &lt;xs:enumeration value="Yes" /&gt;           &lt;xs:enumeration value="No" /&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;     &lt;/xs:attribute&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>																		

## Element SummaryRow

Namespace	No namespace										
Annotations	Element determines if there is a summary row or not.										
Diagram	<p>The diagram shows the <code>SummaryRow</code> element with one attribute: <code>Show</code> (Type: <code>kw_Show_type</code>). A callout box indicates that the element determines if there is a summary row or not.</p>										
Properties	content: complex										
Used by	Element RepeatDef										
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> <th></th> </tr> </thead> <tbody> <tr> <td><code>Show</code></td> <td><code>kw_Show_type</code></td> <td>optional</td> <td></td> </tr> </tbody> </table>	QName	Type	Use		<code>Show</code>	<code>kw_Show_type</code>	optional			
QName	Type	Use									
<code>Show</code>	<code>kw_Show_type</code>	optional									
Source	<pre> &lt;xs:element name="SummaryRow"&gt;   &lt;xs:annotation id="Repeat.Def.SummaryRow"&gt;     &lt;xs:documentation&gt;Element determines if there is a summary row or not.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:attribute ref="Show" /&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>										

## Element RepeatDef / Columns

Namespace	No namespace							
Diagram	<p><b>Attributes</b></p> <ul style="list-style-type: none"> <li><b>NumberColumns</b> Type: xs:integer</li> </ul> <p>Defines the number of columns that should be generated for the repeat structure. These are Visible columns. Each column...</p> <p>1..∞ → Column</p>							
Properties	content: complex							
Model	Column							
Children	Column							
Instance	<pre>&lt;Columns NumberColumns=""&gt;   &lt;Column Seq="" Width=""&gt;{1,1}&lt;/Column&gt; &lt;/Columns&gt;</pre>							
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><b>NumberColumns</b></td> <td>xs:integer</td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Use	<b>NumberColumns</b>	xs:integer	optional	<p>Defines the number of columns that should be generated for the repeat structure. These are Visible columns. Each column has a header row, a data area and a summary area. In each of these areas there may be one or more elements.</p> <p>The Number of columns in the header, data and summary bands need not be equal, but a means must be devised to specify the span.</p>
QName	Type	Use						
<b>NumberColumns</b>	xs:integer	optional						
Source	<pre>&lt;xs:element name="Columns"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence maxOccurs="unbounded"&gt;       &lt;xs:element name="Column"&gt;         &lt;xs:complexType&gt;           &lt;xs:attribute ref="Seq"&gt;             &lt;xs:annotation id="Repeat.RepeatDef.Columns.Column.Seq"&gt;               &lt;xs:documentation&gt;The number of the column&lt;/xs:documentation&gt;             &lt;/xs:annotation&gt;           &lt;/xs:attribute&gt;           &lt;xs:attribute name="Width"&gt;             &lt;xs:annotation id="Repeat.RepeatDef.Columns.Column.Width"&gt;               &lt;xs:documentation&gt;Defines the width of the column&lt;/xs:documentation&gt;             &lt;/xs:annotation&gt;           &lt;/xs:attribute&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute name="NumberColumns" type="xs:integer"&gt;       &lt;xs:annotation id="Repeat.RepeatDef.Columns.NumberColumns"&gt;         &lt;xs:documentation&gt;Defines the number of columns that should be generated for the repeat structure. These are Visible columns. Each column has a header row, a data area and a summary area. In each of these areas there may be one or more elements. The Number of columns in the header, data and summary bands need not be equal, but a means must be devised to specify the span.&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:attribute&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>							

## Element RepeatDef / Columns / Column

Namespace	No namespace		
Diagram	<p><b>Attributes</b></p> <ul style="list-style-type: none"> <li><b>Seq</b></li> <li>The number of the column</li> <li><b>Width</b></li> <li>Defines the width of the column</li> </ul>		
Properties	content: complex		

Attributes	QName	Type	Use
	<b>Seq</b>		optional
	<b>Width</b>		optional
	Defines the width of the column		
Source	<pre>&lt;xs:element name="Column"&gt;   &lt;xs:complexType&gt;     &lt;xs:attribute ref="Seq"&gt;       &lt;xs:annotation id="Repeat.RepeatDef.Columns.Column.Seq"&gt;         &lt;xs:documentation&gt;The number of the column&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:attribute&gt;     &lt;xs:attribute name="Width"&gt;       &lt;xs:annotation id="Repeat.RepeatDef.Columns.Column.Width"&gt;         &lt;xs:documentation&gt;Defines the width of the column&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:attribute&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>		

### Element KeyValue

Namespace	No namespace		
Annotations	The KeyValue determines what creates the unique key for the list of rows. There might still be a sequence number, but if the user chooses a codelist, then there is a unique code and value associated with the code list that will populate the data for each row		
Diagram	<p>The KeyValue determines what creates the unique key for the list of rows. There might still be a sequence number, but...</p>		<p>Built-in derived type. NCName represents XML "non-colonized" Names. The base type of NCName is Name.</p>
Type	extension of xs:NCName		
Properties	content:	complex	
Used by	Element	RepeatDef	
Source	<pre>&lt;xs:element name="KeyValue"&gt;   &lt;xs:annotation id="KeyValue"&gt;     &lt;xs:documentation&gt;The KeyValue determines what creates the unique key for the list of rows. There might still be a sequence number, but if the user chooses a codelist, then there is a unique code and value associated with the code list that will populate the data for each row&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:simpleContent&gt;       &lt;xs:extension base="xs:NCName" /&gt;     &lt;/xs:simpleContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>		

### Element RepeatHeader

Namespace	No namespace		
Diagram			
Properties	content:	complex	
Used by	Complex Type	kw_Repeat_type	
Model	rptColumn+		
Children	rptColumn		

Instance	<pre>&lt;RepeatHeader GroupName="" ID=""&gt;   &lt;rptColumn ColumnNo="" Span=""&gt;{1,unbounded}&lt;/rptColumn&gt; &lt;/RepeatHeader&gt;</pre>		
Attributes	QName	Type	Use
	GroupName	xs:NCName	required
Source	<pre>&lt;xss:element name="RepeatHeader"&gt;   &lt;xss:complexType&gt;     &lt;xss:sequence&gt;       &lt;xss:element maxOccurs="unbounded" name="rptColumn"&gt;         &lt;xss:complexType&gt;           &lt;xss:sequence&gt;             &lt;xss:group ref="BaseElements"/&gt;           &lt;/xss:sequence&gt;           &lt;xss:attribute name="ColumnNo"/&gt;           &lt;xss:attribute name="Span"/&gt;         &lt;/xss:complexType&gt;       &lt;/xss:element&gt;     &lt;/xss:sequence&gt;     &lt;xss:attribute name="GroupName" type="xs:NCName" use="required"/&gt;     &lt;xss:attribute name="ID" type="xs:NCName" use="required"/&gt;   &lt;/xss:complexType&gt; &lt;/xss:element&gt;</pre>		

## Element RepeatHeader / rptColumn

Namespace	No namespace						
Diagram	<pre> classDiagram     class rptColumn {         @Attributes         @ColumnNo         @Span     }     class BaseElements {         elLabel         elField         elSelect         elCalcField         Repeat         elGroup     }     class elLabel     class elField     class elSelect     class elCalcField     class Repeat     class elGroup   </pre> <p>Root element for the structure that defines the repeat or table structure</p>						
Properties	content: complex maxOccurs: unbounded						
Model	(elLabel   elField   elSelect   elCalcField   Repeat   elGroup)						
Children	Repeat, elCalcField, elField, elGroup, elLabel, elSelect						
Instance	<pre>&lt;rptColumn ColumnNo="" Span=""&gt;   &lt;elLabel ID="" Type=""&gt;{1,1}&lt;/elLabel&gt;   &lt;elField ID="" Type=""&gt;{1,1}&lt;/elField&gt;   &lt;elSelect ID="" Type=""&gt;{1,1}&lt;/elSelect&gt;   &lt;elCalcField ID="" Type=""&gt;{1,1}&lt;/elCalcField&gt;   &lt;Repeat ID="" Type=""&gt;{1,1}&lt;/Repeat&gt;   &lt;elGroup ID=""&gt;{1,1}&lt;/elGroup&gt; &lt;/rptColumn&gt;</pre>						
Attributes	<table border="1"> <tr> <td>QName</td> <td>Type</td> <td>Use</td> </tr> <tr> <td>ColumnNo</td> <td></td> <td>optional</td> </tr> </table>	QName	Type	Use	ColumnNo		optional
QName	Type	Use					
ColumnNo		optional					

	<b>QName</b>	<b>Type</b>	<b>Use</b>	
	<b>Span</b>		optional	
Source		<pre>&lt;xs:element maxOccurs="unbounded" name="rptColumn"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:group ref="BaseElements" /&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute name="ColumnNo" /&gt;     &lt;xs:attribute name="Span" /&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>		

### Element elCalcField

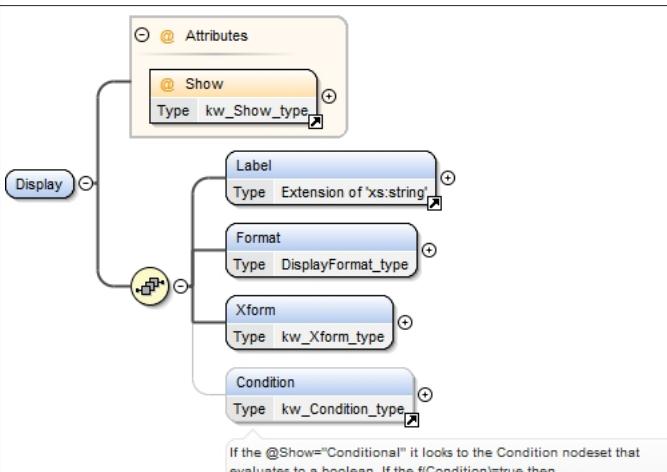
Namespace	No namespace															
Diagram	<pre> classDiagram     class elCalcField {         @ ID : xs.ID         @ Type : Restriction of xs.NCName         UIVersionElements         Display         EntryMode         Validations         Reports         Help     }     elCalcField "1..1" -- "1..1" UIVersionElements     elCalcField "1..1" -- "1..1" Display     elCalcField "1..1" -- "1..1" EntryMode     elCalcField "1..1" -- "1..1" Validations     elCalcField "1..1" -- "1..1" Reports     elCalcField "1..1" -- "1..1" Help   </pre>															
Properties	content: complex															
Used by	Element Group: BaseElements Element: Elements															
Model	UIVersionElements , Display , EntryMode , Validations , Reports , Help															
Children	Display, EntryMode, Help, Reports, UIVersionElements, Validations															
Instance	<pre> &lt;elCalcField ID="" Type=""&gt;   &lt;UIVersionElements&gt;{1,1}&lt;/UIVersionElements&gt;   &lt;Display Show=""&gt;{1,1}&lt;/Display&gt;   &lt;EntryMode Mode=""&gt;{1,1}&lt;/EntryMode&gt;   &lt;Validations&gt;{1,1}&lt;/Validations&gt;   &lt;Reports Include=""&gt;{1,1}&lt;/Reports&gt;   &lt;Help Show=""&gt;{1,1}&lt;/Help&gt; &lt;/elCalcField&gt;   </pre>															
Attributes	<table border="1"> <thead> <tr> <th><b>QName</b></th> <th><b>Type</b></th> <th><b>Use</b></th> <th></th> </tr> </thead> <tbody> <tr> <td><b>ID</b></td> <td>xs:ID</td> <td>required</td> <td></td> </tr> <tr> <td><b>Type</b></td> <td>restriction of xs:NCName</td> <td>required</td> <td></td> </tr> </tbody> </table>	<b>QName</b>	<b>Type</b>	<b>Use</b>		<b>ID</b>	xs:ID	required		<b>Type</b>	restriction of xs:NCName	required				
<b>QName</b>	<b>Type</b>	<b>Use</b>														
<b>ID</b>	xs:ID	required														
<b>Type</b>	restriction of xs:NCName	required														
Source	<pre>&lt;xs:element name="elCalcField"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element ref="UIVersionElements" /&gt;       &lt;xs:element name="Display"&gt;         &lt;xs:complexType&gt;           &lt;xs:sequence&gt;             &lt;xs:element ref="Label" /&gt;             &lt;xs:element name="Format" type="DisplayFormat_type" /&gt;             &lt;xs:element name="Xform" type="kw_Xform_type" /&gt;             &lt;xs:element minOccurs="0" ref="Condition" /&gt;           &lt;/xs:sequence&gt;           &lt;xs:attribute ref="Show" use="required" /&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>															

```

<xs:element name="EntryMode">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="Default" type="Default_type"/>
      <xs:element name="CalcField">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="Calc" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="Mode" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:NCName">
          <xs:enumeration value="ReadOnly"/>
          <xs:enumeration value="Hidden"/>
          <xs:enumeration value="Calculated"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>
<xs:element ref="Validations" />
<xs:element ref="Reports" />
<xs:element ref="Help" />
</xs:sequence>
<xs:attribute use="required" ref="ID" />
<xs:attribute name="Type" use="required">
  <xs:simpleType>
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="CalcField"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>

```

## Element elCalcField / Display

Namespace	No namespace								
Diagram	 <p>If the @Show="Conditional" it looks to the Condition nodeset that evaluates to a boolean. If the f(Condition)=true then...</p>								
Properties	content: complex								
Model	Label , Format , Xform , Condition{0,1}								
Children	Condition, Format, Label, Xform								
Instance	<pre> &lt;Display Show=""&gt;   &lt;Label Show=""&gt;{1,1}&lt;/Label&gt;   &lt;Format&gt;{1,1}&lt;/Format&gt;   &lt;Xform Appearance="" CSS=""&gt;{1,1}&lt;/Xform&gt;   &lt;Condition Operator=""&gt;{0,1}&lt;/Condition&gt; &lt;/Display&gt; </pre>								
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">QName</th> <th style="text-align: left; padding: 2px;">Type</th> <th style="text-align: left; padding: 2px;">Use</th> <th style="text-align: left; padding: 2px;"></th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">Show</td> <td style="padding: 2px;">kw_Show_type</td> <td style="padding: 2px;">required</td> <td style="padding: 2px;"></td> </tr> </tbody> </table>	QName	Type	Use		Show	kw_Show_type	required	
QName	Type	Use							
Show	kw_Show_type	required							
Source	<pre> &lt;xs:element name="Display"&gt;   &lt;xs:complexType&gt; </pre>								

```

<xs:sequence>
  <xs:element ref="Label" />
  <xs:element name="Format" type="DisplayFormat_type" />
  <xs:element name="Xform" type="kw_Xform_type" />
  <xs:element minOccurs="0" ref="Condition" />
</xs:sequence>
<xs:attribute ref="Show" use="required" />
</xs:complexType>
</xs:elements>

```

### Element elCalcField / Display / Format

Namespace	No namespace														
Diagram															
Type	DisplayFormat_type														
Properties	content: simple														
Facets	<table border="1"> <tr><td>enumeration</td><td>Float</td></tr> <tr><td>enumeration</td><td>Numeric</td></tr> <tr><td>enumeration</td><td>Date</td></tr> <tr><td>enumeration</td><td>String</td></tr> <tr><td>enumeration</td><td>Memo</td></tr> <tr><td>enumeration</td><td>Boolean</td></tr> <tr><td>enumeration</td><td>Integer</td></tr> </table>	enumeration	Float	enumeration	Numeric	enumeration	Date	enumeration	String	enumeration	Memo	enumeration	Boolean	enumeration	Integer
enumeration	Float														
enumeration	Numeric														
enumeration	Date														
enumeration	String														
enumeration	Memo														
enumeration	Boolean														
enumeration	Integer														
Source	<xs:element name="Format" type="DisplayFormat_type" />														

### Element elCalcField / Display / Xform

Namespace	No namespace									
Diagram										
Type	kw_Xform_type									
Properties	content: complex									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>Appearance</td> <td>restriction of xs:Name</td> <td>required</td> </tr> <tr> <td>CSS</td> <td>restriction of xs:Name</td> <td>required</td> </tr> </tbody> </table>	QName	Type	Use	Appearance	restriction of xs:Name	required	CSS	restriction of xs:Name	required
QName	Type	Use								
Appearance	restriction of xs:Name	required								
CSS	restriction of xs:Name	required								
Source	<xs:element name="Xform" type="kw_Xform_type" />									

### Element elCalcField / EntryMode

Namespace	No namespace
Diagram	

Properties	content: complex		
Model	Default{0,1} , CalcField		
Children	CalcField, Default		
Instance	<pre>&lt;EntryMode Mode=""&gt;   &lt;Default&gt;{0,1}&lt;/Default&gt;   &lt;CalcField&gt;{1,1}&lt;/CalcField&gt; &lt;/EntryMode&gt;</pre>		
Attributes	QName	Type	Use
	Mode	restriction of xs:NCName	required
Source	<pre>&lt;xs:element name="EntryMode"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element minOccurs="0" name="Default" type="Default_type"/&gt;       &lt;xs:element name="CalcField"&gt;         &lt;xs:complexType&gt;           &lt;xs:sequence&gt;             &lt;xs:element ref="Calc"/&gt;           &lt;/xs:sequence&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute name="Mode" use="required"&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:NCName"&gt;           &lt;xs:enumeration value="ReadOnly"/&gt;           &lt;xs:enumeration value="Hidden"/&gt;           &lt;xs:enumeration value="Calculated"/&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;     &lt;/xs:attribute&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>		

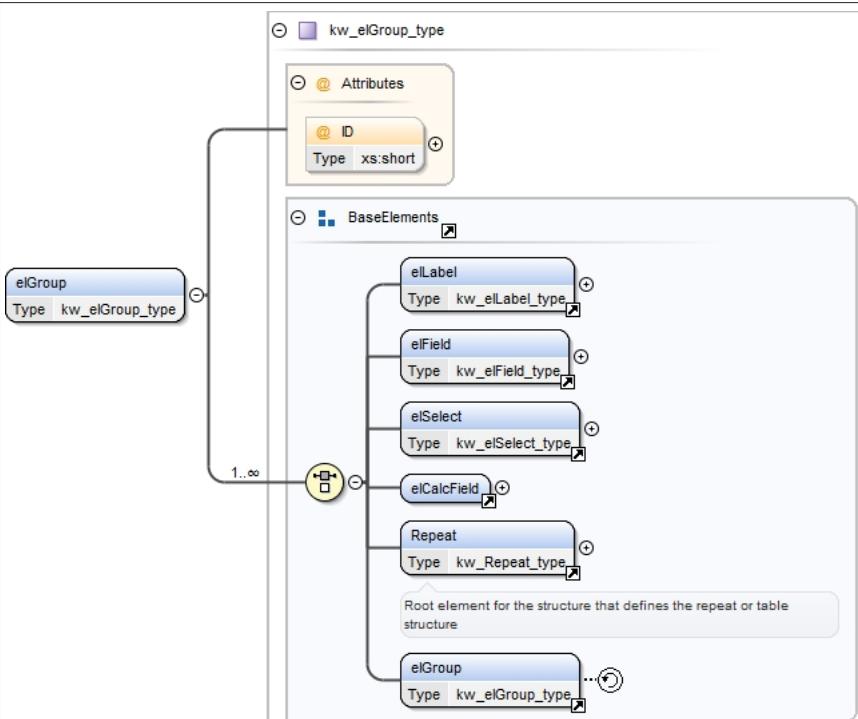
### Element elCalcField / EntryMode / Default

Namespace	No namespace				
Diagram					
Type	Default_type				
Properties	<table border="1"> <tr> <td>content:</td><td>complex</td></tr> <tr> <td>minOccurs:</td><td>0</td></tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Model	Value   (InitiatialValue , Calc)   ElementValue				
Children	Calc, ElementValue, InitiatialValue, Value				
Instance	<pre>&lt;Default&gt;   &lt;Value DataType=""&gt;{1,1}&lt;/Value&gt;   &lt;InitiatialValue DataType=""&gt;{1,1}&lt;/InitiatialValue&gt;   &lt;Calc Seq=""&gt;{1,1}&lt;/Calc&gt;   &lt;ElementValue acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" ElementValue&gt; &lt;/Default&gt;</pre>				
Source	<pre>&lt;xs:element minOccurs="0" name="Default" type="Default_type"/&gt;</pre>				

## Element elCalcField / EntryMode / CalcField

Namespace	No namespace
Diagram	
Properties	content: complex
Model	Calc
Children	Calc
Instance	<pre>&lt;CalcField&gt;   &lt;Calc Seq=""&gt;{1,1}&lt;/Calc&gt; &lt;/CalcField&gt;</pre>
Source	<pre>&lt;xss:element name="CalcField"&gt;   &lt;xss:complexType&gt;     &lt;xss:sequence&gt;       &lt;xss:element ref="Calc"/&gt;     &lt;/xss:sequence&gt;   &lt;/xss:complexType&gt; &lt;/xss:element&gt;</pre>

## Element elGroup

Namespace	No namespace				
Diagram					
Type	kw_elGroup_type				
Properties	content: complex				
Used by	<table border="0"> <tr> <td>Element Group</td> <td>BaseElements</td> </tr> <tr> <td>Element</td> <td>Elements</td> </tr> </table>	Element Group	BaseElements	Element	Elements
Element Group	BaseElements				
Element	Elements				
Model	elLabel   elField   elSelect   elCalcField   Repeat   elGroup				
Children	Repeat, elCalcField, elField, elGroup, elLabel, elSelect				
Instance	<pre>&lt;elGroup ID=""&gt;   &lt;elLabel ID="" Type=""&gt;{1,1}&lt;/elLabel&gt;   &lt;elField ID="" Type=""&gt;{1,1}&lt;/elField&gt;   &lt;elSelect ID="" Types=""&gt;{1,1}&lt;/elSelect&gt;   &lt;elCalcField ID="" Type=""&gt;{1,1}&lt;/elCalcField&gt;   &lt;Repeat ID="" Type=""&gt;{1,1}&lt;/Repeat&gt; &lt;/elGroup&gt;</pre>				

	</elGroup>		
Attributes	<b>QName</b>	Type	Use
	<b>ID</b>	xs:short	optional
Source	<xs:element name="elGroup" type="kw_elGroup_type"/>		

## Element RepeatData

Namespace	No namespace		
Annotations	This is the root element that defines the data block for the repeat table. Everything under this element is repeated for each row in the table.		
Diagram	<pre> classDiagram     class RepeatData {         @ Attributes         @ GroupName         @ ID         rptColumn *--&gt; 1..infinity rptColumn     }   </pre>		
Properties	content:	complex	
Used by	Complex Type	kw_Repeat_type	
Model	rptColumn+		
Children	rptColumn		
Instance	<RepeatData GroupName="" ID="">   <rptColumn ColumnNo="" Span="">{1,unbounded}</rptColumn> </RepeatData>		
Attributes	<b>QName</b>	Type	Use
	<b>GroupName</b>	xs:NCName	required
	<b>ID</b>	xs:NCName	required
Source	<pre> &lt;xs:element name="RepeatData"&gt;   &lt;xs:annotation id="Repeat.RepeatData"&gt;     &lt;xs:documentation&gt;This is the root element that defines the data block for the repeat table. Everything under this element is repeated for each row in the table.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element maxOccurs="unbounded" name="rptColumn"&gt;         &lt;xs:complexType&gt;           &lt;xs:sequence&gt;             &lt;xs:group ref="BaseElements"/&gt;           &lt;/xs:sequence&gt;           &lt;xs:attribute name="ColumnNo"/&gt;           &lt;xs:attribute name="Span"/&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute name="GroupName" type="xs:NCName" use="required"/&gt;     &lt;xs:attribute name="ID" type="xs:NCName" use="required"/&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>		

## Element RepeatData / rptColumn

Namespace	No namespace
-----------	--------------

Diagram	<pre> classDiagram     class rptColumn {         @ ColumnNo         @ Span         --&gt; rptColumn         --&gt; BaseElements     }     class BaseElements {         elLabel         elField         elSelect         elCalcField         Repeat         elGroup     }     class Attributes {         @ ColumnNo         @ Span     }   </pre>									
Properties	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">content:</td><td style="padding: 2px;">complex</td></tr> <tr> <td style="padding: 2px;">maxOccurs:</td><td style="padding: 2px;">unbounded</td></tr> </table>	content:	complex	maxOccurs:	unbounded					
content:	complex									
maxOccurs:	unbounded									
Model	(elLabel   elField   elSelect   elCalcField   Repeat   elGroup)									
Children	Repeat, elCalcField, elField, elGroup, elLabel, elSelect									
Instance	<pre> &lt;rptColumn ColumnNo="" Span=""&gt;     &lt;elLabel ID="" Type=""&gt;{1,1}&lt;/elLabel&gt;     &lt;elField ID="" Type=""&gt;{1,1}&lt;/elField&gt;     &lt;elSelect ID="" Type=""&gt;{1,1}&lt;/elSelect&gt;     &lt;elCalcField ID="" Type=""&gt;{1,1}&lt;/elCalcField&gt;     &lt;Repeat ID="" Type=""&gt;{1,1}&lt;/Repeat&gt;     &lt;elGroup ID=""&gt;{1,1}&lt;/elGroup&gt; &lt;/rptColumn&gt;   </pre>									
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="padding: 2px;">QName</th><th style="padding: 2px;">Type</th><th style="padding: 2px;">Use</th></tr> </thead> <tbody> <tr> <td style="padding: 2px;">ColumnNo</td><td></td><td style="padding: 2px;">optional</td></tr> <tr> <td style="padding: 2px;">Span</td><td></td><td style="padding: 2px;">optional</td></tr> </tbody> </table>	QName	Type	Use	ColumnNo		optional	Span		optional
QName	Type	Use								
ColumnNo		optional								
Span		optional								
Source	<pre> &lt;xs:element maxOccurs="unbounded" name="rptColumn"&gt;     &lt;xs:complexType&gt;         &lt;xs:sequence&gt;             &lt;xs:group ref="BaseElements" /&gt;         &lt;/xs:sequence&gt;         &lt;xs:attribute name="ColumnNo" /&gt;         &lt;xs:attribute name="Span" /&gt;     &lt;/xs:complexType&gt; &lt;/xs:element&gt;   </pre>									

## Element RepeatSummary

Namespace	No namespace		
Diagram	<pre> classDiagram     class RepeatSummary {         @ GroupName         @ ID         --&gt; rptColumn     }     class Attributes {         @ GroupName         @ ID     }   </pre>		
Properties	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">content:</td><td style="padding: 2px;">complex</td></tr> </table>	content:	complex
content:	complex		

Used by	Complex Type kw_Repeat_type		
Model	rptColumn+		
Children	rptColumn		
Instance	<pre>&lt;RepeatSummary GroupName="" ID=""&gt;   &lt;rptColumn ColumnNo="" Span=""&gt;{1,unbounded}&lt;/rptColumn&gt; &lt;/RepeatSummary&gt;</pre>		
Attributes	QName	Type	Use
	GroupName	xs:NCName	required
	ID	xs:NCName	required
Source	<pre>&lt;xs:element name="RepeatSummary"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element maxOccurs="unbounded" ref="rptColumn"/&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute name="GroupName" type="xs:NCName" use="required"/&gt;     &lt;xs:attribute name="ID" type="xs:NCName" use="required"/&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>		

## Element rptColumn

Namespace	No namespace		
Diagram	<pre> classDiagram     class rptColumn {         @ColumnNo         @Span         elLabel         elField         elSelect         elCalcField         Repeat         elGroup     }     class elLabel     class elField     class elSelect     class elCalcField     class Repeat     class elGroup     rptColumn &lt; -- elLabel     rptColumn &lt; -- elField     rptColumn &lt; -- elSelect     rptColumn &lt; -- elCalcField     rptColumn &lt; -- Repeat     rptColumn &lt; -- elGroup     Note over Repeat: Root element for the structure that defines the repeat or table structure   </pre>		
Properties	content:	complex	
Used by	Element	RepeatSummary	
Model	(elLabel   elField   elSelect   elCalcField   Repeat   elGroup)		
Children	Repeat, elCalcField, elField, elGroup, elLabel, elSelect		
Instance	<pre>&lt;rptColumn ColumnNo="" Span=""&gt;   &lt;elLabel ID="" Type=""&gt;{1,1}&lt;/elLabel&gt;   &lt;elField ID="" Type=""&gt;{1,1}&lt;/elField&gt;   &lt;elSelect ID="" Type=""&gt;{1,1}&lt;/elSelect&gt;   &lt;elCalcField ID="" Type=""&gt;{1,1}&lt;/elCalcField&gt;   &lt;Repeat ID="" Type=""&gt;{1,1}&lt;/Repeat&gt;   &lt;elGroup ID=""&gt;{1,1}&lt;/elGroup&gt; &lt;/rptColumn&gt;</pre>		
Attributes	QName	Type	Use
	ColumnNo		optional
	Span		optional

Source	<pre>&lt;xs:element name="rptColumn"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:group ref="BaseElements" /&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute name="ColumnNo" /&gt;     &lt;xs:attribute name="Span" /&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>
--------	--

## Element Hierarchy

Namespace	No namespace															
Diagram	<pre> classDiagram     class Hierarchy {         @ Attributes         @ Levels         Type Restriction of 'xs:short'         @ Taxonomy         Type xs:string         @ TopLevel         Type xs:token         @ Type         Type Restriction of 'xs:NCName'     }     class HierarchyNode     Hierarchy "1..∞" -- "HierarchyNode"   </pre>															
Properties	content: complex															
Used by	Element Elements															
Model	HierarchyNode+															
Children	HierarchyNode															
Instance	<pre>&lt;Hierarchy Levels="" Taxonomy="" TopLevel="" Type=""&gt;   &lt;HierarchyNode Level="" Parameter="" SelectElement="" {1,unbounded}&lt;/HierarchyNode&gt; &lt;/Hierarchy&gt;</pre>															
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>Levels</td> <td>restriction of xs:short</td> <td>optional</td> </tr> <tr> <td>Taxonomy</td> <td>xs:string</td> <td>optional</td> </tr> <tr> <td>TopLevel</td> <td>xs:token</td> <td>optional</td> </tr> <tr> <td>Type</td> <td>restriction of xs:NCName</td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Use	Levels	restriction of xs:short	optional	Taxonomy	xs:string	optional	TopLevel	xs:token	optional	Type	restriction of xs:NCName	optional
QName	Type	Use														
Levels	restriction of xs:short	optional														
Taxonomy	xs:string	optional														
TopLevel	xs:token	optional														
Type	restriction of xs:NCName	optional														
Source	<pre>&lt;xs:element name="Hierarchy"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element maxOccurs="unbounded" ref="HierarchyNode" /&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute name="Levels"&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:short"&gt;           &lt;xs:minInclusive value="-32768" /&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;     &lt;/xs:attribute&gt;     &lt;xs:attribute name="Taxonomy" type="xs:string" /&gt;     &lt;xs:attribute name="TopLevel" type="xs:token" /&gt;     &lt;xs:attribute name="Type"&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:NCName"&gt;           &lt;xs:enumeration value="Hierarchy" /&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;     &lt;/xs:attribute&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>															

## Element HierarchyNode

Namespace	No namespace
-----------	--------------

Diagram	<pre> classDiagram     class HierarchyNode {         @ Attributes: Level(xs:short), Parameter(xs:NCName), SelectElement(xs:NCName)         &lt; --&gt; BaseElements     }     class BaseElements {         elLabel(kw_elLabel_type)         elField(kw_elField_type)         elSelect(kw_elSelect_type)         elCalcField()         Repeat(kw_Repeat_type)         elGroup(kw_elGroup_type)     }     HierarchyNode &lt; --&gt; 1..∞ BaseElements   </pre>												
Properties	content: complex												
Used by	Element Hierarchy												
Model	(elLabel   elField   elSelect   elCalcField   Repeat   elGroup)												
Children	Repeat, elCalcField, elField, elGroup, elLabel, elSelect												
Instance	<pre> &lt;HierarchyNode Level="" Parameter="" SelectElement=""&gt;   &lt;elLabel ID="" Type=""&gt;{1,1}&lt;/elLabel&gt;   &lt;elField ID="" Type=""&gt;{1,1}&lt;/elField&gt;   &lt;elSelect ID="" Types=""&gt;{1,1}&lt;/elSelect&gt;   &lt;elCalcField ID="" Type=""&gt;{1,1}&lt;/elCalcField&gt;   &lt;Repeat ID="" Type=""&gt;{1,1}&lt;/Repeat&gt;   &lt;elGroup ID=""&gt;{1,1}&lt;/elGroup&gt; &lt;/HierarchyNode&gt;   </pre>												
Attributes	<table border="1"> <thead> <tr> <th>QName</th><th>Type</th><th>Use</th></tr> </thead> <tbody> <tr> <td><b>Level</b></td><td>xs:short</td><td>optional</td></tr> <tr> <td><b>Parameter</b></td><td>xs:NCName</td><td>required</td></tr> <tr> <td><b>SelectElement</b></td><td>xs:NCName</td><td>optional</td></tr> </tbody> </table>	QName	Type	Use	<b>Level</b>	xs:short	optional	<b>Parameter</b>	xs:NCName	required	<b>SelectElement</b>	xs:NCName	optional
QName	Type	Use											
<b>Level</b>	xs:short	optional											
<b>Parameter</b>	xs:NCName	required											
<b>SelectElement</b>	xs:NCName	optional											
Source	<pre> &lt;xs:element name="HierarchyNode"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:group maxOccurs="unbounded" ref="BaseElements"/&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute name="Level" type="xs:short"/&gt;     &lt;xs:attribute name="Parameter" type="xs:NCName" use="required"/&gt;     &lt;xs:attribute name="SelectElement" type="xs:NCName"/&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;   </pre>												

## Element elSelectGroup

Namespace	No namespace
-----------	--------------

Diagram	<pre> classDiagram     class elSelectGroup {         @ Attributes         @ ID Type xs:ID         @ Type Type Restriction of 'xs:Name'     }     elSelectGroup "1..1" -- "*" SelectGroupKey : SelectGroupKey     elSelectGroup "*" -- "*" SelectGroup : SelectGroup   </pre>												
Properties	content: complex												
Used by	Element Elements												
Model	SelectGroupKey , SelectGroup												
Children	SelectGroup, SelectGroupKey												
Instance	<pre> &lt;elSelectGroup ID="" Type=""&gt;   &lt;SelectGroupKey KeyID=""&gt;{1,1}&lt;/SelectGroupKey&gt;   &lt;SelectGroup&gt;{1,1}&lt;/SelectGroup&gt; &lt;/elSelectGroup&gt;   </pre>												
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">QName</th><th style="text-align: left; padding: 2px;">Type</th><th style="text-align: left; padding: 2px;">Use</th><th style="text-align: left; padding: 2px;"></th></tr> </thead> <tbody> <tr> <td style="padding: 2px;"><b>ID</b></td><td style="padding: 2px;">xs:ID</td><td style="padding: 2px;">required</td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;"><b>Type</b></td><td style="padding: 2px;">restriction of xs:Name</td><td style="padding: 2px;">required</td><td style="padding: 2px;"></td></tr> </tbody> </table>	QName	Type	Use		<b>ID</b>	xs:ID	required		<b>Type</b>	restriction of xs:Name	required	
QName	Type	Use											
<b>ID</b>	xs:ID	required											
<b>Type</b>	restriction of xs:Name	required											
Source	<pre> &lt;xs:element name="elSelectGroup"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="SelectGroupKey"&gt;         &lt;xs:complexType&gt;           &lt;xs:all&gt;             &lt;xs:element ref="elSelect"/&gt;           &lt;/xs:all&gt;           &lt;xs:attribute name="KeyID" type="xs:token" use="required"&gt;             &lt;xs:annotation&gt;               &lt;xs:documentation&gt;This KeyID references the ID of the field on the group that forms the key to link to this section&lt;/xs:documentation&gt;             &lt;/xs:annotation&gt;           &lt;/xs:attribute&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;       &lt;xs:element name="SelectGroup"&gt;         &lt;xs:complexType&gt;           &lt;xs:sequence maxOccurs="unbounded"&gt;             &lt;xs:group ref="BaseElements" /&gt;           &lt;/xs:sequence&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute ref="ID" use="required" /&gt;     &lt;xs:attribute name="Type" use="required"&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:Name"&gt;           &lt;xs:enumeration value="PersonsList" /&gt;           &lt;xs:enumeration value="OrganisationsList" /&gt;           &lt;xs:enumeration value="Rest" /&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;     &lt;/xs:attribute&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;   </pre>												

## Element elSelectGroup / SelectGroupKey

Namespace	No namespace
-----------	--------------

Diagram	<pre> classDiagram     class SelectGroupKey {         @ Attributes         @ KeyID Type xs:token         elSelect Type kw_elSelect_type     }     Note over KeyID: This KeyID references the ID of the field on the group that forms the key to link to this section   </pre>									
Properties	content: complex									
Model	ALL(elSelect)									
Children	elSelect									
Instance	<pre> &lt;SelectGroupKey KeyID=""&gt;   &lt;elSelect ID="" Type=""&gt;{1,1}&lt;/elSelect&gt; &lt;/SelectGroupKey&gt;   </pre>									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><b>KeyID</b></td> <td>xs:token</td> <td>required</td> </tr> <tr> <td></td> <td></td> <td>This KeyID references the ID of the field on the group that forms the key to link to this section</td> </tr> </tbody> </table>	QName	Type	Use	<b>KeyID</b>	xs:token	required			This KeyID references the ID of the field on the group that forms the key to link to this section
QName	Type	Use								
<b>KeyID</b>	xs:token	required								
		This KeyID references the ID of the field on the group that forms the key to link to this section								
Source	<pre> &lt;xs:element name="SelectGroupKey"&gt;   &lt;xs:complexType&gt;     &lt;xs:all&gt;       &lt;xs:element ref="elSelect"/&gt;     &lt;/xs:all&gt;     &lt;xs:attribute name="KeyID" type="xs:token" use="required"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;This KeyID references the ID of the field on the group that forms the key to link to this section&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:attribute&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;   </pre>									

## Element elSelectGroup / SelectGroup

Namespace	No namespace
Diagram	<pre> classDiagram     class SelectGroup {         1..infinity --&gt; BaseElements         elLabel Type kw_elLabel_type         elField Type kw_elField_type         elSelect Type kw_elSelect_type         elCalcField Type kw_elCalcField_type         Repeat Type kw_Repeat_type         elGroup Type kw_elGroup_type     }     Note over SelectGroup: Root element for the structure that defines the repeat or table structure   </pre>
Properties	content: complex
Model	(elLabel   elField   elSelect   elCalcField   Repeat   elGroup)
Children	Repeat, elCalcField, elField, elGroup, elLabel, elSelect
Instance	<pre> &lt;SelectGroup&gt;   &lt;elLabel ID="" Type=""&gt;{1,1}&lt;/elLabel&gt;   &lt;elField ID="" Type=""&gt;{1,1}&lt;/elField&gt;   &lt;elSelect ID="" Type=""&gt;{1,1}&lt;/elSelect&gt;   &lt;elCalcField ID="" Type=""&gt;{1,1}&lt;/elCalcField&gt;   &lt;Repeat ID="" Type=""&gt;{1,1}&lt;/Repeat&gt;   </pre>

	<pre>&lt;elGroup ID=""&gt;{1,1}&lt;/elGroup&gt; &lt;/SelectGroup&gt;</pre>
Source	<pre>&lt;xs:element name="SelectGroup"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence maxOccurs="unbounded"&gt;       &lt;xs:group ref="BaseElements" /&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>

## Element TableName

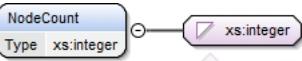
Namespace	No namespace
Diagram	<p>The diagram shows a class named 'TableName' with a multiplicity circle at its end, indicating it is derived from another type. A line connects 'TableName' to a box labeled 'xs:NCName'. A callout bubble provides the following information: 'Built-in derived type. NCName represents XML "non-colonized" Names. The base type of NCName is Name.'</p>
Type	xs:NCName
Properties	content: simple
Source	<pre>&lt;xs:element name="TableName" type="xs:NCName" /&gt;</pre>

## Element Signature

Namespace	No namespace																								
Diagram	<p>The diagram shows a class named 'Signature' with a multiplicity circle at its end, indicating it is a restriction of 'xs:NCName'. A line connects 'Signature' to a box labeled 'restricts: xs:NCName'. A plus sign (+) is at the end of the line, indicating inheritance.</p>																								
Type	restriction of xs:NCName																								
Properties	content: simple																								
Facets	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">ColumnTotal</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">DateDiff</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">GetLocation</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">GetLatitude</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">GetLongitude</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">RowTotal</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">ElementValue</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">Sum</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">Product</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">Max</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">Average</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">SelectGroupKey</td> </tr> </table>	enumeration	ColumnTotal	enumeration	DateDiff	enumeration	GetLocation	enumeration	GetLatitude	enumeration	GetLongitude	enumeration	RowTotal	enumeration	ElementValue	enumeration	Sum	enumeration	Product	enumeration	Max	enumeration	Average	enumeration	SelectGroupKey
enumeration	ColumnTotal																								
enumeration	DateDiff																								
enumeration	GetLocation																								
enumeration	GetLatitude																								
enumeration	GetLongitude																								
enumeration	RowTotal																								
enumeration	ElementValue																								
enumeration	Sum																								
enumeration	Product																								
enumeration	Max																								
enumeration	Average																								
enumeration	SelectGroupKey																								
Source	<pre>&lt;xs:element name="Signature"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="ColumnTotal" /&gt;       &lt;xs:enumeration value="DateDiff" /&gt;       &lt;xs:enumeration value="GetLocation" /&gt;       &lt;xs:enumeration value="GetLatitude" /&gt;       &lt;xs:enumeration value="GetLongitude" /&gt;       &lt;xs:enumeration value="RowTotal" /&gt;       &lt;xs:enumeration value="ElementValue" /&gt;       &lt;xs:enumeration value="Sum" /&gt;       &lt;xs:enumeration value="Product" /&gt;       &lt;xs:enumeration value="Max" /&gt;       &lt;xs:enumeration value="Average" /&gt;       &lt;xs:enumeration value="SelectGroupKey" /&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:element&gt;</pre>																								

## Element NodeCount

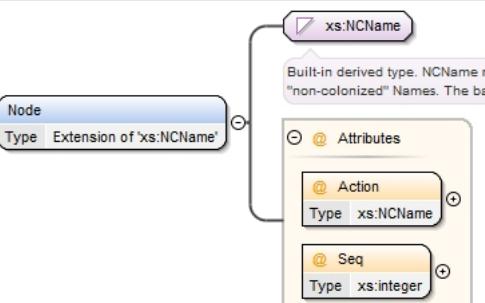
Namespace	No namespace
-----------	--------------

Diagram	 Built-in derived type. The integer datatype is derived from decimal by fixing the value of fractionDigits to be 0. This...
Type	xs:integer
Properties	content: simple
Source	<xs:element name="NodeCount" type="xs:integer"/>

## Element Nodes

Namespace	No namespace
Diagram	
Properties	content: complex
Model	Node+
Children	Node
Instance	<Nodes> <Node Action="" Seq="" />{1,unbounded}</Node> </Nodes>
Source	<xs:element name="Nodes"> <xs:complexType> <xs:sequence> <xs:element maxOccurs="unbounded" ref="Node" /> </xs:sequence> </xs:complexType> </xs:element>

## Element Node

Namespace	No namespace												
Diagram	 Built-in derived type. NCName represents XML "non-colonized" Names. The base type of NCName is Name.												
Type	extension of xs:NCName												
Properties	content: complex												
Used by	Element Nodes												
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">QName</th> <th style="width: 25%;">Type</th> <th style="width: 25%;">Use</th> <th style="width: 25%;"> </th> </tr> </thead> <tbody> <tr> <td>Action</td> <td>xs:NCName</td> <td>required</td> <td></td> </tr> <tr> <td>Seq</td> <td>xs:integer</td> <td>required</td> <td></td> </tr> </tbody> </table>	QName	Type	Use		Action	xs:NCName	required		Seq	xs:integer	required	
QName	Type	Use											
Action	xs:NCName	required											
Seq	xs:integer	required											
Source	<xs:element name="Node"> <xs:complexType> <xs:simpleContent> <xs:extension base="xs:NCName"> <xs:attribute name="Action" use="required" type="xs:NCName" /> <xs:attribute name="Seq" use="required" type="xs:integer" /> </xs:extension> </xs:simpleContent> </xs:complexType> </xs:element>												

## Element FieldName

Namespace	No namespace
-----------	--------------

Diagram	
Type	xs:string
Properties	content: simple
Source	<xs:element name="FieldName" type="xs:string" />

## Element NumberColumns

Namespace	No namespace
Diagram	
Type	xs:integer
Properties	content: simple
Source	<xs:element name="NumberColumns" type="xs:integer" />

## Element CalcProcessType / ValidDate

Namespace	No namespace
Diagram	
Type	fnProcessPeriodValidDate
Properties	content: complex
Model	Calculation , Number , Unit
Children	Calculation, Number, Unit
Instance	<pre>&lt;ValidDate&gt;   &lt;Calculation&gt;{1,1}&lt;/Calculation&gt;   &lt;Number&gt;{1,1}&lt;/Number&gt;   &lt;Unit&gt;{1,1}&lt;/Unit&gt; &lt;/ValidDate&gt;</pre>
Source	<xs:element name="ValidDate" type="fnProcessPeriodValidDate" />

## Element fnProcessPeriodValidDate / Calculation

Namespace	No namespace
Annotations	<p>This element specifies which calculation should be performed</p> <ul style="list-style-type: none"> <li>- Sum: Sum all numeric values</li> <li>- Max: Return the maximum numeric value</li> <li>- Min: Return the minimum numeric value</li> <li>- Count: Return the number of rows in the column</li> <li>- CountUnique: Return the number of Unique values in the column</li> </ul>

Diagram	<p>This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum...</p>				
Type	restriction of xs:NCName				
Properties	content: simple				
Facets	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">enumeration</td><td style="width: 85%;">AfterFirst</td></tr> <tr> <td>enumeration</td><td>BeforeLast</td></tr> </table>	enumeration	AfterFirst	enumeration	BeforeLast
enumeration	AfterFirst				
enumeration	BeforeLast				
Source	<pre>&lt;xss:element name="Calculation"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum numeric value - Min: Return the minimum numeric value - Count: Return the number of rows in the column - CountUnique: Return the number of Unique values in the column&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt;   &lt;xss:simpleType&gt;     &lt;xss:restriction base="xs:NCName"&gt;       &lt;xss:enumeration value="AfterFirst"/&gt;       &lt;xss:enumeration value="BeforeLast"/&gt;     &lt;/xss:restriction&gt;   &lt;/xss:simpleType&gt; &lt;/xss:element&gt;</pre>				

### Element fnProcessPeriodValidDate / Number

Namespace	No namespace				
Annotations	This is the element in the repeat table that is the basis for the calculation.				
Diagram	<p>This is the element in the repeat table that is the basis for the calculation.</p> <p>Built-in derived type. The int datatype is derived from long by setting the value of maxInclusive to be 2147483647 and...</p>				
Type	xs:int				
Properties	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">content:</td><td style="width: 85%;">simple</td></tr> <tr> <td>default:</td><td>0</td></tr> </table>	content:	simple	default:	0
content:	simple				
default:	0				
Source	<pre>&lt;xss:element default="0" name="Number" type="xs:int"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;This is the element in the repeat table that is the basis for the calculation.&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt; &lt;/xss:element&gt;</pre>				

### Element fnProcessPeriodValidDate / Unit

Namespace	No namespace										
Diagram											
Type	restriction of xs:NCName										
Properties	content: simple										
Facets	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">enumeration</td><td style="width: 85%;">Days</td></tr> <tr> <td>enumeration</td><td>WorkingDays</td></tr> <tr> <td>enumeration</td><td>WeekDays</td></tr> <tr> <td>enumeration</td><td>Weeks</td></tr> <tr> <td>enumeration</td><td>Months</td></tr> </table>	enumeration	Days	enumeration	WorkingDays	enumeration	WeekDays	enumeration	Weeks	enumeration	Months
enumeration	Days										
enumeration	WorkingDays										
enumeration	WeekDays										
enumeration	Weeks										
enumeration	Months										
Source	<pre>&lt;xss:element name="Unit"&gt;   &lt;xss:simpleType&gt;     &lt;xss:restriction base="xs:NCName"&gt;</pre>										

```

<xs:enumeration value="Days" />
<xs:enumeration value="WorkingDays" />
<xs:enumeration value="WeekDays" />
<xs:enumeration value="Weeks" />
<xs:enumeration value="Months" />
</xs:restriction>
</xs:simpleType>
</xs:element>

```

### **Element CalcProcessType / DueDate**

Namespace	No namespace
Diagram	
Type	fnProcessPeriodValidDate
Properties	content: complex
Model	Calculation , Number , Unit
Children	Calculation, Number, Unit
Instance	<DueDate>   <Calculation>{1,1}</Calculation>   <Number>{1,1}</Number>   <Unit>{1,1}</Unit> </DueDate>
Source	<xs:element name="DueDate" type="fnProcessPeriodValidDate"/>

### **Element fnElementValue / Calculation**

Namespace	No namespace				
Annotations	This element specifies how to deal with various sequence numbers when accessing the element value: LatestSeq: - Checks what the latest sequence number for the ac is, and uses that sequence number SumAllSeq: - Calculates the value accross all sequence numbers				
Diagram					
Type	restriction of xs:NCName				
Properties	content: simple				
Facets	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">LatestSeq</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">SumAllSeq</td> </tr> </table>	enumeration	LatestSeq	enumeration	SumAllSeq
enumeration	LatestSeq				
enumeration	SumAllSeq				
Source	<xs:element name="Calculation">   <xs:annotation>     <xs:documentation>This element specifies how to deal with various sequence numbers when accessing the element value: LatestSeq: - Checks what the latest sequence number for the ac is, and uses that sequence number SumAllSeq: - Calculates the value accross all sequence numbers</xs:documentation>   </xs:annotation>   <xs:simpleType>     <xs:restriction base="xs:NCName">       <xs:enumeration value="LatestSeq"/>       <xs:enumeration value="SumAllSeq"/>     </xs:restriction>   </xs:simpleType> </xs:element>				

```
</xs:restriction>
</xs:simpleType>
</xs:element>
```

## Element fnElementValue / Element

Namespace	No namespace																					
Annotations	This is the element that contains the first date (Earliest date)																					
Diagram	<p>The diagram illustrates the structure of the KwantuElement_type. It starts with a box labeled "Element" (Type: KwantuElement_type) which points to a larger box for "KwantuElement_type". Inside this box, the "Base Type" is listed as "xs:string". Below this, the "xs:string" type is described as a built-in primitive type representing character strings in XML. The "Attributes" section contains several items:</p> <ul style="list-style-type: none"> <li>"@ Seq" (Type: xs:int): Described as a sequence number for multiple elements, requiring renaming to "Seq".</li> <li>"@ Id" (Type: xs:int): Described as a sequence number for multiple elements, requiring renaming to "Seq".</li> <li>"@ ID":</li> <li>"@ Component" (Type: xs:NCName):</li> <li>"@ Element" (Type: xs:NCName):</li> <li>"@ SubElement" (Type: xs:NCName):</li> <li>"@ acSequenceRule" (Type: Restriction of 'xs:NCName'): Contains a note about being a sequence rule.</li> <li>"@ RepeatKeyRule" (Type: Restriction of 'xs:NCName'): Contains a note about being a repeat key rule.</li> <li>"@ Scope" (Type: kw_ElementScope_type):</li> <li>"@ DataType" (Type: KwantuFormDataTypes):</li> <li>"@ Xpath" (Type: kw_Xpath_type):</li> </ul> <p>A note at the bottom states: "This is the base type for all places where an element is referenced".</p>																					
Type	KwantuElement_type																					
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	complex	maxOccurs:	1																	
content:	complex																					
maxOccurs:	1																					
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>Component</td> <td>xs:NCName</td> <td>optional</td> </tr> <tr> <td>DataType</td> <td>KwantuFormDataTypes</td> <td>required</td> </tr> <tr> <td>Element</td> <td>xs:NCName</td> <td>optional</td> </tr> <tr> <td>ID</td> <td></td> <td>optional</td> </tr> <tr> <td>Id</td> <td>xs:int</td> <td>optional</td> </tr> <tr> <td></td> <td colspan="2">This is used as a sequence number to access multiple elements. Must be renamed to Seq.</td></tr> </tbody> </table>	QName	Type	Use	Component	xs:NCName	optional	DataType	KwantuFormDataTypes	required	Element	xs:NCName	optional	ID		optional	Id	xs:int	optional		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
QName	Type	Use																				
Component	xs:NCName	optional																				
DataType	KwantuFormDataTypes	required																				
Element	xs:NCName	optional																				
ID		optional																				
Id	xs:int	optional																				
	This is used as a sequence number to access multiple elements. Must be renamed to Seq.																					

	<b>QName</b>	<b>Type</b>	<b>Use</b>	
	<b>RepeatKeyRule</b>	restriction of xs:NCName	optional	
	<b>Scope</b>	kw_ElementScope_type	optional	
	<b>Seq</b>	xs:int	optional	
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.		
	<b>SubElement</b>	xs:NCName	optional	
	<b>Xpath</b>	kw_Xpath_type	optional	
	<b>acSequenceRule</b>	restriction of xs:NCName	required	
<b>Source</b>	<pre>&lt;xs:element maxOccurs="1" name="Element" type="KwantuElement_type"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This is the element that contains the first date (Earliest date)&lt;/   xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>			

### Element fnProcessPeriodDueDate / CalcDueDate

Namespace	No namespace								
Annotations	<p>This element specifies which calculation should be performed</p> <ul style="list-style-type: none"> <li>- Sum: Sum all numeric values</li> <li>- Max: Return the maximum numeric value</li> <li>- Min: Return the minimum numeric value</li> <li>- Count: Return the number of rows in the column</li> <li>- CountUnique: Return the number of Unique values in the column</li> </ul>								
Diagram									
Properties	content: complex								
Model	DueDate								
Children	DueDate								
Instance	<pre>&lt;CalcDueDate Type=""&gt;   &lt;DueDate Seq=""&gt;{1,1}&lt;/DueDate&gt; &lt;/CalcDueDate&gt;</pre>								
Attributes	<table border="1"> <thead> <tr> <th><b>QName</b></th> <th><b>Type</b></th> <th><b>Use</b></th> <th></th> </tr> </thead> <tbody> <tr> <td><b>Type</b></td> <td>restriction of xs:NCName</td> <td>optional</td> <td></td> </tr> </tbody> </table>	<b>QName</b>	<b>Type</b>	<b>Use</b>		<b>Type</b>	restriction of xs:NCName	optional	
<b>QName</b>	<b>Type</b>	<b>Use</b>							
<b>Type</b>	restriction of xs:NCName	optional							
Source	<pre>&lt;xs:element name="CalcDueDate"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum numeric value - Min: Return the minimum numeric value - Count: Return the number of rows in the column - CountUnique: Return the number of Unique values in the column&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="DueDate"&gt;         &lt;xs:complexType&gt;           &lt;xs:sequence&gt;             &lt;xs:element name="Calculation"&gt;               &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum numeric value - Min: Return the minimum numeric value - Count: Return the number of rows in the column - CountUnique: Return the number of Unique values in the column&lt;/xs:documentation&gt;               &lt;/xs:annotation&gt;               &lt;xs:simpleType&gt;                 &lt;xs:restriction base="xs:NCName"&gt;                   &lt;xs:enumeration value="After"/&gt;                   &lt;xs:enumeration value="Before"/&gt;                 &lt;/xs:restriction&gt;               &lt;/xs:simpleType&gt;             &lt;/xs:element&gt;           &lt;/xs:sequence&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>								

```

        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element default="0" name="Number" type="xs:int">
    <xs:annotation>
        <xs:documentation>This is the element in the repeat table that is the basis for the calculation.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="Unit">
    <xs:simpleType>
        <xs:restriction base="xs:NCName">
            <xs:enumeration value="Days"/>
            <xs:enumeration value="WorkingDays"/>
            <xs:enumeration value="WeekDays"/>
            <xs:enumeration value="Weeks"/>
            <xs:enumeration value="Months"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="Reference">
    <xs:simpleType>
        <xs:restriction base="xs:NCName">
            <xs:enumeration value="CurrentPeriodStart"/>
            <xs:enumeration value="PreviousPeriodEnd"/>
            <xs:enumeration value="CurrentPeriodEnd"/>
            <xs:enumeration value="ValidDate"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
</xs:sequence>
<xs:attribute name="Seq">
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="Type">
    <xs:simpleType>
        <xs:restriction base="xs:NCName">
            <xs:enumeration value="Earliest"/>
            <xs:enumeration value="Latest"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>

```

### Element fnProcessPeriodDueDate / CalcDueDate / DueDate

Namespace	No namespace
Diagram	<pre> classDiagram     class DueDate {         @ Seq         Calculation         Number         Unit         Reference     }     class Calculation {         Type: Restriction of xs:NCName     }     class Number {         Type: xs:int         Default: 0     }     class Unit {         Type: Restriction of xs:NCName     }     class Reference {         Type: Restriction of xs:NCName     } </pre>
Properties	content: complex
Model	Calculation , Number , Unit , Reference
Children	Calculation, Number, Reference, Unit
Instance	<DueDate Seq="">

	<pre>&lt;Calculation&gt;{1,1}&lt;/Calculation&gt; &lt;Number&gt;{1,1}&lt;/Number&gt; &lt;Unit&gt;{1,1}&lt;/Unit&gt; &lt;Reference&gt;{1,1}&lt;/Reference&gt; &lt;/DueDate&gt;</pre>		
Attributes	QName	Type	Use
	Seq		optional
Source	<pre>&lt;xs:element name="DueDate"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="Calculation"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum numeric value - Min: Return the minimum numeric value - Count: Return the number of rows in the column - CountUnique: Return the number of Unique values in the column&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;         &lt;xs:simpleType&gt;           &lt;xs:restriction base="xs:NCName"&gt;             &lt;xs:enumeration value="After"/&gt;             &lt;xs:enumeration value="Before"/&gt;           &lt;/xs:restriction&gt;         &lt;/xs:simpleType&gt;       &lt;/xs:element&gt;       &lt;xs:element default="0" name="Number" type="xs:int"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;This is the element in the repeat table that is the basis for the calculation.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="Unit"&gt;         &lt;xs:simpleType&gt;           &lt;xs:restriction base="xs:NCName"&gt;             &lt;xs:enumeration value="Days"/&gt;             &lt;xs:enumeration value="WorkingDays"/&gt;             &lt;xs:enumeration value="WeekDays"/&gt;             &lt;xs:enumeration value="Weeks"/&gt;             &lt;xs:enumeration value="Months"/&gt;           &lt;/xs:restriction&gt;         &lt;/xs:simpleType&gt;       &lt;/xs:element&gt;       &lt;xs:element name="Reference"&gt;         &lt;xs:simpleType&gt;           &lt;xs:restriction base="xs:NCName"&gt;             &lt;xs:enumeration value="CurrentPeriodStart"/&gt;             &lt;xs:enumeration value="PreviousPeriodEnd"/&gt;             &lt;xs:enumeration value="CurrentPeriodEnd"/&gt;             &lt;xs:enumeration value="ValidDate"/&gt;           &lt;/xs:restriction&gt;         &lt;/xs:simpleType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute name="Seq"/&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>		

### Element fnProcessPeriodDueDate / CalcDueDate / DueDate / Calculation

Namespace	No namespace
Annotations	<p>This element specifies which calculation should be performed</p> <ul style="list-style-type: none"> <li>- Sum: Sum all numeric values</li> <li>- Max: Return the maximum numeric value</li> <li>- Min: Return the minimum numeric value</li> <li>- Count: Return the number of rows in the column</li> <li>- CountUnique: Return the number of Unique values in the column</li> </ul>
Diagram	<pre> classDiagram     class Calculation {         &lt;&lt;This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum...&gt;&gt;         &lt;&lt;restriction of xs:NCName&gt;&gt;     } </pre>
Type	restriction of xs:NCName
Properties	content: simple
Facets	enumeration After

	enumeration	Before
Source	<pre>&lt;xs:element name="Calculation"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum numeric value - Min: Return the minimum numeric value - Count: Return the number of rows in the column - CountUnique: Return the number of Unique values in the column&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="After"/&gt;       &lt;xs:enumeration value="Before"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:element&gt;</pre>	

#### Element fnProcessPeriodDueDate / CalcDueDate / DueDate / Number

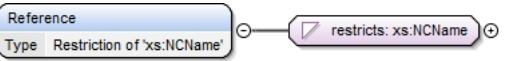
Namespace	No namespace				
Annotations	This is the element in the repeat table that is the basis for the calculation.				
Diagram	<p>This is the element in the repeat table that is the basis for the calculation.</p> <p>Built-in derived type. The int datatype is derived from long by setting the value of maxInclusive to be 2147483647 and...</p>				
Type	xs:int				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>default:</td> <td>0</td> </tr> </table>	content:	simple	default:	0
content:	simple				
default:	0				
Source	<pre>&lt;xs:element default="0" name="Number" type="xs:int"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This is the element in the repeat table that is the basis for the calculation.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>				

#### Element fnProcessPeriodDueDate / CalcDueDate / DueDate / Unit

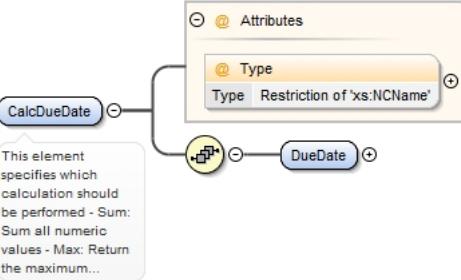
Namespace	No namespace										
Diagram	<p>Unit</p> <p>Type Restriction of xs:NCName</p> <p>restricts: xs:NCName</p>										
Type	restriction of xs:NCName										
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> </table>	content:	simple								
content:	simple										
Facets	<table border="1"> <tr> <td>enumeration</td> <td>Days</td> </tr> <tr> <td>enumeration</td> <td>WorkingDays</td> </tr> <tr> <td>enumeration</td> <td>WeekDays</td> </tr> <tr> <td>enumeration</td> <td>Weeks</td> </tr> <tr> <td>enumeration</td> <td>Months</td> </tr> </table>	enumeration	Days	enumeration	WorkingDays	enumeration	WeekDays	enumeration	Weeks	enumeration	Months
enumeration	Days										
enumeration	WorkingDays										
enumeration	WeekDays										
enumeration	Weeks										
enumeration	Months										
Source	<pre>&lt;xs:element name="Unit"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="Days"/&gt;       &lt;xs:enumeration value="WorkingDays"/&gt;       &lt;xs:enumeration value="WeekDays"/&gt;       &lt;xs:enumeration value="Weeks"/&gt;       &lt;xs:enumeration value="Months"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:element&gt;</pre>										

#### Element fnProcessPeriodDueDate / CalcDueDate / DueDate / Reference

Namespace	No namespace
-----------	--------------

Diagram	
Type	restriction of xs:NCName
Properties	content: simple
Facets	enumeration CurrentPeriodStart
	enumeration PreviousPeriodEnd
	enumeration CurrentPeriodEnd
	enumeration ValidDate
Source	<pre>&lt;xs:element name="Reference"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="CurrentPeriodStart"/&gt;       &lt;xs:enumeration value="PreviousPeriodEnd"/&gt;       &lt;xs:enumeration value="CurrentPeriodEnd"/&gt;       &lt;xs:enumeration value="ValidDate"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:element&gt;</pre>

### Element fnProcessScheduleDueDate / CalcDueDate

Namespace	No namespace		
Annotations	<p>This element specifies which calculation should be performed</p> <ul style="list-style-type: none"> <li>- Sum: Sum all numeric values</li> <li>- Max: Return the maximum numeric value</li> <li>- Min: Return the minimum numeric value</li> <li>- Count: Return the number of rows in the column</li> <li>- CountUnique: Return the number of Unique values in the column</li> </ul>		
Diagram	 <p>This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum numeric value - Min: Return the minimum numeric value - Count: Return the number of rows in the column - CountUnique: Return the number of Unique values in the column...</p>		
Properties	content: complex		
Model	DueDate		
Children	DueDate		
Instance	<pre>&lt;CalcDueDate Type=""&gt;   &lt;DueDate Seq=""&gt;{1,1}&lt;/DueDate&gt; &lt;/CalcDueDate&gt;</pre>		
Attributes	QName	Type	Use
	Type	restriction of xs:NCName	optional
Source	<pre>&lt;xs:element name="CalcDueDate"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum numeric value - Min: Return the minimum numeric value - Count: Return the number of rows in the column - CountUnique: Return the number of Unique values in the column&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="DueDate" type="xs:complexType"&gt;         &lt;xs:sequence&gt;           &lt;xs:element name="Calculation" type="xs:string"&gt;             &lt;xs:annotation&gt;               &lt;xs:documentation&gt;This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum numeric value - Min: Return the minimum numeric value - Count: Return the number of rows in the column - CountUnique: Return the number of Unique values in the column&lt;/xs:documentation&gt;             &lt;/xs:annotation&gt;           &lt;/xs:element&gt;         &lt;/xs:sequence&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>		

```

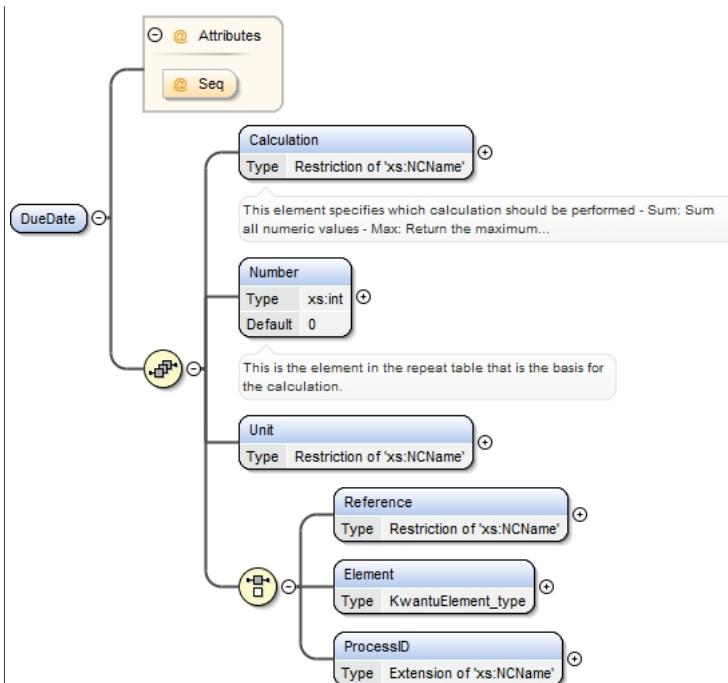
<xs:simpleType>
  <xs:restriction base="xs:NCName">
    <xs:enumeration value="After"/>
    <xs:enumeration value="Before"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element default="0" name="Number" type="xs:int">
  <xs:annotation>
    <xs:documentation>This is the element in the repeat table that is the basis for the calculation.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="Unit">
  <xs:simpleType>
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="Days"/>
      <xs:enumeration value="WorkingDays"/>
      <xs:enumeration value="WeekDays"/>
      <xs:enumeration value="Weeks"/>
      <xs:enumeration value="Months"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:choice>
  <xs:element name="Reference">
    <xs:simpleType>
      <xs:restriction base="xs:NCName">
        <xs:enumeration value="KwantuObjectCreate"/>
        <xs:enumeration value="ElementID"/>
        <xs:enumeration value="CurrentPeriodEnd"/>
        <xs:enumeration value="ValidDate"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="Element" type="KwantuElement_type"/>
  <xs:element name="ProcessID">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:NCName">
          <xs:attribute name="Aspect">
            <xs:simpleType>
              <xs:restriction base="xs:NCName">
                <xs:enumeration value="DueDate"/>
                <xs:enumeration value="ValidDate"/>
                <xs:enumeration value="ReceivedDate"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:choice>
    <xs:sequence>
      <xs:attribute name="Seq"/>
    </xs:sequence>
    <xs:attribute name="Type">
      <xs:simpleType>
        <xs:restriction base="xs:NCName">
          <xs:enumeration value="Max"/>
          <xs:enumeration value="Min"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:choice>
</xs:complexType>
</xs:element>

```

### **Element fnProcessScheduleDueDate / CalcDueDate / DueDate**

Namespace	No namespace
-----------	--------------

Diagram



Properties

content: complex

Model

Calculation , Number , Unit , (Reference | Element | ProcessID)

Children

Calculation, Element, Number, ProcessID, Reference, Unit

Instance

```

<DueDate Seq="">
    <Calculation>{1,1}</Calculation>
    <Number>{1,1}</Number>
    <Unit>{1,1}</Unit>
    <Reference>{1,1}</Reference>
    <Element acSequenceRule="" Component="" DataType="" Element="" Id="" ID="" RepeatKeyRule="" Scope="" Seq="" SubE...
    <ProcessID Aspect="">{1,1}</ProcessID>
</DueDate>

```

Attributes

QName	Type	Use	
Seq		optional	

Source

```

<xs:element name="DueDate">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Calculation">
                <xs:annotation>
                    <xs:documentation>This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum numeric value - Min: Return the minimum numeric value - Count: Return the number of rows in the column - CountUnique: Return the number of Unique values in the column</xs:documentation>
                </xs:annotation>
                <xs:simpleType>
                    <xs:restriction base="xs:NCName">
                        <xs:enumeration value="After"/>
                        <xs:enumeration value="Before"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element default="0" name="Number" type="xs:int">
                <xs:annotation>
                    <xs:documentation>This is the element in the repeat table that is the basis for the calculation.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="Unit">
                <xs:simpleType>
                    <xs:restriction base="xs:NCName">
                        <xs:enumeration value="Days"/>
                        <xs:enumeration value="WorkingDays"/>
                        <xs:enumeration value="WeekDays"/>
                        <xs:enumeration value="Weeks"/>
                        <xs:enumeration value="Months"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```

</xs:simpleType>
</xs:element>
<xs:choice>
  <xs:element name="Reference">
    <xs:simpleType>
      <xs:restriction base="xs:NCName">
        <xs:enumeration value="KwantuObjectCreate"/>
        <xs:enumeration value="ElementID"/>
        <xs:enumeration value="CurrentPeriodEnd"/>
        <xs:enumeration value="ValidDate"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="Element" type="KwantuElement_type"/>
  <xs:element name="ProcessID">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:NCName">
          <xs:attribute name="Aspect">
            <xs:simpleType>
              <xs:restriction base="xs:NCName">
                <xs:enumeration value="DueDate"/>
                <xs:enumeration value="ValidDate"/>
                <xs:enumeration value="ReceivedDate"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:choice>
    <xs:sequence>
      <xs:attribute name="Seq"/>
    </xs:sequence>
  </xs:choice>
</xs:sequence>
<xs:attribute name="Seq"/>
</xs:complexType>
</xs:element>

```

### **Element fnProcessScheduleDueDate / CalcDueDate / DueDate / Calculation**

Namespace	No namespace				
Annotations	<p>This element specifies which calculation should be performed</p> <ul style="list-style-type: none"> <li>- Sum: Sum all numeric values</li> <li>- Max: Return the maximum numeric value</li> <li>- Min: Return the minimum numeric value</li> <li>- Count: Return the number of rows in the column</li> <li>- CountUnique: Return the number of Unique values in the column</li> </ul>				
Diagram	<p>This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum...</p>				
Type	restriction of xs:NCName				
Properties	content: simple				
Facets	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">After</td> </tr> <tr> <td style="padding: 2px;">enumeration</td> <td style="padding: 2px;">Before</td> </tr> </table>	enumeration	After	enumeration	Before
enumeration	After				
enumeration	Before				
Source	<pre> &lt;xs:element name="Calculation"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum numeric value - Min: Return the minimum numeric value - Count: Return the number of rows in the column - CountUnique: Return the number of Unique values in the column&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="After"/&gt;       &lt;xs:enumeration value="Before"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:element&gt; </pre>				

### **Element fnProcessScheduleDueDate / CalcDueDate / DueDate / Number**

Namespace	No namespace
-----------	--------------

Annotations	This is the element in the repeat table that is the basis for the calculation.
Diagram	
Type	xs:int
Properties	content: simple default: 0
Source	<pre>&lt;xs:element default="0" name="Number" type="xs:int"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This is the element in the repeat table that is the basis for the calculation.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

### Element fnProcessScheduleDueDate / CalcDueDate / DueDate / Unit

Namespace	No namespace
Diagram	
Type	restriction of xs:NCName
Properties	content: simple
Facets	enumeration Days enumeration WorkingDays enumeration WeekDays enumeration Weeks enumeration Months
Source	<pre>&lt;xs:element name="Unit"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="Days"/&gt;       &lt;xs:enumeration value="WorkingDays"/&gt;       &lt;xs:enumeration value="WeekDays"/&gt;       &lt;xs:enumeration value="Weeks"/&gt;       &lt;xs:enumeration value="Months"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:element&gt;</pre>

### Element fnProcessScheduleDueDate / CalcDueDate / DueDate / Reference

Namespace	No namespace
Diagram	
Type	restriction of xs:NCName
Properties	content: simple
Facets	enumeration KwantuObjectCreate enumeration ElementID enumeration CurrentPeriodEnd enumeration ValidDate
Source	<pre>&lt;xs:element name="Reference"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="KwantuObjectCreate"/&gt;       &lt;xs:enumeration value="ElementID"/&gt;       &lt;xs:enumeration value="CurrentPeriodEnd"/&gt;       &lt;xs:enumeration value="ValidDate"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:element&gt;</pre>

```
</xs:simpleType>
</xs:element>
```

### Element fnProcessScheduleDueDate / CalcDueDate / DueDate / Element

Namespace	No namespace																														
Diagram	<p>The diagram illustrates the schema type <code>KwantuElement_type</code>. It inherits from the built-in primitive type <code>xs:string</code>. The type is annotated with several attributes:</p> <ul style="list-style-type: none"> <li><code>@ Seq</code>: Type <code>xs:int</code>. Description: This is used as a sequence number to access multiple elements. Must be renamed to Seq.</li> <li><code>@ Id</code>: Type <code>xs:int</code>. Description: This is used as a sequence number to access multiple elements. Must be renamed to Seq.</li> <li><code>@ ID</code></li> <li><code>@ Component</code>: Type <code>xs:NCName</code></li> <li><code>@ Element</code>: Type <code>xs:NCName</code></li> <li><code>@ SubElement</code>: Type <code>xs:NCName</code></li> <li><code>@ acSequenceRule</code>: Type <code>Restriction of 'xs:NCName'</code></li> <li><code>@ RepeatKeyRule</code>: Type <code>Restriction of 'xs:NCName'</code></li> <li><code>@ Scope</code>: Type <code>kw_ElementScope_type</code></li> <li><code>@ DataType</code>: Type <code>KwantuFormDataTypes</code></li> <li><code>@ Xpath</code>: Type <code>kw_Xpath_type</code></li> </ul> <p>A note at the bottom states: "This is the base type for all places where an element is referenced".</p>																														
Type	<code>KwantuElement_type</code>																														
Properties	content: complex																														
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><b>Component</b></td> <td><code>xs:NCName</code></td> <td>optional</td> </tr> <tr> <td><b>DataType</b></td> <td><code>KwantuFormDataTypes</code></td> <td>required</td> </tr> <tr> <td><b>Element</b></td> <td><code>xs:NCName</code></td> <td>optional</td> </tr> <tr> <td><b>ID</b></td> <td></td> <td>optional</td> </tr> <tr> <td><b>Id</b></td> <td><code>xs:int</code></td> <td>optional</td> </tr> <tr> <td></td> <td colspan="2">This is used as a sequence number to access multiple elements. Must be renamed to Seq.</td></tr> <tr> <td><b>RepeatKeyRule</b></td> <td><code>restriction of xs:NCName</code></td> <td>optional</td> </tr> <tr> <td><b>Scope</b></td> <td><code>kw_ElementScope_type</code></td> <td>optional</td> </tr> <tr> <td><b>Seq</b></td> <td><code>xs:int</code></td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Use	<b>Component</b>	<code>xs:NCName</code>	optional	<b>DataType</b>	<code>KwantuFormDataTypes</code>	required	<b>Element</b>	<code>xs:NCName</code>	optional	<b>ID</b>		optional	<b>Id</b>	<code>xs:int</code>	optional		This is used as a sequence number to access multiple elements. Must be renamed to Seq.		<b>RepeatKeyRule</b>	<code>restriction of xs:NCName</code>	optional	<b>Scope</b>	<code>kw_ElementScope_type</code>	optional	<b>Seq</b>	<code>xs:int</code>	optional
QName	Type	Use																													
<b>Component</b>	<code>xs:NCName</code>	optional																													
<b>DataType</b>	<code>KwantuFormDataTypes</code>	required																													
<b>Element</b>	<code>xs:NCName</code>	optional																													
<b>ID</b>		optional																													
<b>Id</b>	<code>xs:int</code>	optional																													
	This is used as a sequence number to access multiple elements. Must be renamed to Seq.																														
<b>RepeatKeyRule</b>	<code>restriction of xs:NCName</code>	optional																													
<b>Scope</b>	<code>kw_ElementScope_type</code>	optional																													
<b>Seq</b>	<code>xs:int</code>	optional																													

QName	Type	Use	
This is used as a sequence number to access multiple elements. Must be renamed to Seq.			
<b>SubElement</b>	xs:NCName	optional	
<b>Xpath</b>	kw_Xpath_type	optional	
<b>acSequenceRule</b>	restriction of xs:NCName	required	
Source	<xs:element name="Element" type="KwantuElement_type"/>		

### Element fnProcessScheduleDueDate / CalcDueDate / DueDate / ProcessID

Namespace	No namespace								
Diagram	<pre> classDiagram     class ProcessID {         &lt;&lt;Extension of 'xs:NCName'&gt;&gt;         &lt;&lt;@Aspect&lt;/Aspect&gt;&gt;         &lt;&lt;@lang&lt;/lang&gt;&gt;     }     ProcessID &lt; -- xs:NCName   </pre>								
Type	extension of xs:NCName								
Properties	content: complex								
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> <th></th> </tr> </thead> <tbody> <tr> <td><b>Aspect</b></td> <td>restriction of xs:NCName</td> <td>optional</td> <td></td></tr> </tbody> </table>	QName	Type	Use		<b>Aspect</b>	restriction of xs:NCName	optional	
QName	Type	Use							
<b>Aspect</b>	restriction of xs:NCName	optional							
Source	<pre> &lt;xs:element name="ProcessID"&gt;   &lt;xs:complexType&gt;     &lt;xs:simpleContent&gt;       &lt;xs:extension base="xs:NCName"&gt;         &lt;xs:attribute name="Aspect"&gt;           &lt;xs:simpleType&gt;             &lt;xs:restriction base="xs:NCName"&gt;               &lt;xs:enumeration value="DueDate"/&gt;               &lt;xs:enumeration value="ValidDate"/&gt;               &lt;xs:enumeration value="ReceivedDate"/&gt;             &lt;/xs:restriction&gt;           &lt;/xs:simpleType&gt;         &lt;/xs:attribute&gt;       &lt;/xs:extension&gt;     &lt;/xs:simpleContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;   </pre>								

### Element kw\_Label\_type / Int

Namespace	No namespace								
Diagram	<pre> classDiagram     class Int {         &lt;&lt;Extension of 'xs:string'&gt;&gt;         &lt;&lt;@lang&lt;/lang&gt;&gt;     }     Int &lt; -- xs:string   </pre>								
Type	extension of xs:string								
Properties	<p>content: complex</p> <p>maxOccurs: unbounded</p>								
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> <th></th> </tr> </thead> <tbody> <tr> <td><b>lang</b></td> <td></td> <td>optional</td> <td></td></tr> </tbody> </table>	QName	Type	Use		<b>lang</b>		optional	
QName	Type	Use							
<b>lang</b>		optional							
Source	<pre> &lt;xs:element maxOccurs="unbounded" name="Int"&gt;   &lt;xs:complexType&gt;     &lt;xs:simpleContent&gt;       &lt;xs:extension base="xs:string"&gt;         &lt;xs:attribute name="lang"/&gt;       &lt;/xs:extension&gt;     &lt;/xs:simpleContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;   </pre>								

```

</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>

```

## Complex Type(s)

### Complex Type ReportDefaultColumnType

Namespace	No namespace						
Diagram	<pre> classDiagram     class ReportDefaultColumnType {         &lt;&lt;Mixed true&gt;&gt;         &lt;&lt;Attributes&gt;&gt;         @Source Type Source_type         Fieldname Type xs:NName         Fieldtype Type ColumnDefReportFieldtype         Fieldlength Type xs:decimal Default 1         Nullable Type Restriction of 'xs:string'     } </pre>						
Properties	mixed: true						
Used by	Elements ReportDefaults/Tables/Table/Attributes/Attribute, ReportDefaults/Tables/Table/Keys/Key						
Model	ALL(Fieldname Fieldtype Fieldlength Nullable{0,1})						
Children	Fieldlength, Fieldname, Fieldtype, Nullable						
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>Source</td> <td>Source_type</td> <td>required</td> </tr> </tbody> </table>	QName	Type	Use	Source	Source_type	required
QName	Type	Use					
Source	Source_type	required					
Source	<pre> &lt;xs:complexType mixed="true" name="ReportDefaultColumnType"&gt;     &lt;xs:all&gt;         &lt;xs:element name="Fieldname" type="xs:NName"/&gt;         &lt;xs:element name="Fieldtype" type="ColumnDefReportFieldtype"/&gt;         &lt;xs:element default="1" name="Fieldlength" type="xs:decimal"/&gt;         &lt;xs:element minOccurs="0" name="Nullable"&gt;             &lt;xs:simpleType&gt;                 &lt;xs:restriction base="xs:string"&gt;                     &lt;xs:enumeration value="Not Null"/&gt;                     &lt;xs:enumeration value="Nullable"/&gt;                 &lt;/xs:restriction&gt;             &lt;/xs:simpleType&gt;         &lt;/xs:element&gt;     &lt;/xs:all&gt;     &lt;xs:attribute name="Source" use="required" type="Source_type"/&gt; &lt;/xs:complexType&gt; </pre>						

### Complex Type kw\_UIVersions\_type

Namespace	No namespace
Diagram	<pre> sequenceDiagram     class kw_UIVersions_type     class UIVersion     kw_UIVersions_type --&gt; UIVersion : 1..∞ </pre>
Used by	Element UIVersions
Model	UIVersion+
Children	UIVersion
Source	<pre> &lt;xs:complexType name="kw_UIVersions_type"&gt;     &lt;xs:sequence&gt;         &lt;xs:element maxOccurs="unbounded" name="UIVersion"&gt;             &lt;xs:complexType&gt;                 &lt;xs:attribute name="Name" use="required"/&gt;             &lt;/xs:complexType&gt;         &lt;/xs:element&gt;     &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; </pre>

## Complex Type KwantuElement\_type

Namespace	No namespace																																		
Annotations	This is the base type for all places where an element is referenced																																		
Diagram																																			
Type	extension of xs:string																																		
Used by	Elements: Condition_type/Element, Condition_type/ReferenceElement, Default_type/ElementValue, Header/Dependencies/Dependency/Element, Kwantu_Filter_type/Element, Kwantu_Filter_type/Parameters/Element, RepeatDef/AddRows/AutoLoad/Component/Filter/ElementID, RepeatDef/AddRows/AutoLoad/Periods/StartElement/Element, RestType/Parameters/Parameter/Element, fnCalcFinYearDates/Element, fnColumnTotal/Element, fnDateDiff/Element, fnElementCalc/Element, fnElementValue/Element, fnElementsList/Element, fnHistoryTotal/Element, fnHistoryTotal/Filters/Filter/Parameters/Element, fnProcessScheduleDueDate/CalcDueDate/DueDate/Element, fnRowCalc/Element, kw_Condition_type/Element, kw_Condition_type/Element2, kw_elLabel_type/Display/Condition/Element, kw_elLabel_type/Display/Condition/Element2, kw_elLabel_type/Display/Element, kw_elSelect_type/EntryMode/Select/SharedData/Filter/FilterValue/Element																																		
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> <th></th> </tr> </thead> <tbody> <tr> <td>Component</td> <td>xs:NCName</td> <td>optional</td> <td></td> </tr> <tr> <td>DataType</td> <td>KwantuFormDataTypes</td> <td>required</td> <td></td> </tr> <tr> <td>Element</td> <td>xs:NCName</td> <td>optional</td> <td></td> </tr> <tr> <td>ID</td> <td></td> <td>optional</td> <td></td> </tr> <tr> <td>Id</td> <td>xs:int</td> <td>optional</td> <td></td> </tr> <tr> <td></td> <td colspan="2">This is used as a sequence number to access multiple elements. Must be renamed to Seq.</td> <td></td> </tr> <tr> <td>RepeatKeyRule</td> <td>restriction of xs:NCName</td> <td>optional</td> <td></td> </tr> </tbody> </table>			QName	Type	Use		Component	xs:NCName	optional		DataType	KwantuFormDataTypes	required		Element	xs:NCName	optional		ID		optional		Id	xs:int	optional			This is used as a sequence number to access multiple elements. Must be renamed to Seq.			RepeatKeyRule	restriction of xs:NCName	optional	
QName	Type	Use																																	
Component	xs:NCName	optional																																	
DataType	KwantuFormDataTypes	required																																	
Element	xs:NCName	optional																																	
ID		optional																																	
Id	xs:int	optional																																	
	This is used as a sequence number to access multiple elements. Must be renamed to Seq.																																		
RepeatKeyRule	restriction of xs:NCName	optional																																	

	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>Scope</b>	kw_ElementScope_type	optional
	<b>Seq</b>	xs:int	optional
		This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
	<b>SubElement</b>	xs:NCName	optional
	<b>Xpath</b>	kw_Xpath_type	optional
	<b>acSequenceRule</b>	restriction of xs:NCName	required
<b>Source</b>	<pre> &lt;xs:complexType name="KwantuElement_type"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This is the base type for all places where an element is referenced&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:simpleContent&gt;     &lt;xs:extension base="xs:string"&gt;       &lt;xs:attribute name="Seq" type="xs:int"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;This is used as a sequence number to access multiple elements. Must be renamed to Seq.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:attribute&gt;       &lt;xs:attribute name="Id" type="xs:int"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;This is used as a sequence number to access multiple elements. Must be renamed to Seq.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:attribute&gt;       &lt;xs:attribute name="ID"/&gt;       &lt;xs:attribute name="Component" type="xs:NCName"/&gt;       &lt;xs:attribute name="Element" type="xs:NCName"/&gt;       &lt;xs:attribute name="SubElement" type="xs:NCName"/&gt;       &lt;xs:attribute name="acSequenceRule" use="required"&gt;         &lt;xs:simpleType&gt;           &lt;xs:restriction base="xs:NCName"&gt;             &lt;xs:enumeration value="Current"/&gt;             &lt;xs:enumeration value="LatestAuthorized"/&gt;             &lt;xs:enumeration value="First"/&gt;             &lt;xs:enumeration value="LatestSeq"/&gt;             &lt;xs:enumeration value="FirsttoLatestSeq"/&gt;           &lt;/xs:restriction&gt;         &lt;/xs:simpleType&gt;       &lt;/xs:attribute&gt;       &lt;xs:attribute name="RepeatKeyRule"&gt;         &lt;xs:simpleType&gt;           &lt;xs:restriction base="xs:NCName"&gt;             &lt;xs:enumeration value="MatchRepeatKey"/&gt;           &lt;/xs:restriction&gt;         &lt;/xs:simpleType&gt;       &lt;/xs:attribute&gt;       &lt;xs:attribute name="Scope" type="kw_ElementScope_type"/&gt;       &lt;xs:attribute name="DataType" type="KwantuFormDataTypes" use="required"/&gt;       &lt;xs:attribute name="Xpath" type="kw_Xpath_type"/&gt;     &lt;/xs:extension&gt;   &lt;/xs:simpleContent&gt; &lt;/xs:complexType&gt;</pre>		

### Complex Type kw\_elLabel\_type

<b>Namespace</b>	No namespace
<b>Diagram</b>	<p>The diagram illustrates the structure of the <code>kw_elLabel_type</code>. It features a central class box labeled <code>kw_elLabel_type</code> with a hollow square icon. This class has three outgoing associations:</p> <ul style="list-style-type: none"> <li>An association to an attribute box labeled <code>@ ID</code> with a solid square icon, indicating a required relationship. The attribute box also contains the type <code>xs:ID</code>.</li> <li>An association to another attribute box labeled <code>@ Type</code> with a solid square icon, indicating a required relationship. The attribute box also contains the type <code>Restriction of 'xs:NCName'</code>.</li> <li>An association to a composite box labeled <code>UIVersionElements</code> with a hollow square icon, indicating an optional relationship. The composite box contains a slot named <code>Display</code> and another slot with a help icon.</li> </ul>

Used by	elLabel		
Model	UIVersionElements , Display , Help		
Children	Display, Help, UIVersionElements		
Attributes	QName	Type	Use
	ID	xs:ID	required
	Type	restriction of xs:NCName	required
Source	<pre> &lt;xs:complexType name="kw_elLabel_type"&gt;   &lt;xs:sequence&gt;     &lt;xs:element ref="UIVersionElements"/&gt;     &lt;xs:element name="Display"&gt;       &lt;xs:complexType&gt;         &lt;xs:sequence&gt;           &lt;xs:element ref="Label"/&gt;           &lt;xs:element name="Xform" type="kw_Xform_type" /&gt;           &lt;xs:sequence minOccurs="0"&gt;             &lt;xs:element name="Format" type="KwantuFormDataTypes" /&gt;             &lt;xs:element name="Element" type="KwantuElement_type" minOccurs="0" /&gt;           &lt;/xs:sequence&gt;           &lt;xs:element minOccurs="0" name="Condition"&gt;             &lt;xs:annotation&gt;               &lt;xs:documentation&gt;If the @Show="Conditional" it looks to the Condition nodeset that evaluates to a boolean. If the f(Condition)=true then field displays, otherwise not. Ie. to specify that the field must display only if the value of another field is "Other" then the following configuration would be applied: Condition[@Operator="Equal"] Condition/Element: {here the reference to the field that could have the value of other} Condition/Value:"Other" &lt;/xs:documentation&gt;             &lt;/xs:annotation&gt;             &lt;xs:complexType&gt;               &lt;xs:sequence&gt;                 &lt;xs:element name="Element" type="KwantuElement_type" /&gt;                 &lt;xs:choice&gt;                   &lt;xs:element name="Value"&gt;                     &lt;xs:complexType/&gt;                   &lt;/xs:element&gt;                   &lt;xs:element name="Element2" type="KwantuElement_type" /&gt;                 &lt;/xs:choice&gt;               &lt;/xs:sequence&gt;             &lt;xs:attribute name="Operator" use="required"&gt;               &lt;xs:simpleType&gt;                 &lt;xs:restriction base="xs:Name"&gt;                   &lt;xs:enumeration value="GreaterThan" /&gt;                   &lt;xs:enumeration value="LessThan" /&gt;                   &lt;xs:enumeration value="GreaterThanOrEqual" /&gt;                   &lt;xs:enumeration value="LessThanOrEqual" /&gt;                   &lt;xs:enumeration value="Equal" /&gt;                   &lt;xs:enumeration value="NotEqual" /&gt;                 &lt;/xs:restriction&gt;               &lt;/xs:simpleType&gt;             &lt;/xs:attribute&gt;           &lt;/xs:complexType&gt;         &lt;/xs:element&gt;         &lt;xs:sequence&gt;           &lt;xs:attribute ref="Show" use="required"/&gt;           &lt;xs:attribute name="Span" type="xs:int" /&gt;         &lt;/xs:sequence&gt;       &lt;/xs:complexType&gt;     &lt;/xs:element&gt;     &lt;xs:element ref="Help"/&gt;   &lt;/xs:sequence&gt;   &lt;xs:attribute ref="ID" use="required"/&gt;   &lt;xs:attribute name="Type" use="required"&gt;     &lt;xs:simpleType&gt;       &lt;xs:restriction base="xs:NCName"&gt;         &lt;xs:enumeration value="Label" /&gt;       &lt;/xs:restriction&gt;     &lt;/xs:simpleType&gt;   &lt;/xs:attribute&gt; &lt;/xs:complexType&gt; </pre>	<p>If the @Show="Conditional" it looks to the Condition nodeset that evaluates to a boolean. If the f(Condition)=true then field displays, otherwise not. Ie. to specify that the field must display only if the value of another field is "Other" then the following configuration would be applied: Condition[@Operator="Equal"] Condition/Element: {here the reference to the field that could have the value of other} Condition/Value:"Other"</p>	

## Complex Type kw\_UIVersionElements\_type

Namespace	No namespace
Diagram	<p>This element is only required if the UIVersion is Custom. Otherwise all the elements are included in the view</p>

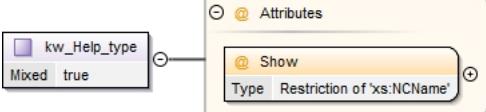
Used by	Element	UIVersionElements
Model	UIElement*	
Children	UIElement	
Source		<pre>&lt;xs:complexType name="kw_UIVersionElements_type"&gt;   &lt;xs:sequence&gt;     &lt;xs:element maxOccurs="unbounded" name="UIElement" minOccurs="0"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;This element is only required if the UIVersion is Custom. Otherwise all the elements are included in the view&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;       &lt;xs:complexType&gt;         &lt;xs:attribute name="Name" use="required"/&gt;         &lt;xs:attribute name="Type" use="required"&gt;           &lt;xs:simpleType&gt;             &lt;xs:restriction base="xs:NCName"&gt;               &lt;xs:enumeration value="Default"/&gt;               &lt;xs:enumeration value="ReadOnly"/&gt;               &lt;xs:enumeration value="Hidden"/&gt;             &lt;/xs:restriction&gt;           &lt;/xs:simpleType&gt;         &lt;/xs:attribute&gt;       &lt;/xs:complexType&gt;     &lt;/xs:element&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;</pre>

## Complex Type kw\_Xform\_type

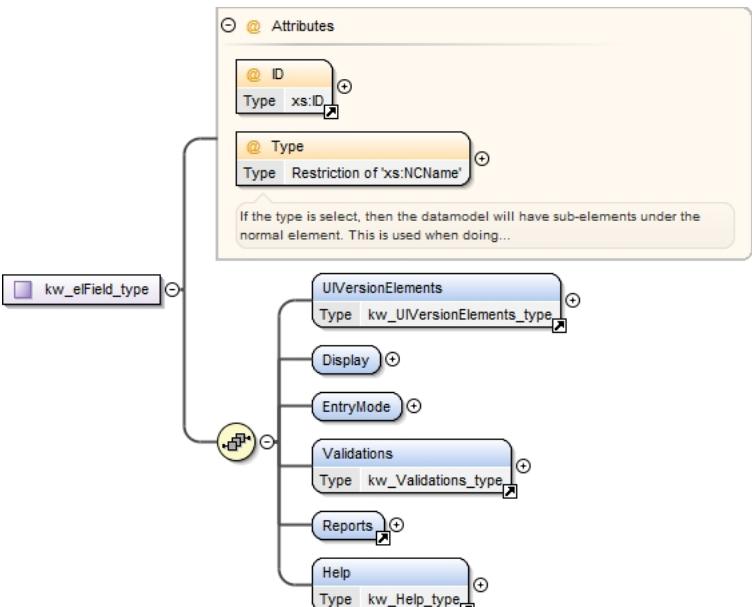
Namespace	No namespace											
Diagram	<pre> classDiagram     class kw_Xform_type {         @ Attributes         @ Appearance         @ CSS     }     kw_Xform_type "1" -- "2" Appearance     kw_Xform_type "1" -- "2" CSS   </pre>											
Used by	Elements	elCalcField/Display/Xform, kw_elLabel_type/Display/Xform										
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>Appearance</td> <td>restriction of xs:Name</td> <td>required</td> </tr> <tr> <td>CSS</td> <td>restriction of xs:Name</td> <td>required</td> </tr> </tbody> </table>	QName	Type	Use	Appearance	restriction of xs:Name	required	CSS	restriction of xs:Name	required		
QName	Type	Use										
Appearance	restriction of xs:Name	required										
CSS	restriction of xs:Name	required										
Source	<pre>&lt;xs:complexType name="kw_Xform_type"&gt;   &lt;xs:attribute name="Appearance" use="required"&gt;     &lt;xs:simpleType&gt;       &lt;xs:restriction base="xs:Name"&gt;         &lt;xs:enumeration value="Output"/&gt;         &lt;xs:enumeration value="Heading"/&gt;         &lt;xs:enumeration value="Subheading"/&gt;         &lt;xs:enumeration value="Currency"/&gt;         &lt;xs:enumeration value="Integer"/&gt;       &lt;/xs:restriction&gt;     &lt;/xs:simpleType&gt;   &lt;/xs:attribute&gt;   &lt;xs:attribute name="CSS" use="required"&gt;     &lt;xs:simpleType&gt;       &lt;xs:restriction base="xs:Name"&gt;         &lt;xs:enumeration value="Subheading"/&gt;         &lt;xs:enumeration value="Kwantu-Standard-Label"/&gt;         &lt;xs:enumeration value="man-label"/&gt;         &lt;xs:enumeration value="Heading"/&gt;         &lt;xs:enumeration value="DatePicker"/&gt;         &lt;xs:enumeration value="None"/&gt;       &lt;/xs:restriction&gt;     &lt;/xs:simpleType&gt;   &lt;/xs:attribute&gt; &lt;/xs:complexType&gt;</pre>											

## Complex Type kw\_Help\_type

Namespace	No namespace
-----------	--------------

Diagram							
Properties	mixed: true						
Used by	Element Help						
Model							
Attributes	<table border="1"> <thead> <tr> <th>QName</th><th>Type</th><th>Use</th></tr> </thead> <tbody> <tr> <td>Show</td><td>restriction of xs:NCName</td><td>required</td></tr> </tbody> </table>	QName	Type	Use	Show	restriction of xs:NCName	required
QName	Type	Use					
Show	restriction of xs:NCName	required					
Source	<pre>&lt;xs:complexType mixed="true" name="kw_Help_type"&gt;   &lt;xs:attribute name="Show" use="required"&gt;     &lt;xs:simpleType&gt;       &lt;xs:restriction base="xs:NCName"&gt;         &lt;xs:enumeration value="Yes"/&gt;         &lt;xs:enumeration value="No"/&gt;       &lt;/xs:restriction&gt;     &lt;/xs:simpleType&gt;   &lt;/xs:attribute&gt; &lt;/xs:complexType&gt;</pre>						

## Complex Type kw\_elField\_type

Namespace	No namespace												
Diagram													
Used by	Element elField												
Model	UIVersionElements , Display , EntryMode , Validations , Reports , Help												
Children	Display, EntryMode, Help, Reports, UVersionElements, Validations												
Attributes	<table border="1"> <thead> <tr> <th>QName</th><th>Type</th><th>Use</th></tr> </thead> <tbody> <tr> <td>ID</td><td>xs:ID</td><td>required</td></tr> <tr> <td>Type</td><td>restriction of xs:NCName</td><td>required</td></tr> <tr> <td></td><td>If the type is select, then the datamodel will have sub-elements under the normal element. This is used when doing assign statements</td><td></td></tr> </tbody> </table>	QName	Type	Use	ID	xs:ID	required	Type	restriction of xs:NCName	required		If the type is select, then the datamodel will have sub-elements under the normal element. This is used when doing assign statements	
QName	Type	Use											
ID	xs:ID	required											
Type	restriction of xs:NCName	required											
	If the type is select, then the datamodel will have sub-elements under the normal element. This is used when doing assign statements												
Source	<pre>&lt;xs:complexType name="kw_elField_type"&gt;   &lt;xs:sequence&gt;     &lt;xs:element ref="UIVersionElements" /&gt;     &lt;xs:element name="Display"&gt;       &lt;xs:complexType&gt;         &lt;xs:sequence&gt;           &lt;xs:element ref="Label" /&gt;           &lt;xs:element name="Format" type="DisplayFormat_type" /&gt;         &lt;/xs:sequence&gt;       &lt;/xs:complexType&gt;     &lt;/xs:element&gt;     &lt;xs:element name="EntryMode" /&gt;     &lt;xs:element name="Help" type="kw_Help_type" /&gt;     &lt;xs:element name="Reports" /&gt;     &lt;xs:element name="Validations" type="kw_Validations_type" /&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;</pre>												

```

<xs:complexType>
  <xs:attribute name="Appearance" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:Name">
        <xs:enumeration value="Currency"/>
        <xs:enumeration value="Numeric"/>
        <xs:enumeration value="Text"/>
        <xs:enumeration value="DatePicker"/>
        <xs:enumeration value="Memo"/>
        <xs:enumeration value="Integer"/>
        <xs:enumeration value="Percentage"/>
        <xs:enumeration value="Output"/>
        <xs:enumeration value="Checkbox"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="CSS">
    <xs:simpleType>
      <xs:restriction base="xs:Name">
        <xs:enumeration value="SmallField"/>
        <xs:enumeration value="MediumField"/>
        <xs:enumeration value="LargeField"/>
        <xs:enumeration value="man-label"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  </xs:complexType>
</xs:elements>
<xs:element minOccurs="0" name="Condition" type="Condition_type">
  <xs:annotation>
    <xs:documentation>If the @Show="Conditional" it looks to the Condition nodeset that evaluates to a boolean. If the f(Condition)=true then field displays, otherwise not. Ie. to specify that the field must display only if the value of another field is "Other" then the following configuration would be applied: Condition[@Operator="Equal"] Condition/Element: {here the reference to the field that could have the value of other} Condition/Value:"Other" </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:sequence>
  <xs:attribute ref="Show" use="required"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="EntryMode">
  <xs:complexType>
    <xs:sequence minOccurs="0">
      <xs:element minOccurs="0" name="Default" type="Default_type"/>
      <xs:choice minOccurs="0">
        <xs:element minOccurs="0" name="AssignElement">
          <xs:annotation>
            <xs:documentation>This provides for mapped fields to map the source element to the element in the current repeat. The source repeat is specified in the Component element Attributes. All the elements that you want to autoload from the source repeat are mapped here to local element ID. These are specific to the local repeat only. </xs:documentation>
          </xs:annotation>
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:normalizedString"/>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="AssignSubElements" type="AssignSubElement_type" minOccurs="0"/>
    </xs:choice>
    <xs:element minOccurs="0" ref="ReadOnlyIf"/>
  </xs:sequence>
  <xs:attribute name="Mode" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:NCName">
        <xs:enumeration value="Capture"/>
        <xs:enumeration value="ReadOnly"/>
        <xs:enumeration value="Hidden"/>
        <xs:enumeration value="Calculated"/>
        <xs:enumeration value="RepeatKey"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  </xs:complexType>
</xs:element>
<xs:element ref="Validations"/>
<xs:element ref="Reports"/>
<xs:element ref="Help"/>
</xs:sequence>
<xs:attribute use="required" ref="ID"/>
<xs:attribute name="Type" use="required">
  <xs:annotation>

```

```

<xs:documentation>If the type is select, then the datamodel will have sub-elements under the
normal element. This is used when doing assign statements</xs:documentation>
</xs:annotation>
<xs:simpleType>
  <xs:restriction base="xs:NCName">
    <xs:enumeration value="Field"/>
    <xs:enumeration value="Select"/>
  </xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:complexType>

```

## Complex Type Condition\_type

Namespace	No namespace												
Diagram	<pre> classDiagram     class Condition_type {         @ Attributes         @ Operator {Type kwantu_comparison_types}         @ Value         @ DataType {Type KwantuFormDataTypes}     }     class Element {Type KwantuElement_type}     class ReferenceElement {Type KwantuElement_type}     Condition_type &lt; -- Element     Condition_type &lt; -- ReferenceElement     Note "Not implemented yet"   </pre>												
Used by	Elements Kwantu_Filter_type/Condition, ReadOnlyIf/Condition, fnHistoryTotal/Filters/Filter/Condition, kw_elField_type/Display/Condition												
Model	Element , ReferenceElement{0,1}												
Children	Element, ReferenceElement												
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>DataType</td> <td>KwantuFormDataTypes</td> <td>optional</td> </tr> <tr> <td>Operator</td> <td>kwantu_comparison_types</td> <td>required</td> </tr> <tr> <td>Value</td> <td></td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Use	DataType	KwantuFormDataTypes	optional	Operator	kwantu_comparison_types	required	Value		optional
QName	Type	Use											
DataType	KwantuFormDataTypes	optional											
Operator	kwantu_comparison_types	required											
Value		optional											
Source	<pre> &lt;xs:complexType name="Condition_type"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="Element" type="KwantuElement_type"/&gt;     &lt;xs:element name="ReferenceElement" type="KwantuElement_type" minOccurs="0"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Not implemented yet&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:element&gt;   &lt;/xs:sequence&gt;   &lt;xs:attribute name="Operator" use="required" type="kwantu_comparison_types"/&gt;   &lt;xs:attribute name="Value"/&gt;   &lt;xs:attribute name="DataType" type="KwantuFormDataTypes"/&gt; &lt;/xs:complexType&gt; </pre>												

## Complex Type Default\_type

Namespace	No namespace
Annotations	A default value is only inserted if the current value of the target element is null(empty). If there is any value in the element, this is not overwritten

Diagram	<pre> graph LR     DT[Default_type] --&gt; Value[Value Type Extension of 'xs:string']     DT --&gt; Init[InitialValue]     DT --&gt; Calc[Calc Type CalcType]     DT --&gt; Ele[ElementValue Type KwantuElement_type]     </pre> <p>A default value is only inserted if the current value of the target element is null(empty). If there is any value in the element, this is not overwritten.</p> <p>This option returns a value from a calculation as the default value.</p> <p>This option returns a value from a different element as the default value.</p>
Used by	Elements elCalcField/EntryMode/Default, kw_elField_type/EntryMode/Default, kw_elSelect_type/EntryMode/Default
Model	Value   (InitialValue , Calc)   ElementValue
Children	Calc, ElementValue, InitialValue, Value
Source	<pre> &lt;xss:complexType name="Default_type"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;A default value is only inserted if the current value of the target element is null(empty). If there is any value in the element, this is not overwritten&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt;   &lt;xss:choice&gt;     &lt;xss:element name="Value"&gt;       &lt;xss:complexType&gt;         &lt;xss:simpleContent&gt;           &lt;xss:extension base="xs:string"&gt;             &lt;xss:attribute name="DataType" type="KwantuFormDataTypes"/&gt;           &lt;/xss:extension&gt;         &lt;/xss:simpleContent&gt;       &lt;/xss:complexType&gt;     &lt;/xss:element&gt;     &lt;xss:sequence&gt;       &lt;xss:element name="InitialValue"&gt;         &lt;xss:complexType&gt;           &lt;xss:attribute name="DataType" type="KwantuFormDataTypes"/&gt;         &lt;/xss:complexType&gt;       &lt;/xss:element&gt;       &lt;xss:element ref="Calc"&gt;         &lt;xss:annotation&gt;           &lt;xss:documentation&gt;This option returns a value from a calculation as the default value&lt;/xss:documentation&gt;         &lt;/xss:annotation&gt;       &lt;/xss:element&gt;     &lt;/xss:sequence&gt;     &lt;xss:element name="ElementValue" type="KwantuElement_type"&gt;       &lt;xss:annotation&gt;         &lt;xss:documentation&gt;This option returns a value from a different element as the default value&lt;/xss:documentation&gt;       &lt;/xss:annotation&gt;     &lt;/xss:element&gt;   &lt;/xss:choice&gt; &lt;/xss:complexType&gt; </pre>

## Complex Type CalcType

Namespace	No namespace
-----------	--------------

Diagram	<pre> classDiagram     class CalcType {         @ Attributes         @ Seq Type xs:integer         ColumnTotal         DateDiff         ElementCalc         RowTotal         ElementsList         CalcFinYearDates         CalcHistory     }     Note over CalcType: This can only be used in a repeat data section     Note over CalcType: This is a repeating section   </pre>						
Used by	Element Calc						
Model	ColumnTotal   DateDiff   ElementCalc   RowTotal   ElementsList   CalcFinYearDates   CalcHistory						
Children	CalcFinYearDates, CalcHistory, ColumnTotal, DateDiff, ElementCalc, ElementsList, RowTotal						
Attributes	<table border="1"> <thead> <tr> <th>QName</th><th>Type</th><th>Use</th></tr> </thead> <tbody> <tr> <td>Seq</td><td>xs:integer</td><td>required</td></tr> </tbody> </table>	QName	Type	Use	Seq	xs:integer	required
QName	Type	Use					
Seq	xs:integer	required					
Source	<pre> &lt;xs:complexType name="CalcType"&gt;   &lt;xs:choice&gt;     &lt;xs:element name="ColumnTotal" type="fnColumnTotal"/&gt;     &lt;xs:element name="DateDiff" type="fnDateDiff"/&gt;     &lt;xs:element name="ElementCalc" type="fnElementCalc"/&gt;     &lt;xs:element name="RowTotal" type="fnRowCalc"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;This can only be used in a repeat data section&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:element&gt;     &lt;xs:element name="ElementsList" type="fnElementsList"/&gt;     &lt;xs:element name="CalcFinYearDates" type="fnCalcFinYearDates"/&gt;     &lt;xs:element name="CalcHistory" type="fnHistoryTotal"/&gt;   &lt;/xs:choice&gt;   &lt;xs:attribute name="Seq" type="xs:integer" use="required"/&gt; &lt;/xs:complexType&gt;   </pre>						

## Complex Type fnColumnTotal

Namespace	No namespace
Diagram	<pre> classDiagram     class fnColumnTotal {         Calculation         Type Restriction of xs:NCName         RepeatID         Type xs:NCName         Element         Type KwantuElement_type         Filters     }     Note over fnColumnTotal: This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum...     Note over fnColumnTotal: This is the RepeatID that specifies the repeat table that is the basis for the calculation.     Note over fnColumnTotal: This is the element in the repeat table that is the basis for the calculation.   </pre>

Used by	Element	CalcType/ColumnTotal
Model	Calculation , RepeatID , Element , Filters{0,1}	
Children	Calculation, Element, Filters, RepeatID	
Source		<pre> &lt;xs:complexType name="fnColumnTotal"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="Calculation"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum numeric value - Min: Return the minimum numeric value - Count: Return the number of rows in the column - CountUnique: Return the number of Unique values in the column&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:NCName"&gt;           &lt;xs:enumeration value="Sum"/&gt;           &lt;xs:enumeration value="Max"/&gt;           &lt;xs:enumeration value="Min"/&gt;           &lt;xs:enumeration value="Count"/&gt;           &lt;xs:enumeration value="CountUnique"/&gt;           &lt;xs:enumeration value="AccumulativeSum"/&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;     &lt;/xs:element&gt;     &lt;xs:element maxOccurs="1" name="RepeatID" type="xs:NCName"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;This is the RepeatID that specifies the repeat table that is the basis for the calculation.&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:element&gt;     &lt;xs:element maxOccurs="1" name="Element" type="KwantuElement_type"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;This is the element in the repeat table that is the basis for the calculation.&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:element&gt;     &lt;xs:element minOccurs="0" name="Filters"&gt;       &lt;xs:complexType&gt;         &lt;xs:sequence&gt;           &lt;xs:element name="Filter" type="Kwantu_Filter_type" maxOccurs="unbounded"&gt;             &lt;xs:annotation&gt;               &lt;xs:documentation&gt;This uses the same basic type as the Validation calculations&lt;/xs:documentation&gt;             &lt;/xs:annotation&gt;           &lt;/xs:element&gt;         &lt;/xs:sequence&gt;       &lt;/xs:complexType&gt;     &lt;/xs:element&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; </pre>

## Complex Type Kwantu\_Filter\_type

Namespace	No namespace
-----------	--------------

Diagram	<pre> classDiagram     class Kwantu_Filter_type {         @ Attributes         @ Seq xs:integer         @ Type kwantu_comparison_types         @ Parameters         @ Reference         @ FilterElement FilterElement_type     }     FilterElement &lt; -- Condition : Condition_type     FilterElement &lt; -- Element : KwantuElement_type     note over Condition : If the conditional element is present, it evaluates to a boolean. If the f(Conditional)=true then the validation rule...   </pre>																		
Used by	Elements Validation, fnColumnTotal/Filters/Filter																		
Model	Parameters{0,1}   Condition{0,1}   Element																		
Children	Condition, Element, Parameters																		
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">QName</th> <th style="text-align: left; padding: 2px;">Type</th> <th style="text-align: left; padding: 2px;">Use</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;"><b>FilterElement</b></td><td style="padding: 2px;">FilterElement_type</td><td style="padding: 2px;">optional</td></tr> <tr> <td style="padding: 2px;"><b>Parameters</b></td><td style="padding: 2px;"></td><td style="padding: 2px;">optional</td></tr> <tr> <td style="padding: 2px;"><b>Reference</b></td><td style="padding: 2px;"></td><td style="padding: 2px;">optional</td></tr> <tr> <td style="padding: 2px;"><b>Seq</b></td><td style="padding: 2px;">xs:integer</td><td style="padding: 2px;">required</td></tr> <tr> <td style="padding: 2px;"><b>Type</b></td><td style="padding: 2px;">kwantu_comparison_types</td><td style="padding: 2px;">required</td></tr> </tbody> </table>	QName	Type	Use	<b>FilterElement</b>	FilterElement_type	optional	<b>Parameters</b>		optional	<b>Reference</b>		optional	<b>Seq</b>	xs:integer	required	<b>Type</b>	kwantu_comparison_types	required
QName	Type	Use																	
<b>FilterElement</b>	FilterElement_type	optional																	
<b>Parameters</b>		optional																	
<b>Reference</b>		optional																	
<b>Seq</b>	xs:integer	required																	
<b>Type</b>	kwantu_comparison_types	required																	
Source	<pre> &lt;xss:complexType name="Kwantu_Filter_type"&gt;   &lt;xss:choice minOccurs="0"&gt;     &lt;xss:element minOccurs="0" name="Parameters"&gt;       &lt;xss:complexType&gt;         &lt;xss:choice&gt;           &lt;xss:element name="Element" type="KwantuElement_type" /&gt;           &lt;xss:element name="Constant"&gt;             &lt;xss:complexType&gt;               &lt;xss:simpleContent&gt;                 &lt;xss:extension base="xs:string"&gt;                   &lt;xss:attribute name="DataType" type="KwantuFormDataTypes" /&gt;                 &lt;/xss:extension&gt;               &lt;/xss:simpleContent&gt;             &lt;/xss:complexType&gt;           &lt;/xss:element&gt;         &lt;/xss:choice&gt;       &lt;/xss:complexType&gt;     &lt;/xss:element&gt;     &lt;xss:element minOccurs="0" name="Condition" type="Condition_type"&gt;       &lt;xss:annotation&gt;         &lt;xss:documentation&gt;If the conditional element is present, it evaluates to a boolean. If the f(Conditional)=true then the validation rule applies, otherwise it does not apply. Ie. to specify that the field is required if the value of another field is "Other" then the following configuration would be applied: Validation: @Seq=1; @Type=NotNull ---&gt; this specifies that the current element value should not be null Condition[@Operator="Equal"] Condition/Element: {here the reference to the field that could have the value of other} Condition/Value: "Other" &lt;/xss:documentation&gt;       &lt;/xss:annotation&gt;     &lt;/xss:element&gt;     &lt;xss:element name="Element" type="KwantuElement_type" /&gt;   &lt;/xss:choice&gt;   &lt;xss:attribute name="Seq" type="xs:integer" use="required"/&gt;   &lt;xss:attribute name="Type" use="required" type="kwantu_comparison_types" /&gt;   &lt;xss:attribute name="Parameters" /&gt;   &lt;xss:attribute name="Reference" /&gt;   &lt;xss:attribute name="FilterElement" type="FilterElement_type" /&gt; &lt;/xss:complexType&gt;   </pre>																		

## Complex Type fnDateDiff

Namespace	No namespace
Diagram	
Used by	Element CalcType/DateDiff
Model	Calculation , Element{2,2}
Children	Calculation, Element
Source	<pre> &lt;xs:complexType name="fnDateDiff"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="Calculation"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;This element specifies which calculation should be performed - CalendarDays: calculates the number of calendar days between date1 and date2 - WorkDays: calculates the number of work days between date1 and date2. Work days includes Mondays to Fridays&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:element&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="CalendarDays" /&gt;       &lt;xs:enumeration value="WorkDays" /&gt;     &lt;/xs:restriction&gt;   &lt;/xs:sequence&gt;   &lt;xs:element maxOccurs="2" name="Element" type="KwantuElement_type" minOccurs="2"&gt;     &lt;xs:annotation&gt;       &lt;xs:documentation&gt;This is the element that contains the first date (Earliest date)&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;   &lt;/xs:element&gt; &lt;/xs:complexType&gt; </pre>

## Complex Type fnElementCalc

Namespace	No namespace
Diagram	
Used by	Elements CalcType/ElementCalc, RepeatDef/AddRows/AutoLoad/Component/Filter/ElementCalc
Model	Calculation , Element{2,2}
Children	Calculation, Element
Source	<pre> &lt;xs:complexType name="fnElementCalc"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="Calculation"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;This element specifies which calculation should be performed - CalendarDays: calculates the number of calendar days between date1 and date2 - WorkDays: calculates the number of work days between date1 and date2. Work days includes Mondays to Fridays&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:element&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="Add" /&gt;       &lt;xs:enumeration value="Subtract" /&gt;     &lt;/xs:restriction&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; </pre>

```

        <xs:enumeration value="Multiply" />
        <xs:enumeration value="Divide" />
    </xs:restriction>
</xs:simpleType>
</xs:element>
<xselement maxOccurs="2" name="Element" type="KwantuElement_type" minOccurs="2">
    <xselement>
        <xs:annotation>
            <xs:documentation>This is the element that contains the first date (Earliest date)</
        xs:documentation>
        <xs:annotation>
    </xs:element>
</xs:sequence>
</xs:complexType>

```

## Complex Type fnRowCalc

Namespace	No namespace
Diagram	
Used by	Element      CalcType/RowTotal
Model	Calculation , Element+
Children	Calculation, Element
Source	<pre> &lt;xs:complexType name="fnRowCalc"&gt;     &lt;xs:sequence&gt;         &lt;xselement name="Calculation"&gt;             &lt;xselement&gt;                 &lt;xs:annotation&gt;                     &lt;xs:documentation&gt;This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum numeric value - Min: Return the minimum numeric value - Count: Return the number of rows in the column - CountUnique: Return the number of Unique values in the column&lt;/xs:documentation&gt;                 &lt;/xs:annotation&gt;                 &lt;xs:simpleType&gt;                     &lt;xs:restriction base="xs:NCName"&gt;                         &lt;xs:enumeration value="Sum" /&gt;                         &lt;xs:enumeration value="Max" /&gt;                         &lt;xs:enumeration value="Min" /&gt;                         &lt;xs:enumeration value="Multiply" /&gt;                         &lt;xs:enumeration value="Count" /&gt;                         &lt;xs:enumeration value="AccumulativeSum" /&gt;                     &lt;/xs:restriction&gt;                 &lt;/xs:simpleType&gt;             &lt;/xselement&gt;         &lt;xselement maxOccurs="unbounded" name="Element" type="KwantuElement_type"&gt;             &lt;xselement&gt;                 &lt;xs:annotation&gt;                     &lt;xs:documentation&gt;This is the element in the repeat table that is the basis for the calculation.&lt;/xs:documentation&gt;                 &lt;/xs:annotation&gt;             &lt;/xselement&gt;         &lt;/xs:sequence&gt;     &lt;/xs:complexType&gt; </pre>

## Complex Type fnElementsList

Namespace	No namespace
Annotations	Performs Calculations on a list of elements

Diagram	<pre> classDiagram     class fnElementsList {         &lt;&lt;Performs Calculations on a list of elements&gt;&gt;     }     class Calculation {         &lt;&lt;This element specifies which calculation should be performed Sum Max Min Count counts the number of non-null values...&gt;&gt;     }     class Element {         &lt;&lt;This is the element(s) that are included in the list&gt;&gt;         &lt;&lt;1..&gt;&gt;     }     fnElementsList --&gt; Calculation     fnElementsList --&gt; Element   </pre>
Used by	Elements      CalcType/ElementsList, RepeatDef/AddRows/AutoLoad/Component/Filter/ElementsList
Model	Element   (Calculation , Element+)
Children	Calculation, Element
Source	<pre> &lt;xs:complexType name="fnElementsList"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Performs Calculations on a list of elements&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:choice&gt;     &lt;xs:element name="Element" type="KwantuElement_type"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;This is the element(s) that are included in the list&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:element&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="Calculation"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;This element specifies which calculation should be performed Sum Max Min Count counts the number of non-null values Count Unique counts the unique values identified&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;         &lt;xs:simpleType&gt;           &lt;xs:restriction base="xs:NCName"&gt;             &lt;xs:enumeration value="Sum"/&gt;             &lt;xs:enumeration value="Max"/&gt;             &lt;xs:enumeration value="Min"/&gt;             &lt;xs:enumeration value="CountUnique"/&gt;           &lt;/xs:restriction&gt;         &lt;/xs:simpleType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:choice&gt; &lt;/xs:complexType&gt;   </pre>

### Complex Type fnCalcFinYearDates

Namespace	No namespace
Diagram	<pre> classDiagram     class fnCalcFinYearDates {         &lt;&lt;This element specifies which calculation should be performed - CalendarDays: calculates the number of calendar days...&gt;&gt;     }     class Calculation {         &lt;&lt;This element specifies which calculation should be performed - CalendarDays: calculates the number of calendar days...&gt;&gt;     }     class Element {         &lt;&lt;This is the element that contains the date that is provided to calculate the financial Years&gt;&gt;     }     fnCalcFinYearDates --&gt; Calculation     fnCalcFinYearDates --&gt; Element   </pre>
Used by	Element      CalcType/CalcFinYearDates
Model	Calculation , Element
Children	Calculation, Element
Source	<pre> &lt;xs:complexType name="fnCalcFinYearDates"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="Calculation"&gt;   </pre>

```

<xs:annotation>
    <xs:documentation>This element specifies which calculation should be performed -
    CalendarDays: calculates the number of calendar days between date1 and date2 - WorkDays:
    calculates the number of work days between date1 and date2. Work days includes Mondays to Fridays</xs:documentation>
</xs:annotation>
<xs:simpleType>
    <xs:restriction base="xs:NCName">
        <xs:enumeration value="StartDateFY1April"/>
        <xs:enumeration value="StartDateFY1July"/>
        <xs:enumeration value="EndDateFY1April"/>
        <xs:enumeration value="EndDateFY1July"/>
    </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="Element" type="KwantuElement_type">
    <xs:annotation>
        <xs:documentation>This is the element that contains the date that is provided to calculate
        the financial Years</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>

```

## Complex Type fnHistoryTotal

Namespace	No namespace
Diagram	<p>The diagram illustrates the structure of the <code>fnHistoryTotal</code> complex type. It consists of a central node labeled <code>fnHistoryTotal</code>, which is connected to three other nodes: <code>Calculation</code>, <code>Element</code>, and <code>Filters</code>. Each node is accompanied by a tooltip containing its description and type.</p> <ul style="list-style-type: none"> <li><b>Calculation</b>: Type <code>Restriction of 'xs:NCName'</code>. Description: This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum...</li> <li><b>Element</b>: Type <code>KwantuElement_type</code>. Description: This is the element in the repeat table that is the basis for the calculation.</li> <li><b>Filters</b>: Description: Filters</li> </ul>
Used by	Element CalcType/CalcHistory
Model	Calculation , Element , Filters{0,1}
Children	Calculation, Element, Filters
Source	<pre> &lt;xs:complexType name="fnHistoryTotal"&gt;     &lt;xs:sequence&gt;         &lt;xs:element name="Calculation"&gt;             &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;This element specifies which calculation should be performed - Sum: Sum                 all numeric values - Max: Return the maximum numeric value - Min: Return the minimum numeric value                 - Count: Return the number of rows in the column - CountUnique: Return the number of Unique values                 in the column&lt;/xs:documentation&gt;             &lt;/xs:annotation&gt;             &lt;xs:simpleType&gt;                 &lt;xs:restriction base="xs:NCName"&gt;                     &lt;xs:enumeration value="Sum"/&gt;                     &lt;xs:enumeration value="Max"/&gt;                     &lt;xs:enumeration value="Min"/&gt;                     &lt;xs:enumeration value="Count"/&gt;                     &lt;xs:enumeration value="CountUnique"/&gt;                     &lt;xs:enumeration value="AccumulativeSum"/&gt;                 &lt;/xs:restriction&gt;             &lt;/xs:simpleType&gt;         &lt;/xs:element&gt;         &lt;xs:element maxOccurs="1" name="Element" type="KwantuElement_type"&gt;             &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;This is the element in the repeat table that is the basis for the                 calculation.&lt;/xs:documentation&gt;             &lt;/xs:annotation&gt;         &lt;/xs:element&gt;         &lt;xs:element minOccurs="0" name="Filters"&gt;             &lt;xs:complexType&gt;                 &lt;xs:sequence&gt;                     &lt;xs:element maxOccurs="unbounded" name="Filter"&gt;                         &lt;xs:complexType&gt;                             &lt;xs:sequence minOccurs="0"&gt;                                 &lt;xs:element minOccurs="0" name="Parameters"&gt;                                     &lt;xs:complexType&gt; </pre>

```

<xs:sequence>
    <xs:element name="Element" type="KwantuElement_type" minOccurs="0" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element minOccurs="0" name="Condition" type="Condition_type">
    <xs:annotation>
        <xs:documentation>If the conditional element is present, it evaluates to a boolean. If the f(Conditional)=true then the validation rule applies, otherwise it does not apply. Ie. to specify that the field is required if the value of another field is "Other" then the following configuration would be applied: Validation: @Seq=1; @Type=NotNull ---> this specifies that the current element value should not be null Condition[@Operator="Equal"] Condition/Element: {here the reference to the field that could have the value of other} Condition/Value:"Other" </xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
<xs:attribute name="Seq" type="xs:integer" use="required"/>
<xs:attribute name="Type" use="required" type="kwantu_comparison_types"/>
<xs:attribute name="Parameters"/>
<xs:attribute name="Reference"/>
<xs:attribute name="FilterElement" type="FilterElement_type"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

```

## Complex Type AssignSubElement\_type

Namespace	No namespace
Diagram	<pre> classDiagram     class AssignSubElement_type {         &lt;&lt;AssignSubElement_type&gt;&gt;         &lt;&lt;0..&gt;&gt; AssignSubElement         &lt;&lt;AssignCode&gt;&gt;         &lt;&lt;AssignName&gt;&gt;     }     AssignSubElement &lt; -- AssignCode     AssignSubElement &lt; -- AssignName     AssignSubElement &lt; -- AssignSubElement   </pre> <p>The diagram illustrates the <code>AssignSubElement_type</code> complex type. It features three associations: <code>AssignCode</code>, <code>AssignName</code>, and <code>AssignSubElement</code>. Each association is represented by a box labeled with its name and type (<code>Extension of xs:normalizedString</code>). A callout box provides a detailed description for each: <code>AssignCode</code> describes assigning values from taxonomy to element IDs associated with the Select Element ID; <code>AssignName</code> describes assigning values from taxonomy to element IDs associated with the Select Element ID; and <code>AssignSubElement</code> describes assigning values from taxonomy to element IDs associated with the Select Element ID.</p>
Used by	Elements <code>kw_elField_type/EntryMode/AssignSubElements</code> , <code>kw_elSelect_type/EntryMode/AssignSubElements</code>
Model	<code>AssignCode</code> , <code>AssignName{0,1}</code> , <code>AssignSubElement*</code>
Children	<code>AssignCode</code> , <code>AssignName</code> , <code>AssignSubElement</code>
Source	<pre> &lt;xs:complexType name="AssignSubElement_type"&gt;     &lt;xs:sequence&gt;         &lt;xs:element name="AssignCode"&gt;             &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;Here you assign one or more values from the taxonomy to element IDs associated with the Select Element ID&lt;/xs:documentation&gt;             &lt;/xs:annotation&gt;         &lt;/xs:element&gt;         &lt;xs:element name="AssignName" minOccurs="0"&gt;             &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;Here you assign one or more values from the taxonomy to element IDs associated with the Select Element ID&lt;/xs:documentation&gt;             &lt;/xs:annotation&gt;         &lt;/xs:element&gt;         &lt;xs:element name="AssignSubElement" type="xs:normalizedString" maxOccurs="unbounded"&gt;             &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;Here you assign one or more values from the taxonomy to element IDs associated with the Select Element ID&lt;/xs:documentation&gt;             &lt;/xs:annotation&gt;         &lt;/xs:element&gt;     &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;   </pre>

```

<xs:attribute name="SubElementID" type="xs:NCName" use="required"/>
<xs:attribute name="DataType" type="KwantuFormDataTypes" use="required"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element maxOccurs="unbounded" name="AssignSubElement" minOccurs="0">
<xs:annotation>
<xs:documentation>Here you assign one or more values from the taxonomy to element IDs associated with the Select Element ID</xs:documentation>
</xs:annotation>
<xs:complexType>
<xs:simpleContent>
<xs:extension base="xs:normalizedString">
<xs:attribute name="Action">
<xs:annotation>
<xs:documentation>Action="Default" means that the assignment only takes place if the element is currently null.</xs:documentation>
</xs:annotation>
<xs:simpleType>
<xs:restriction base="xs:NCName">
<xs:enumeration value="Default"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="SubElementID" type="xs:NCName" use="required"/>
<xs:attribute name="DataType" type="KwantuFormDataTypes" use="required"/>
</xs:extension>
<xs:simpleContent>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

```

## Complex Type kw\_Validations\_type

Namespace	No namespace
Diagram	<pre> classDiagram     class kw_Validations_type {         Required         DataType         Validation * "0..∞"         ErrorMessage     }     Required &lt; -- kw_Validations_type     DataType &lt; -- kw_Validations_type     Validation &lt; -- kw_Validations_type     ErrorMessage &lt; -- kw_Validations_type </pre>
Used by	Element      Validations
Model	Required , DataType , Validation* , ErrorMessage
Children	DataType, ErrorMessage, Required, Validation
Source	<pre> &lt;xs:complexType name="kw_Validations_type"&gt; &lt;xs:sequence&gt; &lt;xs:element name="Required" type="xs:boolean"/&gt; &lt;xs:element ref="DataType"/&gt; &lt;xs:element maxOccurs="unbounded" ref="Validation" minOccurs="0"/&gt; &lt;xs:element name="ErrorMessage" type="xs:string"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; </pre>

## Complex Type ReportColumnType

Namespace	No namespace
-----------	--------------

Diagram	<pre> classDiagram     class ReportColumnType {         &lt;&lt;Mixed true&gt;&gt;         @Attributes         @TableName Type xs&gt;IDREF         @SubElement Type xs:string         @Source Type Source_type     }     class Fieldname {         Type xs:NCName     }     class Fieldtype {         Type ColumnFieldtype     }     class Fieldlength {         Type xs:string         Default 0     }     class Nullable {         Type Restriction of 'xs:string'     }     ReportColumnType &lt; -- Fieldname     ReportColumnType &lt; -- Fieldtype     ReportColumnType &lt; -- Fieldlength     ReportColumnType &lt; -- Nullable   </pre>												
Properties	mixed: true												
Used by	Element Column												
Model	ALL(Fieldname Fieldtype Fieldlength Nullable{0,1})												
Children	Fieldlength, Fieldname, Fieldtype, Nullable												
Attributes	<table border="1"> <thead> <tr> <th>QName</th><th>Type</th><th>Use</th></tr> </thead> <tbody> <tr> <td>Source</td><td>Source_type</td><td>required</td></tr> <tr> <td>SubElement</td><td>xs:string</td><td>required</td></tr> <tr> <td>TableName</td><td>xs&gt;IDREF</td><td>required</td></tr> </tbody> </table>	QName	Type	Use	Source	Source_type	required	SubElement	xs:string	required	TableName	xs>IDREF	required
QName	Type	Use											
Source	Source_type	required											
SubElement	xs:string	required											
TableName	xs>IDREF	required											
Source	<pre> &lt;xs:complexType mixed="true" name="ReportColumnType"&gt;   &lt;xs:all&gt;     &lt;xs:element name="Fieldname" type="xs:NCName"/&gt;     &lt;xs:element name="Fieldtype" type="ColumnFieldtype"/&gt;     &lt;xs:element default="0" name="Fieldlength" type="xs:string"/&gt;     &lt;xs:element minOccurs="0" name="Nullable"&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:string"&gt;           &lt;xs:enumeration value="Not Null"/&gt;           &lt;xs:enumeration value="Nullable"/&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;     &lt;/xs:element&gt;   &lt;/xs:all&gt;   &lt;xs:attribute name="TableName" type="xs&gt;IDREF" use="required"/&gt;   &lt;xs:attribute name="SubElement" type="xs:string" use="required"/&gt;   &lt;xs:attribute name="Source" use="required" type="Source_type"/&gt; &lt;/xs:complexType&gt;   </pre>												

## Complex Type kw\_elSelect\_type

Namespace	No namespace
-----------	--------------

Diagram	<pre> classDiagram     class kw_elSelect_type {         @ Attributes         @ ID Type xs:ID         @ Type Type restriction of 'xs:NCName'     }     class UIVersionElements {         Type kw_UIVersionElements_type     }     class Display     class EntryMode     class Validations {         Type kw_Validations_type     }     class Reports     class Help {         Type kw_Help_type     }      kw_elSelect_type "3" -- "1" UIVersionElements :      kw_elSelect_type "3" -- "1" Display :      kw_elSelect_type "3" -- "1" EntryMode :      kw_elSelect_type "3" -- "1" Validations :      kw_elSelect_type "3" -- "1" Reports :      kw_elSelect_type "3" -- "1" Help :   </pre>												
Used by	Element elSelect												
Model	UIVersionElements , Display , EntryMode , Validations , Reports , Help												
Children	Display, EntryMode, Help, Reports, UIVersionElements, Validations												
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">QName</th> <th style="text-align: left; padding: 2px;">Type</th> <th style="text-align: left; padding: 2px;">Use</th> <th style="text-align: left; padding: 2px;"></th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">ID</td><td style="padding: 2px;">xs:ID</td><td style="padding: 2px;">required</td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">Type</td><td style="padding: 2px;">restriction of xs:NCName</td><td style="padding: 2px;">required</td><td style="padding: 2px;"></td></tr> </tbody> </table>	QName	Type	Use		ID	xs:ID	required		Type	restriction of xs:NCName	required	
QName	Type	Use											
ID	xs:ID	required											
Type	restriction of xs:NCName	required											
Source	<pre> &lt;xs:complexType name="kw_elSelect_type"&gt;   &lt;xs:sequence&gt;     &lt;xs:element ref="UIVersionElements" /&gt;     &lt;xs:element name="Display"&gt;       &lt;xs:complexType&gt;         &lt;xs:sequence&gt;           &lt;xs:element ref="Label" /&gt;           &lt;xs:element name="Format" type="DisplayFormat_type" /&gt;           &lt;xs:element name="Xform"&gt;             &lt;xs:complexType&gt;               &lt;xs:attribute name="Appearance" use="required"&gt;                 &lt;xs:simpleType&gt;                   &lt;xs:restriction base="xs:Name"&gt;                     &lt;xs:enumeration value="SingleSelect" /&gt;                     &lt;xs:enumeration value="MultiSelect" /&gt;                     &lt;xs:enumeration value="RadioSelect" /&gt;                     &lt;xs:enumeration value="CheckBox" /&gt;                     &lt;xs:enumeration value="Output" /&gt;                     &lt;xs:enumeration value="Search" /&gt;                     &lt;xs:enumeration value="Button" /&gt;                   &lt;/xs:restriction&gt;                 &lt;/xs:simpleType&gt;               &lt;/xs:attribute&gt;               &lt;xs:attribute name="CSS" &gt;                 &lt;xs:simpleType&gt;                   &lt;xs:restriction base="xs:Name"&gt;                     &lt;xs:enumeration value="MediumSelect" /&gt;                     &lt;xs:enumeration value="LargeSelect" /&gt;                     &lt;xs:enumeration value="man-label" /&gt;                     &lt;xs:enumeration value="Kwantu-Standard-Label" /&gt;                   &lt;/xs:restriction&gt;                 &lt;/xs:simpleType&gt;               &lt;/xs:attribute&gt;             &lt;/xs:complexType&gt;           &lt;/xs:element&gt;         &lt;xs:element minOccurs="0" ref="Condition" /&gt;       &lt;/xs:sequence&gt;       &lt;xs:attribute ref="Show" use="required" /&gt;     &lt;/xs:complexType&gt;   &lt;/xs:element&gt;   &lt;xs:element name="EntryMode"&gt;     &lt;xs:complexType&gt;       &lt;xs:sequence&gt;         &lt;xs:element name="Select"&gt;           &lt;xs:complexType&gt;             &lt;xs:sequence&gt;   </pre>												

```

<xs:choice minOccurs="0">
    <xs:element ref="Codelist"/>
    <xs:element name="SharedData">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Collection" type="xs:normalizedString"/>
                <xs:element name="TopElement" type="xs:normalizedString"/>
                <xs:element maxOccurs="unbounded" name="KeyValue"
type="xs:normalizedString">
                    <xs:annotation>
                        <xs:documentation>Key Value is the Unique identifier for the block
of data. This holds the path to the key value. If there are more than one then they are assigned
multiple times</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" name="Filter">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="FilterElement" type="xs:normalizedString">
                                <xs:annotation>
                                    <xs:documentation>This holds the path from the TopElement from
which the data will be searched</xs:documentation>
                                </xs:annotation>
                            </xs:element>
                            <xs:element name="FilterValue">
                                <xs:complexType>
                                    <xs:choice>
                                        <xs:element name="Element" type="KwantuElement_type"/>
                                        <xs:element name="Value" type="xs:Name"/>
                                    </xs:choice>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Rest" type="RestType"/>
    <xs:element name="Taxonomy">
        <xs:complexType>
            <xs:simpleContent>
                <xs:extension base="xs:NCName">
                    <xs:attribute name="Type">
                        <xs:simpleType>
                            <xs:restriction base="xs:NCName">
                                <xs:enumeration value="XSL"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:attribute>
                </xs:extension>
            </xs:simpleContent>
        </xs:complexType>
    </xs:element>
</xs:choice>
<xs:element minOccurs="0" name="Refresh">
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded" name="ElementID" type="xs:IDREF"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:elements>
<xs:element name="AssignSubElements" type="AssignSubElement_type">
    <xs:annotation>
        <xs:documentation>This Assign only assigns to this element node</xs:documentation>
    </xs:annotation>
</xs:elements>
<xs:element name="Assigns" type="Assigns_type" minOccurs="0">
    <xs:annotation>
        <xs:documentation>This assigns value to elements that are not the current element</
xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="Default" type="Default_type"/>
<xs:element minOccurs="0" ref="ReadOnlyIf"/>
</xs:sequence>
<xs:attribute name="Mode" use="required">
    <xs:annotation id="Select.Entrymode.Mode">
        <xs:documentation>The mode defines how the data elements are entered into the field.
Capture - User takes an action ReadOnly - User does not take action but the field can only be

```

```

viewed Hidden - User cannot see the values Calculated - Value is calculated RepeatKey - The value
is picked up from the way that the repeat key is specified.</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:NCName">
            <xs:enumeration value="Capture"/>
            <xs:enumeration value="ReadOnly"/>
            <xs:enumeration value="RepeatKey"/>
        </xs:restriction>
    </xs:simpleType>
    </xs:attribute>
<xs:attribute name="Source" type="Source_type"/>
</xs:complexType>
<xs:element ref="Validations"/>
<xs:element ref="Reports"/>
<xs:element ref="Help"/>
</xs:sequence>
<xs:attribute use="required" ref="ID"/>
<xs:attribute name="Type" use="required">
    <xs:simpleType>
        <xs:restriction base="xs:NCName">
            <xs:enumeration value="Select"/>
            <xs:enumeration value="SelectGroup"/>
            <xs:enumeration value="RepeatKeyField"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
</xs:complexType>

```

## Complex Type kw\_Condition\_type

Namespace	No namespace						
Diagram							
Used by	Element Condition						
Model	Element , (Value   Element2)						
Children	Element, Element2, Value						
Attributes	<table border="1"> <thead> <tr> <th>QName</th><th>Type</th><th>Use</th></tr> </thead> <tbody> <tr> <td>Operator</td><td>restriction of xs:Name</td><td>required</td></tr> </tbody> </table>	QName	Type	Use	Operator	restriction of xs:Name	required
QName	Type	Use					
Operator	restriction of xs:Name	required					
Source	<pre> &lt;xs:complexType name="kw_Condition_type"&gt;     &lt;xs:sequence&gt;         &lt;xs:element name="Element" type="KwantuElement_type"/&gt;         &lt;xs:choice&gt;             &lt;xs:element name="Value"&gt;                 &lt;xs:complexType/&gt;             &lt;/xs:element&gt;             &lt;xs:element name="Element2" type="KwantuElement_type"/&gt;         &lt;/xs:choice&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute name="Operator" use="required"&gt;         &lt;xs:simpleType&gt;             &lt;xs:restriction base="xs:Name"&gt;                 &lt;xs:enumeration value="GreaterThan"/&gt;                 &lt;xs:enumeration value="LessThan"/&gt;                 &lt;xs:enumeration value="GreaterThanOrEqualTo"/&gt;                 &lt;xs:enumeration value="LessThanOrEqualTo"/&gt;                 &lt;xs:enumeration value="Equal"/&gt;                 &lt;xs:enumeration value="NotEqual"/&gt;             &lt;/xs:restriction&gt;         &lt;/xs:simpleType&gt;     &lt;/xs:attribute&gt; &lt;/xs:complexType&gt; </pre>						

## Complex Type RestType

Namespace	No namespace						
Diagram	<pre> classDiagram     class RestType {         @Attributes         @ID         HostID         ServiceURL         Parameters         Return         IfError     }     RestType &lt; -- Parameters     RestType &lt; -- Return     RestType &lt; -- IfError     note over Parameters: Here you specify the parameters that are used to call the rest service     note over Return: Here you specify the return elements and what to do with them.     note over IfError: Here you specify what to do if the rest service returned an error. Ignore - means that you continue Abort - means that...   </pre>						
Used by	Element kw_elSelect_type/EntryMode>Select/Rest						
Model	HostID , ServiceURL , Parameters , Return , IfError						
Children	HostID, IfError, Parameters, Return, ServiceURL						
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><b>ID</b></td> <td></td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Use	<b>ID</b>		optional
QName	Type	Use					
<b>ID</b>		optional					
Source	<pre> &lt;xs:complexType name="RestType"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="HostID" type="xs:NCName" /&gt;     &lt;xs:element name="ServiceURL" type="Extension of 'xs:anyURI'" /&gt;     &lt;xs:complexType&gt;       &lt;xs:simpleContent&gt;         &lt;xs:extension base="xs:anyURI"&gt;           &lt;xs:attribute name="Source" use="required"&gt;             &lt;xs:simpleType&gt;               &lt;xs:restriction base="xs:NCName"&gt;                 &lt;xs:enumeration value="ProcessVariable" /&gt;                 &lt;xs:enumeration value="SubProcessVariable" /&gt;                 &lt;xs:enumeration value="ElementID" /&gt;                 &lt;xs:enumeration value="ProcessObject" /&gt;                 &lt;xs:enumeration value="SubProcessObject" /&gt;                 &lt;xs:enumeration value="ServiceURL" /&gt;               &lt;/xs:restriction&gt;             &lt;/xs:simpleType&gt;           &lt;/xs:attribute&gt;         &lt;/xs:extension&gt;       &lt;/xs:simpleContent&gt;     &lt;/xs:complexType&gt;   &lt;/xs:element&gt;   &lt;xs:element name="Parameters" type="KwantuElement_type"&gt;     &lt;xs:annotation&gt;       &lt;xs:documentation&gt;Here you specify the parameters that are used to call the rest service&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;     &lt;xs:complexType&gt;       &lt;xs:sequence&gt;         &lt;xs:element maxOccurs="unbounded" name="Parameter"&gt;           &lt;xs:complexType&gt;             &lt;xs:choice&gt;               &lt;xs:element name="Element" type="KwantuElement_type" /&gt;               &lt;xs:element name="Constant" type="xs:string" /&gt;             &lt;/xs:choice&gt;             &lt;xs:attribute name="Seq" /&gt;             &lt;xs:attribute name="Name" type="xs:NCName" /&gt;             &lt;xs:attribute name="Format" /&gt;           &lt;/xs:complexType&gt;         &lt;/xs:element&gt;       &lt;/xs:sequence&gt;     &lt;/xs:complexType&gt;   &lt;/xs:element&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;   </pre>						

```

        </xs:complexType>
    </xs:element>
<xs:element name="Return">
    <xs:annotation>
        <xs:documentation>Here you specify the return elements and what to do with them.</xs:documentation>
    </xs:annotation>
<xs:complexType>
    <xs:sequence>
        <xs:element name="Result" type="xs:string">
            <xs:annotation>
                <xs:documentation>This is the xpath to the root of the result set that you are dealing with</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element maxOccurs="unbounded" name="Status">
            <xs:annotation>
                <xs:documentation>This elements contains any status aspects that you want to deal with</xs:documentation>
            </xs:annotation>
<xs:complexType>
    <xs:sequence>
        <xs:element name="StatusCode">
            <xs:annotation>
                <xs:documentation>Here we define the xpath relative to the result that contains the error code, and it compares to the value specified to determine if it is an error or not</xs:documentation>
            </xs:annotation>
<xs:complexType>
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute name="Value" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="StatusMessage" type="xs:string" />
</xs:sequence>
<xs:attribute name="Type">
    <xs:annotation>
        <xs:documentation>If the type="Error" then this a potential error status If the type="Success" then this a not an error</xs:documentation>
    </xs:annotation>
<xs:simpleType>
    <xs:restriction base="xs:NCName">
        <xs:enumeration value="Error"/>
        <xs:enumeration value="Success"/>
    </xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>
<xs:element maxOccurs="unbounded" name="Assign" minOccurs="0">
    <xs:annotation>
        <xs:documentation>If you need to make any assignments of data based on the rest service, then you can do it here.</xs:documentation>
    </xs:annotation>
<xs:complexType>
    <xs:simpleContent>
        <xs:extension base="xs:anyURI">
            <xs:attribute name="AssignTo" use="required">
                <xs:simpleType>
                    <xs:restriction base="xs:NCName">
                        <xs:enumeration value="Variable"/>
                        <xs:enumeration value="Element"/>
                        <xs:enumeration value="Object"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="AssignToID" use="required"/>
            <xs:attribute name="Type" use="required">
                <xs:simpleType>
                    <xs:restriction base="xs:NCName">
                        <xs:enumeration value="XPath"/>
                        <xs:enumeration value="JSON"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
</xs:element>
</xs:sequence>

```

```

</xs:complexType>
</xs:element>
<xs:element name="IfError">
  <xs:annotation>
    <xs:documentation>Here you specify what to do if the rest service returned an error.
    Ignore - means that you continue Abort - means that you abort the operation Warning - issue a
    warning to the user and then continue Retry - retry a number of times specified, and then abort</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="Ignore"/>
      <xs:enumeration value="Abort"/>
      <xs:enumeration value="Warning"/>
      <xs:enumeration value="Retry"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
</xs:sequence>
<xs:attribute name="ID"/>
</xs:complexType>

```

## Complex Type Assigns\_type

Namespace	No namespace
Diagram	
Used by	Element kw_elSelect_type/EntryMode/Assigns
Model	Assign*
Children	Assign
Source	<pre> &lt;xs:complexType name="Assigns_type"&gt;   &lt;xs:sequence&gt;     &lt;xs:element maxOccurs="unbounded" name="Assign" minOccurs="0"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Here you assign one or more values from the taxonomy to element IDs associated         with the Select Element ID&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;       &lt;xs:complexType&gt;         &lt;xs:simpleContent&gt;           &lt;xs:extension base="xs:normalizedString"&gt;             &lt;xs:attribute name="Action"&gt;               &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;Action="Default" means that the assignment only takes place if the                 element is currently null.&lt;/xs:documentation&gt;               &lt;/xs:annotation&gt;               &lt;xs:simpleType&gt;                 &lt;xs:restriction base="xs:NCName"&gt;                   &lt;xs:enumeration value="Default"/&gt;                 &lt;/xs:restriction&gt;               &lt;/xs:simpleType&gt;             &lt;/xs:attribute&gt;             &lt;xs:attribute name="Scope" use="required"&gt;               &lt;xs:simpleType&gt;                 &lt;xs:restriction base="xs:NCName"&gt;                   &lt;xs:enumeration value="SubElement"/&gt;                   &lt;xs:enumeration value="Element"/&gt;                   &lt;xs:enumeration value="Repeat"/&gt;                   &lt;xs:enumeration value="Variable"/&gt;                   &lt;xs:enumeration value="Registry"/&gt;                 &lt;/xs:restriction&gt;               &lt;/xs:simpleType&gt;             &lt;/xs:attribute&gt;             &lt;xs:attribute name="ElementID" type="xs:NCName" use="required"/&gt;             &lt;xs:attribute name="SubElementID" type="xs:NCName" /&gt;             &lt;xs:attribute name="RepeatKeyValue" /&gt;           &lt;/xs:extension&gt;         &lt;/xs:simpleContent&gt;       &lt;/xs:complexType&gt;     &lt;/xs:element&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; </pre>

## Complex Type kw\_Repeat\_type

Namespace	No namespace									
Diagram	<pre> classDiagram     class kw_Repeat_type {         &lt;&lt;@ Attributes&gt;&gt;         ID         Type     }     class RepeatDef {         &lt;&lt;The RepeatDef defines the overall aspects of the repeat table such as how rows are added and what are the key elements&gt;&gt;     }     class RepeatHeader {         &lt;&lt;This is the root element that defines the data block for the repeat table. Everything under this element is repeated...&gt;&gt;     }     class RepeatData     class RepeatSummary      kw_Repeat_type "1" -- "1" ID     kw_Repeat_type "1" -- "1" Type     kw_Repeat_type "1" -- "1" RepeatDef     RepeatDef "*" -- "1" RepeatHeader     RepeatHeader "*" -- "1" RepeatData     RepeatData "*" -- "1" RepeatSummary   </pre>									
Used by	Element      Repeat									
Model	RepeatDef , RepeatHeader , RepeatData , RepeatSummary{0,1}									
Children	RepeatData, RepeatDef, RepeatHeader, RepeatSummary									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>ID</td> <td>xs:ID</td> <td>required</td> </tr> <tr> <td>Type</td> <td>xs:NCName</td> <td>required</td> </tr> </tbody> </table>	QName	Type	Use	ID	xs:ID	required	Type	xs:NCName	required
QName	Type	Use								
ID	xs:ID	required								
Type	xs:NCName	required								
Source	<pre> &lt;xsd:complexType name="kw_Repeat_type"&gt;   &lt;xsd:sequence&gt;     &lt;xsd:element ref="RepeatDef" /&gt;     &lt;xsd:element ref="RepeatHeader" /&gt;     &lt;xsd:element ref="RepeatData" /&gt;     &lt;xsd:element ref="RepeatSummary" minOccurs="0" /&gt;   &lt;/xsd:sequence&gt;   &lt;xsd:attribute use="required" ref="ID" /&gt;   &lt;xsd:attribute name="Type" type="xs:NCName" use="required" /&gt; &lt;/xsd:complexType&gt;   </pre>									

## Complex Type kw\_elGroup\_type

Namespace	No namespace
-----------	--------------

Diagram	<pre> classDiagram     class kw_elGroup_type {         @ ID xs:short     }     class BaseElements {         elLabel         elField         elSelect         elCalcField         Repeat         elGroup     }     kw_elGroup_type "1..∞" --&gt; "1..∞" BaseElements     </pre> <p><b>kw_elGroup_type</b></p> <ul style="list-style-type: none"> <li>Attributes:             <ul style="list-style-type: none"> <li><b>@ ID</b> Type <b>xs:short</b></li> </ul> </li> <li>Relationships:             <ul style="list-style-type: none"> <li>Multiplicity 1..∞ to 1..∞ with <b>BaseElements</b>.</li> </ul> </li> </ul> <p><b>BaseElements</b></p> <ul style="list-style-type: none"> <li>Elements:             <ul style="list-style-type: none"> <li><b>elLabel</b> Type <b>kw_elLabel_type</b></li> <li><b>elField</b> Type <b>kw_elField_type</b></li> <li><b>elSelect</b> Type <b>kw_elSelect_type</b></li> <li><b>elCalcField</b></li> <li><b>Repeat</b> Type <b>kw_Repeat_type</b></li> <li><b>elGroup</b> Type <b>kw_elGroup_type</b></li> </ul> </li> </ul> <p>Root element for the structure that defines the repeat or table structure</p>						
Used by	Element <b>elGroup</b>						
Model	<b>elLabel</b>   <b>elField</b>   <b>elSelect</b>   <b>elCalcField</b>   <b>Repeat</b>   <b>elGroup</b>						
Children	Repeat, <b>elCalcField</b> , <b>elField</b> , <b>elGroup</b> , <b>elLabel</b> , <b>elSelect</b>						
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;"><b>QName</b></th> <th style="text-align: left; padding: 2px;"><b>Type</b></th> <th style="text-align: left; padding: 2px;"><b>Use</b></th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;"><b>ID</b></td> <td style="padding: 2px;"><b>xs:short</b></td> <td style="padding: 2px;">optional</td> </tr> </tbody> </table>	<b>QName</b>	<b>Type</b>	<b>Use</b>	<b>ID</b>	<b>xs:short</b>	optional
<b>QName</b>	<b>Type</b>	<b>Use</b>					
<b>ID</b>	<b>xs:short</b>	optional					
Source	<pre> &lt;xss:complexType name="kw_elGroup_type"&gt;     &lt;xss:group maxOccurs="unbounded" ref="BaseElements" /&gt;     &lt;xss:attribute name="ID" type="xs:short" /&gt; &lt;/xss:complexType&gt; </pre>						

## Complex Type kwElementMap

Namespace	No namespace						
Annotations	<p>The purpose of this generic type is to map a source element in the value of the element to the element name in the current application component.</p> <p>Element value - provides the source component</p> <p>Attribute ElementName - provides the name of the element in the local application component that the source element value is mapped to</p>						
Diagram	<pre> classDiagram     class kwElementMap {         @ LocalacElementName         Base Type xs:NCName     }     xs:NCName     </pre> <p><b>kwElementMap</b></p> <p><b>xs:NCName</b></p> <p>The purpose of this generic type is to map a source element in the value of the element to the element name in the...</p> <p>Built-in derived type. NCName represents XML "non-colonized" Names. The base type of NCName is Name.</p> <p><b>Attributes</b></p> <ul style="list-style-type: none"> <li><b>@ LocalacElementName</b></li> </ul>						
Type	extension of xs:NCName						
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;"><b>QName</b></th> <th style="text-align: left; padding: 2px;"><b>Type</b></th> <th style="text-align: left; padding: 2px;"><b>Use</b></th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;"><b>LocalacElementName</b></td> <td style="padding: 2px;"></td> <td style="padding: 2px;">required</td> </tr> </tbody> </table>	<b>QName</b>	<b>Type</b>	<b>Use</b>	<b>LocalacElementName</b>		required
<b>QName</b>	<b>Type</b>	<b>Use</b>					
<b>LocalacElementName</b>		required					
Source	<pre> &lt;xss:complexType name="kwElementMap"&gt;     &lt;xss:annotation&gt;         &lt;xss:documentation&gt;The purpose of this generic type is to map a source element in the value of the element to the element name in the current application component. Element value - provides the source component Attribute ElementName - provides the name of the element in the local application component that the source element value is mapped to&lt;/xss:documentation&gt;     &lt;/xss:annotation&gt;     &lt;xss:simpleContent&gt;         &lt;xss:extension base="xs:NCName"&gt;             &lt;xss:attribute name="LocalacElementName" use="required" /&gt;         &lt;/xss:extension&gt;     &lt;/xss:simpleContent&gt; &lt;/xss:complexType&gt; </pre>						

```
    </xs:simpleContent>
</xs:complexType>
```

## Complex Type GeoPoint\_type

Namespace	No namespace																											
Diagram	<p>The diagram illustrates the structure of the <code>GeoPoint_type</code>. It is defined as an extension of the built-in primitive type <code>xs:string</code>. The <code>GeoPoint_type</code> itself is a complex type with no specific content, only attributes. The attributes listed are: <code>LongSeconds</code>, <code>LongMinutes</code>, <code>LongDegrees</code>, <code>Longitude</code>, <code>LattSeconds</code>, <code>LattMinutes</code>, <code>LattDegrees</code>, and <code>Latitude</code>. A callout box provides a detailed description of <code>xs:string</code> as a built-in primitive type representing character strings in XML.</p>																											
Type	extension of <code>xs:string</code>																											
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><code>LattDegrees</code></td> <td></td> <td>required</td> </tr> <tr> <td><code>LattMinutes</code></td> <td></td> <td>required</td> </tr> <tr> <td><code>LattSeconds</code></td> <td></td> <td>required</td> </tr> <tr> <td><code>Latitude</code></td> <td></td> <td>optional</td> </tr> <tr> <td><code>LongDegrees</code></td> <td></td> <td>required</td> </tr> <tr> <td><code>LongMinutes</code></td> <td></td> <td>required</td> </tr> <tr> <td><code>LongSeconds</code></td> <td></td> <td>required</td> </tr> <tr> <td><code>Longitude</code></td> <td></td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Use	<code>LattDegrees</code>		required	<code>LattMinutes</code>		required	<code>LattSeconds</code>		required	<code>Latitude</code>		optional	<code>LongDegrees</code>		required	<code>LongMinutes</code>		required	<code>LongSeconds</code>		required	<code>Longitude</code>		optional
QName	Type	Use																										
<code>LattDegrees</code>		required																										
<code>LattMinutes</code>		required																										
<code>LattSeconds</code>		required																										
<code>Latitude</code>		optional																										
<code>LongDegrees</code>		required																										
<code>LongMinutes</code>		required																										
<code>LongSeconds</code>		required																										
<code>Longitude</code>		optional																										
Source	<pre>&lt;xs:complexType name="GeoPoint_type"&gt;   &lt;xs:simpleContent&gt;     &lt;xs:extension base="xs:string"&gt;       &lt;xs:attribute name="LongSeconds" use="required"/&gt;       &lt;xs:attribute name="LongMinutes" use="required"/&gt;       &lt;xs:attribute name="LongDegrees" use="required"/&gt;       &lt;xs:attribute name="Longitude" /&gt;       &lt;xs:attribute name="LattSeconds" use="required"/&gt;       &lt;xs:attribute name="LattMinutes" use="required"/&gt;       &lt;xs:attribute name="LattDegrees" use="required"/&gt;       &lt;xs:attribute name="Latitude" /&gt;     &lt;/xs:extension&gt;   &lt;/xs:simpleContent&gt; &lt;/xs:complexType&gt;</pre>																											

## Complex Type NodeType

Namespace	No namespace
Diagram	<p>The diagram illustrates the structure of the <code>NodeType</code>. It is defined as an extension of the built-in derived type <code>xs:NCName</code>. The <code>NodeType</code> itself is a complex type with no specific content, only attributes. The attributes listed are: <code>Seq</code> and <code>Type</code> (with type <code>xs:integer</code>). A callout box provides a detailed description of <code>xs:NCName</code> as a built-in derived type representing XML "non-colonized" Names. The base type of <code>NCName</code> is <code>Name</code>.</p>
Type	extension of <code>xs:NCName</code>

Attributes	QName	Type	Use
	Seq	xs:integer	required
Source	<pre>&lt;xs:complexType name="NodeType"&gt;   &lt;xs:simpleContent&gt;     &lt;xs:extension base="xs:NCName"&gt;       &lt;xs:attribute name="Seq" type="xs:integer" use="required"/&gt;     &lt;/xs:extension&gt;   &lt;/xs:simpleContent&gt; &lt;/xs:complexType&gt;</pre>		

## Complex Type CalcProcessType

Namespace	No namespace
Diagram	
Model	ValidDate   DueDate
Children	DueDate, ValidDate
Source	<pre>&lt;xs:complexType name="CalcProcessType"&gt;   &lt;xs:choice&gt;     &lt;xs:element name="ValidDate" type="fnProcessPeriodValidDate"/&gt;     &lt;xs:element name="DueDate" type="fnProcessPeriodValidDate"/&gt;   &lt;/xs:choice&gt; &lt;/xs:complexType&gt;</pre>

## Complex Type fnProcessPeriodValidDate

Namespace	No namespace
Diagram	
Used by	Elements CalcProcessType/DueDate, CalcProcessType/ValidDate
Model	Calculation , Number , Unit
Children	Calculation, Number, Unit
Source	<pre>&lt;xs:complexType name="fnProcessPeriodValidDate"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="Calculation"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum numeric value - Min: Return the minimum numeric value - Count: Return the number of rows in the column - CountUnique: Return the number of Unique values in the column&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;xs:simpleType&gt;       &lt;xs:restriction base="xs:NCName"&gt;         &lt;xs:enumeration value="AfterFirst"/&gt;         &lt;xs:enumeration value="BeforeLast"/&gt;       &lt;/xs:restriction&gt;     &lt;/xs:simpleType&gt;   &lt;/xs:element&gt;   &lt;xs:element default="0" name="Number" type="xs:int"&gt;     &lt;xs:annotation&gt;       &lt;xs:documentation&gt;This is the element in the repeat table that is the basis for the calculation.&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;   &lt;/xs:element&gt;   &lt;xs:element name="Unit"&gt;</pre>

```

<xs:simpleType>
  <xs:restriction base="xs:NCName">
    <xs:enumeration value="Days" />
    <xs:enumeration value="WorkingDays" />
    <xs:enumeration value="WeekDays" />
    <xs:enumeration value="Weeks" />
    <xs:enumeration value="Months" />
  </xs:restriction>
</xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>

```

### Complex Type fnElementValue

Namespace	No namespace
Diagram	
Model	Calculation , Element
Children	Calculation, Element
Source	<pre> &lt;xs:complexType name="fnElementValue"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="Calculation"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;This element specifies how to deal with various sequence numbers when accessing the element value: LatestSeq: - Checks what the latest sequence number for the ac is, and uses that sequence number SumAllSeq: - Calculates the value accross all sequence numbers&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:NCName"&gt;           &lt;xs:enumeration value="LatestSeq" /&gt;           &lt;xs:enumeration value="SumAllSeq" /&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;     &lt;/xs:element&gt;     &lt;xs:element maxOccurs="1" name="Element" type="KwantuElement_type"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;This is the element that contains the first date (Earliest date)&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:element&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; </pre>

### Complex Type fnProcessPeriodDueDate

Namespace	No namespace
Diagram	
Model	CalcDueDate
Children	CalcDueDate
Source	<pre> &lt;xs:complexType name="fnProcessPeriodDueDate"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="CalcDueDate"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;This element specifies which calculation should be performed - Sum: Sum all numeric values - Max: Return the maximum numeric value - Min: Return the minimum numeric value - Count: Return the number of rows in the column - CountUnique: Return the number of Unique values in the column&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;       &lt;xs:complexType&gt;         &lt;xs:sequence&gt;           &lt;xs:element name="DueDate"&gt; </pre>

```

<xs:complexType>
  <xs:sequence>
    <xs:element name="Calculation">
      <xs:annotation>
        <xs:documentation>This element specifies which calculation should be performed
        - Sum: Sum all numeric values - Max: Return the maximum numeric value - Min: Return the minimum
        numeric value - Count: Return the number of rows in the column - CountUnique: Return the number of
        Unique values in the column</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:NCName">
          <xs:enumeration value="After"/>
          <xs:enumeration value="Before"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element default="0" name="Number" type="xs:int">
      <xs:annotation>
        <xs:documentation>This is the element in the repeat table that is the basis for
        the calculation.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Unit">
      <xs:simpleType>
        <xs:restriction base="xs:NCName">
          <xs:enumeration value="Days"/>
          <xs:enumeration value="WorkingDays"/>
          <xs:enumeration value="WeekDays"/>
          <xs:enumeration value="Weeks"/>
          <xs:enumeration value="Months"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Reference">
      <xs:simpleType>
        <xs:restriction base="xs:NCName">
          <xs:enumeration value="CurrentPeriodStart"/>
          <xs:enumeration value="PreviousPeriodEnd"/>
          <xs:enumeration value="CurrentPeriodEnd"/>
          <xs:enumeration value="ValidDate"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="Seq"/>
</xs:complexType>
</xs:element>
<xs:sequence>
  <xs:attribute name="Type">
    <xs:simpleType>
      <xs:restriction base="xs:NCName">
        <xs:enumeration value="Earliest"/>
        <xs:enumeration value="Latest"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

```

## Complex Type fnProcessScheduleDueDate

Namespace	No namespace
Diagram	
Model	CalcDueDate
Children	CalcDueDate
Source	<pre> &lt;xs:complexType name="fnProcessScheduleDueDate"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="CalcDueDate"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;This element specifies which calculation should be performed - Sum: Sum         all numeric values - Max: Return the maximum numeric value - Min: Return the minimum numeric value         - Count: Return the number of rows in the column - CountUnique: Return the number of Unique values         in the column&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:element&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; </pre>

```

</xs:annotation>
<xs:complexType>
  <xs:sequence>
    <xs:element name="DueDate">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Calculation">
            <xs:annotation>
              <xs:documentation>This element specifies which calculation should be performed
- Sum: Sum all numeric values - Max: Return the maximum numeric value - Min: Return the minimum
numeric value - Count: Return the number of rows in the column - CountUnique: Return the number of
Unique values in the column</xs:documentation>
            </xs:annotation>
            <xs:simpleType>
              <xs:restriction base="xs:NCName">
                <xs:enumeration value="After"/>
                <xs:enumeration value="Before"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element default="0" name="Number" type="xs:int">
            <xs:annotation>
              <xs:documentation>This is the element in the repeat table that is the basis for
the calculation.</xs:documentation>
            </xs:annotation>
          </xs:element>
        <xs:element name="Unit">
          <xs:simpleType>
            <xs:restriction base="xs:NCName">
              <xs:enumeration value="Days"/>
              <xs:enumeration value="WorkingDays"/>
              <xs:enumeration value="WeekDays"/>
              <xs:enumeration value="Weeks"/>
              <xs:enumeration value="Months"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:choice>
          <xs:element name="Reference">
            <xs:simpleType>
              <xs:restriction base="xs:NCName">
                <xs:enumeration value="KwantuObjectCreate"/>
                <xs:enumeration value="ElementID"/>
                <xs:enumeration value="CurrentPeriodEnd"/>
                <xs:enumeration value="ValidDate"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="Element" type="KwantuElement_type"/>
          <xs:element name="ProcessID">
            <xs:complexType>
              <xs:simpleContent>
                <xs:extension base="xs:NCName">
                  <xs:attribute name="Aspect">
                    <xs:simpleType>
                      <xs:restriction base="xs:NCName">
                        <xs:enumeration value="DueDate"/>
                        <xs:enumeration value="ValidDate"/>
                        <xs:enumeration value="ReceivedDate"/>
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:attribute>
                </xs:extension>
                </xs:simpleContent>
              </xs:complexType>
            </xs:element>
          </xs:choice>
        </xs:sequence>
        <xs:attribute name="Seq"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="Type">
    <xs:simpleType>
      <xs:restriction base="xs:NCName">
        <xs:enumeration value="Max"/>
        <xs:enumeration value="Min"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>

```

| </xs:complexType> |

## Complex Type kw\_Label\_type

Namespace	No namespace
Diagram	<pre> graph LR     kw_Label_type[kw_Label_type] --&gt; Mixed     kw_Label_type --&gt; true     Mixed --&gt; Sequence     Sequence --&gt; Int     Int -- "1..∞" --&gt; Extension     Extension --&gt; Type     Type -- "Extension of 'xs:string'" --&gt; End     </pre>
Properties	mixed: true
Model	Int+
Children	Int
Source	<pre> &lt;xs:complexType mixed="true" name="kw_Label_type"&gt;   &lt;xs:sequence&gt;     &lt;xs:element maxOccurs="unbounded" name="Int"&gt;       &lt;xs:complexType&gt;         &lt;xs:simpleContent&gt;           &lt;xs:extension base="xs:string"&gt;             &lt;xs:attribute name="lang"/&gt;           &lt;/xs:extension&gt;         &lt;/xs:simpleContent&gt;       &lt;/xs:complexType&gt;     &lt;/xs:element&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; </pre>

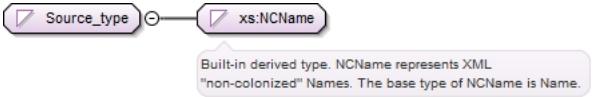
## Simple Type(s)

### Simple Type ColumnDefReportFieldtype

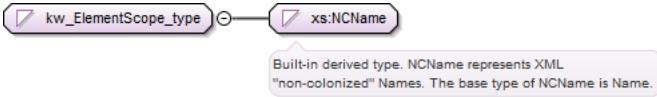
Namespace	No namespace																				
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>																				
Type	restriction of xs:string																				
Facets	<table border="0"> <tr> <td>enumeration</td> <td>Varchar</td> </tr> <tr> <td>enumeration</td> <td>Char</td> </tr> <tr> <td>enumeration</td> <td>Date</td> </tr> <tr> <td>enumeration</td> <td>Time</td> </tr> <tr> <td>enumeration</td> <td>DateTime</td> </tr> <tr> <td>enumeration</td> <td>Numeric</td> </tr> <tr> <td>enumeration</td> <td>Integer</td> </tr> <tr> <td>enumeration</td> <td>Memo</td> </tr> <tr> <td>enumeration</td> <td>Boolean</td> </tr> <tr> <td>enumeration</td> <td>Int</td> </tr> </table>	enumeration	Varchar	enumeration	Char	enumeration	Date	enumeration	Time	enumeration	DateTime	enumeration	Numeric	enumeration	Integer	enumeration	Memo	enumeration	Boolean	enumeration	Int
enumeration	Varchar																				
enumeration	Char																				
enumeration	Date																				
enumeration	Time																				
enumeration	DateTime																				
enumeration	Numeric																				
enumeration	Integer																				
enumeration	Memo																				
enumeration	Boolean																				
enumeration	Int																				
Used by	Element ReportDefaultColumnType/Fieldtype																				
Source	<pre> &lt;xs:simpleType name="ColumnDefReportFieldtype"&gt;   &lt;xs:restriction base="xs:string"&gt;     &lt;xs:enumeration value="Varchar"/&gt;     &lt;xs:enumeration value="Char"/&gt;     &lt;xs:enumeration value="Date"/&gt;     &lt;xs:enumeration value="Time"/&gt;     &lt;xs:enumeration value="DateTime"/&gt;     &lt;xs:enumeration value="Numeric"/&gt;     &lt;xs:enumeration value="Integer"/&gt;     &lt;xs:enumeration value="Memo"/&gt;     &lt;xs:enumeration value="Boolean"/&gt;     &lt;xs:enumeration value="Int"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; </pre>																				

### Simple Type Source\_type

Namespace	No namespace
-----------	--------------

Diagram																	
	Built-in derived type. NCName represents XML "non-colonized" Names. The base type of NCName is Name.																
Type	restriction of xs:NCName																
Facets	<table border="0"> <tr><td>enumeration</td><td>Code</td></tr> <tr><td>enumeration</td><td>Name</td></tr> <tr><td>enumeration</td><td>KwantuObjectID</td></tr> <tr><td>enumeration</td><td>Element</td></tr> <tr><td>enumeration</td><td>SequenceNo</td></tr> <tr><td>enumeration</td><td>ValidDate</td></tr> <tr><td>enumeration</td><td>RepeatSequenceNo</td></tr> <tr><td>enumeration</td><td>ProfileID</td></tr> </table>	enumeration	Code	enumeration	Name	enumeration	KwantuObjectID	enumeration	Element	enumeration	SequenceNo	enumeration	ValidDate	enumeration	RepeatSequenceNo	enumeration	ProfileID
enumeration	Code																
enumeration	Name																
enumeration	KwantuObjectID																
enumeration	Element																
enumeration	SequenceNo																
enumeration	ValidDate																
enumeration	RepeatSequenceNo																
enumeration	ProfileID																
Used by	Attributes ReportColumnType/@Source, ReportDefaultColumnType/@Source, kw_elSelect_type/EntryMode/@Source																
Source	<pre>&lt;xs:simpleType name="Source_type"&gt;   &lt;xs:restriction base="xs:NCName"&gt;     &lt;xs:enumeration value="Code"/&gt;     &lt;xs:enumeration value="Name"/&gt;     &lt;xs:enumeration value="KwantuObjectID"/&gt;     &lt;xs:enumeration value="Element"/&gt;     &lt;xs:enumeration value="SequenceNo"/&gt;     &lt;xs:enumeration value="ValidDate"/&gt;     &lt;xs:enumeration value="RepeatSequenceNo"/&gt;     &lt;xs:enumeration value="ProfileID"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>																

### Simple Type kw\_ElementScope\_type

Namespace	No namespace																				
Diagram																					
	Built-in derived type. NCName represents XML "non-colonized" Names. The base type of NCName is Name.																				
Type	restriction of xs:NCName																				
Facets	<table border="0"> <tr><td>enumeration</td><td>ProcessVariable</td></tr> <tr><td>enumeration</td><td>SubProcessVariable</td></tr> <tr><td>enumeration</td><td>ElementID</td></tr> <tr><td>enumeration</td><td>ProcessObject</td></tr> <tr><td>enumeration</td><td>SubProcessObject</td></tr> <tr><td>enumeration</td><td>ServiceURL</td></tr> <tr><td>enumeration</td><td>Repeat</td></tr> <tr><td>enumeration</td><td>Registry</td></tr> <tr><td>enumeration</td><td>Sequence</td></tr> <tr><td>enumeration</td><td>ValidDate</td></tr> </table>	enumeration	ProcessVariable	enumeration	SubProcessVariable	enumeration	ElementID	enumeration	ProcessObject	enumeration	SubProcessObject	enumeration	ServiceURL	enumeration	Repeat	enumeration	Registry	enumeration	Sequence	enumeration	ValidDate
enumeration	ProcessVariable																				
enumeration	SubProcessVariable																				
enumeration	ElementID																				
enumeration	ProcessObject																				
enumeration	SubProcessObject																				
enumeration	ServiceURL																				
enumeration	Repeat																				
enumeration	Registry																				
enumeration	Sequence																				
enumeration	ValidDate																				
Used by	Attribute KwantuElement_type/@Scope																				
Source	<pre>&lt;xs:simpleType name="kw_ElementScope_type"&gt;   &lt;xs:restriction base="xs:NCName"&gt;     &lt;xs:enumeration value="ProcessVariable"/&gt;     &lt;xs:enumeration value="SubProcessVariable"/&gt;     &lt;xs:enumeration value="ElementID"/&gt;     &lt;xs:enumeration value="ProcessObject"/&gt;     &lt;xs:enumeration value="SubProcessObject"/&gt;     &lt;xs:enumeration value="ServiceURL"/&gt;     &lt;xs:enumeration value="Repeat"/&gt;     &lt;xs:enumeration value="Registry"/&gt;     &lt;xs:enumeration value="Sequence"/&gt;     &lt;xs:enumeration value="ValidDate"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>																				

### Simple Type KwantuFormDataTypes

Namespace	No namespace																								
Diagram	 <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;">         Built-in primitive type. The string datatype represents character strings in XML.       </div>																								
Type	restriction of xs:string																								
Facets	<table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">enumeration</td> <td style="width: 85%;">String</td> </tr> <tr> <td>enumeration</td> <td>Integer</td> </tr> <tr> <td>enumeration</td> <td>Float</td> </tr> <tr> <td>enumeration</td> <td>Memo</td> </tr> <tr> <td>enumeration</td> <td>Email</td> </tr> <tr> <td>enumeration</td> <td>URI</td> </tr> <tr> <td>enumeration</td> <td>Date</td> </tr> <tr> <td>enumeration</td> <td>CoordinateDMS</td> </tr> <tr> <td>enumeration</td> <td>GeoPoint</td> </tr> <tr> <td>enumeration</td> <td>DataTypes</td> </tr> <tr> <td>enumeration</td> <td>YesNo</td> </tr> <tr> <td>enumeration</td> <td>Boolean</td> </tr> </table>	enumeration	String	enumeration	Integer	enumeration	Float	enumeration	Memo	enumeration	Email	enumeration	URI	enumeration	Date	enumeration	CoordinateDMS	enumeration	GeoPoint	enumeration	DataTypes	enumeration	YesNo	enumeration	Boolean
enumeration	String																								
enumeration	Integer																								
enumeration	Float																								
enumeration	Memo																								
enumeration	Email																								
enumeration	URI																								
enumeration	Date																								
enumeration	CoordinateDMS																								
enumeration	GeoPoint																								
enumeration	DataTypes																								
enumeration	YesNo																								
enumeration	Boolean																								
Used by	<table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Attributes</td> <td>AssignSubElement_type/AssignCode/@DataType, AssignSubElement_type/AssignName/@DataType, AssignSubElement_type/AssignSubElement/@DataType, Condition_type/@DataType, Default_type/InitiatialValue/@DataType, Default_type/Value/@DataType, KwantuElement_type/@DataType, Kwantu_Filter_type/Parameters/Constant/@DataType</td> </tr> <tr> <td>Elements</td> <td>DataType, kw_elLabel_type/Display/Format</td> </tr> </table>	Attributes	AssignSubElement_type/AssignCode/@DataType, AssignSubElement_type/AssignName/@DataType, AssignSubElement_type/AssignSubElement/@DataType, Condition_type/@DataType, Default_type/InitiatialValue/@DataType, Default_type/Value/@DataType, KwantuElement_type/@DataType, Kwantu_Filter_type/Parameters/Constant/@DataType	Elements	DataType, kw_elLabel_type/Display/Format																				
Attributes	AssignSubElement_type/AssignCode/@DataType, AssignSubElement_type/AssignName/@DataType, AssignSubElement_type/AssignSubElement/@DataType, Condition_type/@DataType, Default_type/InitiatialValue/@DataType, Default_type/Value/@DataType, KwantuElement_type/@DataType, Kwantu_Filter_type/Parameters/Constant/@DataType																								
Elements	DataType, kw_elLabel_type/Display/Format																								
Source	<pre> &lt;xssimpleType name="KwantuFormDataTypes"&gt;   &lt;xsrrestriction base="xs:string"&gt;     &lt;xsenumeration value="String"/&gt;     &lt;xsenumeration value="Integer"/&gt;     &lt;xsenumeration value="Float"/&gt;     &lt;xsenumeration value="Memo"/&gt;     &lt;xsenumeration value="Email"/&gt;     &lt;xsenumeration value="URI"/&gt;     &lt;xsenumeration value="Date"/&gt;     &lt;xsenumeration value="CoordinateDMS"/&gt;     &lt;xsenumeration value="GeoPoint"/&gt;     &lt;xsenumeration value="DataTypes"/&gt;     &lt;xsenumeration value="YesNo"/&gt;     &lt;xsenumeration value="Boolean"/&gt;   &lt;/xsrrestriction&gt; &lt;/xssimpleType&gt; </pre>																								

### Simple Type kw\_Xpath\_type

Namespace	No namespace
Diagram	 <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;">         Built-in primitive type. The string datatype represents character strings in XML.       </div>
Type	xs:string
Used by	Attribute KwantuElement_type/@Xpath
Source	<pre> &lt;xssimpleType name="kw_Xpath_type"&gt;   &lt;xsrrestriction base="xs:string"/&gt; &lt;/xssimpleType&gt; </pre>

### Simple Type kw\_Show\_type

Namespace	No namespace
Diagram	 <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;">         Built-in derived type. NCName represents XML "non-colonized" Names. The base type of NCName is Name.       </div>

Type	restriction of xs:NCName	
Facets	enumeration	Yes
	enumeration	No
	enumeration	Conditional
Used by	Attribute	@Show
Source	<pre>&lt;xs:simpleType name="kw_Show_type"&gt;   &lt;xs:restriction base="xs:NCName"&gt;     &lt;xs:enumeration value="Yes"/&gt;     &lt;xs:enumeration value="No"/&gt;     &lt;xs:enumeration value="Conditional"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>	

### Simple Type DisplayFormat\_type

Namespace	No namespace	
Diagram	<p>Built-in derived type. NCName represents XML "non-colonized" Names. The base type of NCName is Name.</p>	
Type	restriction of xs:NCName	
Facets	enumeration	Float
	enumeration	Numeric
	enumeration	Date
	enumeration	String
	enumeration	Memo
	enumeration	Boolean
	enumeration	Integer
Used by	Elements	elCalcField/Display/Format, kw_elField_type/Display/Format, kw_elSelect_type/Display/Format
Source	<pre>&lt;xs:simpleType name="DisplayFormat_type"&gt;   &lt;xs:restriction base="xs:NCName"&gt;     &lt;xs:enumeration value="Float"/&gt;     &lt;xs:enumeration value="Numeric"/&gt;     &lt;xs:enumeration value="Date"/&gt;     &lt;xs:enumeration value="String"/&gt;     &lt;xs:enumeration value="Memo"/&gt;     &lt;xs:enumeration value="Boolean"/&gt;     &lt;xs:enumeration value="Integer"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>	

### Simple Type kwantu\_comparison\_types

Namespace	No namespace	
Diagram	<p>Built-in derived type. Name represents XML Names. The base type of Name is token.</p>	
Type	restriction of xs:Name	
Facets	enumeration	UniqueInColumn
	enumeration	GreaterThan
	enumeration	LessThan
	enumeration	MaxLength
	enumeration	UniqueInDomain
	enumeration	GreaterThanOrEqual
	enumeration	LessThanOrEqual
	enumeration	Equal
	enumeration	NotNull

	enumeration	NotEqual
	enumeration	CompositeUnique
Used by	Attributes	Condition_type/@Operator, Kwantu_Filter_type/@Type, fnHistoryTotal/Filters/Filter/@Type
Source	<pre>&lt;xs:simpleType name="kwantu_comparison_types"&gt;   &lt;xs:restriction base="xs:Name"&gt;     &lt;xs:enumeration value="UniqueInColumn"/&gt;     &lt;xs:enumeration value="GreaterThan"/&gt;     &lt;xs:enumeration value="LessThan"/&gt;     &lt;xs:enumeration value="MaxLength"/&gt;     &lt;xs:enumeration value="UniqueInDomain"/&gt;     &lt;xs:enumeration value="GreaterThanOrEqual"/&gt;     &lt;xs:enumeration value="LessThanOrEqual"/&gt;     &lt;xs:enumeration value="Equal"/&gt;     &lt;xs:enumeration value="NotNull"/&gt;     &lt;xs:enumeration value="NotEqual"/&gt;     &lt;xs:enumeration value="CompositeUnique"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>	

### Simple Type FilterElement\_type

Namespace	No namespace									
Diagram										
Type	restriction of xs:NCName									
Facets	<table border="1"> <tr> <td>enumeration</td> <td>ValidDate</td> </tr> <tr> <td>enumeration</td> <td>SequenceNumber</td> </tr> <tr> <td>enumeration</td> <td>ElementInReference</td> </tr> <tr> <td>enumeration</td> <td>CurrentElement</td> </tr> </table>		enumeration	ValidDate	enumeration	SequenceNumber	enumeration	ElementInReference	enumeration	CurrentElement
enumeration	ValidDate									
enumeration	SequenceNumber									
enumeration	ElementInReference									
enumeration	CurrentElement									
Used by	Attributes Kwantu_Filter_type/@FilterElement, fnHistoryTotal/Filters/Filter/@FilterElement									
Source	<pre>&lt;xs:simpleType name="FilterElement_type"&gt;   &lt;xs:restriction base="xs:NCName"&gt;     &lt;xs:enumeration value="ValidDate"/&gt;     &lt;xs:enumeration value="SequenceNumber"/&gt;     &lt;xs:enumeration value="ElementInReference"/&gt;     &lt;xs:enumeration value="CurrentElement"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>									

### Simple Type ColumnFieldtype

Namespace	No namespace																					
Diagram																						
Type	restriction of xs:string																					
Facets	<table border="1"> <tr> <td>enumeration</td> <td>Varchar</td> </tr> <tr> <td>enumeration</td> <td>Char</td> </tr> <tr> <td>enumeration</td> <td>Date</td> </tr> <tr> <td>enumeration</td> <td>Time</td> </tr> <tr> <td>enumeration</td> <td>DateTime</td> </tr> <tr> <td>enumeration</td> <td>Numeric</td> </tr> <tr> <td>enumeration</td> <td>Boolean</td> </tr> <tr> <td>enumeration</td> <td>Int</td> </tr> <tr> <td>enumeration</td> <td>Decimal</td> </tr> <tr> <td>enumeration</td> <td>Mediumblob</td> </tr> </table>		enumeration	Varchar	enumeration	Char	enumeration	Date	enumeration	Time	enumeration	DateTime	enumeration	Numeric	enumeration	Boolean	enumeration	Int	enumeration	Decimal	enumeration	Mediumblob
enumeration	Varchar																					
enumeration	Char																					
enumeration	Date																					
enumeration	Time																					
enumeration	DateTime																					
enumeration	Numeric																					
enumeration	Boolean																					
enumeration	Int																					
enumeration	Decimal																					
enumeration	Mediumblob																					
Used by	Element ReportColumnType/Fieldtype																					

Source	<pre>&lt;xss:simpleType name="ColumnFieldtype"&gt;   &lt;xss:restriction base="xss:string"&gt;     &lt;xss:enumeration value="Varchar" /&gt;     &lt;xss:enumeration value="Char" /&gt;     &lt;xss:enumeration value="Date" /&gt;     &lt;xss:enumeration value="Time" /&gt;     &lt;xss:enumeration value="DateTime" /&gt;     &lt;xss:enumeration value="Numeric" /&gt;     &lt;xss:enumeration value="Boolean" /&gt;     &lt;xss:enumeration value="Int" /&gt;     &lt;xss:enumeration value="Decimal" /&gt;     &lt;xss:enumeration value="Mediumblob" /&gt;   &lt;/xss:restriction&gt; &lt;/xss:simpleType&gt;</pre>
--------	--

## Attribute(s)

### Attribute Version / @Number

Namespace	No namespace	
Type	xs:decimal	
Properties	use: required	
Used by	Element	Version
Source	<pre>&lt;xss:attribute name="Number" use="required" type="xs:decimal" /&gt;</pre>	

### Attribute Identification / @BaseComponent

Namespace	No namespace	
Type	xs:NCName	
Properties	use: required	
Used by	Element	Identification
Source	<pre>&lt;xss:attribute name="BaseComponent" use="required" type="xs:NCName" /&gt;</pre>	

### Attribute Identification / @CurrentVersion

Namespace	No namespace	
Type	xs:decimal	
Properties	use: required	
Used by	Element	Identification
Source	<pre>&lt;xss:attribute name="CurrentVersion" use="required" type="xs:decimal" /&gt;</pre>	

### Attribute Identification / @DevVersion

Namespace	No namespace	
Type	xs:decimal	
Properties	use: required	
Used by	Element	Identification
Source	<pre>&lt;xss:attribute name="DevVersion" use="required" type="xs:decimal" /&gt;</pre>	

### Attribute Identification / @ID

Namespace	No namespace	
Type	xs:NCName	
Properties	use: required	
Used by	Element	Identification
Source	<pre>&lt;xss:attribute name="ID" use="required" type="xs:NCName" /&gt;</pre>	

### **Attribute ReportDefaultColumnType / @Source**

Namespace	No namespace	
Type	Source_type	
Properties	use: required	
Facets	enumeration	Code
	enumeration	Name
	enumeration	KwantuObjectID
	enumeration	Element
	enumeration	SequenceNo
	enumeration	ValidDate
	enumeration	RepeatSequenceNo
	enumeration	ProfileID
Used by	Complex Type	ReportDefaultColumnType
Source	<code>&lt;xss:attribute name="Source" use="required" type="Source_type" /&gt;</code>	

### **Attribute ReportDefaults / Tables / Table / @Type**

Namespace	No namespace	
Type	restriction of xs:NCName	
Properties	content: simple	
Facets	enumeration	Summary
	enumeration	Detail
Used by	Element	ReportDefaults/Tables/Table
Source	<code>&lt;xss:attribute name="Type"&gt;   &lt;xss:simpleType&gt;     &lt;xss:restriction base="xs:NCName"&gt;       &lt;xss:enumeration value="Summary"/&gt;       &lt;xss:enumeration value="Detail"/&gt;     &lt;/xss:restriction&gt;   &lt;/xss:simpleType&gt; &lt;/xss:attribute&gt;</code>	

### **Attribute ReportDefaults / Tables / Table / @Name**

Namespace	No namespace	
Type	xs:ID	
Properties	content: simple	
Used by	Element	ReportDefaults/Tables/Table
Source	<code>&lt;xss:attribute name="Name" type="xs:ID" /&gt;</code>	

### **Attribute kw\_UIVersions\_type / UIVersion / @Name**

Namespace	No namespace	
Properties	use: required	
Used by	Element	kw_UIVersions_type/UIVersion
Source	<code>&lt;xss:attribute name="Name" use="required" /&gt;</code>	

### **Attribute KwantuElement\_type / @Seq**

Namespace	No namespace	
Annotations	This is used as a sequence number to access multiple elements. Must be renamed to Seq.	
Type	xs:int	

Properties	content:	simple
Used by	Complex Type	KwantuElement_type
Source	<pre>&lt;xs:attribute name="Seq" type="xs:int"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This is used as a sequence number to access multiple elements. Must be renamed to Seq.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>	

#### **Attribute KwantuElement\_type / @Id**

Namespace	No namespace
Annotations	This is used as a sequence number to access multiple elements. Must be renamed to Seq.
Type	xs:int
Properties	content:
Used by	Complex Type KwantuElement_type
Source	<pre>&lt;xs:attribute name="Id" type="xs:int"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This is used as a sequence number to access multiple elements. Must be renamed to Seq.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>

#### **Attribute KwantuElement\_type / @ID**

Namespace	No namespace
Used by	Complex Type KwantuElement_type
Source	<pre>&lt;xs:attribute name="ID" /&gt;</pre>

#### **Attribute KwantuElement\_type / @Component**

Namespace	No namespace
Type	xs:NCName
Properties	content:
Used by	Complex Type KwantuElement_type
Source	<pre>&lt;xs:attribute name="Component" type="xs:NCName" /&gt;</pre>

#### **Attribute KwantuElement\_type / @Element**

Namespace	No namespace
Type	xs:NCName
Properties	content:
Used by	Complex Type KwantuElement_type
Source	<pre>&lt;xs:attribute name="Element" type="xs:NCName" /&gt;</pre>

#### **Attribute KwantuElement\_type / @SubElement**

Namespace	No namespace
Type	xs:NCName
Properties	content:
Used by	Complex Type KwantuElement_type
Source	<pre>&lt;xs:attribute name="SubElement" type="xs:NCName" /&gt;</pre>

#### **Attribute KwantuElement\_type / @acSequenceRule**

Namespace	No namespace
-----------	--------------

Type	restriction of xs:NCName	
Properties	use:	required
Facets	enumeration	Current
	enumeration	LatestAuthorized
	enumeration	First
	enumeration	LatestSeq
	enumeration	FirsttoLatestSeq
Used by	Complex Type	KwantuElement_type
Source	<pre>&lt;xs:attribute name="acSequenceRule" use="required"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="Current"/&gt;       &lt;xs:enumeration value="LatestAuthorized"/&gt;       &lt;xs:enumeration value="First"/&gt;       &lt;xs:enumeration value="LatestSeq"/&gt;       &lt;xs:enumeration value="FirsttoLatestSeq"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</pre>	

#### Attribute KwantuElement\_type / @RepeatKeyRule

Namespace	No namespace	
Type	restriction of xs:NCName	
Properties	content:	simple
Facets	enumeration	MatchRepeatKey
Used by	Complex Type	KwantuElement_type
Source	<pre>&lt;xs:attribute name="RepeatKeyRule"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="MatchRepeatKey"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</pre>	

#### Attribute KwantuElement\_type / @Scope

Namespace	No namespace	
Type	kw_ElementScope_type	
Properties	content:	simple
Facets	enumeration	ProcessVariable
	enumeration	SubProcessVariable
	enumeration	ElementID
	enumeration	ProcessObject
	enumeration	SubProcessObject
	enumeration	ServiceURL
	enumeration	Repeat
	enumeration	Registry
	enumeration	Sequence
	enumeration	ValidDate
Used by	Complex Type	KwantuElement_type
Source	<pre>&lt;xs:attribute name="Scope" type="kw_ElementScope_type"/&gt;</pre>	

#### Attribute KwantuElement\_type / @DataType

Namespace	No namespace
-----------	--------------

Type	KwantuFormDataTypes	
Properties	use:	required
Facets	enumeration	String
	enumeration	Integer
	enumeration	Float
	enumeration	Memo
	enumeration	Email
	enumeration	URI
	enumeration	Date
	enumeration	CoordinateDMS
	enumeration	GeoPoint
	enumeration	DataTypes
Used by	Complex Type	KwantuElement_type
Source	<code>&lt;xss:attribute name="DataType" type="KwantuFormDataTypes" use="required"/&gt;</code>	

#### **Attribute KwantuElement\_type / @Xpath**

Namespace	No namespace	
Type	kw_Xpath_type	
Properties	content: simple	
Used by	Complex Type	KwantuElement_type
Source	<code>&lt;xss:attribute name="Xpath" type="kw_Xpath_type"/&gt;</code>	

#### **Attribute Header / Dependencies / Dependency / @ID**

Namespace	No namespace	
Used by	Element	Header/Dependencies/Dependency
Source	<code>&lt;xss:attribute name="ID"/&gt;</code>	

#### **Attribute Header / Dependencies / Dependency / @ComponentID**

Namespace	No namespace	
Used by	Element	Header/Dependencies/Dependency
Source	<code>&lt;xss:attribute name="Component ID"/&gt;</code>	

#### **Attribute kw\_UIVersionElements\_type / UIElement / @Name**

Namespace	No namespace	
Properties	use:	required
Used by	Element	kw_UIVersionElements_type/UIElement
Source	<code>&lt;xss:attribute name="Name" use="required"/&gt;</code>	

#### **Attribute kw\_UIVersionElements\_type / UIElement / @Type**

Namespace	No namespace	
Type	restriction of xs:NCName	
Properties	use:	required
Facets	enumeration	Default

	enumeration	ReadOnly
	enumeration	Hidden
Used by	Element	kw_UIVersionElements_type/UIElement
Source	<pre>&lt;xs:attribute name="Type" use="required"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="Default"/&gt;       &lt;xs:enumeration value=" Readonly"/&gt;       &lt;xs:enumeration value=" Hidden"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</pre>	

#### Attribute @Show

Namespace	No namespace	
Type	kw_Show_type	
Properties	content:	simple
Facets	enumeration	Yes
	enumeration	No
	enumeration	Conditional
Used by	Elements	Label, RepeatDef/HeaderRow, SummaryRow, elCalcField/Display, kw_elField_type/Display, kw_elLabel_type/Display, kw_elSelect_type/Display
Source	<pre>&lt;xs:attribute name="Show" type="kw_Show_type" /&gt;</pre>	

#### Attribute kw\_Xform\_type / @Appearance

Namespace	No namespace	
Type	restriction of xs:Name	
Properties	use:	required
Facets	enumeration	Output
	enumeration	Heading
	enumeration	Subheading
	enumeration	Currency
	enumeration	Integer
Used by	Complex Type	kw_Xform_type
Source	<pre>&lt;xs:attribute name="Appearance" use="required"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:Name"&gt;       &lt;xs:enumeration value="Output"/&gt;       &lt;xs:enumeration value="Heading"/&gt;       &lt;xs:enumeration value="Subheading"/&gt;       &lt;xs:enumeration value="Currency"/&gt;       &lt;xs:enumeration value="Integer"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</pre>	

#### Attribute kw\_Xform\_type / @CSS

Namespace	No namespace	
Type	restriction of xs:Name	
Properties	use:	required
Facets	enumeration	Subheading
	enumeration	Kwantu-Standard-Label
	enumeration	man-label
	enumeration	Heading

	enumeration	DatePicker
	enumeration	None
Used by	Complex Type	kw_Xform_type
Source	<pre>&lt;xs:attribute name="CSS" use="required"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:Name"&gt;       &lt;xs:enumeration value="Subheading"/&gt;       &lt;xs:enumeration value="Kwantu-Standard-Label"/&gt;       &lt;xs:enumeration value="man-label"/&gt;       &lt;xs:enumeration value="Heading"/&gt;       &lt;xs:enumeration value="DatePicker"/&gt;       &lt;xs:enumeration value="None"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</pre>	

#### **Attribute kw\_elLabel\_type / Display / Condition / @Operator**

Namespace	No namespace	
Type	restriction of xs:Name	
Properties	use:	required
Facets	enumeration	GreaterThan
	enumeration	LessThan
	enumeration	GreaterThanOrEqual
	enumeration	LessThanOrEqual
	enumeration	Equal
	enumeration	NotEqual
Used by	Element	kw_elLabel_type/Display/Condition
Source	<pre>&lt;xs:attribute name="Operator" use="required"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:Name"&gt;       &lt;xs:enumeration value="GreaterThan"/&gt;       &lt;xs:enumeration value="LessThan"/&gt;       &lt;xs:enumeration value="GreaterThanOrEqual"/&gt;       &lt;xs:enumeration value="LessThanOrEqual"/&gt;       &lt;xs:enumeration value="Equal"/&gt;       &lt;xs:enumeration value="NotEqual"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</pre>	

#### **Attribute kw\_elLabel\_type / Display / @Span**

Namespace	No namespace	
Type	xs:int	
Properties	content:	simple
Used by	Element	kw_elLabel_type/Display
Source	<pre>&lt;xs:attribute name="Span" type="xs:int"/&gt;</pre>	

#### **Attribute kw\_Help\_type / @Show**

Namespace	No namespace	
Type	restriction of xs:NCName	
Properties	use:	required
Facets	enumeration	Yes
	enumeration	No
Used by	Complex Type	kw_Help_type
Source	<pre>&lt;xs:attribute name="Show" use="required"&gt;   &lt;xs:simpleType&gt;</pre>	

```

<xs:restriction base="xs:NCName">
  <xs:enumeration value="Yes"/>
  <xs:enumeration value="No"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>

```

## Attribute @ID

Namespace	No namespace
Type	xs:ID
Properties	content: simple
Used by	Elements elCalcField, elSelectGroup
	Complex Types kw_Repeat_type, kw_elField_type, kw_elLabel_type, kw_elSelect_type
Source	<xs:attribute name="ID" type="xs:ID"/>

## Attribute kw\_elLabel\_type / @Type

Namespace	No namespace
Type	restriction of xs:NCName
Properties	use: required
Facets	enumeration Label
Used by	Complex Type kw_elLabel_type
Source	<pre> &lt;xs:attribute name="Type" use="required"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="Label"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; </pre>

## Attribute kw\_elField\_type / Display / Xform / @Appearance

Namespace	No namespace
Type	restriction of xs:Name
Properties	use: required
Facets	enumeration Currency
	enumeration Numeric
	enumeration Text
	enumeration DatePicker
	enumeration Memo
	enumeration Integer
	enumeration Percentage
	enumeration Output
	enumeration Checkbox
Used by	Element kw_elField_type/Display/Xform
Source	<pre> &lt;xs:attribute name="Appearance" use="required"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:Name"&gt;       &lt;xs:enumeration value="Currency"/&gt;       &lt;xs:enumeration value="Numeric"/&gt;       &lt;xs:enumeration value="Text"/&gt;       &lt;xs:enumeration value="DatePicker"/&gt;       &lt;xs:enumeration value="Memo"/&gt;       &lt;xs:enumeration value="Integer"/&gt;       &lt;xs:enumeration value="Percentage"/&gt;       &lt;xs:enumeration value="Output"/&gt;       &lt;xs:enumeration value="Checkbox"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; </pre>

	<code>&lt;/xs:attribute&gt;</code>
--	------------------------------------

### **Attribute kw\_elField\_type / Display / Xform / @CSS**

Namespace	No namespace	
Type	restriction of xs:Name	
Properties	content: simple	
Facets	enumeration	SmallField
	enumeration	MediumField
	enumeration	LargeField
	enumeration	man-label
Used by	Element	kw_elField_type/Display/Xform
Source	<pre>&lt;xs:attribute name="CSS"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:Name"&gt;       &lt;xs:enumeration value="SmallField"/&gt;       &lt;xs:enumeration value="MediumField"/&gt;       &lt;xs:enumeration value="LargeField"/&gt;       &lt;xs:enumeration value="man-label"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</pre>	

### **Attribute Condition\_type / @Operator**

Namespace	No namespace	
Type	kwantu_comparison_types	
Properties	use: required	
Facets	enumeration	UniqueInColumn
	enumeration	GreaterThan
	enumeration	LessThan
	enumeration	MaxLength
	enumeration	UniqueInDomain
	enumeration	GreaterThanOrEqual
	enumeration	LessThanOrEqual
	enumeration	Equal
	enumeration	NotNull
	enumeration	NotEqual
Used by	Complex Type	Condition_type
Source	<pre>&lt;xs:attribute name="Operator" use="required" type="kwantu_comparison_types"/&gt;</pre>	

### **Attribute Condition\_type / @Value**

Namespace	No namespace	
Used by	Complex Type	Condition_type
Source	<pre>&lt;xs:attribute name="Value"/&gt;</pre>	

### **Attribute Condition\_type / @DataType**

Namespace	No namespace	
Type	KwantuForm DataTypes	
Properties	content: simple	
Facets	enumeration	String

	enumeration	Integer
	enumeration	Float
	enumeration	Memo
	enumeration	Email
	enumeration	URI
	enumeration	Date
	enumeration	CoordinateDMS
	enumeration	GeoPoint
	enumeration	DataTypes
	enumeration	YesNo
	enumeration	Boolean
Used by	Complex Type	Condition_type
Source	<code>&lt;xss:attribute name="DataType" type="KwantuFormDataTypes" /&gt;</code>	

#### **Attribute Default\_type / Value / @DataType**

Namespace	No namespace	
Type	KwantuFormDataTypes	
Properties	content: simple	
Facets	enumeration String enumeration Integer enumeration Float enumeration Memo enumeration Email enumeration URI enumeration Date enumeration CoordinateDMS enumeration GeoPoint enumeration DataTypes enumeration YesNo enumeration Boolean	
Used by	Element Default_type/Value	
Source	<code>&lt;xss:attribute name="DataType" type="KwantuFormDataTypes" /&gt;</code>	

#### **Attribute Default\_type / InitiatialValue / @DataType**

Namespace	No namespace
Type	KwantuFormDataTypes
Properties	content: simple
Facets	enumeration String enumeration Integer enumeration Float enumeration Memo enumeration Email enumeration URI enumeration Date enumeration CoordinateDMS enumeration GeoPoint enumeration DataTypes enumeration YesNo

	enumeration	Boolean
Used by	Element	Default_type/InitiatialValue
Source	<xs:attribute name="DataType" type="KwantuFormDataTypes" />	

#### **Attribute Kwantu\_Filter\_type / Parameters / Constant / @DataType**

Namespace	No namespace	
Type	KwantuFormDataTypes	
Properties	content: simple	
Facets	enumeration String enumeration Integer enumeration Float enumeration Memo enumeration Email enumeration URI enumeration Date enumeration CoordinateDMS enumeration GeoPoint enumeration DataTypes enumeration YesNo enumeration Boolean	
Used by	Element	Kwantu_Filter_type/Parameters/Constant
Source	<xs:attribute name="DataType" type="KwantuFormDataTypes" />	

#### **Attribute Kwantu\_Filter\_type / @Seq**

Namespace	No namespace	
Type	xs:integer	
Properties	use: required	
Used by	Complex Type Kwantu_Filter_type	
Source	<xs:attribute name="Seq" type="xs:integer" use="required" />	

#### **Attribute Kwantu\_Filter\_type / @Type**

Namespace	No namespace	
Type	kwantu_comparison_types	
Properties	use: required	
Facets	enumeration UniqueInColumn enumeration GreaterThan enumeration LessThan enumeration MaxLength enumeration UniqueInDomain enumeration GreaterThanOrEqual enumeration LessThanOrEqual enumeration Equal enumeration NotNull enumeration NotEqual enumeration CompositeUnique	
Used by	Complex Type	Kwantu_Filter_type

Source	<code>&lt;xss:attribute name="Type" use="required" type="kwantu_comparison_types"/&gt;</code>
--------	---

#### **Attribute Kwantu\_Filter\_type / @Parameters**

Namespace	No namespace
Used by	Complex Type      Kwantu_Filter_type
Source	<code>&lt;xss:attribute name="Parameters"/&gt;</code>

#### **Attribute Kwantu\_Filter\_type / @Reference**

Namespace	No namespace
Used by	Complex Type      Kwantu_Filter_type
Source	<code>&lt;xss:attribute name="Reference"/&gt;</code>

#### **Attribute Kwantu\_Filter\_type / @FilterElement**

Namespace	No namespace
Type	FilterElement_type
Properties	content: simple
Facets	enumeration      ValidDate enumeration      SequenceNumber enumeration      ElementInReference enumeration      CurrentElement
Used by	Complex Type      Kwantu_Filter_type
Source	<code>&lt;xss:attribute name="FilterElement" type="FilterElement_type"/&gt;</code>

#### **Attribute fnHistoryTotal / Filters / Filter / @Seq**

Namespace	No namespace
Type	xs:integer
Properties	use: required
Used by	Element      fnHistoryTotal/Filters/Filter
Source	<code>&lt;xss:attribute name="Seq" type="xs:integer" use="required"/&gt;</code>

#### **Attribute fnHistoryTotal / Filters / Filter / @Type**

Namespace	No namespace
Type	kwantu_comparison_types
Properties	use: required
Facets	enumeration      UniqueInColumn enumeration      GreaterThan enumeration      LessThan enumeration      MaxLength enumeration      UniqueInDomain enumeration      GreaterThanOrEqual enumeration      LessThanOrEqual enumeration      Equal enumeration      NotNull enumeration      NotEqual enumeration      CompositeUnique
Used by	Element      fnHistoryTotal/Filters/Filter

Source	<code>&lt;xss:attribute name="Type" use="required" type="kwantu_comparison_types"/&gt;</code>
--------	---

#### **Attribute fnHistoryTotal / Filters / Filter / @Parameters**

Namespace	No namespace
Used by	Element fnHistoryTotal/Filters/Filter
Source	<code>&lt;xss:attribute name="Parameters"/&gt;</code>

#### **Attribute fnHistoryTotal / Filters / Filter / @Reference**

Namespace	No namespace
Used by	Element fnHistoryTotal/Filters/Filter
Source	<code>&lt;xss:attribute name="Reference"/&gt;</code>

#### **Attribute fnHistoryTotal / Filters / Filter / @FilterElement**

Namespace	No namespace
Type	FilterElement_type
Properties	content: simple
Facets	enumeration ValidDate
	enumeration SequenceNumber
	enumeration ElementInReference
	enumeration CurrentElement
Used by	Element fnHistoryTotal/Filters/Filter
Source	<code>&lt;xss:attribute name="FilterElement" type="FilterElement_type"/&gt;</code>

#### **Attribute CalcType / @Seq**

Namespace	No namespace
Type	xs:integer
Properties	use: required
Used by	Complex Type CalcType
Source	<code>&lt;xss:attribute name="Seq" type="xs:integer" use="required"/&gt;</code>

#### **Attribute AssignSubElement\_type / AssignCode / @SubElementID**

Namespace	No namespace
Type	xs:NCName
Properties	use: required
Used by	Element AssignSubElement_type/AssignCode
Source	<code>&lt;xss:attribute name="SubElementID" type="xs:NCName" use="required"/&gt;</code>

#### **Attribute AssignSubElement\_type / AssignCode / @DataType**

Namespace	No namespace
Type	KwantuForm DataTypes
Properties	use: required
Facets	enumeration String
	enumeration Integer
	enumeration Float
	enumeration Memo
	enumeration Email

	enumeration	URI
	enumeration	Date
	enumeration	CoordinateDMS
	enumeration	GeoPoint
	enumeration	DataTypes
	enumeration	YesNo
	enumeration	Boolean
Used by	Element	AssignSubElement_type/AssignCode
Source	<code>&lt;xss:attribute name="DataType" type="KwantuFormDataTypes" use="required"/&gt;</code>	

#### **Attribute AssignSubElement\_type / AssignName / @SubElementID**

Namespace	No namespace
Type	xs:NCName
Properties	use: required
Used by	Element AssignSubElement_type/AssignName
Source	<code>&lt;xss:attribute name="SubElementID" type="xs:NCName" use="required"/&gt;</code>

#### **Attribute AssignSubElement\_type / AssignName / @DataType**

Namespace	No namespace
Type	KwantuFormDataTypes
Properties	use: required
Facets	enumeration String enumeration Integer enumeration Float enumeration Memo enumeration Email enumeration URI enumeration Date enumeration CoordinateDMS enumeration GeoPoint enumeration DataTypes enumeration YesNo enumeration Boolean
Used by	Element AssignSubElement_type/AssignName
Source	<code>&lt;xss:attribute name="DataType" type="KwantuFormDataTypes" use="required"/&gt;</code>

#### **Attribute AssignSubElement\_type / AssignSubElement / @Action**

Namespace	No namespace
Annotations	Action="Default" means that the assignment only takes place if the element is currently null.
Type	restriction of xs:NCName
Properties	content: simple
Facets	enumeration Default
Used by	Element AssignSubElement_type/AssignSubElement
Source	<code>&lt;xss:attribute name="Action"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;Action="Default" means that the assignment only takes place if the element is currently null.&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt;   &lt;xss:simpleType&gt;</code>

```

<xs:restriction base="xs:NCName">
  <xs:enumeration value="Default" />
</xs:restriction>
</xs:simpleType>
</xs:attribute>

```

#### **Attribute AssignSubElement\_type / AssignSubElement / @SubElementID**

Namespace	No namespace
Type	xs:NCName
Properties	use: required
Used by	Element AssignSubElement_type/AssignSubElement
Source	<xs:attribute name="SubElementID" type="xs:NCName" use="required"/>

#### **Attribute AssignSubElement\_type / AssignSubElement / @DataType**

Namespace	No namespace																								
Type	KwantuFormDataTypes																								
Properties	use: required																								
Facets	<table border="0"> <tr><td>enumeration</td><td>String</td></tr> <tr><td>enumeration</td><td>Integer</td></tr> <tr><td>enumeration</td><td>Float</td></tr> <tr><td>enumeration</td><td>Memo</td></tr> <tr><td>enumeration</td><td>Email</td></tr> <tr><td>enumeration</td><td>URI</td></tr> <tr><td>enumeration</td><td>Date</td></tr> <tr><td>enumeration</td><td>CoordinateDMS</td></tr> <tr><td>enumeration</td><td>GeoPoint</td></tr> <tr><td>enumeration</td><td>DataTypes</td></tr> <tr><td>enumeration</td><td>YesNo</td></tr> <tr><td>enumeration</td><td>Boolean</td></tr> </table>	enumeration	String	enumeration	Integer	enumeration	Float	enumeration	Memo	enumeration	Email	enumeration	URI	enumeration	Date	enumeration	CoordinateDMS	enumeration	GeoPoint	enumeration	DataTypes	enumeration	YesNo	enumeration	Boolean
enumeration	String																								
enumeration	Integer																								
enumeration	Float																								
enumeration	Memo																								
enumeration	Email																								
enumeration	URI																								
enumeration	Date																								
enumeration	CoordinateDMS																								
enumeration	GeoPoint																								
enumeration	DataTypes																								
enumeration	YesNo																								
enumeration	Boolean																								
Used by	Element AssignSubElement_type/AssignSubElement																								
Source	<xs:attribute name="DataType" type="KwantuFormDataTypes" use="required"/>																								

#### **Attribute kw\_elField\_type / EntryMode / @Mode**

Namespace	No namespace										
Type	restriction of xs:NCName										
Properties	use: required										
Facets	<table border="0"> <tr><td>enumeration</td><td>Capture</td></tr> <tr><td>enumeration</td><td>ReadOnly</td></tr> <tr><td>enumeration</td><td>Hidden</td></tr> <tr><td>enumeration</td><td>Calculated</td></tr> <tr><td>enumeration</td><td>RepeatKey</td></tr> </table>	enumeration	Capture	enumeration	ReadOnly	enumeration	Hidden	enumeration	Calculated	enumeration	RepeatKey
enumeration	Capture										
enumeration	ReadOnly										
enumeration	Hidden										
enumeration	Calculated										
enumeration	RepeatKey										
Used by	Element kw_elField_type/EntryMode										
Source	<pre> &lt;xs:attribute name="Mode" use="required"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="Capture" /&gt;       &lt;xs:enumeration value="Readonly" /&gt;       &lt;xs:enumeration value="Hidden" /&gt;       &lt;xs:enumeration value="Calculated" /&gt;       &lt;xs:enumeration value="RepeatKey" /&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; </pre>										

### **Attribute ReportColumnType / @TableName**

Namespace	No namespace
Type	xs:IDREF
Properties	use: required
Used by	Complex Type ReportColumnType
Source	<code>&lt;xs:attribute name="TableName" type="xs:IDREF" use="required"/&gt;</code>

### **Attribute ReportColumnType / @SubElement**

Namespace	No namespace
Type	xs:string
Properties	use: required
Used by	Complex Type ReportColumnType
Source	<code>&lt;xs:attribute name="SubElement" type="xs:string" use="required"/&gt;</code>

### **Attribute ReportColumnType / @Source**

Namespace	No namespace																
Type	Source_type																
Properties	use: required																
Facets	<table> <tr> <td>enumeration</td> <td>Code</td> </tr> <tr> <td>enumeration</td> <td>Name</td> </tr> <tr> <td>enumeration</td> <td>KwantuObjectID</td> </tr> <tr> <td>enumeration</td> <td>Element</td> </tr> <tr> <td>enumeration</td> <td>SequenceNo</td> </tr> <tr> <td>enumeration</td> <td>ValidDate</td> </tr> <tr> <td>enumeration</td> <td>RepeatSequenceNo</td> </tr> <tr> <td>enumeration</td> <td>ProfileID</td> </tr> </table>	enumeration	Code	enumeration	Name	enumeration	KwantuObjectID	enumeration	Element	enumeration	SequenceNo	enumeration	ValidDate	enumeration	RepeatSequenceNo	enumeration	ProfileID
enumeration	Code																
enumeration	Name																
enumeration	KwantuObjectID																
enumeration	Element																
enumeration	SequenceNo																
enumeration	ValidDate																
enumeration	RepeatSequenceNo																
enumeration	ProfileID																
Used by	Complex Type ReportColumnType																
Source	<code>&lt;xs:attribute name="Source" use="required" type="Source_type"/&gt;</code>																

### **Attribute Reports / @Include**

Namespace	No namespace				
Type	restriction of xs:NCName				
Properties	use: required				
Facets	<table> <tr> <td>enumeration</td> <td>Yes</td> </tr> <tr> <td>enumeration</td> <td>No</td> </tr> </table>	enumeration	Yes	enumeration	No
enumeration	Yes				
enumeration	No				
Used by	Element Reports				
Source	<code>&lt;xs:attribute name="Include" use="required"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="Yes"/&gt;       &lt;xs:enumeration value="No"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</code>				

### **Attribute kw\_elField\_type / @Type**

Namespace	No namespace
Annotations	If the type is select, then the datamodel will have sub-elements under the normal element. This is used when doing assign statements
Type	restriction of xs:NCName

Properties	use:	required
Facets	enumeration	Field
	enumeration	Select
Used by	Complex Type	kw_elField_type
Source	<pre>&lt;xs:attribute name="Type" use="required"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;If the type is select, then the datamodel will have sub-elements under the normal element. This is used when doing assign statements&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="Field"/&gt;       &lt;xs:enumeration value="Select"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</pre>	

#### **Attribute kw\_elSelect\_type / Display / Xform / @Appearance**

Namespace	No namespace	
Type	restriction of xs:Name	
Properties	use:	required
Facets	enumeration	SingleSelect
	enumeration	MultiSelect
	enumeration	RadioSelect
	enumeration	CheckBox
	enumeration	Output
	enumeration	Search
	enumeration	Button
Used by	Element	kw_elSelect_type/Display/Xform
Source	<pre>&lt;xs:attribute name="Appearance" use="required"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:Name"&gt;       &lt;xs:enumeration value="SingleSelect"/&gt;       &lt;xs:enumeration value="MultiSelect"/&gt;       &lt;xs:enumeration value="RadioSelect"/&gt;       &lt;xs:enumeration value="CheckBox"/&gt;       &lt;xs:enumeration value="Output"/&gt;       &lt;xs:enumeration value="Search"/&gt;       &lt;xs:enumeration value="Button"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</pre>	

#### **Attribute kw\_elSelect\_type / Display / Xform / @CSS**

Namespace	No namespace	
Type	restriction of xs:Name	
Properties	content:	simple
Facets	enumeration	MediumSelect
	enumeration	LargeSelect
	enumeration	man-label
	enumeration	Kwantu-Standard-Label
Used by	Element	kw_elSelect_type/Display/Xform
Source	<pre>&lt;xs:attribute name="CSS"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:Name"&gt;       &lt;xs:enumeration value="MediumSelect"/&gt;       &lt;xs:enumeration value="LargeSelect"/&gt;       &lt;xs:enumeration value="man-label"/&gt;       &lt;xs:enumeration value="Kwantu-Standard-Label"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</pre>	

```

        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
```

### **Attribute kw\_Condition\_type / @Operator**

Namespace	No namespace	
Type	restriction of xs:Name	
Properties	use: required	
Facets	enumeration	GreaterThan
	enumeration	LessThan
	enumeration	GreaterThanOrEqual
	enumeration	LessThanOrEqual
	enumeration	Equal
	enumeration	NotEqual
Used by	Complex Type	kw_Condition_type
Source	<pre style="font-family: monospace; margin: 0;"> &lt;xs:attribute name="Operator" use="required"&gt;     &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:Name"&gt;             &lt;xs:enumeration value="GreaterThan"/&gt;             &lt;xs:enumeration value="LessThan"/&gt;             &lt;xs:enumeration value="GreaterThanOrEqual"/&gt;             &lt;xs:enumeration value="LessThanOrEqual"/&gt;             &lt;xs:enumeration value="Equal"/&gt;             &lt;xs:enumeration value="NotEqual"/&gt;         &lt;/xs:restriction&gt;     &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</pre>	

### **Attribute Codelist / Taxonomy / @Type**

Namespace	No namespace	
Type	restriction of xs:NCName	
Properties	content: simple	
Facets	enumeration	XSL
	enumeration	Rest
	enumeration	Default
Used by	Element	Codelist/Taxonomy
Source	<pre style="font-family: monospace; margin: 0;"> &lt;xs:attribute name="Type"&gt;     &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:NCName"&gt;             &lt;xs:enumeration value="XSL"/&gt;             &lt;xs:enumeration value="Rest"/&gt;             &lt;xs:enumeration value="Default"/&gt;         &lt;/xs:restriction&gt;     &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</pre>	

### **Attribute Codelist / @SaveCode**

Namespace	No namespace	
Type	restriction of xs:NCName	
Properties	content: simple	
Facets	enumeration	Both
	enumeration	Default
	enumeration	Name
	enumeration	Code
Used by	Element	Codelist
Source	<pre style="font-family: monospace; margin: 0;"> &lt;xs:attribute name="SaveCode"&gt;</pre>	

```

<xs:simpleType>
  <xs:restriction base="xs:NCName">
    <xs:enumeration value="Both"/>
    <xs:enumeration value="Default"/>
    <xs:enumeration value="Name"/>
    <xs:enumeration value="Code"/>
  </xs:restriction>
</xs:simpleType>
</xs:attribute>

```

#### **Attribute RestType / ServiceURL / @Source**

Namespace	No namespace	
Type	restriction of xs:NCName	
Properties	use: required	
Facets	enumeration	ProcessVariable
	enumeration	SubProcessVariable
	enumeration	ElementID
	enumeration	ProcessObject
	enumeration	SubProcessObject
	enumeration	ServiceURL
Used by	Element	RestType/ServiceURL
Source	<xs:attribute name="Source" use="required">   <xs:simpleType>     <xs:restriction base="xs:NCName">       <xs:enumeration value="ProcessVariable"/>       <xs:enumeration value="SubProcessVariable"/>       <xs:enumeration value="ElementID"/>       <xs:enumeration value="ProcessObject"/>       <xs:enumeration value="SubProcessObject"/>       <xs:enumeration value="ServiceURL"/>     </xs:restriction>   </xs:simpleType> </xs:attribute>	

#### **Attribute RestType / Parameters / Parameter / @Seq**

Namespace	No namespace	
Used by	Element RestType/Parameters/Parameter	
Source	<xs:attribute name="Seq" />	

#### **Attribute RestType / Parameters / Parameter / @Name**

Namespace	No namespace	
Type	xs:NCName	
Properties	content: simple	
Used by	Element RestType/Parameters/Parameter	
Source	<xs:attribute name="Name" type="xs:NCName" />	

#### **Attribute RestType / Parameters / Parameter / @Format**

Namespace	No namespace	
Used by	Element RestType/Parameters/Parameter	
Source	<xs:attribute name="Format" />	

#### **Attribute RestType / Return / Status / StatusCode / @Value**

Namespace	No namespace	
Used by	Element RestType/Return/Status/StatusCode	
Source	<xs:attribute name="Value" />	

### **Attribute RestType / Return / Status / @Type**

Namespace	No namespace	
Annotations	If the type="Error" then this a potential error status If the type="Success" then this a not an error	
Type	restriction of xs:NCName	
Properties	content: simple	
Facets	enumeration	Error
	enumeration	Success
Used by	Element RestType/Return/Status	
Source	<pre> &lt;xs:attribute name="Type"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;If the type="Error" then this a potential error status If the type="Success" then this a not an error&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="Error"/&gt;       &lt;xs:enumeration value="Success"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</pre>	

### **Attribute RestType / Return / Assign / @AssignTo**

Namespace	No namespace	
Type	restriction of xs:NCName	
Properties	use: required	
Facets	enumeration	Variable
	enumeration	Element
	enumeration	Object
Used by	Element RestType/Return/Assign	
Source	<pre> &lt;xs:attribute name="AssignTo" use="required"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="Variable"/&gt;       &lt;xs:enumeration value="Element"/&gt;       &lt;xs:enumeration value="Object"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</pre>	

### **Attribute RestType / Return / Assign / @AssignToID**

Namespace	No namespace	
Properties	use: required	
Used by	Element RestType/Return/Assign	
Source	<pre> &lt;xs:attribute name="AssignToID" use="required"/&gt;</pre>	

### **Attribute RestType / Return / Assign / @Type**

Namespace	No namespace	
Type	restriction of xs:NCName	
Properties	use: required	
Facets	enumeration	XPath
	enumeration	JSON
Used by	Element RestType/Return/Assign	
Source	<pre> &lt;xs:attribute name="Type" use="required"&gt;   &lt;xs:simpleType&gt;</pre>	

```

<xs:restriction base="xs:NCName">
  <xs:enumeration value="XPath"/>
  <xs:enumeration value="JSON"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>

```

#### **Attribute RestType / @ID**

Namespace	No namespace	
Used by	Complex Type	RestType
Source	<code>&lt;xs:attribute name="ID"/&gt;</code>	

#### **Attribute kw\_elSelect\_type / EntryMode / Select / Taxonomy / @Type**

Namespace	No namespace	
Type	restriction of xs:NCName	
Properties	content: simple	
Facets	enumeration XSL	
Used by	Element kw_elSelect_type/EntryMode>Select/Taxonomy	
Source	<code>&lt;xs:attribute name="Type"&gt;</code> <code>&lt;xs:simpleType&gt;</code> <code>&lt;xs:restriction base="xs:NCName"&gt;</code> <code>&lt;xs:enumeration value="XSL"/&gt;</code> <code>&lt;/xs:restriction&gt;</code> <code>&lt;/xs:simpleType&gt;</code> <code>&lt;/xs:attribute&gt;</code>	

#### **Attribute Assigns\_type / Assign / @Action**

Namespace	No namespace	
Annotations	Action="Default" means that the assignment only takes place if the element is currently null.	
Type	restriction of xs:NCName	
Properties	content: simple	
Facets	enumeration Default	
Used by	Element Assigns_type/Assign	
Source	<code>&lt;xs:attribute name="Action"&gt;</code> <code>&lt;xs:annotation&gt;</code> <code>&lt;xs:documentation&gt;Action="Default" means that the assignment only takes place if the element is currently null.&lt;/xs:documentation&gt;</code> <code>&lt;/xs:annotation&gt;</code> <code>&lt;xs:simpleType&gt;</code> <code>&lt;xs:restriction base="xs:NCName"&gt;</code> <code>&lt;xs:enumeration value="Default"/&gt;</code> <code>&lt;/xs:restriction&gt;</code> <code>&lt;/xs:simpleType&gt;</code> <code>&lt;/xs:attribute&gt;</code>	

#### **Attribute Assigns\_type / Assign / @Scope**

Namespace	No namespace	
Type	restriction of xs:NCName	
Properties	use: required	
Facets	enumeration SubElement enumeration Element enumeration Repeat enumeration Variable enumeration Registry	
Used by	Element Assigns_type/Assign	
Source	<code>&lt;xs:attribute name="Scope" use="required"&gt;</code>	

```

<xs:simpleType>
  <xs:restriction base="xs:NCName">
    <xs:enumeration value="SubElement"/>
    <xs:enumeration value="Element"/>
    <xs:enumeration value="Repeat"/>
    <xs:enumeration value="Variable"/>
    <xs:enumeration value="Registry"/>
  </xs:restriction>
</xs:simpleType>
</xs:attribute>

```

#### **Attribute Assigns\_type / Assign / @ElementID**

Namespace	No namespace
Type	xs:NCName
Properties	use: required
Used by	Element Assigns_type/Assign
Source	<xs:attribute name="ElementID" type="xs:NCName" use="required"/>

#### **Attribute Assigns\_type / Assign / @SubElementID**

Namespace	No namespace
Type	xs:NCName
Properties	content: simple
Used by	Element Assigns_type/Assign
Source	<xs:attribute name="SubElementID" type="xs:NCName"/>

#### **Attribute Assigns\_type / Assign / @RepeatKeyValue**

Namespace	No namespace
Used by	Element Assigns_type/Assign
Source	<xs:attribute name="RepeatKeyValue"/>

#### **Attribute kw\_elSelect\_type / EntryMode / @Mode**

Namespace	No namespace
Annotations	The mode defines how the data elements are entered into the field. Capture - User takes an action ReadOnly - User does not take action but the field can only be viewed Hidden - User cannot see the values Calculated - Value is calculated RepeatKey - The value is picked up from the way that the repeat key is specified.
Type	restriction of xs:NCName
Properties	use: required
Facets	enumeration Capture enumeration ReadOnly enumeration RepeatKey
Used by	Element kw_elSelect_type/EntryMode
Source	<pre> &lt;xs:attribute name="Mode" use="required"&gt;   &lt;xs:annotation id="Select.Entrymode.Mode"&gt;     &lt;xs:documentation&gt;The mode defines how the data elements are entered into the field. Capture - User takes an action ReadOnly - User does not take action but the field can only be viewed Hidden - User cannot see the values Calculated - Value is calculated RepeatKey - The value is picked up from the way that the repeat key is specified.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="Capture"/&gt;       &lt;xs:enumeration value="ReadOnly"/&gt;       &lt;xs:enumeration value="RepeatKey"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; </pre>

### **Attribute kw\_elSelect\_type / EntryMode / @Source**

Namespace	No namespace	
Type	Source_type	
Properties	content: simple	
Facets	enumeration	Code
	enumeration	Name
	enumeration	KwantuObjectID
	enumeration	Element
	enumeration	SequenceNo
	enumeration	ValidDate
	enumeration	RepeatSequenceNo
	enumeration	ProfileID
Used by	Element	kw_elSelect_type/EntryMode
Source	<code>&lt;xs:attribute name="Source" type="Source_type"/&gt;</code>	

### **Attribute kw\_elSelect\_type / @Type**

Namespace	No namespace	
Type	restriction of xs:NCName	
Properties	use: required	
Facets	enumeration	Select
	enumeration	SelectGroup
	enumeration	RepeatKeyField
Used by	Complex Type	kw_elSelect_type
Source	<code>&lt;xs:attribute name="Type" use="required"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="Select"/&gt;       &lt;xs:enumeration value="SelectGroup"/&gt;       &lt;xs:enumeration value="RepeatKeyField"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</code>	

### **Attribute RepeatDef / AddRows / AutoLoad / Periods / @PeriodType**

Namespace	No namespace	
Type	restriction of xs:Name	
Properties	use: required	
Facets	enumeration	Month
	enumeration	Quarter
	enumeration	YearCalendar
	enumeration	YearAprilToMay
	enumeration	YearJulyToJune
Used by	Element	RepeatDef/AddRows/AutoLoad/Periods
Source	<code>&lt;xs:attribute name="PeriodType" use="required"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:Name"&gt;       &lt;xs:enumeration value="Month"/&gt;       &lt;xs:enumeration value="Quarter"/&gt;       &lt;xs:enumeration value="YearCalendar"/&gt;       &lt;xs:enumeration value="YearAprilToMay"/&gt;       &lt;xs:enumeration value="YearJulyToJune"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</code>	

### **Attribute RepeatDef / AddRows / AutoLoad / Component / Filter / @ElementID**

Namespace	No namespace
Used by	Element RepeatDef/AddRows/AutoLoad/Component/Filter
Source	<code>&lt;xs:attribute name="ElementID"/&gt;</code>

### **Attribute RepeatDef / AddRows / AutoLoad / Component / Filter / @SubElement**

Namespace	No namespace
Used by	Element RepeatDef/AddRows/AutoLoad/Component/Filter
Source	<code>&lt;xs:attribute name="SubElement"/&gt;</code>

### **Attribute RepeatDef / AddRows / AutoLoad / Component / @ComponentName**

Namespace	No namespace
Annotations	Name of the component from which the rows will be retrieved.
Type	xs:NCName
Properties	use: required
Used by	Element RepeatDef/AddRows/AutoLoad/Component
Source	<code>&lt;xs:attribute name="ComponentName" type="xs:NCName" use="required"&gt;     &lt;xs:annotation id="AddRows.Component.ComponentName"&gt;         &lt;xs:documentation&gt;Name of the component from which the rows will be retrieved.&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</code>

### **Attribute RepeatDef / AddRows / AutoLoad / Component / @RepeatID**

Namespace	No namespace
Annotations	Name of the RepeatID from which the rows will be retrieved.
Type	xs:NCName
Properties	use: required
Used by	Element RepeatDef/AddRows/AutoLoad/Component
Source	<code>&lt;xs:attribute name="RepeatID" type="xs:NCName" use="required"&gt;     &lt;xs:annotation id="AddRows.Component.RepeatID"&gt;         &lt;xs:documentation&gt;Name of the RepeatID from which the rows will be retrieved.&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</code>

### **Attribute RepeatDef / HeaderRow / @LabelsFromData**

Namespace	No namespace
Annotations	If Yes, then use the Label components from the first elements in the columns for the data section to determine the labels
Type	restriction of xs:NCName
Properties	content: simple
Facets	enumeration Yes
	enumeration No
Used by	Element RepeatDef/HeaderRow
Source	<code>&lt;xs:attribute name="LabelsFromData"&gt;     &lt;xs:annotation id="HeaderRow.LabelsFromData"&gt;         &lt;xs:documentation&gt;If Yes, then use the Label components from the first elements in the columns for the data section to determine the labels&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt; &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;         &lt;xs:enumeration value="Yes"/&gt;         &lt;xs:enumeration value="No"/&gt;     &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</code>

<pre>&lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</pre>
---

### **Attribute @Seq**

Namespace	No namespace
Used by	Element RepeatDef/Columns/Column
Source	<pre>&lt;xs:attribute name="Seq" /&gt;</pre>

### **Attribute RepeatDef / Columns / Column / @Width**

Namespace	No namespace
Annotations	Defines the width of the column
Used by	Element RepeatDef/Columns/Column
Source	<pre>&lt;xs:attribute name="Width"&gt;   &lt;xs:annotation id="Repeat.RepeatDef.Columns.Column.Width"&gt;     &lt;xs:documentation&gt;Defines the width of the column&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>

### **Attribute RepeatDef / Columns / @NumberColumns**

Namespace	No namespace
Annotations	Defines the number of columns that should be generated for the repeat structure. These are Visible columns. Each column has a header row, a data area and a summary area. In each of these areas there may be one or more elements. The Number of columns in the header, data and summary bands need not be equal, but a means must be devised to specify the span.
Type	xs:integer
Properties	content: simple
Used by	Element RepeatDef/Columns
Source	<pre>&lt;xs:attribute name="NumberColumns" type="xs:integer"&gt;   &lt;xs:annotation id="Repeat.RepeatDef.Columns.NumberColumns"&gt;     &lt;xs:documentation&gt;Defines the number of columns that should be generated for the repeat structure. These are Visible columns. Each column has a header row, a data area and a summary area. In each of these areas there may be one or more elements. The Number of columns in the header, data and summary bands need not be equal, but a means must be devised to specify the span.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>

### **Attribute elCalcField / EntryMode / @Mode**

Namespace	No namespace
Type	restriction of xs:NCName
Properties	use: required
Facets	enumeration ReadOnly enumeration Hidden enumeration Calculated
Used by	Element elCalcField/EntryMode
Source	<pre>&lt;xs:attribute name="Mode" use="required"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="ReadOnly"/&gt;       &lt;xs:enumeration value="Hidden"/&gt;       &lt;xs:enumeration value="Calculated"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</pre>

### **Attribute elCalcField / @Type**

Namespace	No namespace
-----------	--------------

Type	restriction of xs:NCName	
Properties	use:	required
Facets	enumeration	CalcField
Used by	Element	elCalcField
Source	<pre>&lt;xs:attribute name="Type" use="required"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="CalcField"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</pre>	

#### **Attribute BaseElements / @ID**

Namespace	No namespace	
Type	xs:short	
Properties	content:	simple
Used by	Complex Type	kw_elGroup_type
Source	<pre>&lt;xs:attribute name="ID" type="xs:short"/&gt;</pre>	

#### **Attribute kw\_elGroup\_type / @ID**

Namespace	No namespace	
Type	xs:short	
Properties	content:	simple
Source	<pre>&lt;xs:attribute name="ID" type="xs:short"/&gt;</pre>	

#### **Attribute RepeatHeader / rptColumn / @ColumnNo**

Namespace	No namespace	
Used by	Element	RepeatHeader/rptColumn
Source	<pre>&lt;xs:attribute name="ColumnNo"/&gt;</pre>	

#### **Attribute RepeatHeader / rptColumn / @Span**

Namespace	No namespace	
Used by	Element	RepeatHeader/rptColumn
Source	<pre>&lt;xs:attribute name="Span"/&gt;</pre>	

#### **Attribute RepeatHeader / @GroupName**

Namespace	No namespace	
Type	xs:NCName	
Properties	use:	required
Used by	Element	RepeatHeader
Source	<pre>&lt;xs:attribute name="GroupName" type="xs:NCName" use="required"/&gt;</pre>	

#### **Attribute RepeatHeader / @ID**

Namespace	No namespace	
Type	xs:NCName	
Properties	use:	required
Used by	Element	RepeatHeader
Source	<pre>&lt;xs:attribute name="ID" type="xs:NCName" use="required"/&gt;</pre>	

#### **Attribute RepeatData / rptColumn / @ColumnNo**

Namespace	No namespace
Used by	Element      RepeatData/rptColumn
Source	<code>&lt;xs:attribute name="ColumnNo"/&gt;</code>

#### **Attribute RepeatData / rptColumn / @Span**

Namespace	No namespace
Used by	Element      RepeatData/rptColumn
Source	<code>&lt;xs:attribute name="Span"/&gt;</code>

#### **Attribute RepeatData / @GroupName**

Namespace	No namespace
Type	xs:NCName
Properties	use:      required
Used by	Element      RepeatData
Source	<code>&lt;xs:attribute name="GroupName" type="xs:NCName" use="required"/&gt;</code>

#### **Attribute RepeatData / @ID**

Namespace	No namespace
Type	xs:NCName
Properties	use:      required
Used by	Element      RepeatData
Source	<code>&lt;xs:attribute name="ID" type="xs:NCName" use="required"/&gt;</code>

#### **Attribute rptColumn / @ColumnNo**

Namespace	No namespace
Used by	Element      rptColumn
Source	<code>&lt;xs:attribute name="ColumnNo"/&gt;</code>

#### **Attribute rptColumn / @Span**

Namespace	No namespace
Used by	Element      rptColumn
Source	<code>&lt;xs:attribute name="Span"/&gt;</code>

#### **Attribute RepeatSummary / @GroupName**

Namespace	No namespace
Type	xs:NCName
Properties	use:      required
Used by	Element      RepeatSummary
Source	<code>&lt;xs:attribute name="GroupName" type="xs:NCName" use="required"/&gt;</code>

#### **Attribute RepeatSummary / @ID**

Namespace	No namespace
Type	xs:NCName
Properties	use:      required

Used by	Element	RepeatSummary
Source		<xs:attribute name="ID" type="xs:NCName" use="required"/>

#### **Attribute kw\_Repeat\_type / @Type**

Namespace	No namespace	
Type	xs:NCName	
Properties	use: required	
Used by	Complex Type	kw_Repeat_type
Source		<xs:attribute name="Type" type="xs:NCName" use="required"/>

#### **Attribute HierarchyNode / @Level**

Namespace	No namespace	
Type	xs:short	
Properties	content: simple	
Used by	Element	HierarchyNode
Source		<xs:attribute name="Level" type="xs:short"/>

#### **Attribute HierarchyNode / @Parameter**

Namespace	No namespace	
Type	xs:NCName	
Properties	use: required	
Used by	Element	HierarchyNode
Source		<xs:attribute name="Parameter" type="xs:NCName" use="required"/>

#### **Attribute HierarchyNode / @SelectElement**

Namespace	No namespace	
Type	xs:NCName	
Properties	content: simple	
Used by	Element	HierarchyNode
Source		<xs:attribute name="SelectElement" type="xs:NCName"/>

#### **Attribute Hierarchy / @Levels**

Namespace	No namespace	
Type	restriction of xs:short	
Properties	content: simple	
Facets	minInclusive -32768	
Used by	Element	Hierarchy
Source		<xs:attribute name="Levels"> <xs:simpleType> <xs:restriction base="xs:short"> <xs:minInclusive value="-32768"/> </xs:restriction> </xs:simpleType> </xs:attribute>

#### **Attribute Hierarchy / @Taxonomy**

Namespace	No namespace
Type	xs:string

Properties	content:	simple
Used by	Element	Hierarchy
Source	<code>&lt;xs:attribute name="Taxonomy" type="xs:string"/&gt;</code>	

### Attribute Hierarchy / @TopLevel

Namespace	No namespace
Type	xs:token
Properties	content: simple
Used by	Element Hierarchy
Source	<code>&lt;xs:attribute name="TopLevel" type="xs:token"/&gt;</code>

### Attribute Hierarchy / @Type

Namespace	No namespace
Type	restriction of xs:NCName
Properties	content: simple
Facets	enumeration Hierarchy
Used by	Element Hierarchy
Source	<code>&lt;xs:attribute name="Type"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="Hierarchy"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</code>

### Attribute elSelectGroup / SelectGroupKey / @KeyID

Namespace	No namespace
Annotations	This KeyID references the ID of the field on the group that forms the key to link to this section
Type	xs:token
Properties	use: required
Used by	Element elSelectGroup/SelectGroupKey
Source	<code>&lt;xs:attribute name="KeyID" type="xs:token" use="required"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This KeyID references the ID of the field on the group that forms the key to link to this section&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</code>

### Attribute elSelectGroup / @Type

Namespace	No namespace
Type	restriction of xs:Name
Properties	use: required
Facets	enumeration PersonsList enumeration OrganisationsList enumeration Rest
Used by	Element elSelectGroup
Source	<code>&lt;xs:attribute name="Type" use="required"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:Name"&gt;       &lt;xs:enumeration value="PersonsList"/&gt;       &lt;xs:enumeration value="OrganisationsList"/&gt;       &lt;xs:enumeration value="Rest"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</code>

<pre>&lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</pre>
---

#### **Attribute ConfigurationData / @Version**

Namespace	No namespace
Type	xs:decimal
Properties	use: required
Used by	Element ConfigurationData
Source	<pre>&lt;xs:attribute name="Version" use="required" type="xs:decimal"/&gt;</pre>

#### **Attribute Node / @Action**

Namespace	No namespace
Type	xs:NCName
Properties	use: required
Used by	Element Node
Source	<pre>&lt;xs:attribute name="Action" use="required" type="xs:NCName"/&gt;</pre>

#### **Attribute Node / @Seq**

Namespace	No namespace
Type	xs:integer
Properties	use: required
Used by	Element Node
Source	<pre>&lt;xs:attribute name="Seq" use="required" type="xs:integer"/&gt;</pre>

#### **Attribute @No**

Namespace	No namespace
Type	xs:integer
Properties	content: simple
Source	<pre>&lt;xs:attribute name="No" type="xs:integer"/&gt;</pre>

#### **Attribute @Source**

Namespace	No namespace
Type	xs:NCName
Properties	content: simple
Source	<pre>&lt;xs:attribute name="Source" type="xs:NCName"/&gt;</pre>

#### **Attribute kwElementMap / @LocalacElementName**

Namespace	No namespace
Properties	use: required
Used by	Complex Type kwElementMap
Source	<pre>&lt;xs:attribute name="LocalacElementName" use="required"/&gt;</pre>

#### **Attribute GeoPoint\_type / @LongSeconds**

Namespace	No namespace
Properties	use: required
Used by	Complex Type GeoPoint_type

Source	<code>&lt;xss:attribute name="LongSeconds" use="required" /&gt;</code>
--------	--

#### **Attribute GeoPoint\_type / @LongMinutes**

Namespace	No namespace
Properties	use: required
Used by	Complex Type GeoPoint_type
Source	<code>&lt;xss:attribute name="LongMinutes" use="required" /&gt;</code>

#### **Attribute GeoPoint\_type / @LongDegrees**

Namespace	No namespace
Properties	use: required
Used by	Complex Type GeoPoint_type
Source	<code>&lt;xss:attribute name="LongDegrees" use="required" /&gt;</code>

#### **Attribute GeoPoint\_type / @Longitude**

Namespace	No namespace
Used by	Complex Type GeoPoint_type
Source	<code>&lt;xss:attribute name="Longitude" /&gt;</code>

#### **Attribute GeoPoint\_type / @LattSeconds**

Namespace	No namespace
Properties	use: required
Used by	Complex Type GeoPoint_type
Source	<code>&lt;xss:attribute name="LattSeconds" use="required" /&gt;</code>

#### **Attribute GeoPoint\_type / @LattMinutes**

Namespace	No namespace
Properties	use: required
Used by	Complex Type GeoPoint_type
Source	<code>&lt;xss:attribute name="LattMinutes" use="required" /&gt;</code>

#### **Attribute GeoPoint\_type / @LattDegrees**

Namespace	No namespace
Properties	use: required
Used by	Complex Type GeoPoint_type
Source	<code>&lt;xss:attribute name="LattDegrees" use="required" /&gt;</code>

#### **Attribute GeoPoint\_type / @Latitude**

Namespace	No namespace
Used by	Complex Type GeoPoint_type
Source	<code>&lt;xss:attribute name="Latitude" /&gt;</code>

#### **Attribute NodeType / @Seq**

Namespace	No namespace
Type	xs:integer

Properties	use:	required
Used by	Complex Type	NodeType
Source	<code>&lt;xs:attribute name="Seq" type="xs:integer" use="required"/&gt;</code>	

#### **Attribute fnProcessPeriodDueDate / CalcDueDate / DueDate / @Seq**

Namespace	No namespace
Used by	Element fnProcessPeriodDueDate/CalcDueDate/DueDate
Source	<code>&lt;xs:attribute name="Seq"/&gt;</code>

#### **Attribute fnProcessPeriodDueDate / CalcDueDate / @Type**

Namespace	No namespace
Type	restriction of xs:NCName
Properties	content: simple
Facets	enumeration Earliest
	enumeration Latest
Used by	Element fnProcessPeriodDueDate/CalcDueDate
Source	<code>&lt;xs:attribute name="Type"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="Earliest"/&gt;       &lt;xs:enumeration value="Latest"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</code>

#### **Attribute fnProcessScheduleDueDate / CalcDueDate / DueDate / ProcessID / @Aspect**

Namespace	No namespace
Type	restriction of xs:NCName
Properties	content: simple
Facets	enumeration DueDate
	enumeration ValidDate
	enumeration ReceivedDate
Used by	Element fnProcessScheduleDueDate/CalcDueDate/DueDate/ProcessID
Source	<code>&lt;xs:attribute name="Aspect"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="DueDate"/&gt;       &lt;xs:enumeration value="ValidDate"/&gt;       &lt;xs:enumeration value="ReceivedDate"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</code>

#### **Attribute fnProcessScheduleDueDate / CalcDueDate / DueDate / @Seq**

Namespace	No namespace
Used by	Element fnProcessScheduleDueDate/CalcDueDate/DueDate
Source	<code>&lt;xs:attribute name="Seq"/&gt;</code>

#### **Attribute fnProcessScheduleDueDate / CalcDueDate / @Type**

Namespace	No namespace
Type	restriction of xs:NCName
Properties	content: simple

Facets	enumeration	Max
	enumeration	Min
Used by	Element	fnProcessScheduleDueDate/CalcDueDate
Source	<pre>&lt;xs:attribute name="Type"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="Max"/&gt;       &lt;xs:enumeration value="Min"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt;</pre>	

### Attribute kw\_Label\_type / Int / @lang

Namespace	No namespace
Used by	Element kw_Label_type/Int
Source	<pre>&lt;xs:attribute name="lang"&gt;/&gt;</pre>

## Element Group(s)

### Element Group BaseElements

Namespace	No namespace				
Diagram	<pre> graph LR     BE[BaseElements] --&gt; elLabel     BE --&gt; elField     BE --&gt; elSelect     BE --&gt; elCalcField     BE --&gt; Repeat     BE --&gt; elGroup     </pre> <p>The diagram illustrates the structure of the 'BaseElements' element group. It branches into six sub-elements: 'elLabel', 'elField', 'elSelect', 'elCalcField', 'Repeat', and 'elGroup'. Each sub-element is represented by a rounded rectangle with a plus sign (+) indicating it can appear multiple times. A callout box points to the 'Repeat' element, stating: 'Root element for the structure that defines the repeat or table structure'.</p>				
Used by	<table border="1"> <tr> <td>Elements</td> <td>HierarchyNode, RepeatData/rptColumn, RepeatHeader/rptColumn, elSelectGroup/SelectGroup, rptColumn</td> </tr> <tr> <td>Complex Type</td> <td>kw_elGroup_type</td> </tr> </table>	Elements	HierarchyNode, RepeatData/rptColumn, RepeatHeader/rptColumn, elSelectGroup/SelectGroup, rptColumn	Complex Type	kw_elGroup_type
Elements	HierarchyNode, RepeatData/rptColumn, RepeatHeader/rptColumn, elSelectGroup/SelectGroup, rptColumn				
Complex Type	kw_elGroup_type				
Model	elLabel   elField   elSelect   elCalcField   Repeat   elGroup				
Children	Repeat, elCalcField, elField, elGroup, elLabel, elSelect				
Source	<pre>&lt;xs:group name="BaseElements"&gt;   &lt;xs:choice&gt;     &lt;xs:element ref="elLabel"/&gt;     &lt;xs:element ref="elField"/&gt;     &lt;xs:element ref="elSelect"/&gt;     &lt;xs:element ref="elCalcField"/&gt;     &lt;xs:element ref="Repeat"/&gt;     &lt;xs:element ref="elGroup"/&gt;   &lt;/xs:choice&gt; &lt;/xs:group&gt;</pre>				