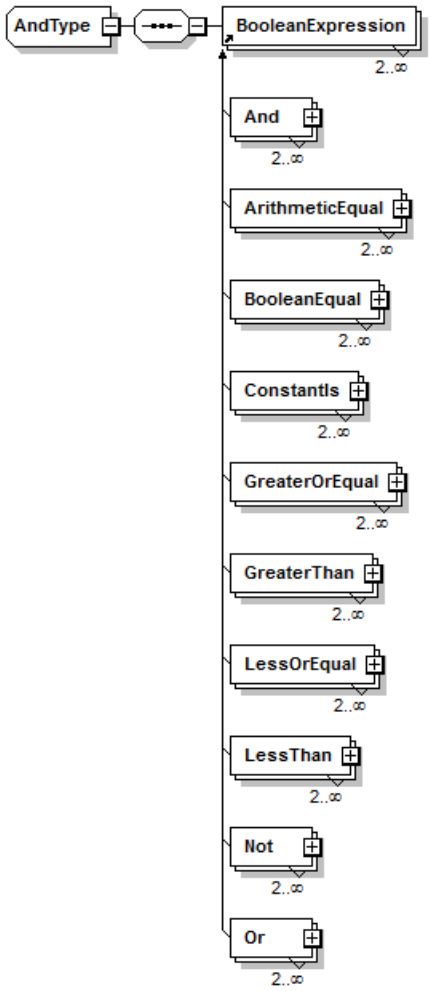
Data dictionary of GenericExpressions.xsd (normative)

schema location: **..\QIFLibrary\GenericExpressions.xsd**
attributeFormDefault: **unqualified**
elementFormDefault: **qualified**
targetNamespace: **http://qifstandards.org/xsd/qif2**

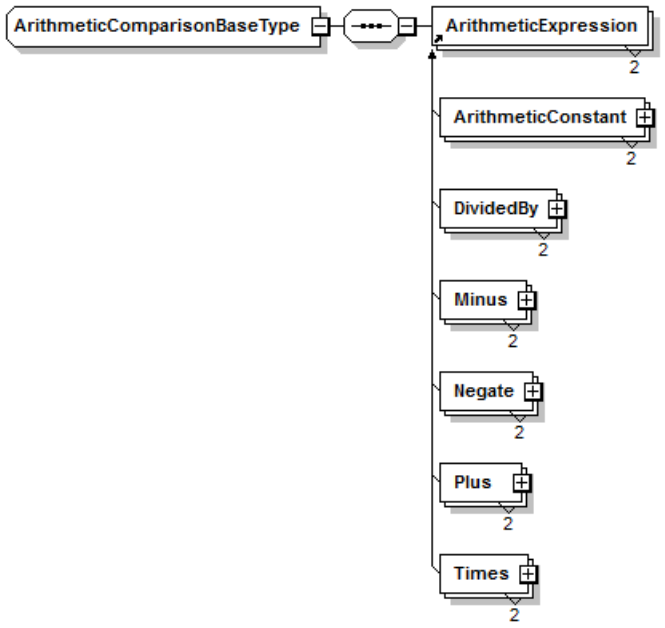
Complex types

Simple types

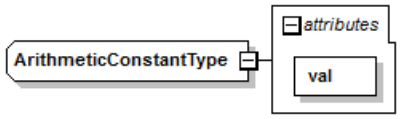
complexType **AndType**

diagram	
type	extension of **BooleanExpressionBaseType**
properties	base BooleanExpressionBaseType
children	**BooleanExpression**
used by	element **And**
annotation	documentation The AndType defines an 'and' Boolean expression. The AndType evaluates to true if all of the BooleanExpression elements it contains evaluate to true. Otherwise, it evaluates to false. The BooleanExpression elements must be evaluated in order. If any BooleanExpression element evaluates to false, the remaining BooleanExpression elements must not be evaluated.

complexType **ArithmeticComparisonBaseType**

diagram	
type	extension of **BooleanExpressionBaseType**
properties	base BooleanExpressionBaseType abstract true
children	**ArithmeticExpression**
used by	complexTypes **ArithmeticEqualType** **GreaterOrEqualType** **GreaterThanType** **LessOrEqualType** **LessThanType**
annotation	documentation The ArithmeticComparisonBaseType defines the base type for arithmetic comparisons. The evaluation environment for arithmetic comparisons must include the arithmetic constant, ComparisonTiny.

complexType **ArithmeticConstantType**

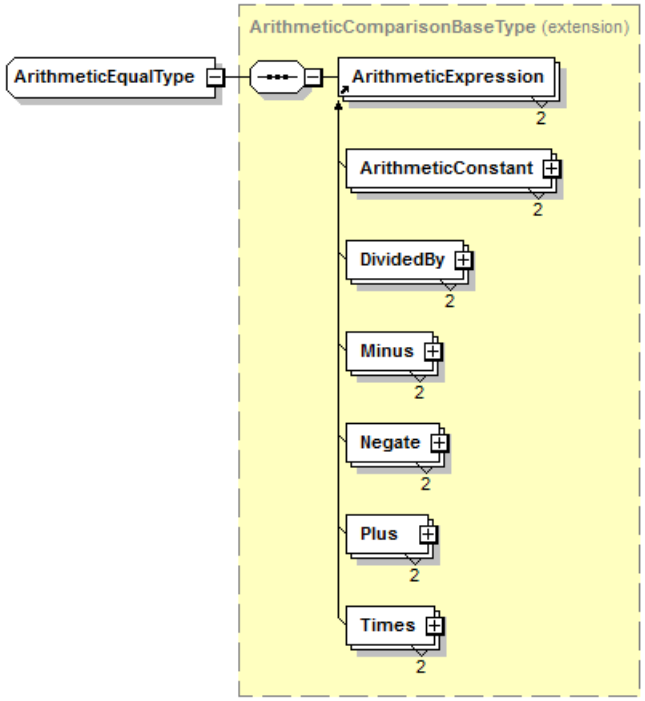
diagram	
type	extension of **ArithmeticExpressionBaseType**
properties	base ArithmeticExpressionBaseType
used by	element **ArithmeticConstant**
attributes	Name: val; Type: **xs:decimal**; Use: required; Default: ; Fixed: ; Annotation: documentation: The required val attribute is an arithmetic constant.
annotation	documentation The ArithmeticConstantType defines an arithmetic constant. The value of an ArithmeticConstant is the value of the Val element.

attribute **ArithmeticConstantType/@val**

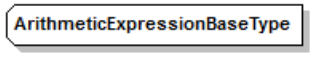
type	**xs:decimal**
properties	use required

annotation	documentation The required val attribute is an arithmetic constant.

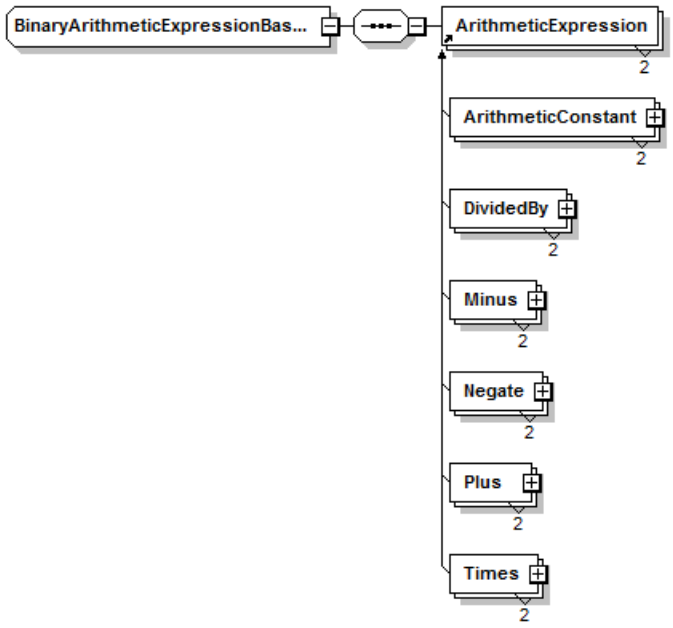
complexType **ArithmeticEqualType**

diagram	
type	extension of **ArithmeticComparisonBaseType**
properties	base ArithmeticComparisonBaseType abstract true
children	**ArithmeticExpression**
used by	element **ArithmeticEqual**
annotation	documentation The ArithmeticEqualType defines a test if the two ArithmeticExpression elements are equal. The ArithmeticEqualType evaluates to true if the first ArithmeticExpression element is (1) less than the second ArithmeticExpression element plus ComparisonTiny and (2) greater then the second ArithmeticExpression element minus ComparisonTiny. Otherwise, it evaluates to false.

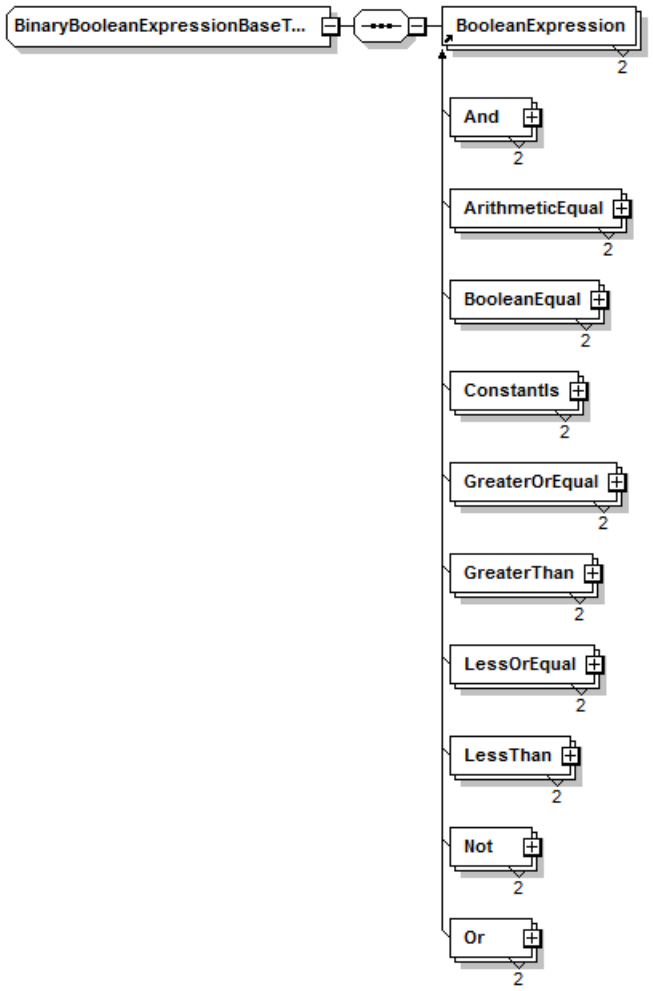
complexType **ArithmeticExpressionBaseType**

diagram	
properties	abstract true
used by	element **ArithmeticExpression** complexTypes **ArithmeticConstantType** **BinaryArithmeticExpressionBaseType** **NegateType**
annotation	documentation The ArithmeticExpressionBaseType defines the base type for arithmetic expressions. All derived types of ArithmeticExpressionBaseType evaluate to an xs:decimal.

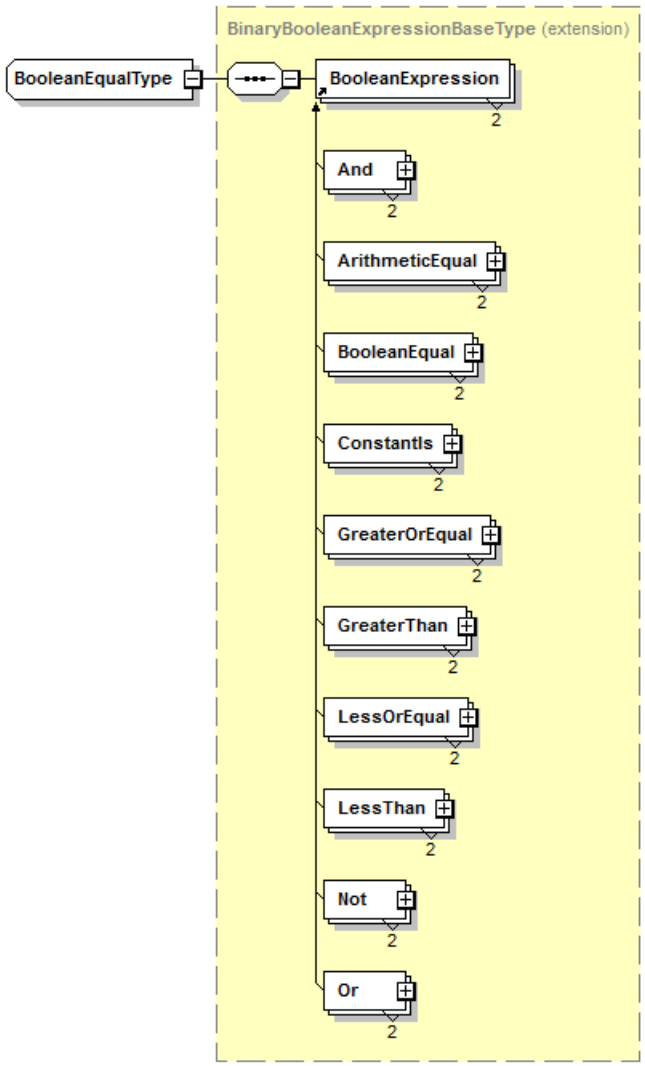
complexType **BinaryArithmeticExpressionBaseType**

diagram	
type	extension of **ArithmeticExpressionBaseType**
properties	base ArithmeticExpressionBaseType abstract true
children	**ArithmeticExpression**
used by	complexTypes **DividedByType** **MinusType** **PlusType** **TimesType**
annotation	documentation The BinaryArithmeticExpressionBaseType is the base type for binary arithmetic expressions.

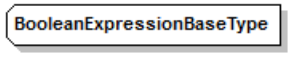
complexType **BinaryBooleanExpressionBaseType**

diagram	
type	extension of **BooleanExpressionBaseType**
properties	base BooleanExpressionBaseType abstract true
children	**BooleanExpression**
used by	complexType **BooleanEqualType**
annotation	documentation The BinaryBooleanExpressionBaseType is the base type for binary Boolean expressions.

complexType **BooleanEqualType**

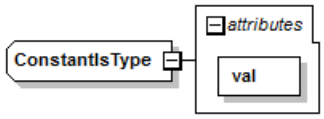
diagram	
type	extension of **BinaryBooleanExpressionBaseType**
properties	base BinaryBooleanExpressionBaseType
children	**BooleanExpression**
used by	element **BooleanEqual**
annotation	documentation The BooleanEqualType defines a test of whether two Boolean expressions are the same. The BooleanEqualType evaluates to true if the two BooleanExpression elements both evaluate to true or both evaluate to false. Otherwise, the BooleanEqualType evaluates to false.

complexType **BooleanExpressionBaseType**

diagram	
properties	abstract true
used by	element **BooleanExpression** complexTypes **AndType** **ArithmeticComparisonBaseType** **BinaryBooleanExpressionBaseType** **ConstantIsType** **NotType** **OrType**

annotation	documentation The BooleanExpressionBaseType defines the base type for Boolean expressions. All derived types of BooleanExpressionBaseType evaluate to true or false.

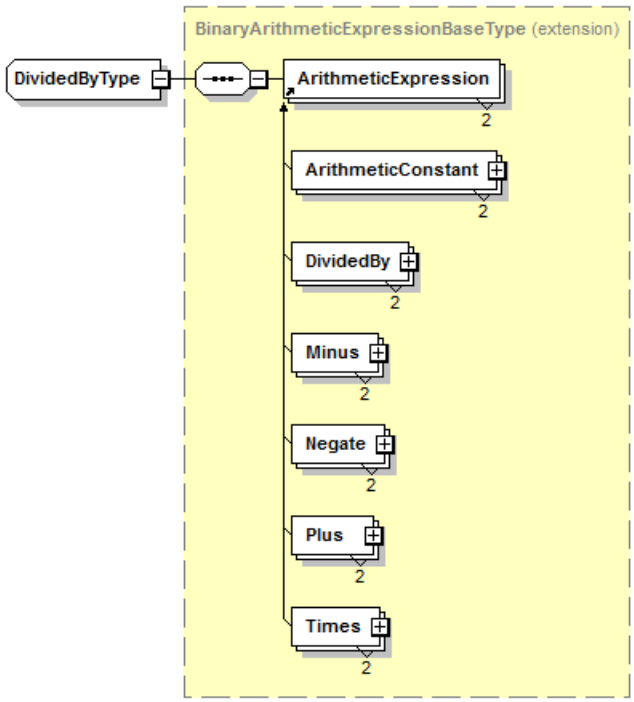
complexType **ConstantIsType**

diagram	
type	extension of **BooleanExpressionBaseType**
properties	base BooleanExpressionBaseType
used by	element **ConstantIs**
attributes	Name: val Type: **BooleanConstantEnumType** Use: required Default: Fixed: Annotation: documentation The required val attribute is a Boolean constant given as an enumeration.
annotation	documentation The ConstantIsType evaluates to true if the val is QIF_TRUE and evaluates to false if the val is QIF_FALSE.

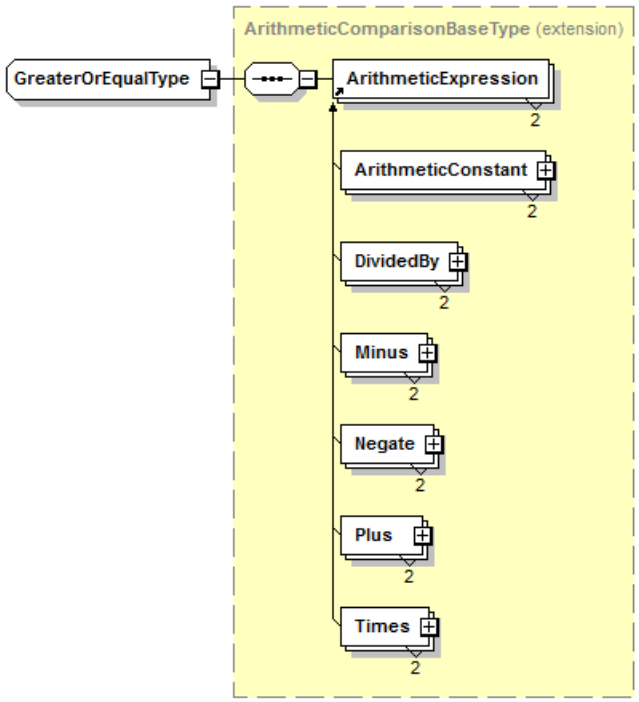
attribute **ConstantIsType/@val**

type	**BooleanConstantEnumType**
properties	use required
facets	Kind: enumeration Value: QIF_TRUE Annotation: Kind: enumeration Value: QIF_FALSE Annotation:
annotation	documentation The required val attribute is a Boolean constant given as an enumeration.

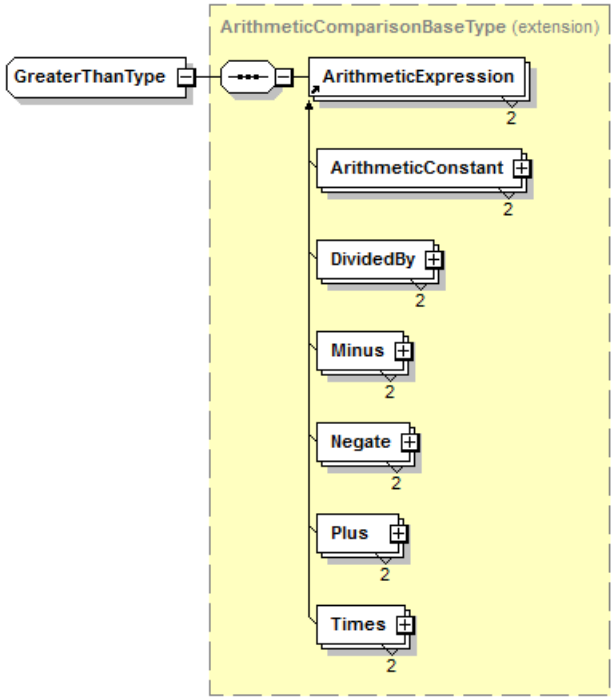
complexType **DividedByType**

diagram	
type	extension of **BinaryArithmeticExpressionBaseType**
properties	base BinaryArithmeticExpressionBaseType
children	**ArithmeticExpression**
used by	element **DividedBy**
annotation	documentation The DividedByType defines division. A DividedByType evaluates to the quotient of the values of its two ArithmeticExpression elements (the first divided by the second). The second ArithmeticExpression element must not evaluate to zero.

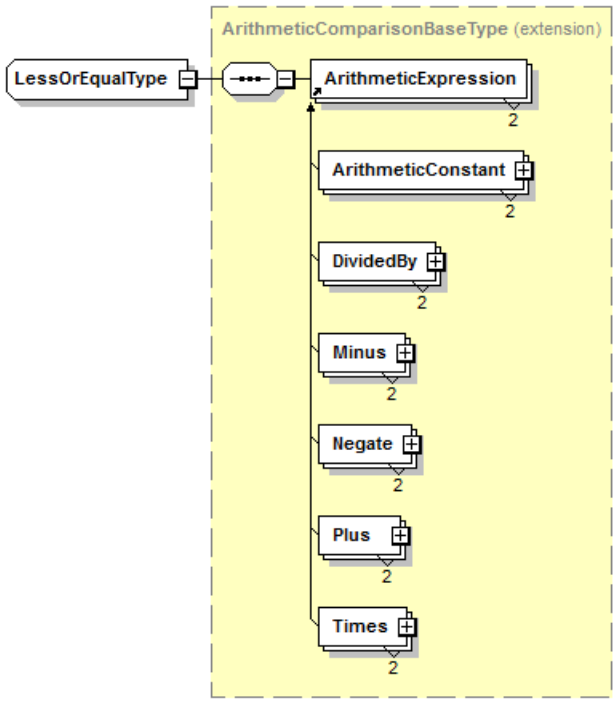
complexType **GreaterOrEqualType**

diagram	
type	extension of **ArithmeticComparisonBaseType**
properties	base ArithmeticComparisonBaseType
children	**ArithmeticExpression**
used by	element **GreaterOrEqual**
annotation	documentation The GreaterOrEqualType defines a test if the first ArithmeticExpression element is greater than or equal to the second ArithmeticExpression element. The GreaterOrEqualType evaluates to true if the value of the first ArithmeticExpression element is greater than the value of the second ArithmeticExpression element minus ComparisonTiny. Otherwise, it evaluates to false.

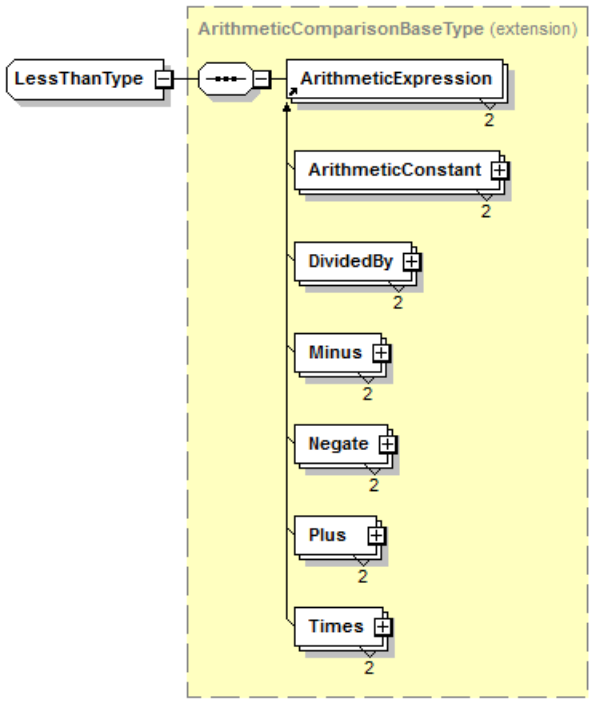
complexType **GreaterThanType**

diagram	
type	extension of **ArithmeticComparisonBaseType**
properties	base ArithmeticComparisonBaseType
children	**ArithmeticExpression**
used by	element **GreaterThan**
annotation	documentation The GreaterThanType defines a test if the first ArithmeticExpression element is greater than the second ArithmeticExpression element. The GreaterThanType evaluates to true if the value of the first ArithmeticExpression element is greater than or equal to the value of the second ArithmeticExpression element plus ComparisonTiny. Otherwise, it evaluates to false.

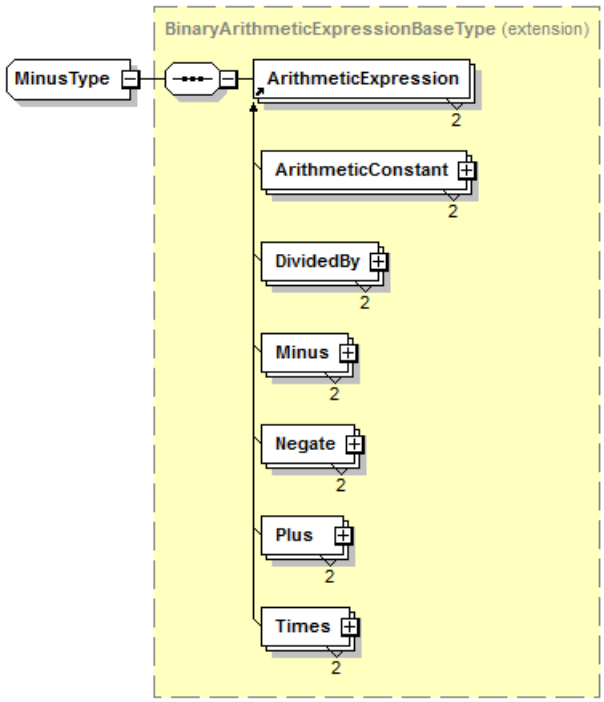
complexType **LessOrEqualType**

diagram	
type	extension of **ArithmeticComparisonBaseType**
properties	base ArithmeticComparisonBaseType
children	**ArithmeticExpression**
used by	element **LessOrEqual**
annotation	documentation The LessOrEqualType defines a test if the first ArithmeticExpression element is less than or equal to the second ArithmeticExpression element. The LessOrEqualType evaluates to true if the value of the first ArithmeticExpression element is less than the value of the second ArithmeticExpression element plus ComparisonTiny. Otherwise, it evaluates to false.

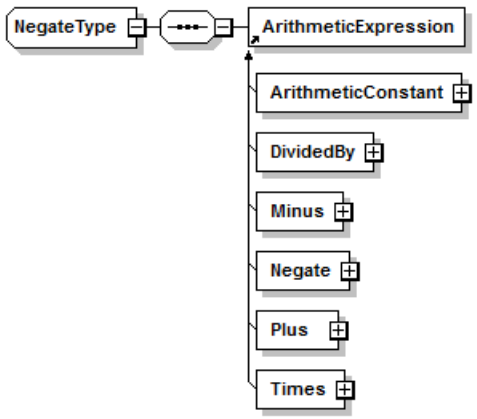
complexType **LessThanType**

diagram	
type	extension of **ArithmeticComparisonBaseType**
properties	base ArithmeticComparisonBaseType
children	**ArithmeticExpression**
used by	element **LessThan**
annotation	documentation The LessThanType defines a test if the first ArithmeticExpression element is less than the second ArithmeticExpression element. The LessThanType evaluates to true if the value of the first ArithmeticExpression element is less than or equal to the value of the second ArithmeticExpression element minus ComparisonTiny. Otherwise, it evaluates to false.

complexType **MinusType**

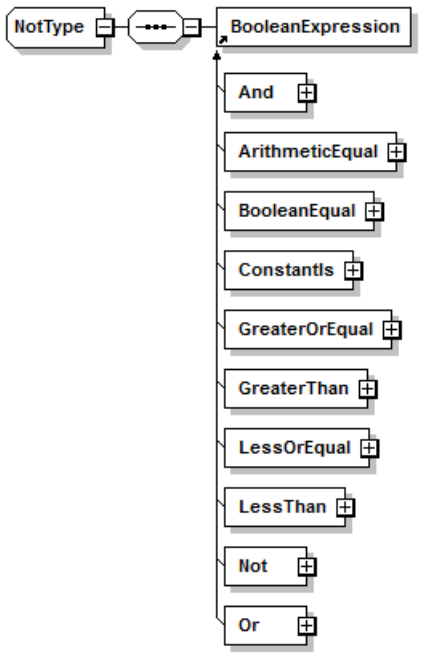
diagram	
type	extension of **BinaryArithmeticExpressionBaseType**
properties	base BinaryArithmeticExpressionBaseType
children	**ArithmeticExpression**
used by	element **Minus**
annotation	documentation The MinusType defines subtraction. A minus type evaluates to the value of the first ArithmeticExpression element minus the value of the second ArithmeticExpression element.

complexType **NegateType**

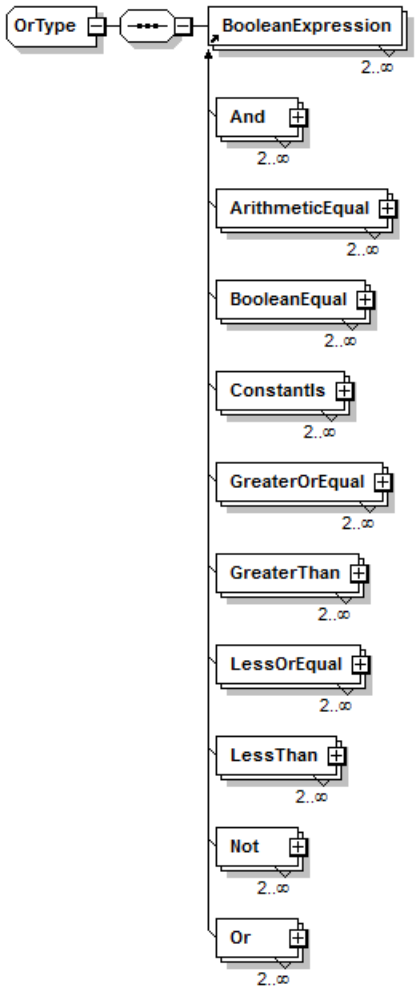
diagram	
type	extension of **ArithmeticExpressionBaseType**
properties	base ArithmeticExpressionBaseType
children	**ArithmeticExpression**

used by	element **Negate**
annotation	documentation The NegateType defines an arithmetic operations that changes the sign of a number. The value of a NegateType is the negative of the value of the ArithmeticExpression element.

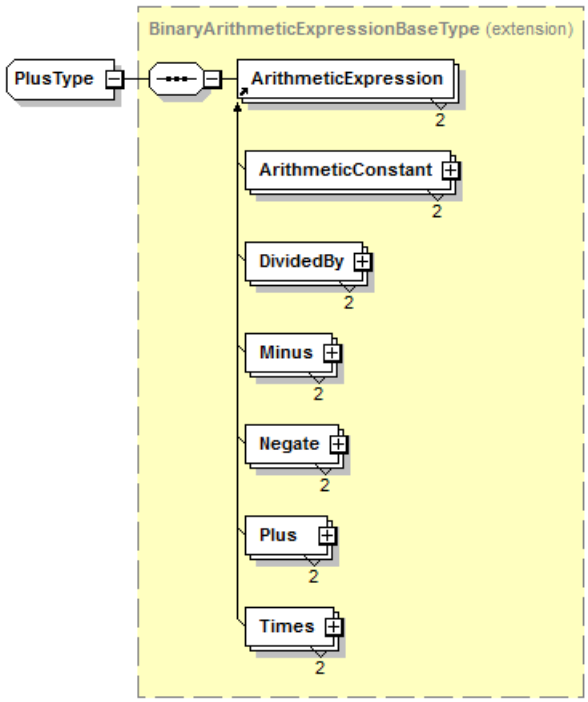
complexType **NotType**

diagram	
type	extension of **BooleanExpressionBaseType**
properties	base BooleanExpressionBaseType
children	**BooleanExpression**
used by	element **Not**
annotation	documentation The NotType defines a Boolean expression that negates another Boolean expression. The NotType evaluates to true if the BooleanExpression element evaluates to false, and the NotType evaluates to false if the BooleanExpression element evaluates to true.

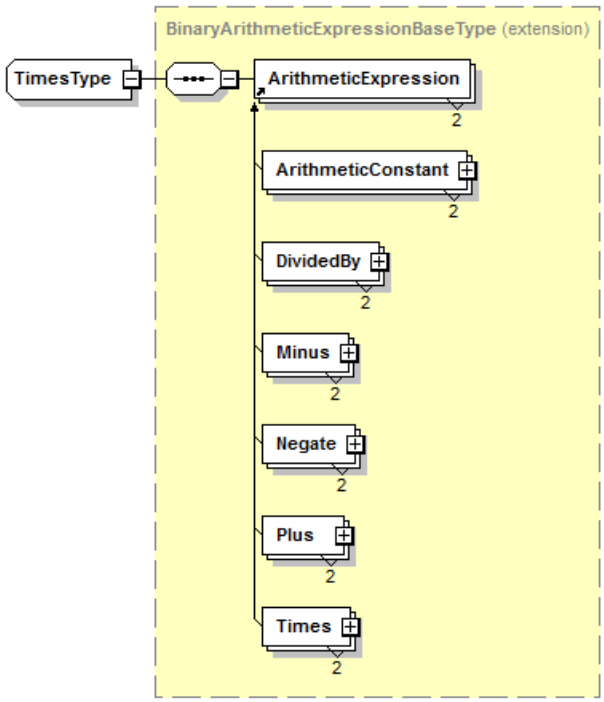
complexType **OrType**

diagram	
type	extension of **BooleanExpressionBaseType**
properties	base BooleanExpressionBaseType
children	**BooleanExpression**
used by	element **Or**
annotation	documentation The OrType defines an 'or' Boolean expression. The OrType evaluates to true if any of the BooleanExpression elements it contains evaluate to true. Otherwise, it evaluates to false. The BooleanExpression elements must be evaluated in order. If any BooleanExpression element evaluates to true, the remaining BooleanExpression elements must not be evaluated.

complexType **PlusType**

diagram	
type	extension of **BinaryArithmeticExpressionBaseType**
properties	base BinaryArithmeticExpressionBaseType
children	**ArithmeticExpression**
used by	element **Plus**
annotation	documentation The PlusType defines addition. A plus type evaluates to the sum of the values of its two ArithmeticExpression elements.

complexType **TimesType**

diagram	
type	extension of **BinaryArithmeticExpressionBaseType**
properties	base BinaryArithmeticExpressionBaseType
children	**ArithmeticExpression**
used by	element **Times**
annotation	documentation The TimesType defines multiplication. A TimesType evaluates to the product of the values of its two ArithmeticExpression elements.

simpleType **BooleanConstantEnumType**

type	restriction of **xs:NMTOKEN**
properties	base xs:NMTOKEN
used by	attribute **ConstantIsType/@val**
facets	Kind Value Annotation enumeration QIF_TRUE enumeration QIF_FALSE
annotation	documentation The BooleanConstantEnumType enumerates values of the two Boolean constants. (The names are of these enumerated values are chosen to avoid conflicts with reserved words and macro definitions in common computer languages.)

