

Abstraktes Strukturmodell des Wavis-Editors.

Table of Contents

1. General Description	3
2. Language Constructs	3
2.1. DataSource	3
2.2. DataSourceDB	3
2.3. DataBase	3
2.4. DBColumn	4
2.5. DBColumnType	5
2.6. DBString	5
2.7. DBInteger	5
2.8. DBFloat	5
2.9. DBDouble	5
2.10. DBBigInt	5
2.11. DBDate	5
2.12. DBTimeStamp	6
2.13. DBBoolean	6
2.14. DBBlob	6
2.15. DataSourceProjectionFilter	6
2.16. Projection	6
2.17. ColumnProjection	7
2.18. ComputationProjection	7
2.19. LimitDataSource	7
2.20. Expression	8
2.21. BinaryExpression	8
2.22. UnaryExpression	8
2.23. ConstantExpression	8
2.24. ColumnValueExpression	9
2.25. ParameterValueExpression	9
2.26. ComputationExpression	9
2.27. DataSourceValue	9
2.28. DataSourceJoin	10
2.29. DataSourceModifyOperation	10
2.30. DataSourceInsert	11
2.31. DataSourceUpdate	11
2.32. DataSourceDelete	11
2.33. DataSourceActualParam	12
2.34. DataSourceFormalParam	12
2.35. Formular	12
2.36. FormularElement	13
2.37. FormularRadioGroupElement	13
2.38. FormularTextElement	14
2.39. FormularTextAreaElement	14
2.40. FormularPasswordElement	14
2.41. FormularCheckboxElement	14
2.42. FormularSelectElement	14
2.43. FormularSelectOption	15
2.44. FormularConstantSelectOption	15
2.45. FormularDataSourceSelectOption	15
2.46. FormularPHPScriptSelectOption	16
2.47. FormularRadioElement	16
2.48. FormularCondition	16
2.49. FormularConditionDefinition	17
2.50. FormularConditionFormalParam	17
2.51. FormularConditionActualParam	17
2.52. WebNodeVariableDefinition	18
2.53. WebNodeActualParam	18

2.54. WebNodeFormalParam	18
2.55. WebNodeExpression	18
2.56. WebNodeConstantExpression	19
2.57. WebNodeVariableExpression	19
2.58. WebNodeColumnExpression	19
2.59. Component	19
2.60. Paragraph	20
2.61. Fragment	20
2.62. TextFragment	20
2.63. ImageFragment	20
2.64. Heading	21
2.65. Table	21
2.66. TableRow	21
2.67. TableColumn	22
2.68. TableCell	22
2.69. LinkTarget	22
2.70. LinkTargetIntern	22
2.71. LinkTargetExtern	23
2.72. Root	23
2.73. Library	23
2.74. NavigationElement	24
2.75. WebApplication	24
2.76. RootWebZone	25
2.77. SubWebZone	26
2.78. WebRole	26
2.79. WebRoleRef	27
2.80. WebNode	27
2.81. WebPage	27
2.82. Frame	28
2.83. GlobalFrame	28
2.84. NavigationBar	28
2.85. NavigationBarElement	29
2.86. FrameRef	29
2.87. MetaInformation	29
2.88. MetaInformationElement	30
2.89. OutputElement	30
2.90. ListOutputElement	30
2.91. StyleSheet	31
2.92. StyleSheetBlock	31
2.93. CustomStyleSheetBlock	31
2.94. StyleSheetPair	32
2.95. AttrValue	32
2.96. SizeValue	32
2.97. ColorValue	33
2.98. FamilyValue	33
2.99. StyleValue	33
2.100. HeadingValue	33
2.101. AlignmentValue	33
2.102. ListValue	34
2.103. CustomValue	34
2.104. HtmlStyleSheetBlock	34
2.105. Operation	34
2.106. ActionHandler	35
2.107. WebAction	35
2.108. PHPOperation	35
2.109. DataBaseOperation	36

1. General Description

Dieses Dokument beschreibt das Strukturmodell des Wavis-Editors. Mit dem Wavis-Editor ist der Benutzer in der Lage, Web-Anwendungen, basierend auf HTML und PHP, visuell zu modellieren.

2. Language Constructs

2.1. DataSource

Eine DataSource ist ein Objekt aus dem oder in das Daten ein- oder ausgegeben werden können. Eine DataSource kann eine Datenbank, TXT-Datei o.ä. sein. Sie wird mit einem aufrufenden Skript verknüpft.

```
ABSTRACT CLASS DataSource {  
    name: VAL VLString;  
    position: VAL VLPoint?;  
}
```

Direct subclasses: DataSourceDB, DataSourceProjectionFilter, LimitDataSource, DataSourceJoin.

Subnode of: DataBase.

Referred by: ListOutputElement, Formular, DataSourceJoin, LimitDataSource, DataSourceProjectionFilter.

name

Name der DataSource.

position

Position für visuelle Darstellung.

2.2. DataSourceDB

Eine DataSourceDB ist ein spezieller Typ der DataSource, nämlich eine auf eine Datenbank zugreifende DataSource.

```
CLASS DataSourceDB INHERITS DataSource {  
    columns: SUB DBColumn*;  
}
```

Referred by: DataSourceModifyOperation.

columns

Die Spalten der Datenbank DataSource

2.3. DataBase

Dient der Festlegung diverser Datenbank-Parameter.

```
CLASS DataBase {  
    name: VAL VLString;  
    userName: VAL VLString?;  
    userPassword: VAL VLString?;  
    hostName: VAL VLString;  
    port: VAL VLInt;
```

Abstraktes Strukturmodell des Wavis-Editors.

```
dataBaseType: VAL VLString;  
dataSources: SUB DataSource*;  
dataSourceModifications: SUB DataSourceModifyOperation*;  
setSize: VAL VLPoint?;  
modifySetSize: VAL VLPoint?;  
}
```

Subnode of: WebApplication.

name

Der Name der Datenbank.

userName

Der Name des Datenbank-Benutzers.

userPassword

Das Passwort für den Datenbank-Benutzer.

hostName

Der Hostname des Datenbankservers.

port

Der Port des Datenbankservers.

dataBaseType

Der Typ der Datenbank (MySQL, Oracle, ...).

dataSources

Es kann eine Menge von Datenquellen definiert werden.

dataSourceModifications

Eine Menge von Datenbankoperationen, die eine DataSource veraendern

setSize

Layout-Attribut für grafische Darstellung.

modifySetSize

No description available.

2.4. DBColumn

Definiert die Spalte einer Datenbank-Tabelle.

```
CLASS DBColumn {  
  name: VAL VLString;  
  type: SUB DBColumnType!;  
}
```

Subnode of: DataSourceDB.

Referred by: WebNodeColumnExpression, DataSourceJoin, DataSourceValue, ComputationExpression, ColumnValueExpression, ComputationProjection, ColumnProjection.

name

Der Name der Spalte.

type

Legt den Typ des Spalten-Inhaltes fest.

2.5. DBColumnType

Abstrakte Basis-Klasse für Werttypen von Datenbank-Spalten.

```
ABSTRACT CLASS DBColumnType {  
}
```

Direct subclasses: DBString, DBInteger, DBFloat, DBDouble, DBBigInt, DBDate, DBTimeStamp, DBBoolean, DBBlob.

Subnode of: WebNodeFormalParam, DataSourceFormalParam, ConstantExpression, DBColumn.

2.6. DBString

Werte vom Typ String.

```
CLASS DBString INHERITS DBColumnType {  
}
```

2.7. DBInteger

Werte vom Typ Integer.

```
CLASS DBInteger INHERITS DBColumnType {  
}
```

2.8. DBFloat

Werte vom Typ Float.

```
CLASS DBFloat INHERITS DBColumnType {  
}
```

2.9. DBDouble

Werte vom Typ Double.

```
CLASS DBDouble INHERITS DBColumnType {  
}
```

2.10. DBBigInt

Werte vom Typ BigInt.

```
CLASS DBBigInt INHERITS DBColumnType {  
}
```

2.11. DBDate

Werte vom Typ Date.

```
CLASS DBDate INHERITS DBColumnType {  
}
```

2.12. DBTimeStamp

Werte vom Typ TimeStamp.

```
CLASS DBTimeStamp INHERITS DBColumnType {  
}
```

2.13. DBBoolean

Werte vom Typ Boolean.

```
CLASS DBBoolean INHERITS DBColumnType {  
}
```

2.14. DBBlob

Zur Aufnahme von Binärdaten, zum Beispiel Bilder.

```
CLASS DBBlob INHERITS DBColumnType {  
}
```

2.15. DataSourceProjectionFilter

Wählt Spalten einer Datenquelle aus

```
CLASS DataSourceProjectionFilter INHERITS DataSource {  
  sourceDataSource: REF DataSource;  
  condition: SUB Expression!;  
  formalParameters: SUB DataSourceFormalParam*;  
  projectionColumns: SUB Projection*;  
}
```

Referred by: FormularDataSourceSelectOption.

sourceDataSource

Die DataSource, die als Quelle dienen soll.

condition

Die Where Klausel der Datenbankabfrage

formalParameters

Formale Parameter fuer die Where Klausel

projectionColumns

Die Spalten, die ausgewaehlt werden sollen

2.16. Projection

Abstrakte Oberklasse fuer die Projektionsspalten

```
ABSTRACT CLASS Projection {  
}
```

Direct subclasses: ColumnProjection, ComputationProjection.

Subnode of: DataSourceProjectionFilter.

2.17. ColumnProjection

Eine Spalte zur Projektion auswaehlen

```
CLASS ColumnProjection INHERITS Projection {  
  column: REF DBColumn;  
}
```

column

Referenz auf eine Spalte, die ausgewaehlt werden soll

2.18. ComputationProjection

Eine Projektion, die eine Berechnung darstellt, z.B. das Aufsummieren von Zeilen einer Spalte

```
CLASS ComputationProjection INHERITS Projection {  
  column: REF DBColumn;  
  computation: VAL VLString;  
}
```

column

Die Spalte, auf die eine Berechnung angewendet werden soll.

computation

Die Berechnung, die ausgefuehrt werden soll.

2.19. LimitDataSource

Wählt Zeilen einer DataSource aus, je nach beliebigen fest eingebaute Werte oder durch Parameter übergebene.

```
CLASS LimitDataSource INHERITS DataSource {  
  sourceDataSource: REF DataSource;  
  minRow: SUB Expression!;  
  maxRow: SUB Expression!;  
  formalParameters: SUB DataSourceFormalParam*;  
}
```

sourceDataSource

Die Quelle, aus der Zeilen ausgewaehlt werden sollen.

minRow

Der erste Parameter fuer die Limit Klausel, untere Schranke

maxRow

Der zweite Parameter fuer die Limit Klausel, obere Schranke

formalParameters

Formale Parameter fuer minRow und maxRow

2.20. Expression

Der zu filternde Ausdruck, kann eine Baumstruktur aufbauen

```
ABSTRACT CLASS Expression {  
}
```

Direct subclasses: BinaryExpression, UnaryExpression, ConstantExpression, ColumnValueExpression, ParameterValueExpression, ComputationExpression.

Subnode of: DataSourceDelete, DataSourceUpdate, DataSourceValue, UnaryExpression, BinaryExpression, LimitDataSource, DataSourceProjectionFilter.

2.21. BinaryExpression

Verknüpft zwei Ausdrücke

```
CLASS BinaryExpression INHERITS Expression {  
  leftOperand: SUB Expression!;  
  rightOperand: SUB Expression!;  
  operator: VAL VLString;  
}
```

leftOperand

Linker Teilbaum der Verknuepfung.

rightOperand

Rechter Teilbaum der Verknuepfung.

operator

Beliebiger Operator, der auch in SQL existiert.

2.22. UnaryExpression

No description available.

```
CLASS UnaryExpression INHERITS Expression {  
  operand: SUB Expression!;  
  operator: VAL VLString;  
}
```

operand

No description available.

operator

No description available.

2.23. ConstantExpression

Filtert nach einem konstanten Wert

```
CLASS ConstantExpression INHERITS Expression {  
  type: SUB DBColumnType!;  
  value: VAL VLString;  
}
```

type

Der Typ des Wertes.

value

Der Wert

2.24. ColumnValueExpression

Eine Spalte auf die ein Filter angewendet werden soll

```
CLASS ColumnValueExpression INHERITS Expression {  
  column: REF DBColumn;  
}
```

column

Die Spalte

2.25. ParameterValueExpression

Filtert nach einem übergebenen Parameter.

```
CLASS ParameterValueExpression INHERITS Expression {  
  parameter: REF DataSourceFormalParam;  
}
```

parameter

Referenz auf einen formalen Parameter, der in der engsten Umgebung definiert ist.

2.26. ComputationExpression

Berechnet Werte ueber die Tupel einer Spalte

```
CLASS ComputationExpression INHERITS Expression {  
  column: REF DBColumn;  
  computation: VAL VLString;  
}
```

column

Referenz auf die Spalte, mit der eine Berechnung ausgefuehrt werden soll.

computation

Die Berechnung, die ausgefuehrt werden soll, muss in SQL definiert sein!

2.27. DataSourceValue

Stellt einen Wert für den DataSourceInsert bzw. DataSourceUpdate dar

```
CLASS DataSourceValue {  
    column: REF DBColumn;  
    expression: SUB Expression!;  
}
```

Subnode of: DataSourceUpdate, DataSourceInsert.

column

Referenz auf eine Spalte der Datenbank

expression

Der Ausdruck, der im Funktionsaufruf steht, hier darf nur eine Constant oder eine ParameterValue Expression stehen

2.28. DataSourceJoin

Verknuepft zwei DataSources oder zwei DataSourceJoins

```
CLASS DataSourceJoin INHERITS DataSource {  
    dataSource1: REF DataSource;  
    dataSource2: REF DataSource;  
    columnRef1: REF DBColumn;  
    columnRef2: REF DBColumn;  
    joinOperator: VAL VLString;  
}
```

dataSource1

Referenz auf die erste DataSource.

dataSource2

Referenz auf die zweite DataSource.

columnRef1

Referenz auf eine Spalte der ersten DataSource, ON Operator in SQL.

columnRef2

Referenz auf eine Spalte der zweiten DataSource.

joinOperator

Art des Joins, bisher wird nur JOIN unterstuetzt.

2.29. DataSourceModifyOperation

Entweder ein Update oder ein Insert

```
ABSTRACT CLASS DataSourceModifyOperation {  
    targetDataSource: REF DataSourceDB;  
    formalParams: SUB DataSourceFormalParam*;  
    name: VAL VLString;  
    position: VAL VLPoint?;  
}
```

Direct subclasses: DataSourceInsert, DataSourceUpdate, DataSourceDelete.

Subnode of: DataBase.

Referred by: DataBaseOperation.

targetDataSource

DataSource die modifiziert werden soll.

formalParams

Formale Parameter ggf. fuer die Where Klausel der Update Operation.

name

No description available.

position

No description available.

2.30. DataSourceInsert

Eine Insert Operation

```
CLASS DataSourceInsert INHERITS DataSourceModifyOperation {  
    values: SUB DataSourceValue*;  
}
```

values

Die Werte, die eingesetzt werden sollen.

2.31. DataSourceUpdate

SQL Update Operation.

```
CLASS DataSourceUpdate INHERITS DataSourceModifyOperation {  
    conditions: SUB Expression!;  
    values: SUB DataSourceValue*;  
}
```

conditions

Die Bedingung, die Tupel selektiert, die veraendert werden sollen.

values

Die einzusetzenden Werte

2.32. DataSourceDelete

Löscht Tupel aus einer DataSource

```
CLASS DataSourceDelete INHERITS DataSourceModifyOperation {  
    conditions: SUB Expression!;  
}
```

conditions

Selektiert die Tupel, die gelöscht werden sollen.

2.33. DataSourceActualParam

Ein aktueller Parameter für die DataSource

```
CLASS DataSourceActualParam {  
  formalParam: REF DataSourceFormalParam;  
  expressions: SUB WebNodeExpression!;  
}
```

Subnode of: DataBaseOperation, ListOutputElement, FormularDataSourceSelectOption.

formalParam

Der formale Parameter auf den sich dieser aktuelle Parameter bezieht

expressions

Der Ausdruck, mit dem der aktuelle Parameter gefüllt werden muß, kann entweder konstant sein, oder sich mit einem WebNodeVariabExpression auf zum Beispiel Formularelemente beziehen.

2.34. DataSourceFormalParam

Der formale Parameter für den DataSourceFilter.

```
CLASS DataSourceFormalParam {  
  name: VAL VLString;  
  type: SUB DBColumnType!;  
}
```

Subnode of: DataSourceModifyOperation, LimitDataSource, DataSourceProjectionFilter.

Referred by: DataSourceActualParam, ParameterValueExpression.

name

Der Name des formalen Parameters

type

Der Typ des formalen Parameters, es sind alle Typen, die MySQL versteht möglich.

2.35. Formular

Die Klasse Formular dient der Anzeige oder Eingabe von Daten in oder aus einem Datenbestand (Datenbank, XML-Datei, Textdatei).

```
CLASS Formular INHERITS Component {  
  action: REF WebNode;  
  method: VAL VLString;  
  elements: SUB FormularElement*;  
  datasource: REF DataSource?;  
  actualParams: SUB WebNodeActualParam*;  
}
```

action	Verweis auf ein Skript an welches die eingegebenen Daten geschickt werden (falls Eingabeformular).
method	Methode mit der die eingegebenen Daten an das Skript übergeben werden (GET/POST).
elements	Eine Liste von Formularelementen.
datasource	Verweis auf die Datenbanktabelle aus bzw. in die die Daten geschrieben / gelesen werden.
actualParams	Aktuelle Parameter für die Parameterübergabe.

2.36. FormularElement

Ein Formularelement dient der Ausgabe / Eingabe von einzelnen Werten. Es besteht aus einem Namen, der vor dem Eingabe-/Ausgabefeld angezeigt wird, dem Eingabe-/Ausgabefeld, sowie einer darauf folgenden Beschreibung).

```
ABSTRACT CLASS FormularElement INHERITS WebNodeVariableDefinition {  
  name: VAL VLString;  
  label: VAL VLString?;  
  description: VAL VLString?;  
  pvalue: REF WebNodeFormalParam?;  
  condition: SUB FormularCondition%;  
}
```

Direct subclasses: FormularRadioGroupElement, FormularTextElement, FormularTextAreaElement, FormularPasswordElement, FormularCheckboxElement, FormularSelectElement.

Subnode of: Formular.

name	Interner HTML-Name der für die Parameterübergabe verwendet wird.
label	Beschriftung des Formular-Elementes.
description	Die Beschreibung nach dem Eingabefeld.
pvalue	Formaler Parameter, mit dessen Hilfe ein Formular-Element mit einem Wert vorgelegt werden kann.
condition	Konsistenz-Check zur Überprüfung der Eingabewerte aus dem Formular-Element.

2.37. FormularRadioGroupElement

Formular-Element zur Gruppierung von Radiobutton-Elementen.

```
CLASS FormularRadioGroupElement INHERITS FormularElement {  
    elements: SUB FormularRadioElement*;  
    checked: REF FormularRadioElement;  
}
```

elements

Liste der Radiobutton-Elemente.

checked

Radiobutton, der bereits vorselektiert sein soll.

2.38. FormularTextElement

Repräsentation des HTML-INPUT-Elementes vom Typ 'text'.

```
CLASS FormularTextElement INHERITS FormularElement {  
}
```

2.39. FormularTextAreaElement

Repräsentation des HTML-Elementes vom Typ 'textarea'.

```
CLASS FormularTextAreaElement INHERITS FormularElement {  
}
```

2.40. FormularPasswordElement

Repräsentation des HTML-INPUT-Elementes vom Typ 'password'.

```
CLASS FormularPasswordElement INHERITS FormularElement {  
}
```

2.41. FormularCheckboxElement

Repräsentation des HTML-INPUT-Elementes vom Typ 'checkbox'.

```
CLASS FormularCheckboxElement INHERITS FormularElement {  
    value: VAL VLString;  
    checked: VAL VLBoolean;  
}
```

value

Wert, der bei der Parameterübergabe übergeben werden soll.

checked

2.42. FormularSelectElement

Repräsentation des HTML-Elementes vom Typ 'select'.

```
CLASS FormularSelectElement INHERITS FormularElement {  
  options: SUB FormularSelectOption!;  
  size: VAL VLInt;  
  multiple: VAL VLBoolean;  
}
```

options

Objekt, welches die Option-Einträge innerhalb des Select-Elementes repräsentiert.

size

Gibt die Anzahl der gleichzeitig sichtbaren Option-Elemente an.

multiple

Gibt an, ob Mehrfachauswahl im Select-Element erlaubt ist.

2.43. FormularSelectOption

Abstrakte Klasse für Option-Elemente innerhalb eines HTML-Select-Elementes.

```
ABSTRACT CLASS FormularSelectOption {  
}
```

Direct subclasses: FormularConstantSelectOption, FormularDataSourceSelectOption, FormularPHPScriptSelectOption.

Subnode of: FormularSelectElement.

2.44. FormularConstantSelectOption

Repräsentiert eine Liste von Option-Elementen, die aus einer Liste mit konstanten Werten erzeugt wird.

```
CLASS FormularConstantSelectOption INHERITS FormularSelectOption {  
  values: VAL VLString;  
}
```

values

Liste mit konstanten Werten für die Option-Elemente.

2.45. FormularDataSourceSelectOption

Repräsentiert eine Liste von Option-Elementen, die aus Werten, die aus eine Datenbank stammen, erzeugt werden.

```
CLASS FormularDataSourceSelectOption INHERITS FormularSelectOption {  
  values: REF DataSourceProjectionFilter;  
  params: SUB DataSourceActualParam*;  
}
```

values

Referenz auf einen DataSourceProjectionFilter. ACHTUNG! Darf nur Werte aus genau einer Spalte zurück liefern!!!

params

No description available.

2.46. FormularPHPScriptSelectOption

Repräsentiert eine Liste von Option-Elementen, die aus Werten, die aus einem PHP-Skript stammen, erzeugt werden.

```
CLASS FormularPHPScriptSelectOption INHERITS FormularSelectOption {  
    script: VAL VLString;  
}
```

script

Das PHP-Skript, in dem die Werte für das Option-Elemente erzeugt werden. Um die Werte für die Option-Elemente zu erzeugen muß das PHP-Array \$values mit Werten belegt werden.

2.47. FormularRadioElement

Repräsentation des HTML-INPUT-Elementes vom Typ 'radio'.

```
CLASS FormularRadioElement {  
    value: VAL VLString;  
    label: VAL VLString;  
    name: VAL VLString;  
}
```

Subnode of: FormularRadioGroupElement.

Referred by: FormularRadioGroupElement.

value

Wert, der bei der Parameterübergabe übergeben werden soll.

label

Beschriftung des Radiobutton-Elementes.

name

Name, der bei der Parameterübergebe übergeben werden soll.

2.48. FormularCondition

Konsistenzchecks für Daten aus einem Eingabefeld.

```
CLASS FormularCondition {  
    condition: REF FormularConditionDefinition?;  
    params: SUB FormularConditionActualParam*;  
}
```

Subnode of: FormularElement.

condition

Definition der Konsistenzcheck-Funktion.

params

Aktuelle Parameter, die an die Konsistenzcheck-Funktion übergeben werden.

2.49. FormularConditionDefinition

Definition der Konsistenzcheck-Funktion.

```
CLASS FormularConditionDefinition {  
  name: VAL VLString;  
  params: SUB FormularConditionFormalParam*;  
  body: VAL VLString;  
}
```

Subnode of: Library.

Referred by: FormularCondition.

name

Name der Funktion.

params

Formale Parameter der Funktion.

body

Rumpf der Funktion.

2.50. FormularConditionFormalParam

Formaler Parameter für Konsistenzcheck-Funktion.

```
CLASS FormularConditionFormalParam {  
  name: VAL VLString;  
}
```

Subnode of: FormularConditionDefinition.

Referred by: FormularConditionActualParam.

name

Name der formalen Parameters.

2.51. FormularConditionActualParam

Aktueller Parameter für Konsistenzcheck-Funktionsaufruf.

```
CLASS FormularConditionActualParam {  
  value: VAL VLString;  
  formalparam: REF FormularConditionFormalParam;  
}
```

Subnode of: FormularCondition.

value

Beim Funktionsaufruf übergebener Wert.

formalparam

Formaler Parameter an den der Wert übergeben wird.

2.52. WebNodeVariableDefinition

Abstrakte Oberklasse für FormularElemente und WebNodeFormalParams.

```
ABSTRACT CLASS WebNodeVariableDefinition {  
}
```

Direct subclasses: FormularElement, WebNodeFormalParam.

Referred by: WebNodeVariableExpression.

2.53. WebNodeActualParam

Aktueller Parameter für Parameterübergabe zwischen WebNodes.

```
CLASS WebNodeActualParam {  
  formalParam: REF WebNodeFormalParam;  
  expressions: SUB WebNodeExpression!;  
}
```

Subnode of: ActionHandler, Formular.

formalParam

Formaler Parameter, an den der Wert übergeben wird.

expressions

Wert des aktuellen Parameters.

2.54. WebNodeFormalParam

Formaler Parameter für Parameterübergabe zwischen WebNodes.

```
CLASS WebNodeFormalParam INHERITS WebNodeVariableDefinition {  
  name: VAL VLString;  
  type: SUB DBColumnType!;  
}
```

Subnode of: WebNode, SubWebZone.

Referred by: OutputElement, WebNodeActualParam, FormularElement.

name

Name des formalen Parameters.

type

Typ des formalen Parameters.

2.55. WebNodeExpression

Abstrakte Oberklasse für Ausdrücke, die bei der Parameterübergabe übergeben werden können.

```
ABSTRACT CLASS WebNodeExpression {  
}
```

Direct subclasses: WebNodeConstantExpression, WebNodeVariableExpression, WebNodeColumnExpression.

Subnode of: WebNodeActualParam, DataSourceActualParam.

2.56. WebNodeConstantExpression

Konstanter Ausdruck. Übergibt bei der Parameterübergabe einen konstanten Wert.

```
CLASS WebNodeConstantExpression INHERITS WebNodeExpression {  
    value: VAL VLString;  
}
```

value

Konstanter Wert, der übergeben werden soll.

2.57. WebNodeVariableExpression

Variabler Ausdruck. Kann den Wert aus einem Formularfeld oder aus einem formalen Parameter enthalten.

```
CLASS WebNodeVariableExpression INHERITS WebNodeExpression {  
    content: REF WebNodeVariableDefinition;  
}
```

content

Die Referenz auf ein Formular-Element oder einen formalen Parameter.

2.58. WebNodeColumnExpression

Ausdruck, der mit den Werten einer Datenbankspalte verknüpft ist.

```
CLASS WebNodeColumnExpression INHERITS WebNodeExpression {  
    content: REF DBColumn;  
}
```

content

Datenbankspalte, aus welcher die Werte ausgelesen werden.

2.59. Component

Die Klasse Component bezeichnet eine Komponente einer Webseite. Sie ist abstrakt und aus ihr werden z.B. Bilder (Images), Formulare oder Textblöcke abgeleitet.

```
ABSTRACT CLASS Component {  
    styleSheetBlock: REF StyleSheetBlock?;  
}
```

Abstraktes Strukturmodell des Wavis-Editors.

Direct subclasses: Formular, Paragraph, Heading, Table, FrameRef, ListOutputElement.

Subnode of: GlobalFrame, Frame.

styleSheetBlock

Referenz auf ein Stylesheet-Block.

2.60. Paragraph

Dient der Einbindung von Fließtext in die Webseite. Dabei können in dem Fließtext weitere Objekte wie zum Beispiel Links und/oder Bilder eingebunden werden.

```
CLASS Paragraph INHERITS Component {  
    fragments: SUB Fragment*;  
}
```

fragments

Die einzelnen Fragmente aus denen der Paragraph besteht.

2.61. Fragment

Abstrakte Klasse für Fragment.

```
ABSTRACT CLASS Fragment {  
    name: VAL VLString;  
    link: REF LinkTarget?;  
}
```

Direct subclasses: TextFragment, ImageFragment, OutputElement.

Subnode of: Paragraph.

name

No description available.

link

Verweis auf einen Link (Verweis auf eine andere interne oder externe Seite.)

2.62. TextFragment

Das Text-Fragment dient der Einbindung von Fließtext, welcher auch Html-Links enthalten kann.

```
CLASS TextFragment INHERITS Fragment {  
    value: VAL VLString;  
}
```

value

Der Text.

2.63. ImageFragment

Dient der Einbettung von Bildern in Fließtext. Das Bild kann auch ein Link einbinden.

```
CLASS ImageFragment INHERITS Fragment {  
  width: VAL VLInt;  
  height: VAL VLInt;  
  text: VAL VLString;  
  imageName: VAL VLString;  
}
```

width

Höhe des Bildes (optional).

height

Die Breite des Bildes (optional).

text

Der ALT-Text (alternativer Text) des Bildes, falls es nicht angezeigt wird/werden kann.

imageName

Bilder

2.64. Heading

Das Sprachelement dient der Definition von Überschriften.

```
CLASS Heading INHERITS Component {  
  text: VAL VLString;  
  level: VAL VLInt;  
}
```

text

Der Text der Überschrift.

level

Die Ebene der Überschrift (1..6).

2.65. Table

Das Sprachelement Table dient zum grafischen anordnen von Komponenten.

```
CLASS Table INHERITS Component {  
  rows: SUB TableRow*;  
  columns: SUB TableColumn*;  
}
```

rows

Enthält die Tabellenzeilen.

columns

Enthält die Tabellenspalten.

2.66. TableRow

Die Zeile einer Tabelle

```
CLASS TableRow {  
  cells: SUB TableCell*;  
}
```

Subnode of: Table.

cells

Eine Tabellenzeile besteht wiederum aus Zellen.

2.67. TableColumn

Die Spalte einer Tabelle.

```
CLASS TableColumn {  
}
```

Subnode of: Table.

Referred by: TableCell.

2.68. TableCell

Die einzelnen Zellen einer Tabellenzeile.

```
CLASS TableCell {  
  columnRef: REF TableColumn;  
  content: SUB Frame%;  
}
```

Subnode of: TableRow.

columnRef

Ein Verweis auf die zugehörige Tabellenspalte.

content

Der Inhalt der Tabellenspalte.

2.69. LinkTarget

Abstrakter Typ für Links.

```
ABSTRACT CLASS LinkTarget {  
  name: VAL VLString;  
}
```

Direct subclasses: LinkTargetIntern, LinkTargetExtern.

Referred by: Fragment.

name

Der Name des Elements.

2.70. LinkTargetIntern

Abstrakter Typ für interne Links.

```
ABSTRACT CLASS LinkTargetIntern INHERITS LinkTarget {  
}
```

Direct subclasses: SubWebZone, WebNode.

Subnode of: SubWebZone, RootWebZone.

Referred by: SubWebZone, RootWebZone.

2.71. LinkTargetExtern

Abstrakter Typ für externe Links.

```
CLASS LinkTargetExtern INHERITS LinkTarget {  
  url: VAL VLString;  
}
```

Subnode of: WebApplication.

url

Die URL des Sprungziels.

2.72. Root

Das Root-Element ist das Wurzelement des abstrakten Strukturmodells. Es besitzt genau eine Webanwendung als Kindelement.

```
CLASS Root {  
  webApp: SUB WebApplication%;  
  library: SUB Library?;  
}
```

webApp

Die Web-Anwendung.

library

2.73. Library

No description available.

```
CLASS Library {  
  name: VAL VLString;  
  formularconditions: SUB FormularConditionDefinition*;  
  navigationElement: SUB NavigationElement*;  
}
```

Subnode of: Root.

name

No description available.

formularconditions

No description available.

navigationElement

No description available.

2.74. NavigationElement

No description available.

```
CLASS NavigationElement {  
  code: VAL VLString;  
  name: VAL VLString;  
}
```

Subnode of: Library.

Referred by: ListOutputElement.

code

No description available.

name

No description available.

2.75. WebApplication

Das Sprachelement definiert die Navigationsebene und die Ebene zur Definition globaler Eigenschaften der Webanwendung. Es können die einzelnen Webseiten definiert werden, sowie globale Eigenschaften festgelegt werden, wie zum Beispiel Formatvorlagen (StyleSheets) und/oder Datenbank-Parameter.

```
CLASS WebApplication {  
  name: VAL VLString;  
  webZone: SUB RootWebZone%;  
  navigationBar: SUB NavigationBar%;  
  styleSheet: SUB StyleSheet%;  
  frames: SUB GlobalFrame*;  
  meta: SUB MetaInformation%;  
  database: SUB DataBase%;  
  roles: SUB WebRole*;  
  linksExtern: SUB LinkTargetExtern*;  
  imagePath: VAL VLString?;  
}
```

Subnode of: Root.

name

Der Name der Webanwendung.

webZone

Eine Webanwendung besitzt genau einen 'geschützten' Bereich.

navigationBar

Dient der Modellierung einer Navigationsleiste.

styleSheet	Es gibt ein globales StyleSheet bestehend aus beliebigen StyleSheet-Blöcken.
frames	Es kann eine Menge von Frames definiert werden, auf die referenziert werden kann.
meta	Es können Meta-Informationen global definiert werden.
database	Bei der Verwendung einer Datenbank können hier die Parameter festgelegt werden.
roles	Es können verschiedene Rollen definiert werden, die für die Kontrolle des Zugriffs auf WebZone's benutzt werden. WebZone's können mehrere Rollen zugewiesen werden. Benutzer können verschiedene Rollen spielen.
linksExtern	Es kann eine Menge von externen Links definiert werden.
imagePath	Das Wurzelverzeichnis für Bilder der Web-Anwendung.

2.76. RootWebZone

Eine WebZone beschreibt einen 'geschützten' oder 'frei' zugänglichen Bereich der Webanwendung. Auf die in einer 'geschützten' WebZone enthaltenen Webseiten darf nur zugegriffen werden, wenn der Benutzer mindestens eine der 'Rollen' besitzt, welche für die WebZone angegeben sind. Sind keine Rollen für die WebZone angegeben, darf jeder Aufrufer auf den Inhalt der WebZone zugreifen. Ein WebZone kann wiederum Web-Elemente, wie zum Beispiel WebZones, Webseiten und Skripte enthalten.

```
CLASS RootWebZone {  
  webTargets: SUB LinkTargetIntern*;  
  errorLink: REF LinkTargetIntern;  
  roles: SUB WebRoleRef*;  
  setSize: VAL VLPoint?;  
}
```

Subnode of: WebApplication.

webTargets	Die einzelnen Knoten (WebPage/WebAction) der Webanwendung, die innerhalb der WebZone liegen.
errorLink	Wenn der Aufrufer kein Zugriffsrecht für die WebZone besitzt, wird zu diesem Verweis gesprungen.
roles	Die Liste mit den Rollen, welche auf die WebZone zugreifen darf.

setSize

No description available.

2.77. SubWebZone

Eine WebZone beschreibt einen 'geschützten' oder 'frei' zugänglichen Bereich der Webanwendung. Auf die in einer 'geschützten' WebZone enthaltenen Webseiten darf nur zugegriffen werden, wenn der Benutzer mindestens eine der 'Rollen' besitzt, welche für die WebZone angegeben sind. Sind keine Rollen für die WebZone angegeben, darf jeder Aufrufer auf den Inhalt der WebZone zugreifen. Ein WebZone kann wiederum Web-Elemente, wie zum Beispiel WebZones, Webseiten und Skripte enthalten.

```
CLASS SubWebZone INHERITS LinkTargetIntern {  
  webTargets: SUB LinkTargetIntern*;  
  errorLink: REF LinkTargetIntern;  
  globalparams: SUB WebNodeFormalParam*;  
  roles: SUB WebRoleRef*;  
  setSize: VAL VLPoint?;  
  position: VAL VLPoint?;  
}
```

webTargets

Die einzelnen Knoten (WebPage/WebAction) der Webanwendung, die innerhalb der WebZone liegen.

errorLink

Wenn der Aufrufer kein Zugriffsrecht für die WebZone besitzt, wird zu diesem Verweis gesprungen.

globalparams

No description available.

roles

Die Liste mit den Rollen, welche auf die WebZone zugreifen darf.

setSize

No description available.

position

No description available.

2.78. WebRole

Dient der Definition verschiedener Rollen. Damit können Zugriffsrechte auf WebZone's modelliert werden.

```
CLASS WebRole {  
  name: VAL VLString;  
  color: VAL VLString;  
}
```

Subnode of: WebApplication.

Referred by: WebRoleRef.

name

Der Name der Rolle.

color

No description available.

2.79. WebRoleRef

Verweist auf ein WebRole-Objekt. Wird benutzt, um WebZone's mehrere Rollen zuweisen zu können.

```
CLASS WebRoleRef {  
    role: REF WebRole;  
}
```

Subnode of: SubWebZone, RootWebZone.

role

Das referenzierte WebRole-Objekt.

2.80. WebNode

Abstrakte Basis-Klasse für Verhaltens- und Darstellungsknoten.

```
ABSTRACT CLASS WebNode INHERITS LinkTargetIntern {  
    formalParameter: SUB WebNodeFormalParam*;  
}
```

Direct subclasses: WebPage, WebAction.

Referred by: NavigationBarElement, Formular.

formalParameter

Formalparameter für WebPage

2.81. WebPage

Das Sprachelement WebPage definiert in der Webanwendung einen Knoten zur Visualisierung in Form einer einzelnen Webseite.

```
CLASS WebPage INHERITS WebNode {  
    title: VAL VLString;  
    frame: SUB Frame%;  
    styleSheetBlock: REF StyleSheetBlock?;  
    position: VAL VLPoint?;  
}
```

Referred by: ActionHandler.

title

Der Titel der Webseite.

frame

Eine Webseite enthält genau einen Rahmen.

styleSheetBlock

Eine Referenz auf ein StyleSheet-Block. Dieses wird bei der Darstellung der Webseite verwendet.

position

No description available.

2.82. Frame

Der Frame bzw. Rahmen dient der inhaltlichen und sichtbaren Separierung von Bereichen einer Webseite. Mit ihm ist z.B. die Trennung von Navigation, Infobox und Inhalt möglich. Dieser ist nicht zu verwechseln mit HTML-Frames.

```
CLASS Frame {  
  components: SUB Component*;  
  styleSheetBlock: REF StyleSheetBlock?;  
}
```

Subnode of: WebPage, TableCell.

components

Die Inhalte (genannt Komponenten) des Frames.

styleSheetBlock

No description available.

2.83. GlobalFrame

Ein Goba- Frame bzw. Rahmen dient der inhaltlichen und sichtbaren Separierung von Bereichen einer Webseite. Mit ihm ist z.B. die Trennung von Navigation, Infobox und Inhalt möglich. Dieser ist nicht zu verwechseln mit HTML-Frames. Im Gegensatz zu den 'normalen' Rahmen kann ein GlobalFrame mittels des Sprachelementes FrameRef wiederverwendet werden.

```
CLASS GlobalFrame {  
  name: VAL VLString;  
  components: SUB Component*;  
  setSize: VAL VLPoint?;  
}
```

Subnode of: WebApplication.

Referred by: FrameRef.

name

Der Name des Frames.

components

Die Inhalte (genannt Komponenten) des Frames.

setSize

No description available.

2.84. NavigationBar

Hier können Eigenschaften der Navigationsleiste festgelegt werden.

```
CLASS NavigationBar {  
  styleSheetBlock: REF StyleSheetBlock?;  
  elements: SUB NavigationBarElement*;  
  alignment: VAL VLString;  
}
```

Subnode of: WebApplication.

styleSheetBlock

Referenz auf ein Stylesheet-Block.

elements

Die Elemente einer NavigationBar (also die Links auf andere Webseiten)

alignment

Die Anordnung der NavigationBar (also top, bottom, left, right) anhand der auch die Ausrichtung (horizontal/vertikal) berechnet wird

2.85. NavigationBarElement

Definiert ein Element einer NavigationBar, also einen Verweis auf eine Webseite bzw. einen WebNode

```
CLASS NavigationBarElement {  
  target: REF WebNode;  
}
```

Subnode of: NavigationBar.

target

Die Referenz auf eine Webseite

2.86. FrameRef

Dient der Referenzierung von Frames.

```
CLASS FrameRef INHERITS Component {  
  frame: REF GlobalFrame;  
}
```

frame

Das referenzierte Frame.

2.87. MetaInformation

Mit dem Sprachelement kann eine Gruppe von Meta-Informationen für Webseiten definiert werden.

```
CLASS MetaInformation {  
  name: VAL VLString;  
  elements: SUB MetaInformationElement*;  
}
```

Subnode of: WebApplication.

name

Der Name der Meta-Information.

elements

Eine Liste von Elementen, welche die Meta-Informationen beinhalten.

2.88. MetaInformationElement

Definiert eine Meta-Information (property=value)

```
CLASS MetaInformationElement {  
    property: VAL VLString;  
    value: VAL VLString;  
}
```

Subnode of: MetaInformation.

property

Der Name der Eigenschaft.

value

Der Wert der Eigenschaft.

2.89. OutputElement

Gibt den Wert eines formalen Parameters zum Beispiel einer Parameterausgabe aus

```
CLASS OutputElement INHERITS Fragment {  
    content: REF WebNodeFormalParam;  
}
```

content

No description available.

2.90. ListOutputElement

Listet Tupel einer Datenbank Abfrage auf.

```
CLASS ListOutputElement INHERITS Component {  
    dataSourceActualParams: SUB DataSourceActualParam*;  
    dataSource: REF DataSource;  
    maxNumberOfElements: VAL VInt;  
    navigationElement: REF NavigationElement;  
}
```

dataSourceActualParams

Aktuelle Parameter, die an den Filter uebergeben werden

dataSource

Der Filter, der als Quelle fuer das ListOutputElement dient

maxNumberOfElements

Falls eine LimitDataSource angegeben wird, koennen ihr die maximalen Zeilen uebergeben werden.

navigationElement

Falls eine LimitDataSource angegeben wird, so kann aus der Bibliothek aus vorgefertigten Navigationselementen gewaehlt werden.

2.91. StyleSheet

Die Klasse StyleSheet legt eine Gruppe von Stylesheetelementen fest, welches später z.B. einer Webseite, einer Komponente oder sonstigen Elementen zugewiesen werden können.

```
CLASS StyleSheet {  
  name: VAL VLString;  
  custom: SUB CustomStyleSheetBlock*;  
  default: SUB HtmlStyleSheetBlock*;  
}
```

Subnode of: WebApplication.

name

Der Name für dieses StyleSheet Dokument.

custom

Liste enthält die benutzerdefinierten StyleSheet-Blöcke.

default

Liste enthält die vordefinierten StyleSheet-Blöcke.

2.92. StyleSheetBlock

Ein CustomStyleSheetBlock ist äquivalent zu einem CSS-Anweisungsblock. Damit kann der Benutzer Stylesheets für eigene Elemente definieren. Damit ein Custom-StyleSheetBlock verwendet wird, muss es explizit von dem Element referenziert werden.

```
ABSTRACT CLASS StyleSheetBlock {  
  pairs: SUB StyleSheetPair*;  
}
```

Direct subclasses: CustomStyleSheetBlock, HtmlStyleSheetBlock.

Referred by: NavigationBar, Frame, WebPage, Component.

pairs

Wertepaare in einem CSS-Block.

2.93. CustomStyleSheetBlock

No description available.

```
CLASS CustomStyleSheetBlock INHERITS StyleSheetBlock {  
  name: VAL VLString;  
}
```

Subnode of: StyleSheet.

name

Der Name des Blocks (selector), auf den referenziert werden kann.

2.94. StyleSheetPair

Ein StyleSheetPair ist ein Wertepaar aus einem Property und einem Value.

```
CLASS StyleSheetPair {  
  category: VAL VLString;  
  attribute: VAL VLString;  
  valueCategory: VAL VLString;  
  value: SUB AttrValue!;  
}
```

Subnode of: StyleSheetBlock.

category

Die auszuwählende Kategorie (z.B. font).

attribute

Die Attribute der ausgewählten Kategorie (z.B. font-size, font-style, font-family).

valueCategory

No description available.

value

No description available.

2.95. AttrValue

Abstrakte Klasse für die verschiedenen Typen von Stylesheet-Werten.

```
ABSTRACT CLASS AttrValue {  
}
```

Direct subclasses: SizeValue, ColorValue, FamilyValue, StyleValue, HeadingValue, AlignmentValue, ListValue, CustomValue.

Subnode of: StyleSheetPair.

2.96. SizeValue

Typ für Größenangaben.

```
CLASS SizeValue INHERITS AttrValue {  
  measure: VAL VLString;  
  value: VAL VLString;  
}
```

measure

Die verwendete Maßeinheit.

value

Der Wert.

2.97. ColorValue

Typ für Farbwerte.

```
CLASS ColorValue INHERITS AttrValue {  
    color: VAL VLString;  
}
```

color

Der Farbwert.

2.98. FamilyValue

Typ für Zeichensätze (Font). Mögliche Werte: Times New Roman, Helvetica, Arial, Courier.

```
CLASS FamilyValue INHERITS AttrValue {  
    family: VAL VLString;  
}
```

family

Der Name des Zeichensatzes.

2.99. StyleValue

Typ für Stile von Zeichensätze. Mögliche Werte: normal, bold, italic.

```
CLASS StyleValue INHERITS AttrValue {  
    style: VAL VLString;  
}
```

style

Der Name des Zeichensatz-Stils.

2.100. HeadingValue

Typ für die verschiedenen Heading-Klassen. Mögliche Werte: h1, h2, h3, h4, h5, h6.

```
CLASS HeadingValue INHERITS AttrValue {  
    heading: VAL VLString;  
}
```

heading

Die verwendete Heading-Klasse.

2.101. AlignmentValue

Typ für die verschiedenen Ausrichtungen. Mögliche Werte: left, center, right.

```
CLASS AlignmentValue INHERITS AttrValue {  
    alignment: VAL VLString;  
}
```

alignment

Die verwendete Ausrichtung.

2.102. ListValue

Sonstiger Typ.

```
CLASS ListValue INHERITS AttrValue {  
    listed: VAL VLString;  
}
```

listed

2.103. CustomValue

Typ für benutzerspezifische Werte.

```
CLASS CustomValue INHERITS AttrValue {  
    value: VAL VLString;  
}
```

value

Der Wert.

2.104. HtmlStyleSheetBlock

Ein HtmlStyleSheetBlock ist äquivalent zu einem CSS-Anweisungsblock, welcher auf ein spezielles Html-Element Auswirkung hat. Damit können für bestimmte Html-Elemente vordefinierte Stylesheets mit voreingestellten Attributen erzeugt werden. Die Funktionalität wird für folgende Html-Elemente angeboten: Body, H1-H6, Paragraph, Table, TableHeader, TableRow, TableData und Div.

```
CLASS HtmlStyleSheetBlock INHERITS StyleSheetBlock {  
    name: VAL VLString;  
}
```

Subnode of: StyleSheet.

name

Dies ist der Name des Html-Elementes für welches das CSS definiert wird.

2.105. Operation

Abstrakte Klasse für Operationen, die innerhalb einer WebAction ausgeführt werden können.

```
ABSTRACT CLASS Operation {
    errorhandler: SUB ActionHandler?;
}
```

Direct subclasses: PHPOperation, DataBaseOperation.

Subnode of: WebAction.

errorhandler

Handler, der im Fehlerfall ausgeführt wird.

2.106. ActionHandler

Aktion, die nach dem erfolgreichen Ablauf einer WebAction oder nach einem Fehler in einer Operation ausgeführt wird.

```
CLASS ActionHandler {
    targetpage: REF WebPage;
    params: SUB WebNodeActualParam*;
}
```

Subnode of: WebAction, Operation.

targetpage

Zielseite, auf die nach erfolgreichem Ablauf der WebAction oder nach einem Fehler in einer Operation weitergeleitet wird.

params

Parameter, die bei der Weiterleitung übergeben werden können.

2.107. WebAction

Folge von Operationen, die nacheinander abgearbeitet werden.

```
CLASS WebAction INHERITS WebNode {
    successpage: SUB ActionHandler%;
    operations: SUB Operation*;
    position: VAL VLPPoint?;
}
```

successpage

Zielseite, auf die nach erfolgreicher Abarbeitung der WebAction weitergeleitet wird.

operations

Operationen, die nacheinander abgearbeitet werden.

position

No description available.

2.108. PHPOperation

Benutzerdefiniertes PHP-Skript, welches innerhalb einer WebAction abgearbeitet wird.

```
CLASS PHPOperation INHERITS Operation {  
    content: VAL VLString;  
}
```

content

Benutzerdefiniertes PHP-Skript.

2.109. DataBaseOperation

Datenbank-Operation, die innerhalb einer WebAction abgearbeitet wird.

```
CLASS DataBaseOperation INHERITS Operation {  
    filter: REF DataSourceModifyOperation;  
    params: SUB DataSourceActualParam*;  
}
```

filter

Referenz auf eine Datenbank-Operation.

params

Parameter, die an die Datenbank-Operation Ã¼bergeben werden kÃ¶nnen.