



bankmark UG (haftungsbeschränkt)
Bahnhofstrasse 10 | 94032 Passau | Deutschland
www.bankmark.de | info@bankmark.de
T +49 851 205 490 490 | F +49 851 205 490 499

PDGF

-Plugin Documentation-

for 2.5-#1397_b4631 June 12, 2015

All information in this document and its appendices are property of bankmark.
bankmark requires that these documents will be kept confidential and will not be disclosed to third parties.

Contents

1	PDGF Plugins	3
1.1	pdgf.core.dataGenerator.UpdateBlackBox	3
1.2	pdgf.core.dataGenerator.scheduler.DbDumpUpdateScheduler	3
1.3	pdgf.core.dataGenerator.scheduler.DefaultPartitioner	4
1.4	pdgf.core.dataGenerator.scheduler.DefaultScheduler	5
1.5	pdgf.core.dataGenerator.scheduler.TemplatePartitioner	6
1.6	pdgf.core.dataGenerator.scheduler.UpdateScheduler	7
1.7	pdgf.core.dbSchema.Field	8
1.8	pdgf.core.dbSchema.Project	12
1.9	pdgf.core.dbSchema.ReferenceIMPL	15
1.10	pdgf.core.dbSchema.Table	17
1.11	pdgf.distribution.Beta	19
1.12	pdgf.distribution.Binomial	19
1.13	pdgf.distribution.Exponential	20
1.14	pdgf.distribution.Logarithmic	21
1.15	pdgf.distribution.Normal	22
1.16	pdgf.distribution.Zeta	22
1.17	pdgf.generator.ActionDeletesAlsoUpdateGenerator	23
1.18	pdgf.generator.AddRandomSecondsGenerator	24
1.19	pdgf.generator.BigBenchReviewGenerator	27
1.20	pdgf.generator.BuildListGenerator	28
1.21	pdgf.generator.CDCSequenceGenerator	30
1.22	pdgf.generator.ComputeGenerator	31
1.23	pdgf.generator.ConvertNumberToStringGenerator	33
1.24	pdgf.generator.DateTimeGenerator	34
1.25	pdgf.generator.DefaultReferenceGenerator	36
1.26	pdgf.generator.DetailFromDateGenerator	38
1.27	pdgf.generator.DictList	40
1.28	pdgf.generator.DoubleGenerator	42
1.29	pdgf.generator.EmailGenerator	43
1.30	pdgf.generator.ExtendedFormulaGenerator	44
1.31	pdgf.generator.ForFormulaGenerator	46
1.32	pdgf.generator.ForGenerator	48
1.33	pdgf.generator.FormatNumberAsDateGenerator	50
1.34	pdgf.generator.FormattedNumberGenerator	51
1.35	pdgf.generator.FormulaGenerator	53
1.36	pdgf.generator.GenderGenerator	55
1.37	pdgf.generator.GlobalRowGenerator	55
1.38	pdgf.generator.HashGenerator	56
1.39	pdgf.generator.IFGenerator	58
1.40	pdgf.generator.IdGenerator	60
1.41	pdgf.generator.LastChoiceGenerator	61
1.42	pdgf.generator.LongGenerator	61
1.43	pdgf.generator.MarkovChainGenerator	62
1.44	pdgf.generator.NameGenerator	63
1.45	pdgf.generator.NullGenerator	64
1.46	pdgf.generator.OtherFieldValueGenerator	65
1.47	pdgf.generator.PaddingGenerator	66
1.48	pdgf.generator.PermutationReferenceGenerator	68
1.49	pdgf.generator.PrePostfixGenerator	70
1.50	pdgf.generator.ProbabilityGenerator	71
1.51	pdgf.generator.RandomAString	72
1.52	pdgf.generator.RandomListItemsGenerator	73
1.53	pdgf.generator.RandomSentence	75
1.54	pdgf.generator.RandomUniqueStringGenerator	76
1.55	pdgf.generator.RandomValueXY	77
1.56	pdgf.generator.ReferenceGenerator	78
1.57	pdgf.generator.RelativeGenerator	81

1.58	pdgf.generator.SequentialGenerator	82
1.59	pdgf.generator.SetQuery	84
1.60	pdgf.generator.SimpleFormatNumberGenerator	85
1.61	pdgf.generator.StaticValueGenerator	86
1.62	pdgf.generator.SwitchGenerator	87
1.63	pdgf.generator.TemplateGenerator	88
1.64	pdgf.generator.UUIDgenerator	90
1.65	pdgf.generator.UpdateActionTypeGenerator	91
1.66	pdgf.generator.UpperLowerCaseGenerator	92
1.67	pdgf.generator.WeightedListGenerator	93
1.68	pdgf.generator.WeightedSubListGenerator	96
1.69	pdgf.output.CSVRowOutput	99
1.70	pdgf.output.CompiledTemplateOutput	102
1.71	pdgf.output.DummyOutput	103
1.72	pdgf.output.HierarchicalXMLRowOutput	104
1.73	pdgf.output.MultiLineOutput	106
1.74	pdgf.output.MultiLineOutputWrapper	109
1.75	pdgf.output.SortedFileOutputWrapper	110
1.76	pdgf.output.SortedOutputWrapper	111
1.77	pdgf.output.SplitFileOutputWrapper	112
1.78	pdgf.output.SplitOutputWrapper	113
1.79	pdgf.output.StatisticsOutput	113
1.80	pdgf.output.XMLTemplateOutput	115

1 PDGF Plugins

1.1 pdgf.core.dataGenerator.UpdateBlackBox

main part of the update system

1.1.1 Attributes

Name	Req.	Min/- Max count	Description	Values
name	x	1/1	DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.1.2 Nodes

Name	Req.	Min/- Max count	Description	Values
------	------	-----------------------	-------------	--------

1.1.3 XML-Example

```
<!--Description: main part of the update system -->
<updateBlackBox_pdgf.core.dataGenerator.UpdateBlackBox
  name="Desc: DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name
    is required. Example: com.en.myPluginPackage.myPuginClass"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
</updateBlackBox_pdgf.core.dataGenerator.UpdateBlackBox>
```

Listing 1: XML example with all options

1.2 pdgf.core.dataGenerator.scheduler.DbDumpUpdateScheduler

Update capable: true

Requires an 'Update' table. Simulates a complete database dump (snapshot) in each update generation. Default: deleted tuples/ID's are excluded from the dump.

1.2.1 Attributes

Name	Req.	Min/- Max count	Description	Values
name	x	1/1	DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.2.2 Nodes

Name	Req.	Min/- Max count	Description	Values
partitioner		0/1	Partitioner plugins specify how the workload is distributed between independent compute nodes. Default partitioner plugin if no one is explicitly specified: <code>pdgf.core.dataGenerator.scheduler.DefaultPartitioner</code>	
excldeDeletedIDs		0/1	If value=True, deleted ID's will be excluded from the generated dump.	true false 0 1

1.2.3 XML-Example

```

<!--Description: Update capable: true
Requires an 'Update' table. Simulates a complete database dump (snapshot) in each update generation.
Default: deleted tuples/ID's are excluded from the dump. -->
<scheduler_pdgf.core.dataGenerator.scheduler.DbDumpUpdateScheduler
  name="Desc: DefaultParser. (ClassName) of this element. Used to identify plugin Class. Full name
    is required. Example: com.en.myPluginPackage.myPuginClass"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: If value=True, deleted ID's will be excluded from the generated dump.-->
  <excludeDeletedIDs>Allowed values: {
    true
    false
    0
    1
  }
</excludeDeletedIDs>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Partitioner plugins specify how the workload is distributed between independent
    compute nodes. Default partitioner plugin if no one is explicitly specified:
    pdgf.core.dataGenerator.scheduler.DefaultPartitioner -->
  <partitioner></partitioner>
</scheduler_pdgf.core.dataGenerator.scheduler.DbDumpUpdateScheduler>

```

Listing 2: XML example with all options

1.3 pdgf.core.dataGenerator.scheduler.DefaultPartitioner

Default partitioner

1.3.1 Attributes

Name	Req.	Min/- Max count	Description	Values
name	x	1/1	DefaultParser. (ClassName) of this element. Used to identify plugin Class. Full name is required. Example: <code>com.en.myPluginPackage.myPuginClass</code>	
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	
staticTableOn-AllNodes		0/1	Generate static tables on all nodes. Static tables are not split by scheduler. If true, each node replicates the same table. If false, only the first node will generate the table.	true false 0 1

1.3.2 Nodes

Name	Req.	Min/- Max count	Description	Values
------	------	-----------------------	-------------	--------

1.3.3 XML-Example

```
<!--Description: Default partitioner -->
<partitioner_pdgf.core.dataGenerator.scheduler.DefaultPartitioner
  name="Desc: DefaultParser. (ClassName) of this element. Used to identify plugin Class. Full name
    is required. Example: com.en.myPluginPackage.myPuginClass"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element."
  staticTableOnAllNodes="(optional) Desc: Generate static tables on all nodes. Static tables are
    not split by scheduler. If true, each node replicates the same table. If false, only the
    first node will generate the table. Allowed values: {
      true
      false
      0
      1
    }">
</partitioner_pdgf.core.dataGenerator.scheduler.DefaultPartitioner>
```

Listing 3: XML example with all options

1.4 pdgf.core.dataGenerator.scheduler.DefaultScheduler

Update capable: false

Scheduler for 'historical' (non update) tables. Assigns work in WorkUnits (default WorkUnits size in rows: 100) in a round robin fashion to workers and assigns sorting ID's to each junk.

1.4.1 Attributes

Name	Req.	Min/- Max count	Description	Values
name	x	1/1	DefaultParser. (ClassName) of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.4.2 Nodes

Name	Req.	Min/- Max count	Description	Values
partitioner		0/1	Partitioner plugins specify how the workload is distributed between independent compute nodes. Default partitioner plugin if no one is explicitly specified: pdgf.core.dataGenerator.scheduler.DefaultPartitioner	

1.4.3 XML-Example

```
<!--Description: Update capable: false
Scheduler for 'historical' (non update) tables. Assigns work in WorkUnits (default WorkUnits size in
rows: 100) in a round robin fashion to workers and assigns sorting ID's to each junk. -->
<scheduler_pdgf.core.dataGenerator.scheduler.DefaultScheduler
  name="Desc: DefaultParser. (ClassName) of this element. Used to identify plugin Class. Full name
  is required. Example: com.en.myPluginPackage.myPuginClass"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element.">
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Partitioner plugins specify how the workload is distributed between independent
  compute nodes. Default partitioner plugin if no one is explicitly specified:
  pdgf.core.dataGenerator.scheduler.DefaultPartitioner -->
  <partitioner</partitioner>
</scheduler_pdgf.core.dataGenerator.scheduler.DefaultScheduler>
```

Listing 4: XML example with all options

1.5 pdgf.core.dataGenerator.scheduler.TemplatePartitioner

Code template based partitioner

1.5.1 Attributes

Name	Req.	Min/- Max count	Description	Values
name	x	1/1	DefaultParser. (ClassName) of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	
staticTableOn- AllNodes		0/1	Generate static tables on all nodes. Static tables are not split by scheduler. If true, each node replicates the same table. If false, only the first node will generate the table.	true false 0 1

1.5.2 Nodes

Name	Req.	Min/- Max count	Description	Values
prePartition		0/1	foo	
nodePartition		0/1	foo	

1.5.3 XML-Example

```
<!--Description: Code template based partitioner -->
<partitioner_pdgf.core.dataGenerator.scheduler.TemplatePartitioner
  name="Desc: DefaultParser. (ClassName) of this element. Used to identify plugin Class. Full name
  is required. Example: com.en.myPluginPackage.myPuginClass"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element.">
```

```

staticTableOnAllNodes="(optional) Desc: Generate static tables on all nodes. Static tables are
not split by scheduler. If true, each node replicates the same table. If false, only the
first node will generate the table. Allowed values: {
    true
    false
    0
    1
}">
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: foo-->
<nodePartition></nodePartition>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: foo-->
<prePartition></prePartition>
</partitioner_pdgf.core.dataGenerator.scheduler.TemplatePartitioner>

```

Listing 5: XML example with all options

1.6 pdgf.core.dataGenerator.scheduler.UpdateScheduler

Update capable: true

Requires an 'Update' table. Simulates 'log file' or 'change data capture' behavior. For every 'UpdateID' this scheduler outputs the the changes made to the database: NEW tuples, DELETED tuples and CHANGED tuples.

1.6.1 Attributes

Name	Req.	Min/- Max count	Description	Values
name	x	1/1	DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.6.2 Nodes

Name	Req.	Min/- Max count	Description	Values
partitioner		0/1	Partitioner plugins specify how the workload is distributed between independent compute nodes. Default partitioner plugin if no one is explicitly specified: pdgf.core.dataGenerator.scheduler.DefaultPartitioner	

1.6.3 XML-Example

```

<!--Description: Update capable: true
Requires an 'Update' table. Simulates 'log file' or 'change data capture' behavior. For every 'UpdateID'
this scheduler outputs the the changes made to the database: NEW tuples, DELETED tuples and CHANGED
tuples. -->
<scheduler_pdgf.core.dataGenerator.scheduler.UpdateScheduler
    name="Desc: DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name
    is required. Example: com.en.myPluginPackage.myPuginClass"
    id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
<!--Required: [ ] Executions(min|max): (0|1) -->

```



```

<!--Description: Partitioner plugins specify how the workload is distributed between independent
compute nodes. Default partitioner plugin if no one is explicitly specified:
pdgf.core.dataGenerator.scheduler.DefaultPartitioner-->
<partitioner></partitioner>
</scheduler_pdgf.core.dataGenerator.scheduler.UpdateScheduler>

```

Listing 6: XML example with all options

1.7 pdgf.core.dbSchema.Field

Represents a Database Tables Field or Entry

1.7.1 Attributes

Name	Req.	Min/- Max count	Description	Values
size		0/1	Field size. Most important for types like 'VARCHAR'. Some default generators (like RandomAString) may use this value to automatically determine their output size. Output plugins, like CSVRowOutput, may use this value to store rows in flat files as fixed length formatted lines. They may do padding or truncate generated values based on this size parameter.	
unique		0/1	Specifies if this field should be unique {true false}. (must be supported by the used Generator), maybe used by output plugins	true false 0 1
name	x	1/1	DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	
updatePer-centage		0/1	percentage that this field will be changed during an update	
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

type	x	1/1	Specifies type of this field. Type must be one of java.sql.Types.* f.e.: java.sql.Types.INTEGER. Maybe used by output plugins	VARCHAR BIT TINYINT SMALLINT INTEGER BIGINT FLOAT REAL DOUBLE NUMERIC DECIMAL CHAR LONGVARCHAR DATE TIME TIMESTAMP BINARY VARBINARY LONGVARBINARY NULL OTHER JAVA_OBJECT DISTINCT STRUCT ARRAY BLOB CLOB REF DATALINK BOOLEAN ROWID NCHAR NVARCHAR LONGNVARCHAR NCLOB SQLXML REF_CURSOR TIME_WITH_TIMEZONE TIMESTAMP_WITH_TIMEZONE
primary		0/1	Specifies if this field should be primary {true false}. (must be supported by the used Generator), maybe used by output plugins	true false 0 1

1.7.2 Nodes

Name	Req.	Min/- Max count	Description	Values
gen	x	1/1	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgf.generator.IdGenerator
seed		0/1	DefaultParser. Random number generator seed of this Element. Overrides default seeding behavior.	

rng		0/1	DefaultParser. Name of random number generator class to be used for calculations in this element. Example: com.en.myRNG	pdgf.util.random.PdgfDefaultRandom
-----	--	-----	---	------------------------------------

1.7.3 XML-Example

```

<!--Description: Represents a Database Tables Field or Entry -->
<field_pdgf.core.dbSchema.Field
  name="Desc: DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name
    is required. Example: com.en.myPluginPackage.myPuginClass"
  type="Desc: Specifies type of this field. Type must be one of java.sql.Types.* f.e.:
    java.sql.Types.INTEGER. Maybe used by output plugins Allowed values: {
      VARCHAR
      BIT
      TINYINT
      SMALLINT
      INTEGER
      BIGINT
      FLOAT
      REAL
      DOUBLE
      NUMERIC
      DECIMAL
      CHAR
      LONGVARCHAR
      DATE
      TIME
      TIMESTAMP
      BINARY
      VARBINARY
      LONGVARBINARY
      NULL
      OTHER
      JAVA_OBJECT
      DISTINCT
      STRUCT
      ARRAY
      BLOB
      CLOB
      REF
      DATALINK
      BOOLEAN
      ROWID
      NCHAR
      NVARCHAR
      LONGNVARCHAR
      NCLOB
      SQLXML
      REF_CURSOR
      TIME_WITH_TIMEZONE
      TIMESTAMP_WITH_TIMEZONE
    }"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element."
  primary="(optional) Desc: Specifies if this field should be primary {true | false}. (must be
    supported by the used Generator), maybe used by output plugins Allowed values: {
    true
    false
    0
    1
    }"
  size="(optional) Desc: Field size. Most important for types like 'VARCHAR'. Some default
    generators (like RandomAString) may use this value to automatically determine their output
    size. Output plugins, like CSVRowOutput, may use this value to store rows in flat files as
    fixed length formatted lines. They may do padding or truncate generated values based on this
    size parameter. "
  unique="(optional) Desc: Specifies if this field should be unique {true | false}. (must be
    supported by the used Generator), maybe used by output plugins Allowed values: {
    true
    false

```

```

        0
        1
    }”
    updatePercentage=”(optional) Desc: percentage that this field will be changed during an update”>
<!--Required:  [x] Executions(min|max): (1|1) -->
<!--Description: Value Generator for this field-->
<gen_(pluginName)>Allowed values: {
    pdgf.generator.HashGenerator
    pdgf.generator.ComputeGenerator
    pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
    pdgf.generator.FormattedNumberGenerator
    pdgf.generator.ForGenerator
    pdgf.generator.MarkovChainGenerator
    pdgf.generator.CDCSequenceGenerator
    pdgf.generator.RandomUniqueStringGenerator
    pdgf.generator.FormatNumberAsDateGenerator
    pdgf.generator.SimpleFormatNumberGenerator
    pdgf.generator.BuildListGenerator
    pdgf.generator.DictList
    pdgf.generator.DoubleGenerator
    pdgf.generator.LastChoiceGenerator
    pdgf.generator.IFGenerator
    pdgf.generator.WeightedSubListGenerator
    pdgf.generator.DetailFromDateGenerator
    pdgf.generator.DateTimeGenerator
    pdgf.generator.UpdateActionTypeGenerator
    pdgf.generator.IdGenerator
    pdgf.generator.OtherFieldValueGenerator
    pdgf.generator.TemplateGenerator
    pdgf.generator.ForFormulaGenerator
    pdgf.generator.WeightedListGenerator
    pdgf.generator.AddRandomSecondsGenerator
    pdgf.generator.FormulaGenerator
    pdgf.generator.PaddingGenerator
    pdgf.generator.BigBenchReviewGenerator
    pdgf.generator.PrePostfixGenerator
    pdgf.generator.DefaultReferenceGenerator
    pdgf.generator.RandomAString
    pdgf.generator.SwitchGenerator
    pdgf.generator.ConvertNumberToStringGenerator
    pdgf.generator.ExtendedFormulaGenerator
    pdgf.generator.SetQuery
    pdgf.generator.StaticValueGenerator
    pdgf.generator.UUIDgenerator
    pdgf.generator.GlobalRowGenerator
    pdgf.generator.LongGenerator
    pdgf.generator.RelativeGenerator
    pdgf.generator.EmailGenerator
    pdgf.generator.SequentialGenerator
    pdgf.generator.GenderGenerator
    pdgf.generator.RandomListItemsGenerator
    pdgf.generator.PermutationReferenceGenerator
    pdgf.generator.RandomValueXY
    pdgf.generator.NameGenerator
    pdgf.generator.NullGenerator
    pdgf.generator.ReferenceGenerator
    pdgf.generator.UpperLowerCaseGenerator
    pdgf.generator.AbstractDecimalGenerator
    pdgf.generator.ProbabilityGenerator
    pdgf.generator.RandomSentence
}
</gen_(pluginName)>
<!--Required:  [ ] Executions(min|max): (0|1) -->
<!--Description: DefaultParser. Name of random number generator class to be used for calculations
in this element. Example: com.en.myRNG-->
<rng>Allowed values: {
    pdgf.util.random.PdgfDefaultRandom
}
</rng>
<!--Required:  [ ] Executions(min|max): (0|1) -->
<!--Description: DefaultParser. Random number generator seed of this Element. Overrides default
seeding behavior.-->
<seed></seed>
</field_pdgf.core.dbSchema.Field>

```

1.8 pdgf.core.dbSchema.Project

This is the Project

1.8.1 Attributes

Name	Req.	Min/- Max count	Description	Values
name	x	1/1	DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	
targetVersion		0/1	Specifies program version a config file was created for.(optional)	

1.8.2 Nodes

Name	Req.	Min/- Max count	Description	Values
generation		0/1		
seed	x	1/1	DefaultParser. Random number generator seed of this Element. Overrides default seeding behavior.	
property	x	1/Int.max	Properties (variables) in config file(s)	double long longUp longDown datetime
rng	x	1/1	DefaultParser. Name of random number generator class to be used for calculations in this element. Example: com.en.myRNG	pdgf.util.random.PdgfDefaultRandom

loadPlugin		0/Int.max	Invokes the default constructor of this class	pdgf.distribution.Normal pdgf.plugin.AbstractPDGFRandom pdgf.generator.HashGenerator pdgf.output.DummyOutput pdgf.core.dbSchema.Field pdgf.core.dataGenerator.scheduler.Def pdgf.core.dataGenerator.scheduler.Upd pdgf.generator.ComputeGenerator pdgf.generator.ActionDeleteIsAlsoUpd pdgf.generator.FormattedNumberGene pdgf.distribution.Binomial pdgf.output.HierarchicalXMLRowOut pdgf.generator.ForGenerator pdgf.generator.MarkovChainGenerator pdgf.generator.CDCSequenceGenerato pdgf.output.AbstractMultiLineOutput pdgf.generator.RandomUniqueStringGe pdgf.generator.FormatNumberAsDate pdgf.generator.SimpleFormatNumberC pdgf.generator.BuildListGenerator pdgf.generator.DictList pdgf.generator.DoubleGenerator pdgf.util.random.PdgfDefaultRandom pdgf.generator.LastChoiceGenerator pdgf.core.dbSchema.ReferenceIMPL pdgf.distribution.Zeta pdgf.output.SortedOutputWrapper pdgf.generator.IFGenerator pdgf.core.dataGenerator.UpdateBlack pdgf.generator.WeightedSubListGener pdgf.output.MultiLineOutput pdgf.plugin.Generator pdgf.output.CSVRowOutput pdgf.generator.DetailFromDateGenera pdgf.generator.DateTimeGenerator pdgf.plugin.Scheduler pdgf.core.dbSchema.Project pdgf.generator.UpdateActionTypeGen pdgf.distribution.Beta pdgf.generator.IdGenerator pdgf.output.FileOutputSkeleton pdgf.generator.OtherFieldValueGenera pdgf.generator.TemplateGenerator pdgf.generator.ForFormulaGenerator pdgf.generator.WeightedListGenerator pdgf.generator.AddRandomSecondsGe pdgf.generator.FormulaGenerator pdgf.generator.PaddingGenerator pdgf.generator.BigBenchReviewGener pdgf.generator.PrePostfixGenerator pdgf.core.dbSchema.Element pdgf.generator.DefaultReferenceGener pdgf.generator.RandomAString pdgf.generator.SwitchGenerator pdgf.generator.ConvertNumberToStrin pdgf.generator.ExtendedFormulaGene pdgf.generator.SetQuery pdgf.generator.StaticValueGenerator pdgf.output.SplitFileOutputWrapper pdgf.plugin.Distribution pdgf.core.dataGenerator.scheduler.Def pdgf.distribution.Logarithmic pdgf.output.SplitOutputWrapper
------------	--	-----------	---	---

table	x	1/Int.max	Node containing all Tables of this Project	
-------	---	-----------	--	--

1.8.3 XML-Example

```

<!--Description: This is the Project -->
<schema_pdgf.core.dbSchema.Project
  name="Desc: DefaultParser. (ClassName) of this element. Used to identify plugin Class. Full name
    is required. Example: com.en.myPluginPackage.myPuginClass"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element."
  targetVersion="(optional) Desc: Specifies program version a config file was created
    for.(optional)">
  <!--Required: [x] Executions(min|max): (1|INT.MAX) -->
  <!--Description: Properties (variables) in config file(s)-->
  <property
    name="Desc: the name of this property"
    type="Desc: the type of this property">Allowed values: {
      double
      long
      longUp
      longDown
      datetime
    }
  </property>
  <!--Required: [x] Executions(min|max): (1|INT.MAX) -->
  <!--Description: Node containing all Tables of this Project-->
  <table></table>
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: DefaultParser. Name of random number generator class to be used for calculations
    in this element. Example: com.en.myRNG-->
  <rng>Allowed values: {
    pdgf.util.random.PdgdDefaultRandom
  }
  </rng>
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: DefaultParser. Random number generator seed of this Element. Overrides default
    seeding behavior.-->
  <seed></seed>
  <!--Required: [ ] Executions(min|max): (0|INT.MAX) -->
  <!--Description: Invokes the default constructor of this class-->
  <loadPlugin>Allowed values: {
    pdgf.distribution.Normal
    pdgf.plugin.AbstractPDGFRandom
    pdgf.generator.HashGenerator
    pdgf.output.DummyOutput
    pdgf.core.dbSchema.Field
    pdgf.core.dataGenerator.scheduler.DefaultScheduler
    pdgf.core.dataGenerator.scheduler.UpdateScheduler
    pdgf.generator.ComputeGenerator
    pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
    pdgf.generator.FormattedNumberGenerator
    pdgf.distribution.Binomial
    pdgf.output.HierarchicalXMLRowOutput
    pdgf.generator.ForGenerator
    pdgf.generator.MarkovChainGenerator
    pdgf.generator.CDCSequenceGenerator
    pdgf.output.AbstractMultiLineOutput
    pdgf.generator.RandomUniqueStringGenerator
    pdgf.generator.FormatNumberAsDateGenerator
    pdgf.generator.SimpleFormatNumberGenerator
    pdgf.generator.BuildListGenerator
    pdgf.generator.DictList
    pdgf.generator.DoubleGenerator
    pdgf.util.random.PdgdDefaultRandom
    pdgf.generator.LastChoiceGenerator
    pdgf.core.dbSchema.ReferenceIMPL
    pdgf.distribution.Zeta
    pdgf.output.SortedOutputWrapper
    pdgf.generator.IFGenerator
    pdgf.core.dataGenerator.UpdateBlackBox
  }
  </loadPlugin>

```

```

    pdgf.generator.WeightedSubListGenerator
    pdgf.output.MultiLineOutput
    pdgf.plugin.Generator
    pdgf.output.CSVRowOutput
    pdgf.generator.DetailFromDateGenerator
    pdgf.generator.DateTimeGenerator
    pdgf.plugin.Scheduler
    pdgf.core.dbSchema.Project
    pdgf.generator.UpdateActionTypeGenerator
    pdgf.distribution.Beta
    pdgf.generator.IdGenerator
    pdgf.output.FileOutputSkeleton
    pdgf.generator.OtherFieldValueGenerator
    pdgf.generator.TemplateGenerator
    pdgf.generator.ForFormulaGenerator
    pdgf.generator.WeightedListGenerator
    pdgf.generator.AddRandomSecondsGenerator
    pdgf.generator.FormulaGenerator
    pdgf.generator.PaddingGenerator
    pdgf.generator.BigBenchReviewGenerator
    pdgf.generator.PrePostfixGenerator
    pdgf.core.dbSchema.Element
    pdgf.generator.DefaultReferenceGenerator
    pdgf.generator.RandomAString
    pdgf.generator.SwitchGenerator
    pdgf.generator.ConvertNumberToStringGenerator
    pdgf.generator.ExtendedFormulaGenerator
    pdgf.generator.SetQuery
    pdgf.generator.StaticValueGenerator
    pdgf.output.SplitFileOutputWrapper
    pdgf.plugin.Distribution
    pdgf.core.dataGenerator.scheduler.DefaultPartitioner
    pdgf.distribution.Logarithmic
    pdgf.output.SplitOutputWrapper
    pdgf.core.dataGenerator.scheduler.TemplatePartitioner
    pdgf.output.StatisticsOutput
    pdgf.generator.UUIDgenerator
    pdgf.generator.GlobalRowGenerator
    pdgf.generator.LongGenerator
    pdgf.generator.RelativeGenerator
    pdgf.output.SortedFileOutputWrapper
    pdgf.generator.EmailGenerator
    pdgf.generator.SequentialGenerator
    pdgf.generator.GenderGenerator
    pdgf.generator.RandomListItemsGenerator
    pdgf.generator.PermutationReferenceGenerator
    pdgf.core.dataGenerator.scheduler.DbDumpUpdateScheduler
    pdgf.output.MultiLineOutputWrapper
    pdgf.core.dbSchema.Table
    pdgf.generator.RandomValueXY
    pdgf.generator.NameGenerator
    pdgf.output.XMLTemplateOutput
    pdgf.generator.NullGenerator
    pdgf.distribution.Exponential
    pdgf.generator.ReferenceGenerator
    pdgf.generator.UpperLowerCaseGenerator
    pdgf.core.dbSchema.Partitioner
    pdgf.generator.AbstractDecimalGenerator
    pdgf.generator.ProbabilityGenerator
    pdgf.output.CompiledTemplateOutput
    pdgf.plugin.Output
    pdgf.generator.RandomSentence
  }
</loadPlugin>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: -->
<generation</generation>
</schema-pdgf.core.dbSchema.Project>

```

Listing 8: XML example with all options

1.9 pdgf.core.dbSchema.ReferenceIMPL

Represents a reference to another field in a database table

1.9.1 Attributes

Name	Req.	Min/- Max count	Description	Values
maxIDFrom-Scheduler-Prepartitioner		0/1	Use referenced tables Scheduler pre-partitioner plugin 'stop' value as 'maxID'	true false 0 1
field	x	1/1	The referenced Field	
minIDFrom-Scheduler-Prepartitioner		0/1	Use referenced tables Scheduler pre-partitioner plugin 'start' value as 'minID'	true false 0 1
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	
table		0/1	The referenced Table	
generator-ByID		0/1	reference to a specific generator choosen by element id <code><element id=" " ></code>	

1.9.2 Nodes

Name	Req.	Min/- Max count	Description	Values
maxID		0/1	maxID value for referenced ID (Choose Referenced ID in range: [minID, maxID]. Default: [0, Table.getSize()])	
minID		0/1	minID value for referenced ID (Choose Referenced ID in range: [minID, maxID]. Default: [0, Table.getSize()])	

1.9.3 XML-Example

```

<!--Description: Represents a reference to another field in a database table -->
<reference.REFERENCE
  field="Desc: The referenced Field"
  generatorByID="(optional) Desc: reference to a specific generator choosen by element id <element id=" ">"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element."
  maxIDFromSchedulerPrepartitioner="(optional) Desc: Use referenced tables Scheduler pre-partitioner plugin 'stop' value as 'maxID' Allowed values: {
    true
    false
    0
    1
  }"
  minIDFromSchedulerPrepartitioner="(optional) Desc: Use referenced tables Scheduler pre-partitioner plugin 'start' value as 'minID' Allowed values: {
    true
    false
    0
    1
  }"
  table="(optional) Desc: The referenced Table">
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: maxID value for referenced ID (Choose Referenced ID in range: [minID, maxID]. Default: [0, Table.getSize()])-->
  <maxID>/maxID>
  <!--Required: [ ] Executions(min|max): (0|1) -->

```

```

<!--Description: minID value for referenced ID (Choose Referenced ID in range: [minID, maxID].
      Default: [0, Table.getSize()])-->
<minID</minID>
</reference.REFERENCE>

```

Listing 9: XML example with all options

1.10 pdgf.core.dbSchema.Table

Represents a database table

1.10.1 Attributes

Name	Req.	Min/- Max count	Description	Values
name	x	1/1	DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	
type		0/1	Type of this table. Available: Standard, Composite, Update, CompositeUpdate,	Standard Composite Update CompositeUpdate

1.10.2 Nodes

Name	Req.	Min/- Max count	Description	Values
deletePer- centage		0/1	percentage of 'delete' ID entry's to generate per update. (minUpdateSize * deletePercentage) /100]	
seed		0/1	DefaultParser. Random number generator seed of this Element. Overrides default seeding behavior.	
minUpdate- Size		0/1	number of entry's to generate per update e.g 100. In combination with newPercentage, changePercentage, changePercentage this determines how many ID's will be new in this generation, how many will be changed and how many will be deleted.	
subTable		0/1	Node containing all sub tables of this table	
update- BlackBox		0/1	alternative UpdateBlackBox of this table.	
rng		0/1	DefaultParser. Name of random number generator class to be used for calculations in this element. Example: com.en.myRNG	pdgf.util.random.PdgdDefaultRandom
newPercent- age		0/1	percentage of new ID entry's to generate per update. (minUpdateSize * newPercentage) /100]	
same- SeedAsTable		0/1	this table should use the same seed as the specified table. If both tables would contain the same fields and generators, both of them would generate the exact same values if their seed matches.	
update- LastID		0/1	Last updateID to generate. And last valid updateID for generating references.	

size	x	1/1	Number of rows in this Table. In case it is an incremental table, size specifies the number of 'historical' initial IDs in the table. The historical size must be at least minUpdateSize * deletePercentage.	
field	x	1/Int.max	contains all fields of this table	
update-FirstID		0/1	First updateID to begin generating updates.	
changePercentage		0/1	percentage of 'change' ID entry's to generate per update. (minUpdateSize * updatePercentage) /100]	

1.10.3 XML-Example

```

<!--Description: Represents a database table -->
<table_pdgf.core.dbSchema.Table
  name="Desc: DefaultParser. (ClassName)Name of this element. Used to identify plugin Class. Full name
    is required. Example: com.en.myPluginPackage.myPuginClass"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element."
  type="(optional) Desc: Type of this table. Available: Standard, Composite, Update,
    CompositeUpdate, Allowed values: {
      Standard
      Composite
      Update
      CompositeUpdate
    }">
  <!--Required: [x] Executions(min|max): (1|INT.MAX) -->
  <!--Description: contains all fields of this table-->
  <field</field>
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: Number of rows in this Table. In case it is an incremental table, size specifies
    the number of 'historical' initial IDs in the table. The historical size must be at least
    minUpdateSize * deletePercentage.-->
  <size</size>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: percentage of 'change' ID entry's to generate per update. (minUpdateSize *
    updatePercentage) /100]-->
  <changePercentage</changePercentage>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: percentage of 'delete' ID entry's to generate per update. (minUpdateSize *
    deletePercentage) /100]-->
  <deletePercentage</deletePercentage>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: number of entry's to generate per update e.g 100. In combination with
    newPercentage, changePercentage, changePercentage this determines how many ID's will be new
    in this generation, how many will be changed and how many will be deleted.-->
  <minUpdateSize</minUpdateSize>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: percentage of new ID entry's to generate per update. (minUpdateSize *
    newPercentage) /100]-->
  <newPercentage</newPercentage>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: DefaultParser. Name of random number generator class to be used for calculations
    in this element. Example: com.en.myRNG-->
  <rng>Allowed values: {
    pdgf.util.random.PdgfDefaultRandom
  }
</rng>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: this table should use the same seed as the specified table. If both tables would
    contain the same fields and generators, both of them would generate the exact same values if
    their seed matches.-->
  <sameSeedAsTable</sameSeedAsTable>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: DefaultParser. Random number generator seed of this Element. Overrides default
    seeding behavior.-->
  <seed</seed>
  <!--Required: [ ] Executions(min|max): (0|1) -->

```

```

<!--Description: Node containing all sub tables of this table-->
<subTable
  name="Desc: the name of this table"></subTable>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: alternative UpdateBlackBox of this table.-->
<updateBlackBox></updateBlackBox>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: First updateID to begin generating updates.-->
<updateFirstID></updateFirstID>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Last updateID to generate. And last valid updateID for generating references.-->
<updateLastID></updateLastID>
</table_pdgf.core.dbSchema.Table>

```

Listing 10: XML example with all options

1.11 pdgf.distribution.Beta

Beta distribution

1.11.1 Attributes

Name	Req.	Min/- Max count	Description	Values
alpha	x	1/1	alpha value	
name	x	1/1	DefaultParser. (ClassName) of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	
beta	x	1/1	beta value	

1.11.2 Nodes

Name	Req.	Min/- Max count	Description	Values
------	------	-----------------------	-------------	--------

1.11.3 XML-Example

```

<!--Description: Beta distribution -->
<distribution_pdgf.distribution.Beta
  alpha="Desc: alpha value"
  beta="Desc: beta value"
  name="Desc: DefaultParser. (ClassName) of this element. Used to identify plugin Class. Full name
    is required. Example: com.en.myPluginPackage.myPuginClass"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
</distribution_pdgf.distribution.Beta>

```

Listing 11: XML example with all options

1.12 pdgf.distribution.Binomial

Binomial

1.12.1 Attributes

Name	Req.	Min/- Max count	Description	Values
p	x	1/1	p value	
name	x	1/1	DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	
n	x	1/1	n value	

1.12.2 Nodes

Name	Req.	Min/- Max count	Description	Values
------	------	-----------------------	-------------	--------

1.12.3 XML-Example

```
<!--Description: Binomial -->
<distribution_pdgf.distribution.Binomial
  n="Desc: n value"
  name="Desc: DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name
    is required. Example: com.en.myPluginPackage.myPuginClass"
  p="Desc: p value"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
</distribution_pdgf.distribution.Binomial>
```

Listing 12: XML example with all options

1.13 pdgf.distribution.Exponential

Exponential

1.13.1 Attributes

Name	Req.	Min/- Max count	Description	Values
lambda	x	1/1	0 p 1	
name	x	1/1	DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.13.2 Nodes

Name	Req.	Min/- Max count	Description	Values
------	------	-----------------------	-------------	--------

1.13.3 XML-Example

```

<!--Description: Exponential -->
<distribution_pdgf.distribution.Exponential
  lambda="Desc: 0 < p < 1"
  name="Desc: DefaultParser. (ClassName) of this element. Used to identify plugin Class. Full name
    is required. Example: com.en.myPluginPackage.myPuginClass"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
</distribution_pdgf.distribution.Exponential>

```

Listing 13: XML example with all options

1.14 pdgf.distribution.Logarithmic

Logarithmic distribution

1.14.1 Attributes

Name	Req.	Min/- Max count	Description	Values
p	x	1/1	$0 \leq p \leq 1$	
name	x	1/1	DefaultParser. (ClassName) of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.14.2 Nodes

Name	Req.	Min/- Max count	Description	Values
------	------	-----------------------	-------------	--------

1.14.3 XML-Example

```

<!--Description: Logarithmic distribution -->
<distribution_pdgf.distribution.Logarithmic
  name="Desc: DefaultParser. (ClassName) of this element. Used to identify plugin Class. Full name
    is required. Example: com.en.myPluginPackage.myPuginClass"
  p="Desc: 0 < p < 1"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
</distribution_pdgf.distribution.Logarithmic>

```

Listing 14: XML example with all options

1.15 pdgf.distribution.Normal

Normal distribtuion

1.15.1 Attributes

Name	Req.	Min/- Max count	Description	Values
sd	x	1/1	standard deviation value	
mean	x	1/1	median value	
name	x	1/1	DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.15.2 Nodes

Name	Req.	Min/- Max count	Description	Values
------	------	-----------------------	-------------	--------

1.15.3 XML-Example

```
<!--Description: Normal distribtuion -->
<distribution_pdgf.distribution.Normal
  mean="Desc: median value"
  name="Desc: DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name
  is required. Example: com.en.myPluginPackage.myPuginClass"
  sd="Desc: standard deviation value"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element.">
</distribution_pdgf.distribution.Normal>
```

Listing 15: XML example with all options

1.16 pdgf.distribution.Zeta

Zeta distribution with parameters ro and pk. Returns a zeta distributed random number. zeta sample range is defined as [1,n] but this distribution interface requires range [0,n], so generateZeta() subtracts -1 to map results to [0,n]

1.16.1 Attributes

Name	Req.	Min/- Max count	Description	Values
name	x	1/1	DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	
pk	x	1/1	pk value	
ro	x	1/1	ro value	

1.16.2 Nodes

Name	Req.	Min/- Max count	Description	Values
------	------	-----------------------	-------------	--------

1.16.3 XML-Example

```
<!--Description: Zeta distribution with parameters ro and pk. Returns a zeta distributed random number.
zeta sample range is defined as [1,n] but this distribution interface requires range [0,n], so
generateZeta() subtracts -1 to map results to [0,n] -->
<distribution_pdgf.distribution.Zeta
  name="Desc: DefaultParser. (ClassName) of this element. Used to identify plugin Class. Full name
    is required. Example: com.en.myPluginPackage.myPuginClass"
  pk="Desc: pk value"
  ro="Desc: ro value"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
</distribution_pdgf.distribution.Zeta>
```

Listing 16: XML example with all options

1.17 pdgf.generator.ActionDeleteIsAlsoUpdateGenerator

Transforms a 'delete' action into a 'change' action. In Action=='delete': instead of retrieving the last state of a field, re-seed the RandomNumberGenerator and run the specified subgenerator. Because of the re-seed the subgenerator will calculate a new (updated) value.

1.17.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.17.2 Nodes

Name	Req.	Min/- Max count	Description	Values
gen	x	1/1	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgf.generator.IdGenerator

1.17.3 XML-Example


```

<!--Description: Transforms a 'delete' action into a 'change' action. In Action=='delete': instead of
retrieving the last state of a field, re-seed the RandomNumberGenerator and run the specified
subgenerator. Because of the re-seed the subgenerator will calculate a new (updated) value. -->
<gen-pdgm.generator.ActionDeleteIsAlsoUpdateGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: Value Generator for this field -->
  <gen-(pluginName)>Allowed values: {
    pdgm.generator.HashGenerator
    pdgm.generator.ComputeGenerator
    pdgm.generator.ActionDeleteIsAlsoUpdateGenerator
    pdgm.generator.FormattedNumberGenerator
    pdgm.generator.ForGenerator
    pdgm.generator.MarkovChainGenerator
    pdgm.generator.CDCSequenceGenerator
    pdgm.generator.RandomUniqueStringGenerator
    pdgm.generator.FormatNumberAsDateGenerator
    pdgm.generator.SimpleFormatNumberGenerator
    pdgm.generator.BuildListGenerator
    pdgm.generator.DictList
    pdgm.generator.DoubleGenerator
    pdgm.generator.LastChoiceGenerator
    pdgm.generator.IFGenerator
    pdgm.generator.WeightedSubListGenerator
    pdgm.generator.DetailFromDateGenerator
    pdgm.generator.DateTimeGenerator
    pdgm.generator.UpdateActionTypeGenerator
    pdgm.generator.IdGenerator
    pdgm.generator.OtherFieldValueGenerator
    pdgm.generator.TemplateGenerator
    pdgm.generator.ForFormulaGenerator
    pdgm.generator.WeightedListGenerator
    pdgm.generator.AddRandomSecondsGenerator
    pdgm.generator.FormulaGenerator
    pdgm.generator.PaddingGenerator
    pdgm.generator.BigBenchReviewGenerator
    pdgm.generator.PrePostfixGenerator
    pdgm.generator.DefaultReferenceGenerator
    pdgm.generator.RandomAString
    pdgm.generator.SwitchGenerator
    pdgm.generator.ConvertNumberToStringGenerator
    pdgm.generator.ExtendedFormulaGenerator
    pdgm.generator.SetQuery
    pdgm.generator.StaticValueGenerator
    pdgm.generator.UUIDgenerator
    pdgm.generator.GlobalRowGenerator
    pdgm.generator.LongGenerator
    pdgm.generator.RelativeGenerator
    pdgm.generator.EmailGenerator
    pdgm.generator.SequentialGenerator
    pdgm.generator.GenderGenerator
    pdgm.generator.RandomListItemsGenerator
    pdgm.generator.PermutationReferenceGenerator
    pdgm.generator.RandomValueXY
    pdgm.generator.NameGenerator
    pdgm.generator.NullGenerator
    pdgm.generator.ReferenceGenerator
    pdgm.generator.UpperLowerCaseGenerator
    pdgm.generator.AbstractDecimalGenerator
    pdgm.generator.ProbabilityGenerator
    pdgm.generator.RandomSentence
  }
  </gen-(pluginName)>
</gen-pdgm.generator.ActionDeleteIsAlsoUpdateGenerator>

```

Listing 17: XML example with all options

1.18 pdgm.generator.AddRandomSecondsGenerator

Adds a random number (between [min, max] parameters) of seconds to a date retrieved by a reference.

1.18.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.18.2 Nodes

Name	Req.	Min/- Max count	Description	Values
reference	x	1/1	reference	
disableRng		0/1	If set to true, instead of choosing randomly, this generator uses the the current row id as random source. If 'useFixedStepSize' is set to false	true false 0 1
min	x	1/1	minimum value of seconds to add	
endDate		0/1	End of time interval from which values are generated. Expecting either a human readable time format or properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31 23:45:59 or \${someUserDefinedPropertie}	
max	x	1/1	maximum value of seconds to add	
historicalUpdateID- sCount		0/1	Number of updateID's treated logically as 'historical'. Logically historical updateIDs can have a different date/-time range than 'normal' updateID's. For historical updateIDs the date/time range: [historicalStartDate, historicalEndDate] is used	
historical- EndDate		0/1	in case your historical range (range [0,1[or in case 'historicalUpdateIDsCount' is set [0,historicalUpdateIDsCount[) should differ from the following range. Begin of time interval from which values are generated. Expecting either a human readable time format or properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31 23:45:59 or \${someUserDefinedPropertie}	
historical- StartDate		0/1	in case your historical range (range [0,1[or in case 'historicalUpdateIDsCount' is set [0,historicalUpdateIDsCount[) should differ from the following range. End of time interval from which values are generated. Expecting either a human readable time format or properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31 23:45:59 or \${someUserDefinedPropertie}	
inputFormat		0/1	Date or time input format for 'startDate', 'endDate', 'historicalStartDate' and 'historicalEndDate'. Default format: yyyy-MM-dd HH:mm:ss	
outputFormat	x	1/1	Date or time output format. Default format: yyyy-MM-dd HH:mm:ss	
startDate		0/1	Begin of time interval from which values are generated. Expecting either a human readable time format or properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31 23:45:59 or \${someUserDefinedPropertie}	
useFixed- StepSize		0/1	If set to false (default), each date.n will be computed with: date.n = date.n-1 + random . In this case 'disableRng' should be set to true!. If set to true, each date will have a distance of [startDate, endDate]/numberOfRows to its next date.	true false 0 1

1.18.3 XML-Example

```
<!--Description: Adds a random number (between [min, max] parameters) of seconds to a date retrieved by a
reference. -->
<gen.pdgm.generator.AddRandomSecondsGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: maximum value of seconds to add-->
  <max-->/max>
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: minimum value of seconds to add-->
  <min-->/min>
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: Date or time output format. Default format: yyyy-MM-dd HH:mm:ss-->
  <outputFormat-->/outputFormat>
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: reference -->
  <reference-->/reference>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: If set to true, instead of choosing randomly, this generator uses the the
    current row id as random source. If 'useFixedStepSize' is set to false-->
  <disableRng>Allowed values: {
    true
    false
    0
    1
  }
</disableRng>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: End of time interval from which values are generated. Expecting either a human
    readable time format or properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31
    23:45:59 or ${someUserDefinedPropertie}-->
  <endDate-->/endDate>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: in case your historical range (range [0,1[ or in case 'historicalUpdateIDsCount'
    is set [0,historicalUpdateIDsCount[ ) should differ from the following range. Begin of time
    interval from which values are generated. Expecting either a human readable time format or
    properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31 23:45:59 or
    ${someUserDefinedPropertie}-->
  <historicalEndDate-->/historicalEndDate>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: in case your historical range (range [0,1[ or in case 'historicalUpdateIDsCount'
    is set [0,historicalUpdateIDsCount[ ) should differ from the following range. End of time
    interval from which values are generated. Expecting either a human readable time format or
    properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31 23:45:59 or
    ${someUserDefinedPropertie}-->
  <historicalStartDate-->/historicalStartDate>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Number of updateID's treated logically as 'historical'. Logically historical
    updateIDs can have a different date/time range than 'normal' updateID's. For historical
    updateIDs the date/time range: [historicalStartDate, historicalEndDate] is used-->
  <historicalUpdateIDsCount-->/historicalUpdateIDsCount>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Date or time input format for 'startDate', 'endDate', 'historicalStartDate' and
    'historicalEndDate'. Default format: yyyy-MM-dd HH:mm:ss-->
  <inputFormat-->/inputFormat>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Begin of time interval from which values are generated. Expecting either a human
    readable time format or properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31
    23:45:59 or ${someUserDefinedPropertie}-->
  <startDate-->/startDate>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: If set to false (default), each date_n will be computed with: date_n = date_n-1
    + random . In this case 'disableRng' should be set to true!. If set to true, each date will
    have a distance of [startDate, endDate]/numberOfRows to its next date.-->
  <useFixedStepSize>Allowed values: {
    true
    false
    0
    1
  }
</useFixedStepSize>
```

Listing 18: XML example with all options

1.19 pdgf.generator.BigBenchReviewGenerator

generates reviews

1.19.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.19.2 Nodes

Name	Req.	Min/- Max count	Description	Values
gen	x	2/3	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgf.generator.IdGenerator
markovChainsPath	x	1/1	directory containing markov chains in format: jmi-norj-majorj-ratingj.bin	

1.19.3 XML-Example

```
<!--Description: generates reviews -->
<gen_pdgf.generator.BigBenchReviewGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (2|3) -->
  <!--Description: Value Generator for this field-->
  <gen_(pluginName)>Allowed values: {
    pdgf.generator.HashGenerator
    pdgf.generator.ComputeGenerator
    pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
    pdgf.generator.FormattedNumberGenerator
    pdgf.generator.ForGenerator
    pdgf.generator.MarkovChainGenerator
    pdgf.generator.CDCSequenceGenerator
    pdgf.generator.RandomUniqueStringGenerator
    pdgf.generator.FormatNumberAsDateGenerator
    pdgf.generator.SimpleFormatNumberGenerator
    pdgf.generator.BuildListGenerator
    pdgf.generator.DictList
    pdgf.generator.DoubleGenerator
    pdgf.generator.LastChoiceGenerator
    pdgf.generator.IFGenerator
    pdgf.generator.WeightedSubListGenerator
    pdgf.generator.DetailFromDateGenerator
    pdgf.generator.DateTimeGenerator
    pdgf.generator.UpdateActionTypeGenerator
    pdgf.generator.IdGenerator
    pdgf.generator.OtherFieldValueGenerator
```

```

    pdgf.generator.TemplateGenerator
    pdgf.generator.ForFormulaGenerator
    pdgf.generator.WeightedListGenerator
    pdgf.generator.AddRandomSecondsGenerator
    pdgf.generator.FormulaGenerator
    pdgf.generator.PaddingGenerator
    pdgf.generator.BigBenchReviewGenerator
    pdgf.generator.PrePostfixGenerator
    pdgf.generator.DefaultReferenceGenerator
    pdgf.generator.RandomAString
    pdgf.generator.SwitchGenerator
    pdgf.generator.ConvertNumberToStringGenerator
    pdgf.generator.ExtendedFormulaGenerator
    pdgf.generator.SetQuery
    pdgf.generator.StaticValueGenerator
    pdgf.generator.UUIDgenerator
    pdgf.generator.GlobalRowGenerator
    pdgf.generator.LongGenerator
    pdgf.generator.RelativeGenerator
    pdgf.generator.EmailGenerator
    pdgf.generator.SequentialGenerator
    pdgf.generator.GenderGenerator
    pdgf.generator.RandomListItemsGenerator
    pdgf.generator.PermutationReferenceGenerator
    pdgf.generator.RandomValueXY
    pdgf.generator.NameGenerator
    pdgf.generator.NullGenerator
    pdgf.generator.ReferenceGenerator
    pdgf.generator.UpperLowerCaseGenerator
    pdgf.generator.AbstractDecimalGenerator
    pdgf.generator.ProbabilityGenerator
    pdgf.generator.RandomSentence
  }
</gen_(pluginName)>
<!--Required: [x] Executions(min|max): (1|1) -->
<!--Description: directory containing markov chains in format: <minor>.<major>.<rating>.bin-->
<markovChainsPath></markovChainsPath>
</gen_pdgf.generator.BigBenchReviewGenerator>

```

Listing 19: XML example with all options

1.20 pdgf.generator.BuildListGenerator

Aggregates Sub-Generator values into a List. Has the option to unify multiple Sub-Generator into one(See attribute: flattenSubList). This generator returns: List<Object>. Beware: Most "default" output plugins are not aware of 'Lists' and just call toString() on the list!

1.20.1 Attributes

Name	Req.	Min/- Max count	Description	Values
flattenSubList		0/1	flattenSubList	true false 0 1
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	
listName		0/1	name of list. List will be of type pdgf.util.io.generator.NamedArrayList extends java.util.ArrayList and has a getName() method which you can use in your custom output.	

1.20.2 Nodes

Name	Req.	Min/- Max count	Description	Values
gen	x	1/Int.max	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgf.generator.IdGenerator

1.20.3 XML-Example

```

<!--Description: Aggregates Sub-Generator values into a List. Has the option to unify multiple
Sub-Generator into one(See attribute: flattenSubList). This generator returns: List<Object>. Beware:
Most "default" output plugins are not aware of 'Lists' and just call toString() on the list! -->
<gen-pdgf.generator.BuildListGenerator
  flattenSubList="(optional) Desc: flattenSubList Allowed values: {
    true
    false
    0
    1
  }"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element."
  listName="(optional) Desc: name of list. List will be of type
  pdgf.util.io.generator.NamedArrayList extends java.util.ArrayList and has a getName() method
  which you can use in your custom output.">
<!--Required: [x] Executions(min|max): (1|INT.MAX) -->
<!--Description: Value Generator for this field-->
<gen-(pluginName)>Allowed values: {
  pdgf.generator.HashGenerator
  pdgf.generator.ComputeGenerator
  pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
  pdgf.generator.FormattedNumberGenerator
  pdgf.generator.ForGenerator
  pdgf.generator.MarkovChainGenerator
  pdgf.generator.CDCSequenceGenerator
  pdgf.generator.RandomUniqueStringGenerator
  pdgf.generator.FormatNumberAsDateGenerator
  pdgf.generator.SimpleFormatNumberGenerator
  pdgf.generator.BuildListGenerator
  pdgf.generator.DictList
  pdgf.generator.DoubleGenerator
  pdgf.generator.LastChoiceGenerator
  pdgf.generator.IFGenerator
  pdgf.generator.WeightedSubListGenerator
  pdgf.generator.DetailFromDateGenerator
  pdgf.generator.DateTimeGenerator
  pdgf.generator.UpdateActionTypeGenerator
  pdgf.generator.IdGenerator
  pdgf.generator.OtherFieldValueGenerator
  pdgf.generator.TemplateGenerator
  pdgf.generator.ForFormulaGenerator
  pdgf.generator.WeightedListGenerator
  pdgf.generator.AddRandomSecondsGenerator
  pdgf.generator.FormulaGenerator
  pdgf.generator.PaddingGenerator
  pdgf.generator.BigBenchReviewGenerator
  pdgf.generator.PrePostfixGenerator
  pdgf.generator.DefaultReferenceGenerator
  pdgf.generator.RandomAString
  pdgf.generator.SwitchGenerator
  pdgf.generator.ConvertNumberToStringGenerator
  pdgf.generator.ExtendedFormulaGenerator
  pdgf.generator.SetQuery
  pdgf.generator.StaticValueGenerator
  pdgf.generator.UUIDgenerator
  pdgf.generator.GlobalRowGenerator

```

```

        pdgf.generator.LongGenerator
        pdgf.generator.RelativeGenerator
        pdgf.generator.EmailGenerator
        pdgf.generator.SequentialGenerator
        pdgf.generator.GenderGenerator
        pdgf.generator.RandomListItemsGenerator
        pdgf.generator.PermutationReferenceGenerator
        pdgf.generator.RandomValueXY
        pdgf.generator.NameGenerator
        pdgf.generator.NullGenerator
        pdgf.generator.ReferenceGenerator
        pdgf.generator.UpperLowerCaseGenerator
        pdgf.generator.AbstractDecimalGenerator
        pdgf.generator.ProbabilityGenerator
        pdgf.generator.RandomSentence
    }
    </gen_(pluginName)>
</gen-pdgf.generator.BuildListGenerator>

```

Listing 20: XML example with all options

1.21 pdgf.generator.CDCSequenceGenerator

Calculates a continuous dense sequence number (continuing over all updates), based on the current row.

1.21.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.21.2 Nodes

Name	Req.	Min/- Max count	Description	Values
substractHistoricalSize		0/1	Sequence will restart at 0 at updateID 1. Calculate with: generationContext.getCurRow()+ generationContext.getUpdate().getMaxRowID()- parentTableHistoricalSize	true false 0 1

1.21.3 XML-Example

```

<!--Description: Calculates a continuous dense sequence number (continuing over all updates), based on
the current row. -->
<gen-pdgf.generator.CDCSequenceGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
uniquely identify a field within the children of an Element.">
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Sequence will restart at 0 at updateID 1. Calculate with:
generationContext.getCurRow()+ generationContext.getUpdate().getMaxRowID()-
parentTableHistoricalSize -->
  <substractHistoricalSize>Allowed values: {
    true
    false
    0
    1
  }

```

```
</subtractHistoricalSize>
</gen-pdgm.generator.CDCSequenceGenerator>
```

Listing 21: XML example with all options

1.22 pdgm.generator.ComputeGenerator

Supports basic arithmetic operations (+ - * / %). Computes the (double) result of two sub generators using the specified operand. e.g generator[0] + generator[1] or generator[0] * generator[1] ...

1.22.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.22.2 Nodes

Name	Req.	Min/- Max count	Description	Values
gen	x	1/1	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgm.generator.IdGenerator
decimalPlaces		0/1	number of decimal places of output. Example: places=2 -> output: 3.64; places=3 -> 3.642. Default: noLimit	
operator	x	1/1	The operator to use.	

1.22.3 XML-Example

```
<!--Description: Supports basic arithmetic operations ( + - * / % ). Computes the (double) result of two
sub generators using the specified operand. e.g generator[0] + generator[1] or generator[0] *
generator[1] ... -->
<gen-pdgm.generator.ComputeGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: Value Generator for this field-->
  <gen-(pluginName)>Allowed values: {
    pdgm.generator.HashGenerator
    pdgm.generator.ComputeGenerator
    pdgm.generator.ActionDeleteIsAlsoUpdateGenerator
    pdgm.generator.FormattedNumberGenerator
    pdgm.generator.ForGenerator
    pdgm.generator.MarkovChainGenerator
    pdgm.generator.CDCSequenceGenerator
    pdgm.generator.RandomUniqueStringGenerator
    pdgm.generator.FormatNumberAsDateGenerator
    pdgm.generator.SimpleFormatNumberGenerator
    pdgm.generator.BuildListGenerator
    pdgm.generator.DictList
    pdgm.generator.DoubleGenerator
    pdgm.generator.LastChoiceGenerator
    pdgm.generator.IFGenerator
```



```

    pdgf.generator.WeightedSubListGenerator
    pdgf.generator.DetailFromDateGenerator
    pdgf.generator.DateTimeGenerator
    pdgf.generator.UpdateActionTypeGenerator
    pdgf.generator.IdGenerator
    pdgf.generator.OtherFieldValueGenerator
    pdgf.generator.TemplateGenerator
    pdgf.generator.ForFormulaGenerator
    pdgf.generator.WeightedListGenerator
    pdgf.generator.AddRandomSecondsGenerator
    pdgf.generator.FormulaGenerator
    pdgf.generator.PaddingGenerator
    pdgf.generator.BigBenchReviewGenerator
    pdgf.generator.PrePostfixGenerator
    pdgf.generator.DefaultReferenceGenerator
    pdgf.generator.RandomAString
    pdgf.generator.SwitchGenerator
    pdgf.generator.ConvertNumberToStringGenerator
    pdgf.generator.ExtendedFormulaGenerator
    pdgf.generator.SetQuery
    pdgf.generator.StaticValueGenerator
    pdgf.generator.UUIDgenerator
    pdgf.generator.GlobalRowGenerator
    pdgf.generator.LongGenerator
    pdgf.generator.RelativeGenerator
    pdgf.generator.EmailGenerator
    pdgf.generator.SequentialGenerator
    pdgf.generator.GenderGenerator
    pdgf.generator.RandomListItemsGenerator
    pdgf.generator.PermutationReferenceGenerator
    pdgf.generator.RandomValueXY
    pdgf.generator.NameGenerator
    pdgf.generator.NullGenerator
    pdgf.generator.ReferenceGenerator
    pdgf.generator.UpperLowerCaseGenerator
    pdgf.generator.AbstractDecimalGenerator
    pdgf.generator.ProbabilityGenerator
    pdgf.generator.RandomSentence
  }
</gen-(pluginName)>
<!--Required: [x] Executions(min|max): (1|1) -->
<!--Description: The operator to use.-->
<operator></operator>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: number of decimal places of output. Example: places=2 -> output: 3.64; places=3
-> 3.642. Default: noLimit-->
<decimalPlaces
  roundPlainValue="(optional) Desc: Do rounding of generated double. if decimalPlaces
  attribute is specified. Default rounding mode is: HALF_EVEN. Default for
  'roundPlainValue' is:false Allowed values: {
    true
    false
    0
    1
  }"
  roundingMode="(optional) Desc: Set rounding mode. Default rounding mode is: HALF_EVEN
  Possible values: [UP, DOWN, CEILING, FLOOR, HALF_UP, HALF_DOWN, HALF_EVEN,
  UNNECESSARY] Allowed values: {
    UP
    DOWN
    CEILING
    FLOOR
    HALF_UP
    HALF_DOWN
    HALF_EVEN
    UNNECESSARY
  }"</decimalPlaces>
</gen-pdgf.generator.ComputeGenerator>

```

Listing 22: XML example with all options

1.23 pdgf.generator.ConvertNumberToStringGenerator

This generator converts a number like '123456789' into a character representation of specified 'size'. e.g. '123456789' will result in 'aaaaaaaaapHnxu' using the default character set 'a-zA-Z' and 'size'=15.

1.23.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.23.2 Nodes

Name	Req.	Min/- Max count	Description	Values
gen		0/1	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgf.generator.IdGenerator
characters		0/1	Sets characters this generator uses for generating a random string. Default:abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ	
size	x	1/1	length of String to be generated from the number. Example: input number '123456789' will result int : 'aaaaaaaaapHnxu' using the default character set 'a-zA-Z' and 'size'=15.	

1.23.3 XML-Example

```
<!--Description: This generator converts a number like '123456789' into a character representation of
specified 'size'. e.g. '123456789' will result in 'aaaaaaaaapHnxu' using the default character set
'a-zA-Z' and 'size'=15. -->
<gen_pdgf.generator.ConvertNumberToStringGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: length of String to be generated from the number. Example: input number
  '123456789' will result int : 'aaaaaaaaapHnxu' using the default character set 'a-zA-Z' and
  'size'=15.-->
  <size></size>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Sets characters this generator uses for generating a random string.
  Default:abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ-->
  <characters></characters>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Value Generator for this field-->
  <gen_(pluginName)>Allowed values: {
    pdgf.generator.HashGenerator
    pdgf.generator.ComputeGenerator
    pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
    pdgf.generator.FormattedNumberGenerator
    pdgf.generator.ForGenerator
    pdgf.generator.MarkovChainGenerator
    pdgf.generator.CDCSequenceGenerator
    pdgf.generator.RandomUniqeStringGenerator
```

```

    pdgf.generator.FormatNumberAsDateGenerator
    pdgf.generator.SimpleFormatNumberGenerator
    pdgf.generator.BuildListGenerator
    pdgf.generator.DictList
    pdgf.generator.DoubleGenerator
    pdgf.generator.LastChoiceGenerator
    pdgf.generator.IFGenerator
    pdgf.generator.WeightedSubListGenerator
    pdgf.generator.DetailFromDateGenerator
    pdgf.generator.DateTimeGenerator
    pdgf.generator.UpdateActionTypeGenerator
    pdgf.generator.IdGenerator
    pdgf.generator.OtherFieldValueGenerator
    pdgf.generator.TemplateGenerator
    pdgf.generator.ForFormulaGenerator
    pdgf.generator.WeightedListGenerator
    pdgf.generator.AddRandomSecondsGenerator
    pdgf.generator.FormulaGenerator
    pdgf.generator.PaddingGenerator
    pdgf.generator.BigBenchReviewGenerator
    pdgf.generator.PrePostfixGenerator
    pdgf.generator.DefaultReferenceGenerator
    pdgf.generator.RandomAString
    pdgf.generator.SwitchGenerator
    pdgf.generator.ConvertNumberToStringGenerator
    pdgf.generator.ExtendedFormulaGenerator
    pdgf.generator.SetQuery
    pdgf.generator.StaticValueGenerator
    pdgf.generator.UUIDgenerator
    pdgf.generator.GlobalRowGenerator
    pdgf.generator.LongGenerator
    pdgf.generator.RelativeGenerator
    pdgf.generator.EmailGenerator
    pdgf.generator.SequentialGenerator
    pdgf.generator.GenderGenerator
    pdgf.generator.RandomListItemsGenerator
    pdgf.generator.PermutationReferenceGenerator
    pdgf.generator.RandomValueXY
    pdgf.generator.NameGenerator
    pdgf.generator.NullGenerator
    pdgf.generator.ReferenceGenerator
    pdgf.generator.UpperLowerCaseGenerator
    pdgf.generator.AbstractDecimalGenerator
    pdgf.generator.ProbabilityGenerator
    pdgf.generator.RandomSentence
  }
</gen-(pluginName)>
</gen-pdgf.generator.ConvertNumberToStringGenerator>

```

Listing 23: XML example with all options

1.24 pdgf.generator.DateTimeGenerator

DateTime generator. You have to define a start and end date. Either by specifying named references to foreign sources of date values (id="startDateRef", id="endDateRef", id="historicalStartDateRef", id="historicalEndDateRef") or specifying static ranges with: ['startDate', 'endDate'] and optional, if 'historicalUpdateIDsCount' !=0, additionally with ['startDate', 'endDate']

1.24.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.24.2 Nodes

Name	Req.	Min/- Max count	Description	Values
reference		0/2	reference	
disableRng		0/1	If set to true, instead of choosing randomly, this generator uses the the current row id as random source. If 'useFixedStepSize' is set to false	true false 0 1
endDate		0/1	End of time interval from which values are generated. Expecting either a human readable time format or properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31 23:45:59 or \${someUserDefinedPropertie}	
historicalUpdateID- sCount		0/1	Number of updateID's treated logically as 'historical'. Logically historical updateIDs can have a different date/-time range than 'normal' updateID's. For historical updateIDs the date/time range: [historicalStartDate, historicalEndDate] is used	
historical- EndDate		0/1	in case your historical range (range [0,1[or in case 'historicalUpdateIDsCount' is set [0,historicalUpdateIDsCount[) should differ from the following range. Begin of time interval from which values are generated. Expecting either a human readable time format or properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31 23:45:59 or \${someUserDefinedPropertie}	
historical- StartDate		0/1	in case your historical range (range [0,1[or in case 'historicalUpdateIDsCount' is set [0,historicalUpdateIDsCount[) should differ from the following range. End of time interval from which values are generated. Expecting either a human readable time format or properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31 23:45:59 or \${someUserDefinedPropertie}	
inputFormat		0/1	Date or time input format for 'startDate', 'endDate', 'historicalStartDate' and 'historicalEndDate'. Default format: yyyy-MM-dd HH:mm:ss	
outputFor- mat	x	1/1	Date or time output format. Default format: yyyy-MM-dd HH:mm:ss	
startDate		0/1	Begin of time interval from which values are generated. Expecting either a human readable time format or properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31 23:45:59 or \${someUserDefinedPropertie}	
useFixed- StepSize		0/1	If set to false (default), each date _n will be computed with: date _n = date _{n-1} + random . In this case 'disableRng' should be set to true!. If set to true, each date will have a distance of [startDate, endDate]/numberOfRows to its next date.	true false 0 1

1.24.3 XML-Example

```

<!--Description: DateTime generator. You have to define a start and end date. Either by specifying named
references to foreign sources of date values (id="startDateRef", id="endDateRef",
id="historicalStartDateRef", id="historicalEndDateRef") or specifying static ranges with:
['startDate', 'endDate'] and optional, if 'historicalUpdateIDsCount' !=0, additionally with
['startDate', 'endDate'] -->
<gen_pdgf.generator.DateTimeGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->

```

```

<!--Description: Date or time output format. Default format: yyyy-MM-dd HH:mm:ss-->
<outputFormat></outputFormat>
<!--Required: [ ] Executions(min|max): (0|2) -->
<!--Description: reference-->
<reference></reference>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: If set to true, instead of choosing randomly, this generator uses the the
current row id as random source. If 'useFixedStepSize' is set to false-->
<disableRng>Allowed values: {
    true
    false
    0
    1
}
</disableRng>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: End of time interval from which values are generated. Expecting either a human
readable time format or properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31
23:45:59 or ${someUserDefinedPropertie}-->
<endDate></endDate>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: in case your historical range (range [0,1[ or in case 'historicalUpdateIDsCount'
is set [0,historicalUpdateIDsCount[ ) should differ from the following range. Begin of time
interval from which values are generated. Expecting either a human readable time format or
properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31 23:45:59 or
${someUserDefinedPropertie}-->
<historicalEndDate></historicalEndDate>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: in case your historical range (range [0,1[ or in case 'historicalUpdateIDsCount'
is set [0,historicalUpdateIDsCount[ ) should differ from the following range. End of time
interval from which values are generated. Expecting either a human readable time format or
properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31 23:45:59 or
${someUserDefinedPropertie}-->
<historicalStartDate></historicalStartDate>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Number of updateID's treated logically as 'historical'. Logically historical
updateIDs can have a different date/time range than 'normal' updateID's. For historical
updateIDs the date/time range: [historicalStartDate, historicalEndDate] is used-->
<historicalUpdateIDsCount></historicalUpdateIDsCount>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Date or time input format for 'startDate', 'endDate', 'historicalStartDate' and
'historicalEndDate'. Default format: yyyy-MM-dd HH:mm:ss-->
<inputFormat></inputFormat>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Begin of time interval from which values are generated. Expecting either a human
readable time format or properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31
23:45:59 or ${someUserDefinedPropertie}-->
<startDate></startDate>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: If set to false (default), each date_n will be computed with: date_n = date_n-1
+ random . In this case 'disableRng' should be set to true!. If set to true, each date will
have a distance of [startDate, endDate]/numberOfRows to its next date.-->
<useFixedStepSize>Allowed values: {
    true
    false
    0
    1
}
</useFixedStepSize>
</gen-pdgm.generator.DateTimeGenerator>

```

Listing 24: XML example with all options

1.25 pdgm.generator.DefaultReferenceGenerator

Generates values for referencing field by following the reference and randomly (following the specified distribution) picking an ID from the value set of valid ID's from the referenced Table and recalculating its Value. Default updateIDOffset = 0. Default: chose reference from SAME updateID. This may skew time dependencies as it might chooses a reference to an ID yet to be created in this generation. To prevent this, use an updateIDOffset of -1 to choose reference from previous generation of referenced Table.

1.25.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.25.2 Nodes

Name	Req.	Min/- Max count	Description	Values
reference	x	1/1	reference	
same-ChoiceAs		0/1	Requires a <code>field</code> and a <code>generatorByID</code> node (in same Table) to pick the row number from. If specified this DefaultReferenceGenerator does not choose a random row, but it uses the same row as the DefaultReferenceGenerator of the specified field/generator.	
updateID-Offset		0/1	Default: chose reference from SAME updateID. This may skew time dependencies as it might choose a reference to an ID yet to be created in this generation. To prevent this, use an updateIDOffset of -1 to choose reference from previous generation in referenced Table. Default offset is 0.	
distribution		0/1	Distribution to be used by a generator when calculating a value	
includeDeletedIDs		0/1	when generating references to an update table, also reference already deleted IDs.	true false 0 1
useRelative-UpdateIDs		0/1	Use updateIDs relative to the number of updates of the parent table for reference access	true false 0 1

1.25.3 XML-Example

```

<!--Description: Generates values for referencing field by following the reference and randomly
(following the specified distribution) picking an ID from the value set of valid ID's from the
referenced Table and recalculating its Value. Default updateIDOffset = 0. Default: chose reference
from SAME updateID. This may skew time dependencies as it might chooses a reference to an ID yet to
be created in this generation. To prevent this, use an updateIDOffset of -1 to choose reference from
previous generation of referenced Table. -->
<gen_pdgf.generator.DefaultReferenceGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: reference -->
  <reference></reference>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Distribution to be used by a generator when calculating a value-->
  <distribution></distribution>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: when generating references to an update table, also reference already deleted
  IDs.-->
  <includeDeletedIDs>Allowed values: {
    true
    false
    0
    1
  }

```

```

    }
</includeDeletedIDs>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Requires a <field> and a <generatorByID> node ( in same Table) to pick the row
number from. If specified this DefaultReferenceGenerator does not choose a random row, but it
uses the same row as the DefaultReferenceGenerator of the specified field/generator.-->
<sameChoiceAs></sameChoiceAs>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Default: chose reference from SAME updateID. This may skew time dependencies as
it might choose a reference to an ID yet to be created in this generation. To prevent this,
use an updateIDOffset of -1 to choose reference from previous generation in referenced Table.
Default offset is 0.-->
<updateIDOffset></updateIDOffset>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Use updateIDs relative to the number of updates of the parent table for
reference access-->
<useRelativeUpdateIDs>Allowed values: {
    true
    false
    0
    1
}
</useRelativeUpdateIDs>
</gen_pdgf.generator.DefaultReferenceGenerator>

```

Listing 25: XML example with all options

1.26 pdgf.generator.DetailFromDateGenerator

Gets parts from a timestamp in another field (like year, quarter, ...)

1.26.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.26.2 Nodes

Name	Req.	Min/- Max count	Description	Values
reference	x	1/1	reference	
isFiscal		0/1	If this is true, half a year is added to the date set	true false 0 1
disableRng		0/1	If set to true, instead of choosing randomly, this generator uses the the current row id as random source. If 'useFixedStepSize' is set to false	true false 0 1
endDate		0/1	End of time interval from which values are generated. Expecting either a human readable time format or properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31 23:45:59 or \${someUserDefinedPropertie}	

historicalUpdateID-sCount		0/1	Number of updateID's treated logically as 'historical'. Logically historical updateIDs can have a different date/-time range than 'normal' updateID's. For historical updateIDs the date/time range: [historicalStartDate, historicalEndDate] is used	
historical-EndDate		0/1	in case your historical range (range [0,1[or in case 'historicalUpdateIDsCount' is set [0,historicalUpdateIDsCount[) should differ from the following range. Begin of time interval from which values are generated. Expecting either a human readable time format or properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31 23:45:59 or \${someUserDefinedPropertie}	
historical-StartDate		0/1	in case your historical range (range [0,1[or in case 'historicalUpdateIDsCount' is set [0,historicalUpdateIDsCount[) should differ from the following range. End of time interval from which values are generated. Expecting either a human readable time format or properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31 23:45:59 or \${someUserDefinedPropertie}	
inputFormat		0/1	Date or time input format for 'startDate', 'endDate', 'historicalStartDate' and 'historicalEndDate'. Default format: yyyy-MM-dd HH:mm:ss	
outputFormat	x	1/1	Date or time output format. Default format: yyyy-MM-dd HH:mm:ss	
startDate		0/1	Begin of time interval from which values are generated. Expecting either a human readable time format or properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31 23:45:59 or \${someUserDefinedPropertie}	
useFixed-StepSize		0/1	If set to false (default), each date_n will be computed with: date_n = date_n-1 + random . In this case 'disableRng' should be set to true!. If set to true, each date will have a distance of [startDate, endDate]/numberOfRows to its next date.	true false 0 1

1.26.3 XML-Example

```

<!--Description: Gets parts from a timestamp in another field (like year, quarter, ...) -->
<gen.pdgm.generator.DetailFromDateGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: Date or time output format. Default format: yyyy-MM-dd HH:mm:ss-->
  <outputFormat></outputFormat>
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: reference-->
  <reference></reference>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: If set to true, instead of choosing randomly, this generator uses the the
    current row id as random source. If 'useFixedStepSize' is set to false-->
  <disableRng>Allowed values: {
    true
    false
    0
    1
  }
</disableRng>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: End of time interval from which values are generated. Expecting either a human
  readable time format or properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31
  23:45:59 or ${someUserDefinedPropertie}-->
<endDate></endDate>
<!--Required: [ ] Executions(min|max): (0|1) -->

```



```

<!--Description: in case your historical range (range [0,1[ or in case 'historicalUpdateIDsCount'
is set [0,historicalUpdateIDsCount[ ) should differ from the following range. Begin of time
interval from which values are generated. Expecting either a human readable time format or
properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31 23:45:59 or
${someUserDefinedPropertie}-->
<historicalEndDate</historicalEndDate>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: in case your historical range (range [0,1[ or in case 'historicalUpdateIDsCount'
is set [0,historicalUpdateIDsCount[ ) should differ from the following range. End of time
interval from which values are generated. Expecting either a human readable time format or
properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31 23:45:59 or
${someUserDefinedPropertie}-->
<historicalStartDate</historicalStartDate>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Number of updateID's treated logically as 'historical'. Logically historical
updateIDs can have a different date/time range than 'normal' updateID's. For historical
updateIDs the date/time range: [historicalStartDate, historicalEndDate] is used-->
<historicalUpdateIDsCount</historicalUpdateIDsCount>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Date or time input format for 'startDate', 'endDate', 'historicalStartDate' and
'historicalEndDate'. Default format: yyyy-MM-dd HH:mm:ss-->
<inputFormat</inputFormat>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: If this is true, half a year is added to the date set-->
<isFiscal>Allowed values: {
    true
    false
    0
    1
}
</isFiscal>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Begin of time interval from which values are generated. Expecting either a human
readable time format or properties. Default format: yyyy-MM-dd HH:mm:ss Examples: 2013-12-31
23:45:59 or ${someUserDefinedPropertie}-->
<startDate</startDate>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: If set to false (default), each date_n will be computed with: date_n = date_n-1
+ random . In this case 'disableRng' should be set to true!. If set to true, each date will
have a distance of [startDate, endDate]/numberOfRows to its next date.-->
<useFixedStepSize>Allowed values: {
    true
    false
    0
    1
}
</useFixedStepSize>
</gen_pdgf.generator.DetailFromDateGenerator>

```

Listing 26: XML example with all options

1.27 pdgf.generator.DictList

Randomly (using the specified distribution) picking a line from a specified dictionary file. If `disableRng` is true than lines from file are correlate with row number of table (10th row in table = value of 10th row in file. With enabled `uniquei` option you can not choose more distinct lines than available lines in source file (value of `sizei` must not be greater than available lines in specified file

1.27.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.27.2 Nodes

Name	Req.	Min/- Max count	Description	Values
disableRng		0/1	If this is true, picked lines from file correlate with row number of the currently generated row. e.g. Row 1 will pick dictionary entry 1 and in row 2 dictionary entry 2 will be picked.	true false 0 1
file	x	1/1	A path to a file to be used by a generator. See <code>pdgf.generator.DictList</code> source for a example.	
size		0/1	Number of randomly picked dictionary entries to concatenate. Default 1. Should be used in conjunction with 'separator' to define the delimiter to be used between dictionary entries. Specify the 'unique' parameter if you want to avoid that sometimes the same random line gets chosen when concatenating dictionary entries.	
unique		0/1	Specifies if picked lines should be unique. Like taking samples from an urn without replacement	true false 0 1
distribution		0/1	Distribution to be used by a generator when calculating a value	
separator		0/1	Separator string inserted between multiple randomly picked dictionary lines. Standard is a single whitespace ' '. Has only an effect if 'size' is greater 1	

1.27.3 XML-Example

```

<!--Description: Randomly (using the specified distribution) picking a line from a specified dictionary
file. If disableRng is true than lines from file are correlate with row number of table (10th row in
table = value of 10th row in file. With enabled <unique> option you can not choose more distinct
lines than available lines in source file (value of <size> must not be greater than available lines
in specified file -->
<gen.pdgf.generator.DictList
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: A path to a file to be used by a generator. See pdgf.generator.DictList source
  for a example.-->
  <file></file>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: If this is true, picked lines from file correlate with row number of the
  currently generated row. e.g. Row 1 will pick dictionary entry 1 and in row 2 dictionary
  entry 2 will be picked.-->
  <disableRng>Allowed values: {
    true
    false
    0
    1
  }
</disableRng>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Distribution to be used by a generator when calculating a value-->
<distribution></distribution>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Separator string inserted between multiple randomly picked dictionary lines.
Standard is a single whitespace ' '. Has only an effect if 'size' is greater 1-->
<separator></separator>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Number of randomly picked dictionary entries to concatenate. Default 1. Should
be used in conjunction with 'separator' to define the delimiter to be used between dictionary
entries. Specify the 'unique' parameter if you want to avoid that sometimes the same random
line gets chosen when concatenating dictionary entries.-->
<size></size>

```

```

<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Specifies if picked lines should be unique. Like taking samples from an urn
without replacement-->
<unique>Allowed values: {
    true
    false
    0
    1
}
</unique>
</gen-pdgm.generator.DictList>

```

Listing 27: XML example with all options

1.28 pdgm.generator.DoubleGenerator

Generates real double values between min and max

1.28.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.28.2 Nodes

Name	Req.	Min/- Max count	Description	Values
minD	x	1/1	minimal oputput value of this generator	
decimalPlaces		0/1	number of decimal places of output. Example: places=2 -> output: 3.64; places=3 -> 3.642. Default: noLimit	
maxD	x	1/1	maximal oputput value of this generator	
distribution		0/1	Distribution to be used by a generator when calculating a value	

1.28.3 XML-Example

```

<!--Description: Generates real double values between min and max -->
<gen-pdgm.generator.DoubleGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: maximal oputput value of this generator-->
  <maxD></maxD>
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: minimal oputput value of this generator-->
  <minD></minD>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: number of decimal places of output. Example: places=2 -> output: 3.64; places=3
  -> 3.642. Default: noLimit-->
  <decimalPlaces
    roundPlainValue="(optional) Desc: Do rounding of generated double. if decimalPlaces
    attribute is specified. Default rounding mode is: HALF.EVEN. Default for
    'roundPlainValue' is:false Allowed values: {
      true
      false
    }
  >

```

```

        0
        1
    }”
    roundingMode=”(optional) Desc: Set rounding mode. Default rounding mode is: HALF_EVEN
    Possible values: [UP, DOWN, CEILING, FLOOR, HALF_UP, HALF_DOWN, HALF_EVEN,
    UNNECESSARY] Allowed values: {
        UP
        DOWN
        CEILING
        FLOOR
        HALF_UP
        HALF_DOWN
        HALF_EVEN
        UNNECESSARY
    }”</decimalPlaces>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Distribution to be used by a generator when calculating a value-->
<distribution></distribution>
</gen_pdgf.generator.DoubleGenerator>

```

Listing 28: XML example with all options

1.29 pdgf.generator.EmailGenerator

Generates Mail addresses from the name fields

1.29.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.29.2 Nodes

Name	Req.	Min/- Max count	Description	Values
reference	x	2/2	reference	
file	x	1/1	A path to a file to be used by a generator. See pdgf.generator.DictList source for a example.	

1.29.3 XML-Example

```

<!--Description: Generates Mail addresses from the name fields -->
<gen_pdgf.generator.EmailGenerator
    id=”(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.”>
    <!--Required: [x] Executions(min|max): (2|2) -->
    <!--Description: reference-->
    <reference></reference>
    <!--Required: [x] Executions(min|max): (1|1) -->
    <!--Description: A path to a file to be used by a generator. See pdgf.generator.DictList source
    for a example.-->
    <file></file>
</gen_pdgf.generator.EmailGenerator>

```

Listing 29: XML example with all options

1.30 pdgf.generator.ExtendedFormulaGenerator

Computes result by evaluating the specified formula. Can access child generators with generator[index].

1.30.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	
listName		0/1	name of list. List will be of type pdgf.util.io.generator.NamedArrayList extends java.util.ArrayList and has a getName() method which you can use in your custom output.	
ignoreInvalidValues		0/1	tells this generator to ignore invalid values from other generators. Instead of throwing an exception, the invalid value for generator[] is to: Double.NaN	true false 0 1

1.30.2 Nodes

Name	Req.	Min/- Max count	Description	Values
gen		0/Int.max	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgf.generator.IdGenerator
staticValue		0/1	here you can specify static values to be available to use. Example: <code>staticValue<new int{1,2,3,4,5}>/staticValue<</code>	
decimalPlaces		0/1	number of decimal places of output. Example: <code>places=2 -></code> output: 3.64; <code>places=3 -></code> 3.642. Default: noLimit	
formula	x	1/1	The formula to evaluate. Example: <code>(300 * (generator[1] + 4))/ generator[0]</code>	

1.30.3 XML-Example

```
<!--Description: Computes result by evaluating the specified formula. Can access child generators with  
generator[index]. -->  
<gen_pdgf.generator.ExtendedFormulaGenerator  
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to  
    uniquely identify a field within the children of an Element."  
  ignoreInvalidValues="(optional) Desc: tells this generator to ignore invalid values from other  
    generators. Instead of throwing an exception, the invalid value for generator[] is to:  
    Double.NaN Allowed values: {  
      true  
      false  
      0  
      1  
    }"  
  listName="(optional) Desc: name of list. List will be of type  
    pdgf.util.io.generator.NamedArrayList extends java.util.ArrayList and has a getName() method  
    which you can use in your custom output.">  
<!--Required: [x] Executions(min|max): (1|1) -->  
<!--Description: The formula to evaluate. Example: (300 * ( generator[1] + 4 ))/ generator[0]-->
```

```

<formula
  unlockFullAccess="(optional) Desc: omits complexity restriction Allowed values: {
    true
    false
    0
    1
  }"></formula>
<!--Required: [ ] Executions(min|max): (0|INT.MAX) -->
<!--Description: Value Generator for this field-->
<gen_(pluginName)>Allowed values: {
  pdgf.generator.HashGenerator
  pdgf.generator.ComputeGenerator
  pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
  pdgf.generator.FormattedNumberGenerator
  pdgf.generator.ForGenerator
  pdgf.generator.MarkovChainGenerator
  pdgf.generator.CDCSequenceGenerator
  pdgf.generator.RandomUniqueStringGenerator
  pdgf.generator.FormatNumberAsDateGenerator
  pdgf.generator.SimpleFormatNumberGenerator
  pdgf.generator.BuildListGenerator
  pdgf.generator.DictList
  pdgf.generator.DoubleGenerator
  pdgf.generator.LastChoiceGenerator
  pdgf.generator.IFGenerator
  pdgf.generator.WeightedSubListGenerator
  pdgf.generator.DetailFromDateGenerator
  pdgf.generator.DateTimeGenerator
  pdgf.generator.UpdateActionTypeGenerator
  pdgf.generator.IdGenerator
  pdgf.generator.OtherFieldValueGenerator
  pdgf.generator.TemplateGenerator
  pdgf.generator.ForFormulaGenerator
  pdgf.generator.WeightedListGenerator
  pdgf.generator.AddRandomSecondsGenerator
  pdgf.generator.FormulaGenerator
  pdgf.generator.PaddingGenerator
  pdgf.generator.BigBenchReviewGenerator
  pdgf.generator.PrePostfixGenerator
  pdgf.generator.DefaultReferenceGenerator
  pdgf.generator.RandomAString
  pdgf.generator.SwitchGenerator
  pdgf.generator.ConvertNumberToStringGenerator
  pdgf.generator.ExtendedFormulaGenerator
  pdgf.generator.SetQuery
  pdgf.generator.StaticValueGenerator
  pdgf.generator.UUIDgenerator
  pdgf.generator.GlobalRowGenerator
  pdgf.generator.LongGenerator
  pdgf.generator.RelativeGenerator
  pdgf.generator.EmailGenerator
  pdgf.generator.SequentialGenerator
  pdgf.generator.GenderGenerator
  pdgf.generator.RandomListItemsGenerator
  pdgf.generator.PermutationReferenceGenerator
  pdgf.generator.RandomValueXY
  pdgf.generator.NameGenerator
  pdgf.generator.NullGenerator
  pdgf.generator.ReferenceGenerator
  pdgf.generator.UpperLowerCaseGenerator
  pdgf.generator.AbstractDecimalGenerator
  pdgf.generator.ProbabilityGenerator
  pdgf.generator.RandomSentence
}
</gen_(pluginName)>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: number of decimal places of output. Example: places=2 -> output: 3.64; places=3
-> 3.642. Default: noLimit-->
<decimalPlaces
  roundPlainValue="(optional) Desc: Do rounding of generated double. if decimalPlaces
  attribute is specified. Default rounding mode is: HALF_EVEN. Default for
  'roundPlainValue' is:false Allowed values: {
    true
    false
    0
    1

```

```

        }"
        roundingMode="(optional) Desc: Set rounding mode. Default rounding mode is: HALF_EVEN
        Possible values: [UP, DOWN, CEILING, FLOOR, HALF_UP, HALF_DOWN, HALF_EVEN,
        UNNECESSARY] Allowed values: {
            UP
            DOWN
            CEILING
            FLOOR
            HALF_UP
            HALF_DOWN
            HALF_EVEN
            UNNECESSARY
        }"</decimalPlaces>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: here you can specify static values to be available to use. Example:
    <staticValue>new int {1,2,3,4,5}</staticValue>-->
<staticValue
    unlockFullAccess="(optional) Desc: omits complexity restriction Allowed values: {
        true
        false
        0
        1
    }"</staticValue>
</gen_pdgf.generator.ExtendedFormulaGenerator>

```

Listing 30: XML example with all options

1.31 pdgf.generator.ForFormulaGenerator

Computes result by evaluating the specified formula. Can access child generators with generator[index].

1.31.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	
ignoreInvalidValues		0/1	tells this generator to ignore invalid values from other generators. Instead of throwing an exception, the invalid value for generator[] is to: Double.NaN	true false 0 1

1.31.2 Nodes

Name	Req.	Min/- Max count	Description	Values
gen		0/Int.max	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgf.generator.IdGenerator
staticValue		0/1	here you can specify static values to be available to use. Example: <code><staticValue>new int {1,2,3,4,5}</staticValue></code>	
decimalPlaces		0/1	number of decimal places of output. Example: <code>places=2</code> -> output: 3.64; <code>places=3</code> -> 3.642. Default: noLimit	
formula	x	1/1	The formula to evaluate. Example: <code>(300 * (generator[1] + 4))/ generator[0]</code>	

1.31.3 XML-Example

```
<!--Description: Computes result by evaluating the specified formula. Can access child generators with
generator[index]. -->
<gen-pdgf.generator.ForFormulaGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element."
  ignoreInvalidValues="(optional) Desc: tells this generator to ignore invalid values from other
    generators. Instead of throwing an exception, the invalid value for generator[] is to:
    Double.NaN Allowed values: {
      true
      false
      0
      1
    }">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: The formula to evaluate. Example: (300 * ( generator[1] + 4 ))/ generator[0]-->
  <formula
    unlockFullAccess="(optional) Desc: omits complexity restriction Allowed values: {
      true
      false
      0
      1
    }"></formula>
  <!--Required: [ ] Executions(min|max): (0|INT.MAX) -->
  <!--Description: Value Generator for this field-->
  <gen-(pluginName)>Allowed values: {
    pdgf.generator.HashGenerator
    pdgf.generator.ComputeGenerator
    pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
    pdgf.generator.FormattedNumberGenerator
    pdgf.generator.ForGenerator
    pdgf.generator.MarkovChainGenerator
    pdgf.generator.CDCSequenceGenerator
    pdgf.generator.RandomUniqueStringGenerator
    pdgf.generator.FormatNumberAsDateGenerator
    pdgf.generator.SimpleFormatNumberGenerator
    pdgf.generator.BuildListGenerator
    pdgf.generator.DictList
    pdgf.generator.DoubleGenerator
    pdgf.generator.LastChoiceGenerator
    pdgf.generator.IFGenerator
    pdgf.generator.WeightedSubListGenerator
    pdgf.generator.DetailFromDateGenerator
    pdgf.generator.DateTimeGenerator
    pdgf.generator.UpdateActionTypeGenerator
    pdgf.generator.IdGenerator
    pdgf.generator.OtherFieldValueGenerator
    pdgf.generator.TemplateGenerator
    pdgf.generator.ForFormulaGenerator
    pdgf.generator.WeightedListGenerator
    pdgf.generator.AddRandomSecondsGenerator
    pdgf.generator.FormulaGenerator
    pdgf.generator.PaddingGenerator
    pdgf.generator.BigBenchReviewGenerator
    pdgf.generator.PrePostfixGenerator
    pdgf.generator.DefaultReferenceGenerator
    pdgf.generator.RandomAString
    pdgf.generator.SwitchGenerator
    pdgf.generator.ConvertNumberToStringGenerator
    pdgf.generator.ExtendedFormulaGenerator
    pdgf.generator.SetQuery
    pdgf.generator.StaticValueGenerator
    pdgf.generator.UUIDgenerator
    pdgf.generator.GlobalRowGenerator
    pdgf.generator.LongGenerator
    pdgf.generator.RelativeGenerator
    pdgf.generator.EmailGenerator
    pdgf.generator.SequentialGenerator
    pdgf.generator.GenderGenerator
    pdgf.generator.RandomListItemsGenerator
    pdgf.generator.PermutationReferenceGenerator
    pdgf.generator.RandomValueXY
```



```

        pdgf.generator.NameGenerator
        pdgf.generator.NullGenerator
        pdgf.generator.ReferenceGenerator
        pdgf.generator.UpperLowerCaseGenerator
        pdgf.generator.AbstractDecimalGenerator
        pdgf.generator.ProbabilityGenerator
        pdgf.generator.RandomSentence
    }
</gen_(pluginName)>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: number of decimal places of output. Example: places=2 -> output: 3.64; places=3
-> 3.642. Default: noLimit-->
<decimalPlaces
    roundPlainValue="(optional) Desc: Do rounding of generated double. if decimalPlaces
    attribute is specified. Default rounding mode is: HALF_EVEN. Default for
    'roundPlainValue' is:false Allowed values: {
        true
        false
        0
        1
    }"
    roundingMode="(optional) Desc: Set rounding mode. Default rounding mode is: HALF_EVEN
    Possible values: [UP, DOWN, CEILING, FLOOR, HALF_UP, HALF_DOWN, HALF_EVEN,
    UNNECESSARY] Allowed values: {
        UP
        DOWN
        CEILING
        FLOOR
        HALF_UP
        HALF_DOWN
        HALF_EVEN
        UNNECESSARY
    }"></decimalPlaces>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: here you can specify static values to be available to use. Example:
<staticValue>new int {1,2,3,4,5}</staticValue>-->
<staticValue
    unlockFullAccess="(optional) Desc: omits complexity restriction Allowed values: {
        true
        false
        0
        1
    }"></staticValue>
</gen_pdgf.generator.ForFormulaGenerator>

```

Listing 31: XML example with all options

1.32 pdgf.generator.ForGenerator

Example: `jfield name="kung foo" size="50" type="VARCHAR" ;`

`igen.ForGenerator resultsToArray="true" ;`

`ij ;`

`igen.LongGenerator ;`

`jmin ; 1 / min ; jmax ; 3 / max ;`

`i / gen.LongGenerator ;`

`i / i ;`

`ido ;`

`igen.StaticValueGenerator ; jvalue ; foo i / value ; i / gen.StaticValueGenerator ;`

`i / do ;`

`i / gen.ForGenerator ;`

`i / field ;`

1.32.1 Attributes

Name	Req.	Min/- Max count	Description	Values
------	------	-----------------------	-------------	--------

resultsToList		0/1	add each iteration result into an ArrayList	true false 0 1
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	
listName		0/1	name of result list. List will be of type pdgf.util.io.generator.NamedArrayList extends java.util.ArrayList and has a getName() method which you can use in your custom output.	

1.32.2 Nodes

Name	Req.	Min/- Max count	Description	Values
i	x	1/1	Generator that specifies the iteration count	
same-ChoiceAs		0/1		
do	x	1/1	Contains generator node to n times as speciifed by <i>i</i> sub-generator	

1.32.3 XML-Example

```

<!--Description: Example: <field name="kung foo" size="50" type="VARCHAR" >
    <gen_ForGenerator resultsToArray="true">
        <i>
            <gen_LongGenerator>
                <min>1</min><max>3</max>
            </gen_LongGenerator>
        </i>
        <do>
            <gen_StaticValueGenerator<value>foo
            </value></gen_StaticValueGenerator>
        </do>
    </gen_ForGenerator>
</field> -->
<gen_pdgf.generator.ForGenerator
    id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element."
    listName="(optional) Desc: name of result list. List will be of type
    pdgf.util.io.generator.NamedArrayList extends java.util.ArrayList and has a getName() method
    which you can use in your custom output."
    resultsToList="(optional) Desc: add each iteration result into an ArrayList Allowed values: {
        true
        false
        0
        1
    }">
    <!--Required: [x] Executions(min|max): (1|1) -->
    <!--Description: Contains generator node to n times as speciifed by <i> subgenerator-->
    <do></do>
    <!--Required: [x] Executions(min|max): (1|1) -->
    <!--Description: Generator that specifies the iteration count-->
    <i></i>
    <!--Required: [ ] Executions(min|max): (0|1) -->
    <!--Description: -->
    <sameChoiceAs></sameChoiceAs>
</gen_pdgf.generator.ForGenerator>

```

Listing 32: XML example with all options

1.33 pdgf.generator.FormatNumberAsDateGenerator

Formats a number as date value

1.33.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.33.2 Nodes

Name	Req.	Min/- Max count	Description	Values
gen		0/1	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgf.generator.IdGenerator
outputFormat	x	1/1	Date or time output format. Default format: yyyy-MM-dd HH:mm:ss	

1.33.3 XML-Example

```
<!--Description: Formats a number as date value -->
<gen_pdgf.generator.FormatNumberAsDateGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: Date or time output format. Default format: yyyy-MM-dd HH:mm:ss -->
  <outputFormat></outputFormat>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Value Generator for this field -->
  <gen_(pluginName)>Allowed values: {
    pdgf.generator.HashGenerator
    pdgf.generator.ComputeGenerator
    pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
    pdgf.generator.FormattedNumberGenerator
    pdgf.generator.ForGenerator
    pdgf.generator.MarkovChainGenerator
    pdgf.generator.CDCSequenceGenerator
    pdgf.generator.RandomUniqeStringGenerator
    pdgf.generator.FormatNumberAsDateGenerator
    pdgf.generator.SimpleFormatNumberGenerator
    pdgf.generator.BuildListGenerator
    pdgf.generator.DictList
    pdgf.generator.DoubleGenerator
    pdgf.generator.LastChoiceGenerator
    pdgf.generator.IFGenerator
    pdgf.generator.WeightedSubListGenerator
    pdgf.generator.DetailFromDateGenerator
    pdgf.generator.DateTimeGenerator
    pdgf.generator.UpdateActionTypeGenerator
    pdgf.generator.IdGenerator
    pdgf.generator.OtherFieldValueGenerator
    pdgf.generator.TemplateGenerator
    pdgf.generator.ForFormulaGenerator
    pdgf.generator.WeightedListGenerator
```

```

    pdgf.generator.AddRandomSecondsGenerator
    pdgf.generator.FormulaGenerator
    pdgf.generator.PaddingGenerator
    pdgf.generator.BigBenchReviewGenerator
    pdgf.generator.PrePostfixGenerator
    pdgf.generator.DefaultReferenceGenerator
    pdgf.generator.RandomAString
    pdgf.generator.SwitchGenerator
    pdgf.generator.ConvertNumberToStringGenerator
    pdgf.generator.ExtendedFormulaGenerator
    pdgf.generator.SetQuery
    pdgf.generator.StaticValueGenerator
    pdgf.generator.UUIDgenerator
    pdgf.generator.GlobalRowGenerator
    pdgf.generator.LongGenerator
    pdgf.generator.RelativeGenerator
    pdgf.generator.EmailGenerator
    pdgf.generator.SequentialGenerator
    pdgf.generator.GenderGenerator
    pdgf.generator.RandomListItemsGenerator
    pdgf.generator.PermutationReferenceGenerator
    pdgf.generator.RandomValueXY
    pdgf.generator.NameGenerator
    pdgf.generator.NullGenerator
    pdgf.generator.ReferenceGenerator
    pdgf.generator.UpperLowerCaseGenerator
    pdgf.generator.AbstractDecimalGenerator
    pdgf.generator.ProbabilityGenerator
    pdgf.generator.RandomSentence
  }
</gen-(pluginName)>
</gen-pdgf.generator.FormatNumberAsDateGenerator>

```

Listing 33: XML example with all options

1.34 pdgf.generator.FormattedNumberGenerator

Generates a number and formats it according to the format string. Use `pdgf.generator.SimpleFormatNumberGenerator` if only `"%d"` is required

1.34.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.34.2 Nodes

Name	Req.	Min/- Max count	Description	Values
gen		0/1	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: <code>IdGenerator</code> or its full name: <code>pdgf.generator.IdGenerator</code>
format	x	1/1	Date or time format	

1.34.3 XML-Example

```
<!--Description: Generates a number and formats it according to the format string. Use
pdgf.generator.SimpleFormatNumberGenerator if only "%d" is required -->
<gen_pdgf.generator.FormattedNumberGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: Date or time format-->
  <format</format>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Value Generator for this field-->
  <gen_(pluginName)>Allowed values: {
    pdgf.generator.HashGenerator
    pdgf.generator.ComputeGenerator
    pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
    pdgf.generator.FormattedNumberGenerator
    pdgf.generator.ForGenerator
    pdgf.generator.MarkovChainGenerator
    pdgf.generator.CDCSequenceGenerator
    pdgf.generator.RandomUniqueStringGenerator
    pdgf.generator.FormatNumberAsDateGenerator
    pdgf.generator.SimpleFormatNumberGenerator
    pdgf.generator.BuildListGenerator
    pdgf.generator.DictList
    pdgf.generator.DoubleGenerator
    pdgf.generator.LastChoiceGenerator
    pdgf.generator.IFGenerator
    pdgf.generator.WeightedSubListGenerator
    pdgf.generator.DetailFromDateGenerator
    pdgf.generator.DateTimeGenerator
    pdgf.generator.UpdateActionTypeGenerator
    pdgf.generator.IdGenerator
    pdgf.generator.OtherFieldValueGenerator
    pdgf.generator.TemplateGenerator
    pdgf.generator.ForFormulaGenerator
    pdgf.generator.WeightedListGenerator
    pdgf.generator.AddRandomSecondsGenerator
    pdgf.generator.FormulaGenerator
    pdgf.generator.PaddingGenerator
    pdgf.generator.BigBenchReviewGenerator
    pdgf.generator.PrePostfixGenerator
    pdgf.generator.DefaultReferenceGenerator
    pdgf.generator.RandomAString
    pdgf.generator.SwitchGenerator
    pdgf.generator.ConvertNumberToStringGenerator
    pdgf.generator.ExtendedFormulaGenerator
    pdgf.generator.SetQuery
    pdgf.generator.StaticValueGenerator
    pdgf.generator.UUIDgenerator
    pdgf.generator.GlobalRowGenerator
    pdgf.generator.LongGenerator
    pdgf.generator.RelativeGenerator
    pdgf.generator.EmailGenerator
    pdgf.generator.SequentialGenerator
    pdgf.generator.GenderGenerator
    pdgf.generator.RandomListItemsGenerator
    pdgf.generator.PermutationReferenceGenerator
    pdgf.generator.RandomValueXY
    pdgf.generator.NameGenerator
    pdgf.generator.NullGenerator
    pdgf.generator.ReferenceGenerator
    pdgf.generator.UpperLowerCaseGenerator
    pdgf.generator.AbstractDecimalGenerator
    pdgf.generator.ProbabilityGenerator
    pdgf.generator.RandomSentence
  }
  </gen_(pluginName)>
</gen_pdgf.generator.FormattedNumberGenerator>
```

Listing 34: XML example with all options

1.35 pdgf.generator.FormulaGenerator

Computes result by evaluating the specified formula. Can access child generators with generator[index].

1.35.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	
ignoreInvalidValues		0/1	tells this generator to ignore invalid values from other generators. Instead of throwing an exception, the invalid value for generator[] is to: Double.NaN	true false 0 1

1.35.2 Nodes

Name	Req.	Min/- Max count	Description	Values
gen		0/Int.max	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgf.generator.IdGenerator
staticValue		0/1	here you can specify static values to be available to use. Example: <code>staticValue{new int{1,2,3,4,5}}/staticValue{}</code>	
decimalPlaces		0/1	number of decimal places of output. Example: <code>places=2</code> -> output: 3.64; <code>places=3</code> -> 3.642. Default: noLimit	
formula	x	1/1	The formula to evaluate. Example: <code>(300 * (generator[1] + 4))/ generator[0]</code>	

1.35.3 XML-Example

```
<!--Description: Computes result by evaluating the specified formula. Can access child generators with  
generator[index]. -->  
<gen-pdgf.generator.FormulaGenerator  
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to  
    uniquely identify a field within the children of an Element."  
  ignoreInvalidValues="(optional) Desc: tells this generator to ignore invalid values from other  
    generators. Instead of throwing an exception, the invalid value for generator[] is to:  
    Double.NaN Allowed values: {  
      true  
      false  
      0  
      1  
    }">  
  <!--Required: [x] Executions(min|max): (1|1) -->  
  <!--Description: The formula to evaluate. Example: (300 * ( generator[1] + 4 ))/ generator[0]-->  
  <formula  
    unlockFullAccess="(optional) Desc: omits complexity restriction Allowed values: {  
      true  
      false  
      0  
      1  
    }"></formula>  
  <!--Required: [ ] Executions(min|max): (0|INT.MAX) -->  
  <!--Description: Value Generator for this field-->
```

```

<gen-(pluginName)>Allowed values: {
    pdgf.generator.HashGenerator
    pdgf.generator.ComputeGenerator
    pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
    pdgf.generator.FormattedNumberGenerator
    pdgf.generator.ForGenerator
    pdgf.generator.MarkovChainGenerator
    pdgf.generator.CDCSequenceGenerator
    pdgf.generator.RandomUniqueStringGenerator
    pdgf.generator.FormatNumberAsDateGenerator
    pdgf.generator.SimpleFormatNumberGenerator
    pdgf.generator.BuildListGenerator
    pdgf.generator.DictList
    pdgf.generator.DoubleGenerator
    pdgf.generator.LastChoiceGenerator
    pdgf.generator.IFGenerator
    pdgf.generator.WeightedSubListGenerator
    pdgf.generator.DetailFromDateGenerator
    pdgf.generator.DateTimeGenerator
    pdgf.generator.UpdateActionTypeGenerator
    pdgf.generator.IdGenerator
    pdgf.generator.OtherFieldValueGenerator
    pdgf.generator.TemplateGenerator
    pdgf.generator.ForFormulaGenerator
    pdgf.generator.WeightedListGenerator
    pdgf.generator.AddRandomSecondsGenerator
    pdgf.generator.FormulaGenerator
    pdgf.generator.PaddingGenerator
    pdgf.generator.BigBenchReviewGenerator
    pdgf.generator.PrePostfixGenerator
    pdgf.generator.DefaultReferenceGenerator
    pdgf.generator.RandomAString
    pdgf.generator.SwitchGenerator
    pdgf.generator.ConvertNumberToStringGenerator
    pdgf.generator.ExtendedFormulaGenerator
    pdgf.generator.SetQuery
    pdgf.generator.StaticValueGenerator
    pdgf.generator.UUIDgenerator
    pdgf.generator.GlobalRowGenerator
    pdgf.generator.LongGenerator
    pdgf.generator.RelativeGenerator
    pdgf.generator.EmailGenerator
    pdgf.generator.SequentialGenerator
    pdgf.generator.GenderGenerator
    pdgf.generator.RandomListItemsGenerator
    pdgf.generator.PermutationReferenceGenerator
    pdgf.generator.RandomValueXY
    pdgf.generator.NameGenerator
    pdgf.generator.NullGenerator
    pdgf.generator.ReferenceGenerator
    pdgf.generator.UpperLowerCaseGenerator
    pdgf.generator.AbstractDecimalGenerator
    pdgf.generator.ProbabilityGenerator
    pdgf.generator.RandomSentence
}
</gen-(pluginName)>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: number of decimal places of output. Example: places=2 -> output: 3.64; places=3
-> 3.642. Default: noLimit-->
<decimalPlaces
    roundPlainValue="(optional) Desc: Do rounding of generated double. if decimalPlaces
        attribute is specified. Default rounding mode is: HALF_EVEN. Default for
        'roundPlainValue' is:false Allowed values: {
            true
            false
            0
            1
        }"
    roundingMode="(optional) Desc: Set rounding mode. Default rounding mode is: HALF_EVEN
        Possible values: [UP, DOWN, CEILING, FLOOR, HALF_UP, HALF_DOWN, HALF_EVEN,
        UNNECESSARY] Allowed values: {
            UP
            DOWN
            CEILING
            FLOOR
            HALF_UP

```

```

        HALF_DOWN
        HALF_EVEN
        UNNECESSARY
    }"></decimalPlaces>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: here you can specify static values to be available to use. Example:
    <staticValue>new int{1,2,3,4,5}</staticValue>-->
<staticValue
    unlockFullAccess="(optional) Desc: omits complexity restriction Allowed values: {
        true
        false
        0
        1
    }"></staticValue>
</gen_pdgf.generator.FormulaGenerator>

```

Listing 35: XML example with all options

1.36 pdgf.generator.GenderGenerator

Generates a gender flag. M for male and F for female

1.36.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.36.2 Nodes

Name	Req.	Min/- Max count	Description	Values

1.36.3 XML-Example

```

<!--Description: Generates a gender flag. M for male and F for female -->
<gen_pdgf.generator.GenderGenerator
    id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
        uniquely identify a field within the children of an Element.">
</gen_pdgf.generator.GenderGenerator>

```

Listing 36: XML example with all options

1.37 pdgf.generator.GlobalRowGenerator

This generator generates the global row over all updates.

1.37.1 Attributes

Name	Req.	Min/- Max count	Description	Values

id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	
----	--	-----	---	--

1.37.2 Nodes

Name	Req.	Min/- Max count	Description	Values
------	------	-----------------------	-------------	--------

1.37.3 XML-Example

```
<!--Description: This generator generates the global row over all updates. -->
<gen_pdgf.generator.GlobalRowGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
</gen_pdgf.generator.GlobalRowGenerator>
```

Listing 37: XML example with all options

1.38 pdgf.generator.HashGenerator

Calculates the hashcode of a value computed by a subgenerator

1.38.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.38.2 Nodes

Name	Req.	Min/- Max count	Description	Values
gen		0/1	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgf.generator.IdGenerator

digest	x	1/1	Select the message digest algorithm. Available: [SHA-1, SHA1, SHA-384, OID.1.3.14.3.2.26, 2.16.840.1.101.3.4.2.2, SHA, 2.16.840.1.101.3.4.2.1, 2.16.840.1.101.3.4.2.4, 2.16.840.1.101.3.4.2.3, OID.2.16.840.1.101.3.4.2.4, OID.2.16.840.1.101.3.4.2.3, OID.2.16.840.1.101.3.4.2.2, 1.3.14.3.2.26, OID.2.16.840.1.101.3.4.2.1, SHA-224, SHA-256, MD2, SHA-512, MD5]	SHA-1 SHA1 SHA-384 OID.1.3.14.3.2.26 2.16.840.1.101.3.4.2.2 SHA 2.16.840.1.101.3.4.2.1 2.16.840.1.101.3.4.2.4 2.16.840.1.101.3.4.2.3 OID.2.16.840.1.101.3.4.2.4 OID.2.16.840.1.101.3.4.2.3 OID.2.16.840.1.101.3.4.2.2 1.3.14.3.2.26 OID.2.16.840.1.101.3.4.2.1 SHA-224 SHA-256 MD2 SHA-512 MD5
--------	---	-----	--	--

1.38.3 XML-Example

```

<!--Description: Calculates the hashcode of a value computed by a subgenerator -->
<gen_pdgf.generator.HashGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: Select the message digest algorithm. Available: [SHA-1, SHA1, SHA-384,
    OID.1.3.14.3.2.26, 2.16.840.1.101.3.4.2.2, SHA, 2.16.840.1.101.3.4.2.1,
    2.16.840.1.101.3.4.2.4, 2.16.840.1.101.3.4.2.3, OID.2.16.840.1.101.3.4.2.4,
    OID.2.16.840.1.101.3.4.2.3, OID.2.16.840.1.101.3.4.2.2, 1.3.14.3.2.26,
    OID.2.16.840.1.101.3.4.2.1, SHA-224, SHA-256, MD2, SHA-512, MD5]-->
  <digest>Allowed values: {
    SHA-1
    SHA1
    SHA-384
    OID.1.3.14.3.2.26
    2.16.840.1.101.3.4.2.2
    SHA
    2.16.840.1.101.3.4.2.1
    2.16.840.1.101.3.4.2.4
    2.16.840.1.101.3.4.2.3
    OID.2.16.840.1.101.3.4.2.4
    OID.2.16.840.1.101.3.4.2.3
    OID.2.16.840.1.101.3.4.2.2
    1.3.14.3.2.26
    OID.2.16.840.1.101.3.4.2.1
    SHA-224
    SHA-256
    MD2
    SHA-512
    MD5
  }
</digest>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Value Generator for this field-->
  <gen_(pluginName)>Allowed values: {
    pdgf.generator.HashGenerator
    pdgf.generator.ComputeGenerator
    pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
    pdgf.generator.FormattedNumberGenerator
    pdgf.generator.ForGenerator
    pdgf.generator.MarkovChainGenerator
    pdgf.generator.CDCSequenceGenerator
    pdgf.generator.RandomUniqueStringGenerator
    pdgf.generator.FormatNumberAsDateGenerator
  }

```

```

    pdgf.generator.SimpleFormatNumberGenerator
    pdgf.generator.BuildListGenerator
    pdgf.generator.DictList
    pdgf.generator.DoubleGenerator
    pdgf.generator.LastChoiceGenerator
    pdgf.generator.IFGenerator
    pdgf.generator.WeightedSubListGenerator
    pdgf.generator.DetailFromDateGenerator
    pdgf.generator.DateTimeGenerator
    pdgf.generator.UpdateActionTypeGenerator
    pdgf.generator.IdGenerator
    pdgf.generator.OtherFieldValueGenerator
    pdgf.generator.TemplateGenerator
    pdgf.generator.ForFormulaGenerator
    pdgf.generator.WeightedListGenerator
    pdgf.generator.AddRandomSecondsGenerator
    pdgf.generator.FormulaGenerator
    pdgf.generator.PaddingGenerator
    pdgf.generator.BigBenchReviewGenerator
    pdgf.generator.PrePostfixGenerator
    pdgf.generator.DefaultReferenceGenerator
    pdgf.generator.RandomAString
    pdgf.generator.SwitchGenerator
    pdgf.generator.ConvertNumberToStringGenerator
    pdgf.generator.ExtendedFormulaGenerator
    pdgf.generator.SetQuery
    pdgf.generator.StaticValueGenerator
    pdgf.generator.UUIDgenerator
    pdgf.generator.GlobalRowGenerator
    pdgf.generator.LongGenerator
    pdgf.generator.RelativeGenerator
    pdgf.generator.EmailGenerator
    pdgf.generator.SequentialGenerator
    pdgf.generator.GenderGenerator
    pdgf.generator.RandomListItemsGenerator
    pdgf.generator.PermutationReferenceGenerator
    pdgf.generator.RandomValueXY
    pdgf.generator.NameGenerator
    pdgf.generator.NullGenerator
    pdgf.generator.ReferenceGenerator
    pdgf.generator.UpperLowerCaseGenerator
    pdgf.generator.AbstractDecimalGenerator
    pdgf.generator.ProbabilityGenerator
    pdgf.generator.RandomSentence
  }
</gen_(pluginName)>
</gen-pdgf.generator.HashGenerator>

```

Listing 38: XML example with all options

1.39 pdgf.generator.IFGenerator

Generates a value based on the output of another generator.

1.39.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.39.2 Nodes

Name	Req.	Min/- Max count	Description	Values
gen		0/Int.max	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgf.generator.IdGenerator
staticValue		0/1	here you can specify static values to be available to use. Example: <code>staticValue_i new int{1,2,3,4,5}_i/staticValue_i</code>	
else	x	1/1	Contains generator node to execute if <code>else_i</code> formula evaluates to FALSE	
then	x	1/1	Contains generator node to execute if <code>then_i</code> formula evaluates to TRUE	
if	x	1/1	The formula to evaluate. Example: <code>(300 * (generator[1] + 4))/ generator[0]</code>	

1.39.3 XML-Example

```

<!--Description: Generates a value based on the output of another generator. -->
<gen_pdgf.generator.IFGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: Contains generator node to execute if <else> formula evaluates to FALSE-->
  <else></else>
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: The formula to evaluate. Example: (300 * ( generator[1] + 4 ))/ generator[0]-->
  <if
    unlockFullAccess="(optional) Desc: override complexity restrictions Allowed values: {
      true
      false
      0
      1
    }"></if>
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: Contains generator node to execute if <then> formula evaluates to TRUE-->
  <then></then>
  <!--Required: [ ] Executions(min|max): (0|INT.MAX) -->
  <!--Description: Value Generator for this field-->
  <gen_(pluginName)>Allowed values: {
    pdgf.generator.HashGenerator
    pdgf.generator.ComputeGenerator
    pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
    pdgf.generator.FormattedNumberGenerator
    pdgf.generator.ForGenerator
    pdgf.generator.MarkovChainGenerator
    pdgf.generator.CDCSequenceGenerator
    pdgf.generator.RandomUniqueStringGenerator
    pdgf.generator.FormatNumberAsDateGenerator
    pdgf.generator.SimpleFormatNumberGenerator
    pdgf.generator.BuildListGenerator
    pdgf.generator.DictList
    pdgf.generator.DoubleGenerator
    pdgf.generator.LastChoiceGenerator
    pdgf.generator.IFGenerator
    pdgf.generator.WeightedSubListGenerator
    pdgf.generator.DetailFromDateGenerator
    pdgf.generator.DateTimeGenerator
    pdgf.generator.UpdateActionTypeGenerator
    pdgf.generator.IdGenerator
    pdgf.generator.OtherFieldValueGenerator
    pdgf.generator.TemplateGenerator
    pdgf.generator.ForFormulaGenerator
    pdgf.generator.WeightedListGenerator
    pdgf.generator.AddRandomSecondsGenerator
    pdgf.generator.FormulaGenerator

```

```

    pdgf.generator.PaddingGenerator
    pdgf.generator.BigBenchReviewGenerator
    pdgf.generator.PrePostfixGenerator
    pdgf.generator.DefaultReferenceGenerator
    pdgf.generator.RandomAString
    pdgf.generator.SwitchGenerator
    pdgf.generator.ConvertNumberToStringGenerator
    pdgf.generator.ExtendedFormulaGenerator
    pdgf.generator.SetQuery
    pdgf.generator.StaticValueGenerator
    pdgf.generator.UUIDgenerator
    pdgf.generator.GlobalRowGenerator
    pdgf.generator.LongGenerator
    pdgf.generator.RelativeGenerator
    pdgf.generator.EmailGenerator
    pdgf.generator.SequentialGenerator
    pdgf.generator.GenderGenerator
    pdgf.generator.RandomListItemsGenerator
    pdgf.generator.PermutationReferenceGenerator
    pdgf.generator.RandomValueXY
    pdgf.generator.NameGenerator
    pdgf.generator.NullGenerator
    pdgf.generator.ReferenceGenerator
    pdgf.generator.UpperLowerCaseGenerator
    pdgf.generator.AbstractDecimalGenerator
    pdgf.generator.ProbabilityGenerator
    pdgf.generator.RandomSentence
  }
</gen_(pluginName)>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: here you can specify static values to be available to use. Example:
  <staticValue>new int {1,2,3,4,5}</staticValue>-->
<staticValue
  unlockFullAccess="(optional) Desc: omits complexity restriction Allowed values: {
    true
    false
    0
    1
  }"></staticValue>
</gen_pdgf.generator.IFGenerator>

```

Listing 39: XML example with all options

1.40 pdgf.generator.IdGenerator

Simple ID generator. ID equals GenerationContext.getID(), starting at 0. Starting point can be modified with min parameter

1.40.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.40.2 Nodes

Name	Req.	Min/- Max count	Description	Values
min		0/1	start offset for ID's. Default: 0 If min = 5, generated ID's will range from [5,6,...,n]	

1.40.3 XML-Example

```
<!--Description: Simple ID generator. ID equals GenerationType.getID(), starting at 0. Starting point  
can be modified withmin parameter -->  
<gen.pdgf.generator.IdGenerator  
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to  
  uniquely identify a field within the children of an Element.">  
  <!--Required: [ ] Executions(min|max): (0|1) -->  
  <!--Description: start offset for ID's. Default: 0 If min = 5, generated ID's will range from  
  [5,6,...,n]-->  
  <min</min>  
</gen.pdgf.generator.IdGenerator>
```

Listing 40: XML example with all options

1.41 pdgf.generator.LastChoiceGenerator

Outputs the last numeric choice

1.41.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.41.2 Nodes

Name	Req.	Min/- Max count	Description	Values
reference	x	1/1	reference	

1.41.3 XML-Example

```
<!--Description: Outputs the last numeric choice -->  
<gen.pdgf.generator.LastChoiceGenerator  
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to  
  uniquely identify a field within the children of an Element.">  
  <!--Required: [x] Executions(min|max): (1|1) -->  
  <!--Description: reference -->  
  <reference</reference>  
</gen.pdgf.generator.LastChoiceGenerator>
```

Listing 41: XML example with all options

1.42 pdgf.generator.LongGenerator

Generates an 64Bit Integer value within the range [min, max] Distribution of values as specified in parameter distribution

1.42.1 Attributes

Name	Req.	Min/- Max count	Description	Values
------	------	-----------------------	-------------	--------

id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	
----	--	-----	---	--

1.42.2 Nodes

Name	Req.	Min/- Max count	Description	Values
min	x	1/1	minimum value to generate (inclusive)	
max	x	1/1	maximum value to generate (inclusive)	
distribution		0/1	Distribution to be used by a generator when calculating a value	

1.42.3 XML-Example

```

<!--Description: Generates an 64Bit Integer value within the range [min, max] Distribution of values as
specified in parameter distribution -->
<gen_pdgf.generator.LongGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: maximum value to generate (inclusive)-->
  <maxX/max>
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: minimum value to generate (inclusive)-->
  <minX/min>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Distribution to be used by a generator when calculating a value-->
  <distributionX/distribution>
</gen_pdgf.generator.LongGenerator>

```

Listing 42: XML example with all options

1.43 pdgf.generator.MarkovChainGenerator

generates samples based on an loaded chain

1.43.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.43.2 Nodes

Name	Req.	Min/- Max count	Description	Values
min	x	1/1	min sample length	

file	x	1/1		
max	x	1/1	max sample length	

1.43.3 XML-Example

```

<!--Description: generates samples based on an loaded chain -->
<gen_pdgf.generator.MarkovChainGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: -->
  <file</file>
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: max sample length-->
  <max</max>
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: min sample length-->
  <min</min>
</gen_pdgf.generator.MarkovChainGenerator>

```

Listing 43: XML example with all options

1.44 pdgf.generator.NameGenerator

First file must provide a list of Male names, second file must provide a list of female names

1.44.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.44.2 Nodes

Name	Req.	Min/- Max count	Description	Values
reference		0/1	reference	
file	x	1/1	A path to a file to be used by a generator. See pdgf.generator.DictList source for a example.	
file2	x	1/1	A path to a file to be used by a generator. See pdgf.generator.DictList source for a example.	
distribution		0/1	Distribution to be used by a generator when calculating a value	

1.44.3 XML-Example

```

<!--Description: First file must provide a list of Male names, second file must provide a list of female
names -->
<gen_pdgf.generator.NameGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element.">

```



```

<!--Required: [x] Executions(min|max): (1|1) -->
<!--Description: A path to a file to be used by a generator. See pdgf.generator.DictList source
for a example.-->
<fileX/file>
<!--Required: [x] Executions(min|max): (1|1) -->
<!--Description: A path to a file to be used by a generator. See pdgf.generator.DictList source
for a example.-->
<file2X/file2>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Distribution to be used by a generator when calculating a value-->
<distributionX/distribution>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: reference-->
<referenceX/reference>
</gen_pdgf.generator.NameGenerator>

```

Listing 44: XML example with all options

1.45 pdgf.generator.NullGenerator

Generates a NULL value with probability <probability>

1.45.1 Attributes

Name	Req.	Min/- Max count	Description	Values
probability	x	1/1	Sets the probability [0.0, 1.0] for 'NULL'	
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.45.2 Nodes

Name	Req.	Min/- Max count	Description	Values
gen		0/1	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgf.generator.IdGenerator
same-ChoiceAs		0/1	Another 'NullGenerator within the same table to pick his choice from. If specified, this generator does not decide upon 'null' on its own, but it uses the same choice as the referenced NullGenerator made	

1.45.3 XML-Example

```

<!--Description: Generates a NULL value with probability <probability> -->
<gen_pdgf.generator.NullGenerator
  probability="Desc: Sets the probability [0.0, 1.0] for 'NULL'"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element.">
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Value Generator for this field-->
<gen-(pluginName)>Allowed values: {
  pdgf.generator.HashGenerator
  pdgf.generator.ComputeGenerator

```

```

    pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
    pdgf.generator.FormattedNumberGenerator
    pdgf.generator.ForGenerator
    pdgf.generator.MarkovChainGenerator
    pdgf.generator.CDCSequenceGenerator
    pdgf.generator.RandomUniqueStringGenerator
    pdgf.generator.FormatNumberAsDateGenerator
    pdgf.generator.SimpleFormatNumberGenerator
    pdgf.generator.BuildListGenerator
    pdgf.generator.DictList
    pdgf.generator.DoubleGenerator
    pdgf.generator.LastChoiceGenerator
    pdgf.generator.IFGenerator
    pdgf.generator.WeightedSubListGenerator
    pdgf.generator.DetailFromDateGenerator
    pdgf.generator.DateTimeGenerator
    pdgf.generator.UpdateActionTypeGenerator
    pdgf.generator.IdGenerator
    pdgf.generator.OtherFieldValueGenerator
    pdgf.generator.TemplateGenerator
    pdgf.generator.ForFormulaGenerator
    pdgf.generator.WeightedListGenerator
    pdgf.generator.AddRandomSecondsGenerator
    pdgf.generator.FormulaGenerator
    pdgf.generator.PaddingGenerator
    pdgf.generator.BigBenchReviewGenerator
    pdgf.generator.PrePostfixGenerator
    pdgf.generator.DefaultReferenceGenerator
    pdgf.generator.RandomAString
    pdgf.generator.SwitchGenerator
    pdgf.generator.ConvertNumberToStringGenerator
    pdgf.generator.ExtendedFormulaGenerator
    pdgf.generator.SetQuery
    pdgf.generator.StaticValueGenerator
    pdgf.generator.UUIDgenerator
    pdgf.generator.GlobalRowGenerator
    pdgf.generator.LongGenerator
    pdgf.generator.RelativeGenerator
    pdgf.generator.EmailGenerator
    pdgf.generator.SequentialGenerator
    pdgf.generator.GenderGenerator
    pdgf.generator.RandomListItemsGenerator
    pdgf.generator.PermutationReferenceGenerator
    pdgf.generator.RandomValueXY
    pdgf.generator.NameGenerator
    pdgf.generator.NullGenerator
    pdgf.generator.ReferenceGenerator
    pdgf.generator.UpperLowerCaseGenerator
    pdgf.generator.AbstractDecimalGenerator
    pdgf.generator.ProbabilityGenerator
    pdgf.generator.RandomSentence
  }
</gen-(pluginName)>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Another 'NullGenerator' within the same table to pick his choice from. If
    specified, this generator does not decide upon 'null' on its own, but it uses the same choice
    as the referenced NullGenerator made-->
    <sameChoiceAs></sameChoiceAs>
</gen-pdgf.generator.NullGenerator>

```

Listing 45: XML example with all options

1.46 pdgf.generator.OtherFieldValueGenerator

Writes out the field value generated in another field in the same table

1.46.1 Attributes

Name	Req.	Min/- Max count	Description	Values

id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	
----	--	-----	---	--

1.46.2 Nodes

Name	Req.	Min/- Max count	Description	Values
reference	x	1/1	reference	

1.46.3 XML-Example

```

<!--Description: Writes out the field value generated in another field in the same table -->
<gen_pdgf.generator.OtherFieldValueGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: reference -->
  <reference></reference>
</gen_pdgf.generator.OtherFieldValueGenerator>

```

Listing 46: XML example with all options

1.47 pdgf.generator.PaddingGenerator

Pads the value generated by another generator to a fixed length field.

1.47.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.47.2 Nodes

Name	Req.	Min/- Max count	Description	Values
gen		0/1	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgf.generator.IdGenerator
character		0/1	Set character this generator uses for padding fields	

size		0/1	final size of value. If the sub generator of this generator produced String shorter than the specified size the value will be padded with the characters as specified in 'character'. If the generated value is to long, it will be truncated to the specified 'size'	
padToLeft		0/1	If this is true, fields are padded to the left, else to the right	

1.47.3 XML-Example

```

<!--Description: Pads the value generated by another generator to a fixed length field. -->
<gen_pdgf.generator.PaddingGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Set character this generator uses for padding fields-->
  <character></character>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Value Generator for this field-->
  <gen_(pluginName)>Allowed values: {
    pdgf.generator.HashGenerator
    pdgf.generator.ComputeGenerator
    pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
    pdgf.generator.FormattedNumberGenerator
    pdgf.generator.ForGenerator
    pdgf.generator.MarkovChainGenerator
    pdgf.generator.CDCSequenceGenerator
    pdgf.generator.RandomUniqeStringGenerator
    pdgf.generator.FormatNumberAsDateGenerator
    pdgf.generator.SimpleFormatNumberGenerator
    pdgf.generator.BuildListGenerator
    pdgf.generator.DictList
    pdgf.generator.DoubleGenerator
    pdgf.generator.LastChoiceGenerator
    pdgf.generator.IFGenerator
    pdgf.generator.WeightedSubListGenerator
    pdgf.generator.DetailFromDateGenerator
    pdgf.generator.DateTimeGenerator
    pdgf.generator.UpdateActionTypeGenerator
    pdgf.generator.IdGenerator
    pdgf.generator.OtherFieldValueGenerator
    pdgf.generator.TemplateGenerator
    pdgf.generator.ForFormulaGenerator
    pdgf.generator.WeightedListGenerator
    pdgf.generator.AddRandomSecondsGenerator
    pdgf.generator.FormulaGenerator
    pdgf.generator.PaddingGenerator
    pdgf.generator.BigBenchReviewGenerator
    pdgf.generator.PrePostfixGenerator
    pdgf.generator.DefaultReferenceGenerator
    pdgf.generator.RandomAString
    pdgf.generator.SwitchGenerator
    pdgf.generator.ConvertNumberToStringGenerator
    pdgf.generator.ExtendedFormulaGenerator
    pdgf.generator.SetQuery
    pdgf.generator.StaticValueGenerator
    pdgf.generator.UUIDgenerator
    pdgf.generator.GlobalRowGenerator
    pdgf.generator.LongGenerator
    pdgf.generator.RelativeGenerator
    pdgf.generator.EmailGenerator
    pdgf.generator.SequentialGenerator
    pdgf.generator.GenderGenerator
    pdgf.generator.RandomListItemsGenerator
    pdgf.generator.PermutationReferenceGenerator
    pdgf.generator.RandomValueXY
    pdgf.generator.NameGenerator
    pdgf.generator.NullGenerator
    pdgf.generator.ReferenceGenerator
    pdgf.generator.UpperLowerCaseGenerator
  }

```

```

    pdgf.generator.AbstractDecimalGenerator
    pdgf.generator.ProbabilityGenerator
    pdgf.generator.RandomSentence
  }
</gen-(pluginName)>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: If this is true, fields are padded to the left, else to the right-->
<padToLeft</padToLeft>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: final size of value. If the sub generator of this generator produced String
    shorter than the specified size the value will be padded with the characters as specified in
    'character'. If the generated value is to long, it will be truncated to the specified
    'size'-->
    <size</size>
</gen-pdgf.generator.PaddingGenerator>

```

Listing 47: XML example with all options

1.48 pdgf.generator.PermutationReferenceGenerator

Generates values for referencing field by following the reference and randomly (following the specified distribution) picking an ID from the value set of valid ID's from the referenced Table and recalculating its Value. Default updateIDOffset = 0. Default: chose reference from SAME updateID. This may skew time dependencies as it might chooses a reference to an ID yet to be created in this generation. To prevent this, use an updateIDOffset of -1 to choose reference from previous generation of referenced Table.

1.48.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.48.2 Nodes

Name	Req.	Min/- Max count	Description	Values
reference	x	1/1	reference	
same-ChoiceAs		0/1	Requires a <code>jfield_i</code> and a <code>jgeneratorByID_i</code> node (in same Table) to pick the row number from. If specified this Default-ReferenceGenerator does not choose a random row, but it uses the same row as the DefaultReferenceGenerator of the specified field/generator.	
updateID-Offset		0/1	Default: chose reference from SAME updateID. This may skew time dependencies as it might choose a reference to an ID yet to be created in this generation. To prevent this, use an updateIDOffset of -1 to choose reference from previous generation in referenced Table. Default offset is 0.	
override-Bound-sCheck		0/1	disables check, if to much unique samples are requeste form this permuation based reference generator. If the permuations 'wraps' an error is thrown.	true false 0 1
distribution		0/1	Distribution to be used by a generator when calculating a value	

includeDeletedIDs		0/1	when generating references to an update table, also reference already deleted IDs.	true false 0 1
useRelativeUpdateIDs		0/1	Use updateIDs relative to the number of updates of the parent table for reference access	true false 0 1

1.48.3 XML-Example

```

<!--Description: Generates values for referencing field by following the reference and randomly
(following the specified distribution) picking an ID from the value set of valid ID's from the
referenced Table and recalculating its Value. Default updateIDOffset = 0. Default: chose reference
from SAME updateID. This may skew time dependencies as it might chooses a reference to an ID yet to
be created in this generation. To prevent this, use an updateIDOffset of -1 to choose reference from
previous generation of referenced Table. -->
<gen_pdgf.generator.PermutationReferenceGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: reference-->
  <reference></reference>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Distribution to be used by a generator when calculating a value-->
  <distribution></distribution>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: when generating references to an update table, also reference already deleted
  IDs.-->
  <includeDeletedIDs>Allowed values: {
    true
    false
    0
    1
  }
</includeDeletedIDs>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: disables check, if to much unique samples are requeste form this permutation
  based reference generator. If the permutations 'wraps' an error is thrown.-->
  <overrideBoundsCheck>Allowed values: {
    true
    false
    0
    1
  }
</overrideBoundsCheck>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Requires a <field> and a <generatorByID> node ( in same Table) to pick the row
  number from. If specified this DefaultReferenceGenerator does not choose a random row, but it
  uses the same row as the DefaultReferenceGenerator of the specified field/generator.-->
  <sameChoiceAs></sameChoiceAs>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Default: chose reference from SAME updateID. This may skew time dependencies as
  it might choose a reference to an ID yet to be created in this generation. To prevent this,
  use an updateIDOffset of -1 to choose reference from previous generation in referenced Table.
  Default offset is 0.-->
  <updateIDOffset></updateIDOffset>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Use updateIDs relative to the number of updates of the parent table for
  reference access-->
  <useRelativeUpdateIDs>Allowed values: {
    true
    false
    0
    1
  }
</useRelativeUpdateIDs>
</gen_pdgf.generator.PermutationReferenceGenerator>

```

1.49 pdgf.generator.PrePostfixGenerator

Pre- or postfixes a generated value with a static string

1.49.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.49.2 Nodes

Name	Req.	Min/- Max count	Description	Values
gen		0/1	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgf.generator.IdGenerator
prefix		0/1	Sets the prefix	
postfix		0/1	Sets the postfix	

1.49.3 XML-Example

```
<!--Description: Pre- or postfixes a generated value with a static string -->
<gen_pdgf.generator.PrePostfixGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Value Generator for this field-->
  <gen-(pluginName)>Allowed values: {
    pdgf.generator.HashGenerator
    pdgf.generator.ComputeGenerator
    pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
    pdgf.generator.FormattedNumberGenerator
    pdgf.generator.ForGenerator
    pdgf.generator.MarkovChainGenerator
    pdgf.generator.CDCSequenceGenerator
    pdgf.generator.RandomUniqueStringGenerator
    pdgf.generator.FormatNumberAsDateGenerator
    pdgf.generator.SimpleFormatNumberGenerator
    pdgf.generator.BuildListGenerator
    pdgf.generator.DictList
    pdgf.generator.DoubleGenerator
    pdgf.generator.LastChoiceGenerator
    pdgf.generator.IFGenerator
    pdgf.generator.WeightedSubListGenerator
    pdgf.generator.DetailFromDateGenerator
    pdgf.generator.DateTimeGenerator
    pdgf.generator.UpdateActionTypeGenerator
    pdgf.generator.IdGenerator
    pdgf.generator.OtherFieldValueGenerator
```

```

    pdgf.generator.TemplateGenerator
    pdgf.generator.ForFormulaGenerator
    pdgf.generator.WeightedListGenerator
    pdgf.generator.AddRandomSecondsGenerator
    pdgf.generator.FormulaGenerator
    pdgf.generator.PaddingGenerator
    pdgf.generator.BigBenchReviewGenerator
    pdgf.generator.PrePostfixGenerator
    pdgf.generator.DefaultReferenceGenerator
    pdgf.generator.RandomAString
    pdgf.generator.SwitchGenerator
    pdgf.generator.ConvertNumberToStringGenerator
    pdgf.generator.ExtendedFormulaGenerator
    pdgf.generator.SetQuery
    pdgf.generator.StaticValueGenerator
    pdgf.generator.UUIDgenerator
    pdgf.generator.GlobalRowGenerator
    pdgf.generator.LongGenerator
    pdgf.generator.RelativeGenerator
    pdgf.generator.EmailGenerator
    pdgf.generator.SequentialGenerator
    pdgf.generator.GenderGenerator
    pdgf.generator.RandomListItemsGenerator
    pdgf.generator.PermutationReferenceGenerator
    pdgf.generator.RandomValueXY
    pdgf.generator.NameGenerator
    pdgf.generator.NullGenerator
    pdgf.generator.ReferenceGenerator
    pdgf.generator.UpperLowerCaseGenerator
    pdgf.generator.AbstractDecimalGenerator
    pdgf.generator.ProbabilityGenerator
    pdgf.generator.RandomSentence
  }
</gen_(pluginName)>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Sets the postfix-->
<postfix>/postfix>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Sets the prefix-->
<prefix>/prefix>
</gen_pdgf.generator.PrePostfixGenerator>

```

Listing 49: XML example with all options

1.50 pdgf.generator.ProbabilityGenerator

Runs generators with a certain probability

1.50.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.50.2 Nodes

Name	Req.	Min/- Max count	Description	Values
------	------	-----------------------	-------------	--------

disableRng		0/1	If this is true, random mixing of values will be disabled. The values will be generated in the order specified in the xml (in relation to the current ID).	true false 0 1
chunkSize		0/1	Sets a chunk size (useful when disableRng is used)	
probability	x	1/1	Run given generator with the given probability	
same-ChoiceAs		0/1	Uses the same choice as the given ProbabilityGenerator	

1.50.3 XML-Example

```

<!--Description: Runs generators with a certain probability -->
<gen_pdgf.generator.ProbabilityGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: Run given generator with the given probability -->
  <probability></probability>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Sets a chunk size (useful when disableRng is used)-->
  <chunkSize></chunkSize>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: If this is true, random mixing of values will be disabled. The values will be
  generated in the order specified in the xml (in relation to the current ID).-->
  <disableRng>Allowed values: {
    true
    false
    0
    1
  }
</disableRng>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Uses the same choice as the given ProbabilityGenerator -->
  <sameChoiceAs></sameChoiceAs>
</gen_pdgf.generator.ProbabilityGenerator>

```

Listing 50: XML example with all options

1.51 pdgf.generator.RandomAString

Generates random string with a random size between [min, max] using either the default character set: a-zA-Z0-9 or a custom set as specified with the 'characters' parameter

1.51.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.51.2 Nodes

Name	Req.	Min/- Max count	Description	Values
------	------	-----------------------	-------------	--------

characters		0/1	Sets characters this generator uses for generating a random string	
disableRng		0/1	If this is true, picked lines from file correlate with table line numbers	
min		0/1	minimum number of characters to generate (inclusive). Default: 0	
max	x	1/1	maximum number of characters to generate (inclusive)	

1.51.3 XML-Example

```

<!--Description: Generates random string with a random size between [min, max] using either the default
character set: a-zA-Z0-9 or a custom set as specified with the 'characters' parameter -->
<gen_pdgf.generator.RandomAString
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: maximum number of characters to generate (inclusive)-->
  <max</max>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Sets characters this generator uses for generating a random string-->
  <characters</characters>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: If this is true, picked lines from file correlate with table line numbers-->
  <disableRng</disableRng>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: minimum number of characters to generate (inclusive). Default: 0-->
  <min</min>
</gen_pdgf.generator.RandomAString>

```

Listing 51: XML example with all options

1.52 pdgf.generator.RandomListItemsGenerator

Generates a new list with random elements from the list generated by a subgenerator.

1.52.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	
listName		0/1	name of list. List will be of type pdgf.util.io.generator.NamedArrayList extends java.util.ArrayList and has a getName() method which you can use in your custom output.	

1.52.2 Nodes

Name	Req.	Min/- Max count	Description	Values
------	------	-----------------------	-------------	--------

gen	x	1/1	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgf.generator.IdGenerator
same-ChoiceAs		0/1	Another 'RandomListItemsGenerator within the same table to pick his choice from. If specified, this generator does not decide upon the picked list elements itself, but it uses the same choice as the referenced RandomListItemsGenerator made	

1.52.3 XML-Example

```

<!--Description: Generates a new list with random elements from the list generated by a subgenerator. -->
<gen.pdgf.generator.RandomListItemsGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element."
  listName="(optional) Desc: name of list. List will be of type
    pdgf.util.io.generator.NamedArrayList extends java.util.ArrayList and has a getName() method
    which you can use in your custom output.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: Value Generator for this field-->
  <gen-(pluginName)>Allowed values: {
    pdgf.generator.HashGenerator
    pdgf.generator.ComputeGenerator
    pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
    pdgf.generator.FormattedNumberGenerator
    pdgf.generator.ForGenerator
    pdgf.generator.MarkovChainGenerator
    pdgf.generator.CDCSequenceGenerator
    pdgf.generator.RandomUniqueStringGenerator
    pdgf.generator.FormatNumberAsDateGenerator
    pdgf.generator.SimpleFormatNumberGenerator
    pdgf.generator.BuildListGenerator
    pdgf.generator.DictList
    pdgf.generator.DoubleGenerator
    pdgf.generator.LastChoiceGenerator
    pdgf.generator.IFGenerator
    pdgf.generator.WeightedSubListGenerator
    pdgf.generator.DetailFromDateGenerator
    pdgf.generator.DateTimeGenerator
    pdgf.generator.UpdateActionTypeGenerator
    pdgf.generator.IdGenerator
    pdgf.generator.OtherFieldValueGenerator
    pdgf.generator.TemplateGenerator
    pdgf.generator.ForFormulaGenerator
    pdgf.generator.WeightedListGenerator
    pdgf.generator.AddRandomSecondsGenerator
    pdgf.generator.FormulaGenerator
    pdgf.generator.PaddingGenerator
    pdgf.generator.BigBenchReviewGenerator
    pdgf.generator.PrePostfixGenerator
    pdgf.generator.DefaultReferenceGenerator
    pdgf.generator.RandomAString
    pdgf.generator.SwitchGenerator
    pdgf.generator.ConvertNumberToStringGenerator
    pdgf.generator.ExtendedFormulaGenerator
    pdgf.generator.SetQuery
    pdgf.generator.StaticValueGenerator
    pdgf.generator.UUIDgenerator
    pdgf.generator.GlobalRowGenerator
    pdgf.generator.LongGenerator
    pdgf.generator.RelativeGenerator
    pdgf.generator.EmailGenerator
    pdgf.generator.SequentialGenerator
    pdgf.generator.GenderGenerator
    pdgf.generator.RandomListItemsGenerator
    pdgf.generator.PermutationReferenceGenerator
  }

```

```

        pdgf.generator.RandomValueXY
        pdgf.generator.NameGenerator
        pdgf.generator.NullGenerator
        pdgf.generator.ReferenceGenerator
        pdgf.generator.UpperLowerCaseGenerator
        pdgf.generator.AbstractDecimalGenerator
        pdgf.generator.ProbabilityGenerator
        pdgf.generator.RandomSentence
    }
</gen-(pluginName)>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Another 'RandomListItemsGenerator' within the same table to pick his choice from.
    If specified, this generator does not decide upon the picked list elements itself, but it
    uses the same choice as the referenced RandomListItemsGenerator made-->
    <sameChoiceAs></sameChoiceAs>
</gen-pdgf.generator.RandomListItemsGenerator>

```

Listing 52: XML example with all options

1.53 pdgf.generator.RandomSentence

Generates an text based on a pseudo text grammar. The length of the string is varied with a random from min to max. The output of the pseudo text grammar is truncated at the selected length.

1.53.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.53.2 Nodes

Name	Req.	Min/- Max count	Description	Values
min	x	1/1	minimum string length to generate (inclusive)	
max	x	1/1	maximum string length to generate (inclusive)	
complaints		0/1		
recommends		0/1		

1.53.3 XML-Example

```

<!--Description: Generates an text based on a pseudo text grammar. The length of the string is varied
    with a random from min to max. The output of the pseudo text grammar is truncated at the selected
    length. -->
<gen-pdgf.generator.RandomSentence
    id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
    <!--Required: [x] Executions(min|max): (1|1) -->
    <!--Description: maximum string length to generate (inclusive)-->
    <max></max>
    <!--Required: [x] Executions(min|max): (1|1) -->
    <!--Description: minimum string length to generate (inclusive)-->
    <min></min>
    <!--Required: [ ] Executions(min|max): (0|1) -->
    <!--Description: -->
    <complaints></complaints>

```

```

<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: -->
<recommends></recommends>
</gen_pdgf.generator.RandomSentence>

```

Listing 53: XML example with all options

1.54 pdgf.generator.RandomUniqueStringGenerator

Generates random unique string with a specified length using either the default character set: a-zA-Z0-9 or a custom set as specified with the 'characters' parameter. The difference between this generator and the 'pdgf.generator.ConvertNumberToStringGenerator' is: pdgf.generator.ConvertNumberToStringGenerator generates a unique string for each number. But number 0 would produce AAAAA and number 1 would produce AAAAB, and so on. Where this generator would produce e.g. ST4GA for 0 and 8GHS3 for 1.

1.54.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.54.2 Nodes

Name	Req.	Min/- Max count	Description	Values
gen		0/1	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgf.generator.IdGenerator
characters		0/1	Sets characters this generator uses for generating a random string. Default: [a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, v, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9]	
length	x	1/1	length of string to generate	

1.54.3 XML-Example

```

<!--Description: Generates random unique string with a specified length using either the default
character set: a-zA-Z0-9 or a custom set as specified with the 'characters' parameter. The difference
between this generator and the 'pdgf.generator.ConvertNumberToStringGenerator' is:
pdgf.generator.ConvertNumberToStringGenerator generates a unique string for each number. But number 0
would produce AAAAA and number 1 would produce AAAAB, and so on. Where this generator would produce
e.g. ST4GA for 0 and 8GHS3 for 1. -->
<gen_pdgf.generator.RandomUniqueStringGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: length of string to generate-->
  <length></length>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Sets characters this generator uses for generating a random string. Default: [a,
    b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, v, A, B, C, D, E,

```

```

F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, 0, 1, 2, 3, 4, 5, 6, 7, 8,
9]-->
<characters</characters>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Value Generator for this field-->
<gen-(pluginName)>Allowed values: {
    pdgf.generator.HashGenerator
    pdgf.generator.ComputeGenerator
    pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
    pdgf.generator.FormattedNumberGenerator
    pdgf.generator.ForGenerator
    pdgf.generator.MarkovChainGenerator
    pdgf.generator.CDCSequenceGenerator
    pdgf.generator.RandomUniqueStringGenerator
    pdgf.generator.FormatNumberAsDateGenerator
    pdgf.generator.SimpleFormatNumberGenerator
    pdgf.generator.BuildListGenerator
    pdgf.generator.DictList
    pdgf.generator.DoubleGenerator
    pdgf.generator.LastChoiceGenerator
    pdgf.generator.IFGenerator
    pdgf.generator.WeightedSubListGenerator
    pdgf.generator.DetailFromDateGenerator
    pdgf.generator.DateTimeGenerator
    pdgf.generator.UpdateActionTypeGenerator
    pdgf.generator.IdGenerator
    pdgf.generator.OtherFieldValueGenerator
    pdgf.generator.TemplateGenerator
    pdgf.generator.ForFormulaGenerator
    pdgf.generator.WeightedListGenerator
    pdgf.generator.AddRandomSecondsGenerator
    pdgf.generator.FormulaGenerator
    pdgf.generator.PaddingGenerator
    pdgf.generator.BigBenchReviewGenerator
    pdgf.generator.PrePostfixGenerator
    pdgf.generator.DefaultReferenceGenerator
    pdgf.generator.RandomAString
    pdgf.generator.SwitchGenerator
    pdgf.generator.ConvertNumberToStringGenerator
    pdgf.generator.ExtendedFormulaGenerator
    pdgf.generator.SetQuery
    pdgf.generator.StaticValueGenerator
    pdgf.generator.UUIDgenerator
    pdgf.generator.GlobalRowGenerator
    pdgf.generator.LongGenerator
    pdgf.generator.RelativeGenerator
    pdgf.generator.EmailGenerator
    pdgf.generator.SequentialGenerator
    pdgf.generator.GenderGenerator
    pdgf.generator.RandomListItemsGenerator
    pdgf.generator.PermutationReferenceGenerator
    pdgf.generator.RandomValueXY
    pdgf.generator.NameGenerator
    pdgf.generator.NullGenerator
    pdgf.generator.ReferenceGenerator
    pdgf.generator.UpperLowerCaseGenerator
    pdgf.generator.AbstractDecimalGenerator
    pdgf.generator.ProbabilityGenerator
    pdgf.generator.RandomSentence
}
</gen-(pluginName)>
</gen-pdgf.generator.RandomUniqueStringGenerator>

```

Listing 54: XML example with all options

1.55 pdgf.generator.RandomValueXY

The notation random value [x .. y] represents a random value between x and y inclusively, with a mean of $(x+y)/2$, and with the same number of digits of precision as shown. For example, [0.01 .. 100.00] has 10,000 unique values, whereas [1..100] has only 100 unique values.

1.55.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.55.2 Nodes

Name	Req.	Min/- Max count	Description	Values
x	x	1/1	X part of [x .. y] represents a random value between x and y inclusively, with a mean of (x+y)/2, and with the same number of digits of precision as shown. For example, [0.01 .. 100.00] has 10,000 unique values	
y	x	1/1	Y part of [x .. y] represents a random value between x and y inclusively, with a mean of (x+y)/2, and with the same number of digits of precision as shown. For example, [0.01 .. 100.00] has 10,000 unique values	

1.55.3 XML-Example

```

<!--Description: The notation random value [x .. y] represents a random value between x and y
    inclusively , with a mean of (x+y)/2, and with the same number of digits of precision as shown. For
    example, [0.01 .. 100.00] has 10,000 unique values , whereas [1..100] has only 100 unique values. -->
<gen-pdgm.generator.RandomValueXY
    id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
        uniquely identify a field within the children of an Element.">
    <!--Required: [x] Executions(min|max): (1|1) -->
    <!--Description: X part of [x .. y] represents a random value between x and y inclusively , with a
        mean of (x+y)/2, and with the same number of digits of precision as shown. For example, [0.01
        .. 100.00] has 10,000 unique values-->
    <x>/x>
    <!--Required: [x] Executions(min|max): (1|1) -->
    <!--Description: Y part of [x .. y] represents a random value between x and y inclusively , with a
        mean of (x+y)/2, and with the same number of digits of precision as shown. For example, [0.01
        .. 100.00] has 10,000 unique values-->
    <y>/y>
</gen-pdgm.generator.RandomValueXY>

```

Listing 55: XML example with all options

1.56 pdgm.generator.ReferenceGenerator

Generates values for referencing field by following the reference and randomly (following the specified distribution) picking a row from the value set (rows) of the referenced Table and recalculating its Value. Default updateIDOffset = 0

1.56.1 Attributes

Name	Req.	Min/- Max count	Description	Values
updateID-MAX		0/1	Maximal chosen updateID this Generator is going to use relatively to the current UpdateID in GenerationContext.getUpdateID() (or Default max is 'last updateID in table')	

choose	x	1/1	supported modes: [random, relativeRowMapping, relativeUnique_withoutDeletedInNextUpdate, directRowMapping, globalUnique, sameChoiceAs, intraTupel, relativeUnique]	random relativeRowMapping relativeUnique_withoutDeletedInNextUpdate directRowMapping globalUnique sameChoiceAs intraTupel relativeUnique
from	x	1/1	supported modes: [fixedTimeFrame, atInsert, historical, relativeTimeFrame, sameTimeFrame]	fixedTimeFrame atInsert historical relativeTimeFrame sameTimeFrame
updateID-Offset		0/1	Offset for UpdateID this this Generator is going to use. e.g. If fixed time frame mode is used: refUpdateID = offset + gc.updateID. (Default offset is 0)	
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	
includeDeletedIDs		0/1	when generating references to an update table, also include already deleted IDs.	true false 0 1

1.56.2 Nodes

Name	Req.	Min/- Max count	Description	Values
reference	x	1/1	reference	
gen		0/1	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgf.generator.IdGenerator
same-ChoiceAs		0/1	Requires a <code>jfield</code> and a <code>jgeneratorById</code> node (in same Table) to pick the row number from. If specified this ReferenceGenerator does not choose a random row, but it uses the same row as the ReferenceGenerator of the specified field/generator.	
distribution		0/1	Distribution to be used by a generator when calculating a value	

1.56.3 XML-Example

```

<!--Description: Generates values for referencing field by following the reference and randomly
(following the specified distribution) picking a row from the value set (rows) of the referenced
Table and recalculating its Value. Default updateIDOffset = 0 -->
<gen_pdgf.generator.ReferenceGenerator
  choose="Desc: supported modes: [random, relativeRowMapping,
    relativeUnique_withoutDeletedInNextUpdate, directRowMapping, globalUnique, sameChoiceAs,
    intraTupel, relativeUnique] Allowed values: {
    random
    relativeRowMapping
    relativeUnique_withoutDeletedInNextUpdate
    directRowMapping

```



```

        globalUnique
        sameChoiceAs
        intraTupel
        relativeUnique
    }"
from="Desc: supported modes: [fixedTimeFrame, atInsert, historical, relativeTimeFrame,
    sameTimeFrame] Allowed values: {
    fixedTimeFrame
    atInsert
    historical
    relativeTimeFrame
    sameTimeFrame
}"
id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element."
includeDeletedIDs="(optional) Desc: when generating references to an update table, also include
    already deleted IDs. Allowed values: {
    true
    false
    0
    1
    }"
updateIDMAX="(optional) Desc: Maximal chosen updateID this Generator is going to use relatively
    to the current UpdateID in GenerationContext.getUpdateID() (or Default max is 'last updateID
    in table '")
updateIDOffset="(optional) Desc: Offset for UpdateID this this Generator is going to use. e.g.
    If fixed time frame mode is used: refUpdateID = offset + gc.updateID. (Default offset is 0)">
<!--Required: [x] Executions(min|max): (1|1) -->
<!--Description: reference-->
<reference></reference>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Distribution to be used by a generator when calculating a value-->
<distribution></distribution>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Value Generator for this field-->
<gen-(pluginName)>Allowed values: {
    pdgf.generator.HashGenerator
    pdgf.generator.ComputeGenerator
    pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
    pdgf.generator.FormattedNumberGenerator
    pdgf.generator.ForGenerator
    pdgf.generator.MarkovChainGenerator
    pdgf.generator.CDCSequenceGenerator
    pdgf.generator.RandomUniqueStringGenerator
    pdgf.generator.FormatNumberAsDateGenerator
    pdgf.generator.SimpleFormatNumberGenerator
    pdgf.generator.BuildListGenerator
    pdgf.generator.DictList
    pdgf.generator.DoubleGenerator
    pdgf.generator.LastChoiceGenerator
    pdgf.generator.IFGenerator
    pdgf.generator.WeightedSubListGenerator
    pdgf.generator.DetailFromDateGenerator
    pdgf.generator.DateTimeGenerator
    pdgf.generator.UpdateActionTypeGenerator
    pdgf.generator.IdGenerator
    pdgf.generator.OtherFieldValueGenerator
    pdgf.generator.TemplateGenerator
    pdgf.generator.ForFormulaGenerator
    pdgf.generator.WeightedListGenerator
    pdgf.generator.AddRandomSecondsGenerator
    pdgf.generator.FormulaGenerator
    pdgf.generator.PaddingGenerator
    pdgf.generator.BigBenchReviewGenerator
    pdgf.generator.PrePostfixGenerator
    pdgf.generator.DefaultReferenceGenerator
    pdgf.generator.RandomAString
    pdgf.generator.SwitchGenerator
    pdgf.generator.ConvertNumberToStringGenerator
    pdgf.generator.ExtendedFormulaGenerator
    pdgf.generator.SetQuery
    pdgf.generator.StaticValueGenerator
    pdgf.generator.UUIDgenerator
    pdgf.generator.GlobalRowGenerator
    pdgf.generator.LongGenerator
    pdgf.generator.RelativeGenerator

```

```

    pdgf.generator.EmailGenerator
    pdgf.generator.SequentialGenerator
    pdgf.generator.GenderGenerator
    pdgf.generator.RandomListItemsGenerator
    pdgf.generator.PermutationReferenceGenerator
    pdgf.generator.RandomValueXY
    pdgf.generator.NameGenerator
    pdgf.generator.NullGenerator
    pdgf.generator.ReferenceGenerator
    pdgf.generator.UpperLowerCaseGenerator
    pdgf.generator.AbstractDecimalGenerator
    pdgf.generator.ProbabilityGenerator
    pdgf.generator.RandomSentence
  }
</gen_(pluginName)>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Requires a <field> and a <generatorByID> node ( in same Table) to pick the row
    number from. If specified this ReferenceGenerator does not choose a random row, but it uses
    the same row as the ReferenceGenerator of the specified field/generator.-->
    <sameChoiceAs></sameChoiceAs>
</gen_pdgf.generator.ReferenceGenerator>

```

Listing 56: XML example with all options

1.57 pdgf.generator.RelativeGenerator

Generates a target value relative to the size of the source table. $result = (\text{long}) \text{Math.floor}(\text{staticFactor} * \text{relativeFactor} * \text{generatedDouble})$. If not explicitly specified, every factor has a default value of 1.0

1.57.1 Attributes

Name	Req.	Min/- Max count	Description	Values
targetSize		0/1	Sets the size of the target table	
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	
sourceSize		0/1	Sets the size of the source table	
correction-Factor		0/1	Sets a static correction factor	

1.57.2 Nodes

Name	Req.	Min/- Max count	Description	Values
gen	x	1/1	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgf.generator.IdGenerator

1.57.3 XML-Example

```

<!--Description: Generates a target value relative to the size of the source table. result= (long)
    Math.floor( staticFactor * relativeFactor * generatedDouble). If not explicitly specified, every
    factor has a default value of 1.0 -->
<gen_pdgf.generator.RelativeGenerator

```

```

correctionFactor="(optional) Desc: Sets a static correction factor"
id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element."
sourceSize="(optional) Desc: Sets the size of the source table"
targetSize="(optional) Desc: Sets the size of the target table">
<!--Required: [x] Executions(min|max): (1|1) -->
<!--Description: Value Generator for this field-->
<gen_(pluginName)>Allowed values: {
    pdgf.generator.HashGenerator
    pdgf.generator.ComputeGenerator
    pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
    pdgf.generator.FormattedNumberGenerator
    pdgf.generator.ForGenerator
    pdgf.generator.MarkovChainGenerator
    pdgf.generator.CDCSequenceGenerator
    pdgf.generator.RandomUniqueStringGenerator
    pdgf.generator.FormatNumberAsDateGenerator
    pdgf.generator.SimpleFormatNumberGenerator
    pdgf.generator.BuildListGenerator
    pdgf.generator.DictList
    pdgf.generator.DoubleGenerator
    pdgf.generator.LastChoiceGenerator
    pdgf.generator.IFGenerator
    pdgf.generator.WeightedSubListGenerator
    pdgf.generator.DetailFromDateGenerator
    pdgf.generator.DateTimeGenerator
    pdgf.generator.UpdateActionTypeGenerator
    pdgf.generator.IdGenerator
    pdgf.generator.OtherFieldValueGenerator
    pdgf.generator.TemplateGenerator
    pdgf.generator.ForFormulaGenerator
    pdgf.generator.WeightedListGenerator
    pdgf.generator.AddRandomSecondsGenerator
    pdgf.generator.FormulaGenerator
    pdgf.generator.PaddingGenerator
    pdgf.generator.BigBenchReviewGenerator
    pdgf.generator.PrePostfixGenerator
    pdgf.generator.DefaultReferenceGenerator
    pdgf.generator.RandomAString
    pdgf.generator.SwitchGenerator
    pdgf.generator.ConvertNumberToStringGenerator
    pdgf.generator.ExtendedFormulaGenerator
    pdgf.generator.SetQuery
    pdgf.generator.StaticValueGenerator
    pdgf.generator.UUIDgenerator
    pdgf.generator.GlobalRowGenerator
    pdgf.generator.LongGenerator
    pdgf.generator.RelativeGenerator
    pdgf.generator.EmailGenerator
    pdgf.generator.SequentialGenerator
    pdgf.generator.GenderGenerator
    pdgf.generator.RandomListItemsGenerator
    pdgf.generator.PermutationReferenceGenerator
    pdgf.generator.RandomValueXY
    pdgf.generator.NameGenerator
    pdgf.generator.NullGenerator
    pdgf.generator.ReferenceGenerator
    pdgf.generator.UpperLowerCaseGenerator
    pdgf.generator.AbstractDecimalGenerator
    pdgf.generator.ProbabilityGenerator
    pdgf.generator.RandomSentence
}
</gen_(pluginName)>
</gen_pdgf.generator.RelativeGenerator>

```

Listing 57: XML example with all options

1.58 pdgf.generator.SequentialGenerator

Runs multiple generators one after another and optionally keeps interim results.

1.58.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	
concatenateResults		0/1	Concatenate all sub-generator values as string.	true false 0 1

1.58.2 Nodes

Name	Req.	Min/- Max count	Description	Values
gen	x	1/Int.max	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgf.generator.IdGenerator

1.58.3 XML-Example

```

<!--Description: Runs multiple generators one after another and optionally keeps interim results. -->
<gen_pdgf.generator.SequentialGenerator
  concatenateResults="(optional) Desc: Concatenate all sub-generator values as string. Allowed
    values: {
      true
      false
      0
      1
    }"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
<!--Required: [x] Executions(min|max): (1|INT.MAX) -->
<!--Description: Value Generator for this field-->
<gen_(pluginName)>Allowed values: {
  pdgf.generator.HashGenerator
  pdgf.generator.ComputeGenerator
  pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
  pdgf.generator.FormattedNumberGenerator
  pdgf.generator.ForGenerator
  pdgf.generator.MarkovChainGenerator
  pdgf.generator.CDCSequenceGenerator
  pdgf.generator.RandomUniqueStringGenerator
  pdgf.generator.FormatNumberAsDateGenerator
  pdgf.generator.SimpleFormatNumberGenerator
  pdgf.generator.BuildListGenerator
  pdgf.generator.DictList
  pdgf.generator.DoubleGenerator
  pdgf.generator.LastChoiceGenerator
  pdgf.generator.IFGenerator
  pdgf.generator.WeightedSubListGenerator
  pdgf.generator.DetailFromDateGenerator
  pdgf.generator.DateTimeGenerator
  pdgf.generator.UpdateActionTypeGenerator
  pdgf.generator.IdGenerator
  pdgf.generator.OtherFieldValueGenerator
  pdgf.generator.TemplateGenerator
  pdgf.generator.ForFormulaGenerator
  pdgf.generator.WeightedListGenerator

```

```

    pdgf.generator.AddRandomSecondsGenerator
    pdgf.generator.FormulaGenerator
    pdgf.generator.PaddingGenerator
    pdgf.generator.BigBenchReviewGenerator
    pdgf.generator.PrePostfixGenerator
    pdgf.generator.DefaultReferenceGenerator
    pdgf.generator.RandomAString
    pdgf.generator.SwitchGenerator
    pdgf.generator.ConvertNumberToStringGenerator
    pdgf.generator.ExtendedFormulaGenerator
    pdgf.generator.SetQuery
    pdgf.generator.StaticValueGenerator
    pdgf.generator.UUIDgenerator
    pdgf.generator.GlobalRowGenerator
    pdgf.generator.LongGenerator
    pdgf.generator.RelativeGenerator
    pdgf.generator.EmailGenerator
    pdgf.generator.SequentialGenerator
    pdgf.generator.GenderGenerator
    pdgf.generator.RandomListItemsGenerator
    pdgf.generator.PermutationReferenceGenerator
    pdgf.generator.RandomValueXY
    pdgf.generator.NameGenerator
    pdgf.generator.NullGenerator
    pdgf.generator.ReferenceGenerator
    pdgf.generator.UpperLowerCaseGenerator
    pdgf.generator.AbstractDecimalGenerator
    pdgf.generator.ProbabilityGenerator
    pdgf.generator.RandomSentence
  }
</gen-(pluginName)>
</gen-pdgf.generator.SequentialGenerator>

```

Listing 58: XML example with all options

1.59 pdgf.generator.SetQuery

Generates fields K500K, K250K, K100K, K40K, K10K, K1K, K100, K25 K10, K5, K4 and K2 of the set query benchmark. For KSEQ the IdGenerator is used and for S fields the StaticValueGenerator. Does not support use of differnt distributions

1.59.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.59.2 Nodes

Name	Req.	Min/- Max count	Description	Values
------	------	-----------------------	-------------	--------

1.59.3 XML-Example

```

<!--Description: Generates fields K500K, K250K, K100K, K40K, K10K, K1K, K100, K25 K10, K5, K4 and K2 of
the set query benchmark. For KSEQ the IdGenerator is used and for S fields the StaticValueGenerator.
Does not support use of differnt distributions -->
<gen-pdgf.generator.SetQuery

```

```

    id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
        uniquely identify a field within the children of an Element.">
</gen_pdgf.generator.SetQuery>

```

Listing 59: XML example with all options

1.60 pdgf.generator.SimpleFormatNumberGenerator

No other format specifiers then %d are allowed! Generates a number and formats it according to the format string.

1.60.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.60.2 Nodes

Name	Req.	Min/- Max count	Description	Values
gen		0/1	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgf.generator.IdGenerator
format	x	1/1	Date or time format	

1.60.3 XML-Example

```

<!--Description: No other format specifiers then %d are allowed! Generates a number and formats it
    according to the format string. -->
<gen_pdgf.generator.SimpleFormatNumberGenerator
    id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
        uniquely identify a field within the children of an Element.">
    <!--Required: [x] Executions(min|max): (1|1) -->
    <!--Description: Date or time format-->
    <formatX/format>
    <!--Required: [ ] Executions(min|max): (0|1) -->
    <!--Description: Value Generator for this field-->
    <gen-(pluginName)>Allowed values: {
        pdgf.generator.HashGenerator
        pdgf.generator.ComputeGenerator
        pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
        pdgf.generator.FormattedNumberGenerator
        pdgf.generator.ForGenerator
        pdgf.generator.MarkovChainGenerator
        pdgf.generator.CDCSequenceGenerator
        pdgf.generator.RandomUniqeStringGenerator
        pdgf.generator.FormatNumberAsDateGenerator
        pdgf.generator.SimpleFormatNumberGenerator
        pdgf.generator.BuildListGenerator
        pdgf.generator.DictList
        pdgf.generator.DoubleGenerator
        pdgf.generator.LastChoiceGenerator
        pdgf.generator.IFGenerator
        pdgf.generator.WeightedSubListGenerator
    }

```

```

    pdgf.generator.DetailFromDateGenerator
    pdgf.generator.DateTimeGenerator
    pdgf.generator.UpdateActionTypeGenerator
    pdgf.generator.IdGenerator
    pdgf.generator.OtherFieldValueGenerator
    pdgf.generator.TemplateGenerator
    pdgf.generator.ForFormulaGenerator
    pdgf.generator.WeightedListGenerator
    pdgf.generator.AddRandomSecondsGenerator
    pdgf.generator.FormulaGenerator
    pdgf.generator.PaddingGenerator
    pdgf.generator.BigBenchReviewGenerator
    pdgf.generator.PrePostfixGenerator
    pdgf.generator.DefaultReferenceGenerator
    pdgf.generator.RandomAString
    pdgf.generator.SwitchGenerator
    pdgf.generator.ConvertNumberToStringGenerator
    pdgf.generator.ExtendedFormulaGenerator
    pdgf.generator.SetQuery
    pdgf.generator.StaticValueGenerator
    pdgf.generator.UUIDgenerator
    pdgf.generator.GlobalRowGenerator
    pdgf.generator.LongGenerator
    pdgf.generator.RelativeGenerator
    pdgf.generator.EmailGenerator
    pdgf.generator.SequentialGenerator
    pdgf.generator.GenderGenerator
    pdgf.generator.RandomListItemsGenerator
    pdgf.generator.PermutationReferenceGenerator
    pdgf.generator.RandomValueXY
    pdgf.generator.NameGenerator
    pdgf.generator.NullGenerator
    pdgf.generator.ReferenceGenerator
    pdgf.generator.UpperLowerCaseGenerator
    pdgf.generator.AbstractDecimalGenerator
    pdgf.generator.ProbabilityGenerator
    pdgf.generator.RandomSentence
  }
</gen-(pluginName)>
</gen-pdgf.generator.SimpleFormatNumberGenerator>

```

Listing 60: XML example with all options

1.61 pdgf.generator.StaticValueGenerator

For all rows the same value is used as specified in `{value}`. The value string to be used `{value}`. Detects if static value is numerical. If so the 'plainValue' field contains a Number object

1.61.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.61.2 Nodes

Name	Req.	Min/- Max count	Description	Values
value	x	1/1	The static value to be used for all rows. <code>{value}</code> / <code>{value}</code> equals NULL	

1.61.3 XML-Example

```
<!--Description: For all rows the same value is used as specified in <value>The value string to be
used</value>. Detects if static value is numerical. If so the 'plainValue' field contains a Number
object -->
<gen_pdgf.generator.StaticValueGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: The static value to be used for all rows. <value></value> equals NULL-->
  <value></value>
</gen_pdgf.generator.StaticValueGenerator>
```

Listing 61: XML example with all options

1.62 pdgf.generator.SwitchGenerator

Generates a value based on the output of another generator.

1.62.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.62.2 Nodes

Name	Req.	Min/- Max count	Description	Values
gen		0/1	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgf.generator.IdGenerator
default		0/1	The Generator to use to generate the default value, if none of the cases match	
case	x	1/1	A case to check	

1.62.3 XML-Example

```
<!--Description: Generates a value based on the output of another generator. -->
<gen_pdgf.generator.SwitchGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: A case to check-->
  <case></case>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: The Generator to use to generate the default value, if none of the cases match-->
  <default></default>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Value Generator for this field-->
  <gen_(pluginName)>Allowed values: {
```



```

    pdgf.generator.HashGenerator
    pdgf.generator.ComputeGenerator
    pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
    pdgf.generator.FormattedNumberGenerator
    pdgf.generator.ForGenerator
    pdgf.generator.MarkovChainGenerator
    pdgf.generator.CDCSequenceGenerator
    pdgf.generator.RandomUniqueStringGenerator
    pdgf.generator.FormatNumberAsDateGenerator
    pdgf.generator.SimpleFormatNumberGenerator
    pdgf.generator.BuildListGenerator
    pdgf.generator.DictList
    pdgf.generator.DoubleGenerator
    pdgf.generator.LastChoiceGenerator
    pdgf.generator.IFGenerator
    pdgf.generator.WeightedSubListGenerator
    pdgf.generator.DetailFromDateGenerator
    pdgf.generator.DateTimeGenerator
    pdgf.generator.UpdateActionTypeGenerator
    pdgf.generator.IdGenerator
    pdgf.generator.OtherFieldValueGenerator
    pdgf.generator.TemplateGenerator
    pdgf.generator.ForFormulaGenerator
    pdgf.generator.WeightedListGenerator
    pdgf.generator.AddRandomSecondsGenerator
    pdgf.generator.FormulaGenerator
    pdgf.generator.PaddingGenerator
    pdgf.generator.BigBenchReviewGenerator
    pdgf.generator.PrePostfixGenerator
    pdgf.generator.DefaultReferenceGenerator
    pdgf.generator.RandomAString
    pdgf.generator.SwitchGenerator
    pdgf.generator.ConvertNumberToStringGenerator
    pdgf.generator.ExtendedFormulaGenerator
    pdgf.generator.SetQuery
    pdgf.generator.StaticValueGenerator
    pdgf.generator.UUIDgenerator
    pdgf.generator.GlobalRowGenerator
    pdgf.generator.LongGenerator
    pdgf.generator.RelativeGenerator
    pdgf.generator.EmailGenerator
    pdgf.generator.SequentialGenerator
    pdgf.generator.GenderGenerator
    pdgf.generator.RandomListItemsGenerator
    pdgf.generator.PermutationReferenceGenerator
    pdgf.generator.RandomValueXY
    pdgf.generator.NameGenerator
    pdgf.generator.NullGenerator
    pdgf.generator.ReferenceGenerator
    pdgf.generator.UpperLowerCaseGenerator
    pdgf.generator.AbstractDecimalGenerator
    pdgf.generator.ProbabilityGenerator
    pdgf.generator.RandomSentence
  }
</gen_(pluginName)>
</gen-pdgf.generator.SwitchGenerator>

```

Listing 62: XML example with all options

1.63 pdgf.generator.TemplateGenerator

Template based generator. Specify your javacode between `getValue<i>i</i>!`– HERE –`<i>i</i>/getValue<i>i</i>`

You are responsible for implementing the method body of:

```
public void getValue(pdgf.plugin.AbstractPDGFRandom rng, pdgf.core.dataGenerator.beans.FieldValueDTO fvdto, pdgf.core.dataGenerator
gc) throws Exception{

```

```
...
```

```
}

```

1.63.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.63.2 Nodes

Name	Req.	Min/- Max count	Description	Values
gen		0/Int.max	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgf.generator.IdGenerator
getValue	x	1/1	Template for output formatting.	
staticValue		0/1	here you can specify static values to be available to use. Example: <code>staticValue{new int{1,2,3,4,5}}/staticValue</code>	
distribution		0/1	Distribution to be used by a generator when calculating a value	

1.63.3 XML-Example

```

<!--Description: Template based generator. Specify your javacode between <getValue><!-- HERE
--></getValue>
You are responsible for implementing the method body of:
public void getValue(pdgf.plugin.AbstractPDGFRandom rng,pdgf.core.dataGenerator.beans.FieldValueDTO
    fvdto ,      pdgf.core.dataGenerator.beans.GenerationContext gc) throws Exception{
    ...
} -->
<gen-pdgf.generator.TemplateGenerator
    id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
        uniquely identify a field within the children of an Element.">
    <!--Required:  [x] Executions(min|max): (1|1) -->
    <!--Description: Template for output formatting.-->
    <getValue></getValue>
    <!--Required:  [ ] Executions(min|max): (0|INT.MAX) -->
    <!--Description: Value Generator for this field-->
    <gen-(pluginName)>Allowed values: {
        pdgf.generator.HashGenerator
        pdgf.generator.ComputeGenerator
        pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
        pdgf.generator.FormattedNumberGenerator
        pdgf.generator.ForGenerator
        pdgf.generator.MarkovChainGenerator
        pdgf.generator.CDCSequenceGenerator
        pdgf.generator.RandomUniqueStringGenerator
        pdgf.generator.FormatNumberAsDateGenerator
        pdgf.generator.SimpleFormatNumberGenerator
        pdgf.generator.BuildListGenerator
        pdgf.generator.DictList
        pdgf.generator.DoubleGenerator
        pdgf.generator.LastChoiceGenerator
        pdgf.generator.IFGenerator
        pdgf.generator.WeightedSubListGenerator
        pdgf.generator.DetailFromDateGenerator
        pdgf.generator.DateTimeGenerator
        pdgf.generator.UpdateActionTypeGenerator
        pdgf.generator.IdGenerator
    }

```

```

    pdgf.generator.OtherFieldValueGenerator
    pdgf.generator.TemplateGenerator
    pdgf.generator.ForFormulaGenerator
    pdgf.generator.WeightedListGenerator
    pdgf.generator.AddRandomSecondsGenerator
    pdgf.generator.FormulaGenerator
    pdgf.generator.PaddingGenerator
    pdgf.generator.BigBenchReviewGenerator
    pdgf.generator.PrePostfixGenerator
    pdgf.generator.DefaultReferenceGenerator
    pdgf.generator.RandomAString
    pdgf.generator.SwitchGenerator
    pdgf.generator.ConvertNumberToStringGenerator
    pdgf.generator.ExtendedFormulaGenerator
    pdgf.generator.SetQuery
    pdgf.generator.StaticValueGenerator
    pdgf.generator.UUIDgenerator
    pdgf.generator.GlobalRowGenerator
    pdgf.generator.LongGenerator
    pdgf.generator.RelativeGenerator
    pdgf.generator.EmailGenerator
    pdgf.generator.SequentialGenerator
    pdgf.generator.GenderGenerator
    pdgf.generator.RandomListItemsGenerator
    pdgf.generator.PermutationReferenceGenerator
    pdgf.generator.RandomValueXY
    pdgf.generator.NameGenerator
    pdgf.generator.NullGenerator
    pdgf.generator.ReferenceGenerator
    pdgf.generator.UpperLowerCaseGenerator
    pdgf.generator.AbstractDecimalGenerator
    pdgf.generator.ProbabilityGenerator
    pdgf.generator.RandomSentence
  }
</gen-(pluginName)>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Distribution to be used by a generator when calculating a value-->
<distribution></distribution>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: here you can specify static values to be available to use. Example:
  <staticValue>new int {1,2,3,4,5}</staticValue>-->
<staticValue
  unlockFullAccess="(optional) Desc: omits complexity restriction Allowed values: {
    true
    false
    0
    1
  }"></staticValue>
</gen-pdgf.generator.TemplateGenerator>

```

Listing 63: XML example with all options

1.64 pdgf.generator.UUIDgenerator

Generates a random 'universally unique identifier' (UUID) in version 4, variant 2 (Leach-Salz). A UUID represents a 128-bit value. For more information: http://de.wikipedia.org/wiki/Universally_Unique_Identifier#.28Pseudo.29zuf.C3.A4llig-generierte_UUID

1.64.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.64.2 Nodes

Name	Req.	Min/- Max count	Description	Values
------	------	-----------------------	-------------	--------

1.64.3 XML-Example

```

<!--Description: Generates a random 'universally unique identifier' (UUID) in version 4, variant 2
(Leach-Salz). A UUID represents a 128-bit value. For more information:
http://de.wikipedia.org/wiki/Universally_Unique_Identifier#.28Pseudo.29zuf.C3.A4llig-generierte_UUIDs_.28Vers.
-->
<gen_pdgf.generator.UUIDgenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
</gen_pdgf.generator.UUIDgenerator>

```

Listing 64: XML example with all options

1.65 pdgf.generator.UpdateActionTypeGenerator

Simple Action type generator.

1.65.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.65.2 Nodes

Name	Req.	Min/- Max count	Description	Values
changeName		0/1	changeName	
newName		0/1	newName	
deleteName		0/1	deleteName	

1.65.3 XML-Example

```

<!--Description: Simple Action type generator. -->
<gen_pdgf.generator.UpdateActionTypeGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: changeName-->
  <changeName</changeName>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: deleteName-->
  <deleteName</deleteName>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: newName-->
  <newName</newName>
</gen_pdgf.generator.UpdateActionTypeGenerator>

```

1.66 pdgf.generator.UpperLowerCaseGenerator

Reads a string generated by another generator or a subgenerator and formats it 1. with only the first letter uppercase, all uppercase or all lowercase.

1.66.1 Attributes

Name	Req.	Min/- Max count	Description	Values
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.66.2 Nodes

Name	Req.	Min/- Max count	Description	Values
mode	x	1/1	Sets the mode for this generator	
gen		0/1	Value Generator for this field	Allowed values are: a Generator-plugin names. e.g.: IdGenerator or its full name: pdgf.generator.IdGenerator

1.66.3 XML-Example

```
<!--Description: Reads a string generated by another generator or a subgenerator and formats it 1. with
only the first letter uppercase, all uppercase or all lowercase. -->
<gen_pdgf.generator.UpperLowerCaseGenerator
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: Sets the mode for this generator-->
  <mode</mode>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Value Generator for this field-->
  <gen_(pluginName)>Allowed values: {
    pdgf.generator.HashGenerator
    pdgf.generator.ComputeGenerator
    pdgf.generator.ActionDeleteIsAlsoUpdateGenerator
    pdgf.generator.FormattedNumberGenerator
    pdgf.generator.ForGenerator
    pdgf.generator.MarkovChainGenerator
    pdgf.generator.CDCSequenceGenerator
    pdgf.generator.RandomUniqueStringGenerator
    pdgf.generator.FormatNumberAsDateGenerator
    pdgf.generator.SimpleFormatNumberGenerator
    pdgf.generator.BuildListGenerator
    pdgf.generator.DictList
    pdgf.generator.DoubleGenerator
    pdgf.generator.LastChoiceGenerator
    pdgf.generator.IFGenerator
    pdgf.generator.WeightedSubListGenerator
    pdgf.generator.DetailFromDateGenerator
```

```

    pdgf.generator.DateTimeGenerator
    pdgf.generator.UpdateActionTypeGenerator
    pdgf.generator.IdGenerator
    pdgf.generator.OtherFieldValueGenerator
    pdgf.generator.TemplateGenerator
    pdgf.generator.ForFormulaGenerator
    pdgf.generator.WeightedListGenerator
    pdgf.generator.AddRandomSecondsGenerator
    pdgf.generator.FormulaGenerator
    pdgf.generator.PaddingGenerator
    pdgf.generator.BigBenchReviewGenerator
    pdgf.generator.PrePostfixGenerator
    pdgf.generator.DefaultReferenceGenerator
    pdgf.generator.RandomAString
    pdgf.generator.SwitchGenerator
    pdgf.generator.ConvertNumberToStringGenerator
    pdgf.generator.ExtendedFormulaGenerator
    pdgf.generator.SetQuery
    pdgf.generator.StaticValueGenerator
    pdgf.generator.UUIDgenerator
    pdgf.generator.GlobalRowGenerator
    pdgf.generator.LongGenerator
    pdgf.generator.RelativeGenerator
    pdgf.generator.EmailGenerator
    pdgf.generator.SequentialGenerator
    pdgf.generator.GenderGenerator
    pdgf.generator.RandomListItemsGenerator
    pdgf.generator.PermutationReferenceGenerator
    pdgf.generator.RandomValueXY
    pdgf.generator.NameGenerator
    pdgf.generator.NullGenerator
    pdgf.generator.ReferenceGenerator
    pdgf.generator.UpperLowerCaseGenerator
    pdgf.generator.AbstractDecimalGenerator
    pdgf.generator.ProbabilityGenerator
    pdgf.generator.RandomSentence
  }
</gen_(pluginName)>
</gen-pdgf.generator.UpperLowerCaseGenerator>

```

Listing 66: XML example with all options

1.67 pdgf.generator.WeightedListGenerator

A dictionary with weights. example:Example file:

SIMPLE:

```

create terminators;
set types = (varchar);
set weights = 1;
add (".": 10);
add ("\\": 2);
add (" -": 1);
add (":": 1);

```

Complex:- fields weights (aliases in parens)

```

- =====
- 1: FIPS code (fips) 1: uniform (uniform)
- 2: county name (name) 2: population (population)
- 3: state abbreviation (st) 3: timezone weighting (tz)
- 4: full state name (state) 4: in zone1 (tz90)
- 5: ZIP prefix (zone) 5: in zone2 (tz9)
- 6: gmt offset (gmt) 6 in zone3 (tz1)
-
create fips_county;
set types = (int, varchar, varchar, varchar, varchar, int);
set weights = 6;

```

```

set names = (fips, county, st, state, zone, gmt:uniform, population, tz, tz90, tz9, tz1);
add (47187,"Williamson County", "TN", "Tennessee", "3", -5:1, 117569, 1387, 1, 0, 0);
add (46137,"Ziebach County", "SD", "South Dakota", "5", -6:1, 2176, 1148, 1, 0, 0);
add (01127,"Walker County", "AL", "Alabama", "3", -6:1, 71027, 1148, 1, 0, 0);
add (45039,"Fairfield County", "SC", "South Carolina", "2", -5:1, 22394, 1387, 1, 0, 0);
add (39139,"Richland County", "OH", "Ohio", "4", -5:1, 127342, 1387, 1, 0, 0);
add (22041,"Franklin Parish", "LA", "Louisiana", "7", -6:1, 22163, 1148, 1, 0, 0);
add (29061,"Daviess County", "MO", "Mosourri", "6", -6:1, 7842, 1148, 1, 0, 0);

```

1.67.1 Attributes

Name	Req.	Min/- Max count	Description	Values
filename		0/1	<p>Example file:</p> <p>SIMPLE:</p> <pre> create terminators; set types = (varchar); set weights = 1; add (".": 10); add ("\\": 2); add (" -": 1); add (":": 1); </pre> <p>Complex:– fields weights (aliases in parens)</p> <pre> - ===== - 1: FIPS code (fips) 1: uniform (uniform) - 2: county name (name) 2: population (population) - 3: state abbreviation (st) 3: timezone weighting (tz) - 4: full state name (state) 4: in zone1 (tz90) - 5: ZIP prefix (zone) 5: in zone2 (tz9) - 6: gmt offset (gmt) 6 in zone3 (tz1) - create fips_county; set types = (int, varchar, varchar, varchar, varchar, int); set weights = 6; set names = (fips, county, st, state, zone, gmt:uniform, population, tz, tz90, tz9, tz1); add (47187,"Williamson County", "TN", "Tennessee", "3", -5:1, 117569, 1387, 1, 0, 0); add (46137,"Ziebach County", "SD", "South Dakota", "5", -6:1, 2176, 1148, 1, 0, 0); add (01127,"Walker County", "AL", "Alabama", "3", -6:1, 71027, 1148, 1, 0, 0); add (45039,"Fairfield County", "SC", "South Carolina", "2", -5:1, 22394, 1387, 1, 0, 0); add (39139,"Richland County", "OH", "Ohio", "4", -5:1, 127342, 1387, 1, 0, 0); add (22041,"Franklin Parish", "LA", "Louisiana", "7", -6:1, 22163, 1148, 1, 0, 0); add (29061,"Daviess County", "MO", "Mosourri", "6", -6:1, 7842, 1148, 1, 0, 0); </pre>	
valueCol- umn	x	1/1		
weightCol- umn	x	1/1		

id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	
list	x	1/1	list	

1.67.2 Nodes

Name	Req.	Min/- Max count	Description	Values
same-ChoiceAs		0/1	Another 'WeightedListGenerator within the same table to pick his choice from. If specified, this generator does not decide upon the row in the weightedList on its own, but it uses the same choice as the referenced WeightedListGenerator made	

1.67.3 XML-Example

```

<!--Description: A dictionary with weights. example:Example file:

SIMPLE:
create terminators;
set types = (varchar);
set weights = 1;
add (".": 10);
add ("\;": 2);
add (" —": 1);
add (":": 1);

Complex:-- fields      weights (aliases in parens)
-- =====
-- 1: FIPS code (fips)      1: uniform (uniform)
-- 2: county name (name)    2: population (population)
-- 3: state abbreviation (st) 3: timezone weighting (tz)
-- 4: full state name (state) 4: in zone1 (tz90)
-- 5: ZIP prefix (zone)     5: in zone2 (tz9)
-- 6: gmt offset (gmt)      6 in zone3 (tz1)
--
create fips_county;
set types = (int, varchar, varchar, varchar, varchar, int);
set weights = 6;
set names = (fips, county, st, state, zone, gmt:uniform, population, tz, tz90, tz9, tz1);
add (47187,"Williamson County", "TN", "Tennessee", "3", -5:1, 117569, 1387, 1, 0, 0);
add (46137,"Ziebach County", "SD", "South Dakota", "5", -6:1, 2176, 1148, 1, 0, 0);
add (01127,"Walker County", "AL", "Alabama", "3", -6:1, 71027, 1148, 1, 0, 0);
add (45039,"Fairfield County", "SC", "South Carolina", "2", -5:1, 22394, 1387, 1, 0, 0);
add (39139,"Richland County", "OH", "Ohio", "4", -5:1, 127342, 1387, 1, 0, 0);
add (22041,"Franklin Parish", "LA", "Louisiana", "7", -6:1, 22163, 1148, 1, 0, 0);
add (29061,"Daviess County", "MO", "Mosourri", "6", -6:1, 7842, 1148, 1, 0, 0); -->
<gen.pdgm.generator.WeightedListGenerator
  list="Desc: list"
  valueColumn="Desc: "
  weightColumn="Desc: "
  filename="(optional) Desc: Example file:

SIMPLE:
create terminators;
set types = (varchar);
set weights = 1;
add (".": 10);
add ("\;": 2);
add (" —": 1);
add (":": 1);

```



```

Complex:-- fields      weights (aliases in parens)
--
-- 1: FIPS code (fips)      1: uniform (uniform)
-- 2: county name (name)    2: population (population)
-- 3: state abbreviation (st) 3: timezone weighting (tz)
-- 4: full state name (state) 4: in zone1 (tz90)
-- 5: ZIP prefix (zone)     5: in zone2 (tz9)
-- 6: gmt offset (gmt)      6 in zone3 (tz1)
--
create fips_county;
set types = (int, varchar, varchar, varchar, varchar, int);
set weights = 6;
set names = (fips, county, st, state, zone, gmt:uniform, population, tz, tz90, tz9, tz1);
add (47187,"Williamson County", "TN", "Tennessee", "3", -5:1, 117569, 1387, 1, 0, 0);
add (46137,"Ziebach County", "SD", "South Dakota", "5", -6:1, 2176, 1148, 1, 0, 0);
add (01127,"Walker County", "AL", "Alabama", "3", -6:1, 71027, 1148, 1, 0, 0);
add (45039,"Fairfield County", "SC", "South Carolina", "2", -5:1, 22394, 1387, 1, 0, 0);
add (39139,"Richland County", "OH", "Ohio", "4", -5:1, 127342, 1387, 1, 0, 0);
add (22041,"Franklin Parish", "LA", "Louisiana", "7", -6:1, 22163, 1148, 1, 0, 0);
add (29061,"Daviness County", "MO", "Mosourri", "6", -6:1, 7842, 1148, 1, 0, 0);"
    id=(optional) Desc: DefaultParser. Identification String of this element. May be used to
        uniquely identify a field within the children of an Element.">
    <!--Required: [ ] Executions(min|max): (0|1) -->
    <!--Description: Another 'WeightedListGenerator within the same table to pick his choice from. If
        specified, this generator does not decide upon the row in the weightedList on its own, but it
        uses the same choice as the referenced WeightedListGenerator made-->
    <sameChoiceAs></sameChoiceAs>
</gen_pdgf.generator.WeightedListGenerator>

```

Listing 67: XML example with all options

1.68 pdgf.generator.WeightedSubListGenerator

A dictionary with weights. example:Example file:

SIMPLE:

```

create terminators;
set types = (varchar);
set weights = 1;
add (".": 10);
add ("\\": 2);
add (" -": 1);
add (":": 1);

```

Complex:-- fields weights (aliases in parens)

```

- =====
- 1: FIPS code (fips) 1: uniform (uniform)
- 2: county name (name) 2: population (population)
- 3: state abbreviation (st) 3: timezone weighting (tz)
- 4: full state name (state) 4: in zone1 (tz90)
- 5: ZIP prefix (zone) 5: in zone2 (tz9)
- 6: gmt offset (gmt) 6 in zone3 (tz1)
-
create fips_county;
set types = (int, varchar, varchar, varchar, varchar, int);
set weights = 6;
set names = (fips, county, st, state, zone, gmt:uniform, population, tz, tz90, tz9, tz1);
add (47187,"Williamson County", "TN", "Tennessee", "3", -5:1, 117569, 1387, 1, 0, 0);
add (46137,"Ziebach County", "SD", "South Dakota", "5", -6:1, 2176, 1148, 1, 0, 0);
add (01127,"Walker County", "AL", "Alabama", "3", -6:1, 71027, 1148, 1, 0, 0);
add (45039,"Fairfield County", "SC", "South Carolina", "2", -5:1, 22394, 1387, 1, 0, 0);
add (39139,"Richland County", "OH", "Ohio", "4", -5:1, 127342, 1387, 1, 0, 0);
add (22041,"Franklin Parish", "LA", "Louisiana", "7", -6:1, 22163, 1148, 1, 0, 0);

```

add (29061,"Daviness County", "MO", "Mosourri", "6", -6:1, 7842, 1148, 1, 0, 0);

1.68.1 Attributes

Name	Req.	Min/- Max count	Description	Values
filename		0/1	<p>Example file:</p> <p>SIMPLE: create terminators; set types = (varchar); set weights = 1; add (".": 10); add ("\\": 2); add (" -": 1); add (":": 1);</p> <p>Complex:- fields weights (aliases in parens) - ===== - 1: FIPS code (fips) 1: uniform (uniform) - 2: county name (name) 2: population (population) - 3: state abbreviation (st) 3: timezone weighting (tz) - 4: full state name (state) 4: in zone1 (tz90) - 5: ZIP prefix (zone) 5: in zone2 (tz9) - 6: gmt offset (gmt) 6 in zone3 (tz1) - create fips_county; set types = (int, varchar, varchar, varchar, varchar, int); set weights = 6; set names = (fips, county, st, state, zone, gmt:uniform, population, tz, tz90, tz9, tz1); add (47187,"Williamson County", "TN", "Tennessee", "3", -5:1, 117569, 1387, 1, 0, 0); add (46137,"Ziebach County", "SD", "South Dakota", "5", -6:1, 2176, 1148, 1, 0, 0); add (01127,"Walker County", "AL", "Alabama", "3", -6:1, 71027, 1148, 1, 0, 0); add (45039,"Fairfield County", "SC", "South Carolina", "2", -5:1, 22394, 1387, 1, 0, 0); add (39139,"Richland County", "OH", "Ohio", "4", -5:1, 127342, 1387, 1, 0, 0); add (22041,"Franklin Parish", "LA", "Louisiana", "7", -6:1, 22163, 1148, 1, 0, 0); add (29061,"Daviness County", "MO", "Mosourri", "6", -6:1, 7842, 1148, 1, 0, 0);</p>	
same-ChoiceAs-ValueColumn	x	1/1		
valueColumn	x	1/1		
weightColumn	x	1/1		
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

same-ChoiceAsList	x	1/1	list	
-------------------	---	-----	------	--

1.68.2 Nodes

Name	Req.	Min/- Max count	Description	Values
same-ChoiceAs	x	1/1	Another 'WeightedSubListGenerator within the same table to pick his choice from. If specified, this generator does not decide upon the row in the weightedList on its own, but it uses the same choice as the referenced WeightedSubListGenerator made	

1.68.3 XML-Example

```

<!--Description: A dictionary with weights. example:Example file:

SIMPLE:
create terminators;
set types = (varchar);
set weights = 1;
add (".": 10);
add ("\": 2);
add ("—": 1);
add (":": 1);

Complex:-- fields      weights (aliases in parens)
--
-- 1: FIPS code (fips)      1: uniform (uniform)
-- 2: county name (name)    2: population (population)
-- 3: state abbreviation (st) 3: timezone weighting (tz)
-- 4: full state name (state) 4: in zone1 (tz90)
-- 5: ZIP prefix (zone)     5: in zone2 (tz9)
-- 6: gmt offset (gmt)      6 in zone3 (tz1)
--
create fips_county;
set types = (int, varchar, varchar, varchar, varchar, int);
set weights = 6;
set names = (fips, county, st, state, zone, gmt:uniform, population, tz, tz90, tz9, tz1);
add (47187,"Williamson County", "TN", "Tennessee", "3", -5:1, 117569, 1387, 1, 0, 0);
add (46137,"Ziebach County", "SD", "South Dakota", "5", -6:1, 2176, 1148, 1, 0, 0);
add (01127,"Walker County", "AL", "Alabama", "3", -6:1, 71027, 1148, 1, 0, 0);
add (45039,"Fairfield County", "SC", "South Carolina", "2", -5:1, 22394, 1387, 1, 0, 0);
add (39139,"Richland County", "OH", "Ohio", "4", -5:1, 127342, 1387, 1, 0, 0);
add (22041,"Franklin Parish", "LA", "Louisiana", "7", -6:1, 22163, 1148, 1, 0, 0);
add (29061,"Daviness County", "MO", "Mosourri", "6", -6:1, 7842, 1148, 1, 0, 0); -->
<gen_pdgf.generator.WeightedSubListGenerator
    sameChoiceAsList="Desc: list"
    sameChoiceAsValueColumn="Desc: "
    valueColumn="Desc: "
    weightColumn="Desc: "
    filename="(optional) Desc: Example file:

SIMPLE:
create terminators;
set types = (varchar);
set weights = 1;
add (".": 10);
add ("\": 2);
add ("—": 1);
add (":": 1);

```

```

Complex:-- fields      weights (aliases in parens)
--
-- 1: FIPS code (fips)      1: uniform (uniform)
-- 2: county name (name)    2: population (population)
-- 3: state abbreviation (st) 3: timezone weighting (tz)
-- 4: full state name (state) 4: in zone1 (tz90)
-- 5: ZIP prefix (zone)     5: in zone2 (tz9)
-- 6: gmt offset (gmt)      6 in zone3 (tz1)
--
create fips_county;
set types = (int, varchar, varchar, varchar, varchar, int);
set weights = 6;
set names = (fips, county, st, state, zone, gmt:uniform, population, tz, tz90, tz9, tz1);
add (47187,"Williamson County", "TN", "Tennessee", "3", -5:1, 117569, 1387, 1, 0, 0);
add (46137,"Ziebach County", "SD", "South Dakota", "5", -6:1, 2176, 1148, 1, 0, 0);
add (01127,"Walker County", "AL", "Alabama", "3", -6:1, 71027, 1148, 1, 0, 0);
add (45039,"Fairfield County", "SC", "South Carolina", "2", -5:1, 22394, 1387, 1, 0, 0);
add (39139,"Richland County", "OH", "Ohio", "4", -5:1, 127342, 1387, 1, 0, 0);
add (22041,"Franklin Parish", "LA", "Louisiana", "7", -6:1, 22163, 1148, 1, 0, 0);
add (29061,"Daviss County", "MO", "Mosourri", "6", -6:1, 7842, 1148, 1, 0, 0);"
    id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
        uniquely identify a field within the children of an Element.">
<!--Required: [x] Executions(min|max): (1|1) -->
<!--Description: Another 'WeightedSubListGenerator within the same table to pick his choice from.
    If specified, this generator does not decide upon the row in the weightedList on its own, but
    it uses the same choice as the referenced WeightedSubListGenerator made-->
<sameChoiceAs</sameChoiceAs>
</gen_pdgf.generator.WeightedSubListGenerator>

```

Listing 68: XML example with all options

1.69 pdgf.output.CSVRowOutput

Takes a RowDataDTO (containing a generated row) and writes this row into a CSV file. This RowOutput does not provide ordering of rows. Rows are written when they arrive from the worker threads. Two or more programm runs may not result in the same output as it concerns ordering of generated rows.

1.69.1 Attributes

Name	Req.	Min/- Max count	Description	Values
name	x	1/1	DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	
active		0/1	Boolean flag if this output is active/used	true false 0 1
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.69.2 Nodes

Name	Req.	Min/- Max count	Description	Values
quotation-Character		0/1	Puts quotation marks around strings. Default quotation mark:"	
outputDir		0/1	Directory to put the generated files in.	

charset		0/1	Charset used for output encoding. Default: null	
padding		0/1	if true, pads value with whitespace(s) till <code>{field;size}</code> characters are written. Default:false Example <code>{delimiter}</code> is: <code>;</code> Let <code>{field name='a';size;5i/size}</code> and <code>{field name='b';size;6i/size}</code> . Values <code>a='foo'</code> , <code>b='bar'</code> . Output will be: <code>'foo ;bar ;'</code>	true false 0 1
Excluded-Historical-Fields		0/1	<code>[0-FieldNumber[</code> wich is excluded when <code>updateID = 0</code> . <code>FieldNumber</code> is the first field to be written to file. <code>updateID</code> can be specified via the <code>{ExcludedHistoricalFields historicalUpdateIDsCount=""}</code> attribute	
delimiterAt-LineEnd		0/1	Put delimiter char also at line end? Values: {true false} Default: false	true false 0 1
sortBy-RowID		0/1	Enables sorting of RowDataDTOs by rowid. Default: false	true false 0 1
footer		0/1	Footer of output file.	
fileEnding		0/1	Ending of output files. Default:null	
quoteStrings		0/1	Puts quotation marks around strings. Default quotation mark:"	true false 0 1
fileTemplate	x	1/1	Generic directory and filename of file to put the generated data in. Available variables: project, table, tableID, updateID, output, nodeCount, nodeNumber, outputDir, fileEnding	
fieldIDOffset		0/1	exclude <code>[0,fieldIDOffset[</code> ids. These ids are not written.	
delimiter		0/1	Delimiter char used in CSV. examples: { <code>;</code> , <code>;</code> <code>\t</code> }. Default: <code>;</code>	
paddingObjectArray		0/1	if this is true and <code>{padding}</code> is also true, Object <code>[]</code> are also padded automatically. Default: false	true false 0 1
header		0/1	Header of output file.	
bufferSize		0/1	Buffersize for Output. Buffersize is the amount of bytes cached before a write to a file occurs. Default: 0	

1.69.3 XML-Example

```

<!--Description: Takes a RowDataDTO (containing a generated row) and writes this row into a CSV file.
This RowOutput does not provide ordering of rows. Rows are written when they arrive from the worker
threads. Two or more programm runs may not result in the same output as it conceners ordering of
generated rows. -->
<output_pdgf.output.CSVRowOutput
  name="Desc: DefaultParser. (ClassName) of this element. Used to identify plugin Class. Full name
  is required. Example: com.en.myPluginPackage.myPuginClass"
  active="(optional) Desc: Boolean flag if this output is active/used Allowed values: {
    true
    false
    0
    1
  }"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element.">
<!--Required: [x] Executions(min|max): (1|1) -->
<!--Description: Generic directory and filename of file to put the generated data in. Available
variables: project, table, tableID, updateID, output, nodeCount, nodeNumber, outputDir,
fileEnding-->
<fileTemplate></fileTemplate>

```

```

<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: [0-FieldNumber[ wich is excluded when updateID = 0. FieldNumber is the first
    field to be written to file. updateID can be specified via the <ExcludedHistoricalFields
        historicalUpdateIDsCount=""]> attribute-->
<ExcludedHistoricalFields
    historicalUpdateIDsCount="(optional) Desc: ""/ExcludedHistoricalFields>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: BufferSize for Output. BufferSize is the amount of bytes cached before a write
    to a file occurs. Default: 0-->
<bufferSize"/bufferSize>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Charset used for output encoding. Default: null-->
<charset"/charset>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Delimiter char used in CSV. examples: { , ; | \t}. Default: ';'-->
<delimiter"/delimiter>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Put delimiter char also at line end? Values: {true | false} Default: false-->
<delimiterAtLineEnd>Allowed values: {
    true
    false
    0
    1
}
</delimiterAtLineEnd>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: exclude [0,fieldIDOffset[ ids. These ids are not written.-->
<fieldIDOffset"/fieldIDOffset>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Ending of output files. Default:null-->
<fileEnding"/fileEnding>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Footer of output file.-->
<footer"/footer>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Header of output file.-->
<header"/header>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Directory to put the generated files in.-->
<outputDir"/outputDir>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: if true, pads value with whitespace(s) till <field size> characters are
    written. Default:false Example <delimiter> is: ; Let <field name='a' size>5</size> and
    <field name='b' size>6</size>. Values a='foo ', b='bar '. Output will be: 'foo ';bar ';-->
<padding>Allowed values: {
    true
    false
    0
    1
}
</padding>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: if this is true and <padding> is also true, Object [] are also padded
    automatically. Default: false-->
<paddingObjectArray>Allowed values: {
    true
    false
    0
    1
}
</paddingObjectArray>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Puts quotation marks arround strings. Default quotation mark:"-->
<quotationCharacter"/quotationCharacter>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Puts quotation marks arround strings. Default quotation mark:"-->
<quoteStrings>Allowed values: {
    true
    false
    0
    1
}
</quoteStrings>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Enables sorting of RowDataDTOs by rowid. Default: false-->
<sortByRowID>Allowed values: {

```

```

        true
        false
        0
        1
    }
</sortByRowID>
</output_pdgf.output.CSVRowOutput>

```

Listing 69: XML example with all options

1.70 pdgf.output.CompiledTemplateOutput

Generic output. The RowDataDTO's are formatted by a template which can be specified within the xml file. The template is pure Java and represents the body of the method: `FileNameTemplateSuperClass.public void writeLine(pdgf.core.dataGenerator.beans.[] fields, pdgf.util.caching.EncodingCharLineCache buffer)`. The contents of the provided `FieldValueDTO[]` must! be appended to the provided buffer.

1.70.1 Attributes

Name	Req.	Min/- Max count	Description	Values
name	x	1/1	DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name is required. Example: <code>com.en.myPluginPackage.myPuginClass</code>	
active		0/1	Boolean flag if this output is active/used	true false 0 1
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.70.2 Nodes

Name	Req.	Min/- Max count	Description	Values
template	x	1/1	Template for output formatting.	
outputDir		0/1	Directory to put the generated files in.	
charset		0/1	Charset used for output encoding. Default: null	
sortBy- RowID		0/1	Enables sorting of RowDataDTOs by rowid. Default: false	true false 0 1
fileTemplate	x	1/1	Generic directory and filename of file to put the generated data in. Available variables: project, table, tableID, updateID, output, nodeCount, nodeNumber, outputDir, fileEnding	
footer		0/1	Footer of output file.	
delimiter		0/1	Delimiter char used in CSV. examples: { , ; \t}. Default: ','	
fileEnding		0/1	Ending of output files. Default:null	
header		0/1	Header of output file.	
bufferSize		0/1	Buffersize for Output. Buffersize is the amount of bytes cached before a write to a file occurs. Default: 0	

1.70.3 XML-Example

```
<!--Description: Generic output. The RowDataDTO's are formatted by a template which can be specified
within the xml file. The template is pure Java and represents the body of the method:
FileNameTemplateSuperClass.public void writeLine(pdgf.core.dataGenerator.beans.FieldValueDTO []
fields , pdgf.util.caching.EncodingCharLineCache buffer). The contents of the provided
FieldValueDTO[] must! be appended to the provided buffer. -->
<output_pdgf.output.CompiledTemplateOutput
  name="Desc: DefaultParser. (ClassName) of this element. Used to identify plugin Class. Full name
  is required. Example: com.en.myPluginPackage.myPuginClass"
  active="(optional) Desc: Boolean flag if this output is active/used Allowed values: {
    true
    false
    0
    1
  }"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: Generic directory and filename of file to put the generated data in. Available
  variables: project, table, tableID, updateID, output, nodeCount, nodeNumber, outputDir,
  fileEnding-->
  <fileTemplate></fileTemplate>
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: Template for output formatting.-->
  <template></template>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Bufferize for Output. Bufferize is the amount of bytes cached before a write
  to a file occurs. Default: 0-->
  <bufferSize></bufferSize>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Charset used for output encoding. Default: null-->
  <charset></charset>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Delimiter char used in CSV. examples: { , ; | \t}. Default: ';'-->
  <delimiter></delimiter>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Ending of output files. Default: null-->
  <fileEnding></fileEnding>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Footer of output file.-->
  <footer></footer>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Header of output file.-->
  <header></header>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Directory to put the generated files in.-->
  <outputDir></outputDir>
  <!--Required: [ ] Executions(min|max): (0|1) -->
  <!--Description: Enables sorting of RowDataDTOs by rowid. Default: false -->
  <sortByRowID>Allowed values: {
    true
    false
    0
    1
  }
</sortByRowID>
</output_pdgf.output.CompiledTemplateOutput>
```

Listing 70: XML example with all options

1.71 pdgf.output.DummyOutput

Does nothing. Data is not saved or cached in any way. All methods are empty. Return values are always true.

1.71.1 Attributes

Name	Req.	Min/- Max count	Description	Values
name	x	1/1	DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	
active		0/1	Boolean flag if this output is active/used	true false 0 1
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.71.2 Nodes

Name	Req.	Min/- Max count	Description	Values
------	------	-----------------------	-------------	--------

1.71.3 XML-Example

```

<!--Description: Does nothing. Data is not safed or cached in any way. All methods are empty. Return
values are allways true. -->
<output_pdgf.output.DummyOutput
  name="Desc: DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name
is required. Example: com.en.myPluginPackage.myPuginClass"
  active="(optional) Desc: Boolean flag if this output is active/used Allowed values: {
    true
    false
    0
    1
  }"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
uniquely identify a field within the children of an Element.">
</output_pdgf.output.DummyOutput>

```

Listing 71: XML example with all options

1.72 pdgf.output.HierarchicalXMLRowOutput

Prototype of a hierarchical xml output. It does not produce XML-Attribute values and will write all values as separate nodes. You can annotate list producing generators like ForGenerator or BuildListGenerator with a listName attribute, which will be used as a xml node name. Non-List generators can be annotated using the id attribute, which will also serve as XML-Node name. If you see a `null` node, you forgot the annotation!

1.72.1 Attributes

Name	Req.	Min/- Max count	Description	Values
name	x	1/1	DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	

active		0/1	Boolean flag if this output is active/used	true false 0 1
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.72.2 Nodes

Name	Req.	Min/- Max count	Description	Values
outputDir		0/1	Directory to put the generated files in.	
charset		0/1	Charset used for output encoding. Default: null	
sortBy- RowID		0/1	Enables sorting of RowDataDTOs by rowid. Default: false	true false 0 1
fileTemplate	x	1/1	Generic directory and filename of file to put the generated data in. Available variables: project, table, tableID, updateID, output, nodeCount, nodeNumber, outputDir, fileEnding	
footer	x	1/1	Footer of output file.	
delimiter		0/1	Delimiter char used in CSV. examples: { , ; \t}. Default: ','	
fileEnding		0/1	Ending of output files. Default:null	
header	x	1/1	Header of output file.	
bufferSize		0/1	Buffer size for Output. Buffer size is the amount of bytes cached before a write to a file occurs. Default: 0	

1.72.3 XML-Example

```

<!--Description: Prototype of a hierarchical xml output. It does not produce XML-Attribute values and
will write all values as separate nodes. You can annotate list producing generators like ForGenerator
or BuildListGenerator with a listName attribute, which will be used as a xml node name. Non-List
generators can be annotated using the id attribute, which will also serve as XML-Node name. If you
see a <null> node, you forgot the annotation! -->
<output_pdgf.output.HierarchicalXMLRowOutput
  name="Desc: DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name
  is required. Example: com.en.myPluginPackage.myPuginClass"
  active="(optional) Desc: Boolean flag if this output is active/used Allowed values: {
    true
    false
    0
    1
  }"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
  uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: Generic directory and filename of file to put the generated data in. Available
  variables: project, table, tableID, updateID, output, nodeCount, nodeNumber, outputDir,
  fileEnding-->
  <fileTemplate></fileTemplate>
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: Footer of output file.-->
  <footer></footer>
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: Header of output file.-->

```

```

<header></header>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: BufferSize for Output. BufferSize is the amount of bytes cached before a write
to a file occurs. Default: 0-->
<bufferSize></bufferSize>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Charset used for output encoding. Default: null-->
<charset></charset>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Delimiter char used in CSV. examples: { , ; | \t }. Default: ';'-->
<delimiter></delimiter>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Ending of output files. Default: null-->
<fileEnding></fileEnding>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Directory to put the generated files in.-->
<outputDir></outputDir>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Enables sorting of RowDataDTOs by rowid. Default: false -->
<sortByRowID>Allowed values: {
    true
    false
    0
    1
}
</sortByRowID>
</output_pdgf.output.HierarchicalXMLRowOutput>

```

Listing 72: XML example with all options

1.73 pdgf.output.MultiLineOutput

If you used `pdgf.generator.ForGenerator` or a similar construct generating an `List<Object>` of values per field and row, this output will split and replicate this single line into multiple lines.

Example: Original Line:

```
{1,2,{A,B,C},3,{D,E,F}}
```

will result in: {1,2,A,3,D}

```
{1,2,B,3,E}
```

```
{1,2,B,3,F}
```

Additionally this output can interleave the virtual lines in within a batch.

Example with batch size 3:

```
{1,1,{A,B,C},1,{D,E,F}}
```

```
{2,2,{A,B,C},2,{D,E,F}}
```

```
{3,3,{A,B,C},3,{D,E,F}}
```

Will produce 9 virtual lines:

```
{1,1,A,1,D}
```

```
{1,1,B,1,E}
```

```
{1,1,C,1,F}
```

```
{2,2,A,2,D}
```

```
{2,2,B,2,E}
```

```
{2,2,C,2,F}
```

```
{3,3,A,3,D}
```

```
{3,3,B,3,E}
```

```
{3,3,C,3,F}
```

It randomly interleaves the lines (but keeps the order of the sublines):

```
{3,3,A,3,D}
```

```
{3,3,B,3,E}
```

```
{2,2,A,2,D}
```

```
{1,1,A,1,D}
```

```
{1,1,B,1,E}
```

```
{2,2,B,2,E}
```

```
{2,2,C,2,F}
```

```
{1,1,C,1,F}
```

```
{3,3,C,3,F}
```

1.73.1 Attributes

Name	Req.	Min/- Max count	Description	Values
name	x	1/1	DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	
active		0/1	Boolean flag if this output is active/used	true false 0 1
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.73.2 Nodes

Name	Req.	Min/- Max count	Description	Values
outputDir		0/1	Directory to put the generated files in.	
charset		0/1	Charset used for output encoding. Default: null	
sortBy- RowID		0/1	Enables sorting of RowDataDTOs by rowid. Default: false	true false 0 1
fileTemplate	x	1/1	Generic directory and filename of file to put the generated data in. Available variables: project, table, tableID, updateID, output, nodeCount, nodeNumber, outputDir, fileEnding	
multiline- FieldId	x	1/1		
footer		0/1	Footer of output file.	
delimiter		0/1	Delimiter char used in CSV. examples: { , ; \t}. Default: ','	
fileEnding		0/1	Ending of output files. Default:null	
header		0/1	Header of output file.	
bufferSize		0/1	Buffersize for Output. Buffersize is the amount of bytes cached before a write to a file occurs. Default: 0	
interleave- Multilines		0/1	interleaves multilines of a batch. (controlled by scheduler batch size)	true false 0 1

1.73.3 XML-Example

```

<!--Description: If you used apdgm.generator.ForGenerator or a similar construct generating an
List<Object> of values per field and row, this output will split and replicate this single line into
multiple lines.
Example: Original Line:
{1,2,{A,B,C},3,{D,E,F}}
will result in: {1,2,A,3,D}
{1,2,B,3,E}
{1,2,B,3,F}
Additionally this output can interleave the virtual lines in within a batch.

```

Example with batch size 3:

```
{1,1,{A,B,C},1,{D,E,F}}
{2,2,{A,B,C},2,{D,E,F}}
{3,3,{A,B,C},3,{D,E,F}}
```

Will produce 9 virtual lines:

```
{1,1,A,1,D}
{1,1,B,1,E}
{1,1,C,1,F}
{2,2,A,2,D}
{2,2,B,2,E}
{2,2,C,2,F}
{3,3,A,3,D}
{3,3,B,3,E}
{3,3,C,3,F}
```

It randomly interleaves the lines (but keeps the order of the sublines):

```
{3,3,A,3,D}
{3,3,B,3,E}
{2,2,A,2,D}
{1,1,A,1,D}
{1,1,B,1,E}
{2,2,B,2,E}
{2,2,C,2,F}
{1,1,C,1,F}
{3,3,C,3,F}
```

—>

```
<output.pdgif.output.MultiLineOutput
  name="Desc: DefaultParser. (ClassName)Name of this element. Used to identify plugin Class. Full name
    is required. Example: com.en.myPluginPackage.myPuginClass"
  active="(optional) Desc: Boolean flag if this output is active/used Allowed values: {
    true
    false
    0
    1
  }"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
<!--Required: [x] Executions(min|max): (1|1) -->
<!--Description: Generic directory and filename of file to put the generated data in. Available
  variables: project, table, tableID, updateID, output, nodeCount, nodeNumber, outputDir,
  fileEnding-->
<fileTemplate</fileTemplate>
<!--Required: [x] Executions(min|max): (1|1) -->
<!--Description: -->
<multilineFieldId</multilineFieldId>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: BufferSize for Output. BufferSize is the amount of bytes cached before a write
  to a file occurs. Default: 0-->
<bufferSize</bufferSize>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Charset used for output encoding. Default: null-->
<charset</charset>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Delimiter char used in CSV. examples: { , ; | \t}. Default: ';'-->
<delimiter</delimiter>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Ending of output files. Default:null-->
<fileEnding</fileEnding>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Footer of output file.-->
<footer</footer>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Header of output file.-->
<header</header>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: interleaves multilines of a batch. (controlled by scheduler batch size)-->
<interleaveMultilines>Allowed values: {
  true
  false
  0
  1
}
</interleaveMultilines>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Directory to put the generated files in.-->
<outputDir</outputDir>
<!--Required: [ ] Executions(min|max): (0|1) -->
```

```

<!--Description: Enables sorting of RowDataDTOs by rowid. Default: false-->
<sortByRowID>Allowed values: {
    true
    false
    0
    1
}
</sortByRowID>
</output_pdgf.output.MultiLineOutput>

```

Listing 73: XML example with all options

1.74 pdgf.output.MultiLineOutputWrapper

If you used `apdgf.generator.ForGenerator` or a similar construct generating an `List<Object>` of values per field and row, this output will split and replicate this single line into multiple lines.

Example: Original Line:

`{1,2,{A,B,C},3,{D,E,F}}`

will result in: `{1,2,A,3,D}`

`{1,2,B,3,E}`

`{1,2,B,3,F}`

1.74.1 Attributes

Name	Req.	Min/- Max count	Description	Values
name	x	1/1	DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name is required. Example: <code>com.en.myPluginPackage.myPuginClass</code>	
active		0/1	Boolean flag if this output is active/used	true false 0 1
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.74.2 Nodes

Name	Req.	Min/- Max count	Description	Values
output	x	1/1	Specifies the output plugin and its configuration to be used for output of generated value. Required child nodes are specified by the plugin.	
multiline-FieldId	x	1/1		

1.74.3 XML-Example

```

<!--Description: If you used apdgf.generator.ForGenerator or a similar construct generating an
List<Object> of values per field and row, this output will split and replicate this single line into
multiple lines.
Example: Original Line:
{1,2,{A,B,C},3,{D,E,F}}
will result in: {1,2,A,3,D}

```

```

{1,2,B,3,E}
{1,2,B,3,F} -->
<output_pdgf.output.MultiLineOutputWrapper
  name="Desc: DefaultParser. (ClassName)Name of this element. Used to identify plugin Class. Full name
    is required. Example: com.en.myPluginPackage.myPuginClass"
  active="(optional) Desc: Boolean flag if this output is active/used Allowed values: {
    true
    false
    0
    1
  }"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
    uniquely identify a field within the children of an Element.">
<!--Required: [x] Executions(min|max): (1|1) -->
<!--Description: -->
<multilineFieldId></multilineFieldId>
<!--Required: [x] Executions(min|max): (1|1) -->
<!--Description: Specifies the output plugin and its configuration to be used for output of
  generated value. Required child nodes are specified by the plugin.-->
<output></output>
</output_pdgf.output.MultiLineOutputWrapper>

```

Listing 74: XML example with all options

1.75 pdgf.output.SortedFileOutputWrapper

Sorts incoming RowDataDTOs by rowID and forwards them to an underlying output

1.75.1 Attributes

Name	Req.	Min/- Max count	Description	Values
name	x	1/1	DefaultParser. (ClassName)Name of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	
active		0/1	Boolean flag if this output is active/used	true false 0 1
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.75.2 Nodes

Name	Req.	Min/- Max count	Description	Values
output	x	1/1	Specifies the output plugin and its configuration to be used for output of generated value. Required child nodes are specified by the plugin.	

1.75.3 XML-Example

```

<!--Description: Sorts incoming RowDataDTOs by rowID and forwards them to an underlying output -->
<output_pdgf.output.SortedFileOutputWrapper
  name="Desc: DefaultParser. (ClassName)Name of this element. Used to identify plugin Class. Full name
    is required. Example: com.en.myPluginPackage.myPuginClass"

```

```

    active="(optional) Desc: Boolean flag if this output is active/used Allowed values: {
        true
        false
        0
        1
    }"
    id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
        uniquely identify a field within the children of an Element.">
<!--Required: [x] Executions(min|max): (1|1) -->
<!--Description: Specifies the output plugin and its configuration to be used for output of
    generated value. Required child nodes are specified by the plugin.-->
<output></output>
</output_pdgf.output.SortedFileOutputWrapper>

```

Listing 75: XML example with all options

1.76 pdgf.output.SortedOutputWrapper

Sorts incoming RowDataDTOs by rowID and forwards them to an underlying output

1.76.1 Attributes

Name	Req.	Min/- Max count	Description	Values
name	x	1/1	DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	
active		0/1	Boolean flag if this output is active/used	true false 0 1
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.76.2 Nodes

Name	Req.	Min/- Max count	Description	Values
output	x	1/1	Specifies the output plugin and its configuration to be used for output of generated value. Required child nodes are specified by the plugin.	

1.76.3 XML-Example

```

<!--Description: Sorts incoming RowDataDTOs by rowID and forwards them to an underlying output -->
<output_pdgf.output.SortedOutputWrapper
    name="Desc: DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name
        is required. Example: com.en.myPluginPackage.myPuginClass"
    active="(optional) Desc: Boolean flag if this output is active/used Allowed values: {
        true
        false
        0
        1
    }"
    id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
        uniquely identify a field within the children of an Element.">

```



```

<!--Required: [x] Executions(min|max): (1|1) -->
<!--Description: Specifies the output plugin and its configuration to be used for output of
generated value. Required child nodes are specified by the plugin.-->
<output></output>
</output_pdgf.output.SortedOutputWrapper>

```

Listing 76: XML example with all options

1.77 pdgf.output.SplitFileOutputWrapper

Can split the contents of a RowDataDTO to multiple Output sub classes. (And therefore to multiple files)

1.77.1 Attributes

Name	Req.	Min/- Max count	Description	Values
name	x	1/1	DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	
active		0/1	Boolean flag if this output is active/used	true false 0 1
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.77.2 Nodes

Name	Req.	Min/- Max count	Description	Values
output	x	1/1	Specifies the output plugin and its configuration to be used for output of generated value. Required child nodes are specified by the plugin.	

1.77.3 XML-Example

```

<!--Description: Can split the contents of a RowDataDTO to multiple Output sub classes. (And therefore to
multiple files) -->
<output_pdgf.output.SplitFileOutputWrapper
  name="Desc: DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name
is required. Example: com.en.myPluginPackage.myPuginClass"
  active="(optional) Desc: Boolean flag if this output is active/used Allowed values: {
    true
    false
    0
    1
  }"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: Specifies the output plugin and its configuration to be used for output of
generated value. Required child nodes are specified by the plugin.-->
  <output></output>
</output_pdgf.output.SplitFileOutputWrapper>

```

Listing 77: XML example with all options

1.78 pdgf.output.SplitOutputWrapper

Can split the contents of a RowDataDTO to multiple Output sub classes. (And therefore to multiple files)

1.78.1 Attributes

Name	Req.	Min/- Max count	Description	Values
name	x	1/1	DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	
active		0/1	Boolean flag if this output is active/used	true false 0 1
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.78.2 Nodes

Name	Req.	Min/- Max count	Description	Values
output	x	1/1	Specifies the output plugin and its configuration to be used for output of generated value. Required child nodes are specified by the plugin.	

1.78.3 XML-Example

```
<!--Description: Can split the contents of a RowDataDTO to multiple Output sub classes. (And therefore to
multiple files) -->
<output_pdgf.output.SplitOutputWrapper
  name="Desc: DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name
is required. Example: com.en.myPluginPackage.myPuginClass"
  active="(optional) Desc: Boolean flag if this output is active/used Allowed values: {
    true
    false
    0
    1
  }"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
uniquely identify a field within the children of an Element.">
  <!--Required: [x] Executions(min|max): (1|1) -->
  <!--Description: Specifies the output plugin and its configuration to be used for output of
generated value. Required child nodes are specified by the plugin.-->
  <outputX/output>
</output_pdgf.output.SplitOutputWrapper>
```

Listing 78: XML example with all options

1.79 pdgf.output.StatisticsOutput

TODO

1.79.1 Attributes

Name	Req.	Min/- Max count	Description	Values
name	x	1/1	DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	
active		0/1	Boolean flag if this output is active/used	true false 0 1
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.79.2 Nodes

Name	Req.	Min/- Max count	Description	Values
outputDir		0/1	Directory to put the generated files in.	
charset		0/1	Charset used for output encoding. Default: null	
sortBy- RowID		0/1	Enables sorting of RowDataDTOs by rowid. Default: false	true false 0 1
footer		0/1	Footer of output file.	
tableChangedE- ventTem- plate		0/1	Template for tableChange() event.	
fileEnding		0/1	Ending of output files. Default:null	
counterTem- plate	x	1/1	Template for gathering statistics.	
fileTemplate	x	1/1	Generic directory and filename of file to put the generated data in. Available variables: project, table, tableID, updateID, output, nodeCount, nodeNumber, outputDir, fileEnding	
size	x	1/1	number of thread safe counter to allocate	
delimiter		0/1	Delimiter char used in CSV. examples: { , ; \t}. Default: ','	
header		0/1	Header of output file.	
for- materTem- plate		0/1	Template for output statistics forming.	
bufferSize		0/1	Buffer size for Output. Buffer size is the amount of bytes cached before a write to a file occurs. Default: 0	

1.79.3 XML-Example

```

<!--Description: TODO -->
<output_pdgf.output.StatisticsOutput
  name="Desc: DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name
    is required. Example: com.en.myPluginPackage.myPuginClass"
  active="(optional) Desc: Boolean flag if this output is active/used Allowed values: {
    true

```

```

        false
        0
        1
    }"
    id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
        uniquely identify a field within the children of an Element.">
    <!--Required: [x] Executions(min|max): (1|1) -->
    <!--Description: Template for gathering statistics.-->
    <counterTemplate></counterTemplate>
    <!--Required: [x] Executions(min|max): (1|1) -->
    <!--Description: Generic directory and filename of file to put the generated data in. Available
        variables: project, table, tableID, updateID, output, nodeCount, nodeNumber, outputDir,
        fileEnding-->
    <fileTemplate></fileTemplate>
    <!--Required: [x] Executions(min|max): (1|1) -->
    <!--Description: number of thread safe counter to allocate-->
    <size></size>
    <!--Required: [ ] Executions(min|max): (0|1) -->
    <!--Description: BufferSize for Output. BufferSize is the amount of bytes cached before a write
        to a file occurs. Default: 0-->
    <bufferSize></bufferSize>
    <!--Required: [ ] Executions(min|max): (0|1) -->
    <!--Description: Charset used for output encoding. Default: null-->
    <charset></charset>
    <!--Required: [ ] Executions(min|max): (0|1) -->
    <!--Description: Delimiter char used in CSV. examples: { , ; | \t}. Default: ';'-->
    <delimiter></delimiter>
    <!--Required: [ ] Executions(min|max): (0|1) -->
    <!--Description: Ending of output files. Default: null-->
    <fileEnding></fileEnding>
    <!--Required: [ ] Executions(min|max): (0|1) -->
    <!--Description: Footer of output file.-->
    <footer></footer>
    <!--Required: [ ] Executions(min|max): (0|1) -->
    <!--Description: Template for output statistics formating.-->
    <formatterTemplate></formatterTemplate>
    <!--Required: [ ] Executions(min|max): (0|1) -->
    <!--Description: Header of output file.-->
    <header></header>
    <!--Required: [ ] Executions(min|max): (0|1) -->
    <!--Description: Directory to put the generated files in.-->
    <outputDir></outputDir>
    <!--Required: [ ] Executions(min|max): (0|1) -->
    <!--Description: Enables sorting of RowDataDTOs by rowid. Default: false-->
    <sortByRowID>Allowed values: {
        true
        false
        0
        1
    }
    </sortByRowID>
    <!--Required: [ ] Executions(min|max): (0|1) -->
    <!--Description: Template for tableChange() event.-->
    <tableChangedEventTemplate></tableChangedEventTemplate>
</output_pdgf.output.StatisticsOutput>

```

Listing 79: XML example with all options

1.80 pdgf.output.XMLTemplateOutput

Simplistic template based output. Just specify a xml template string and insert generated fields using format specifiers. <http://docs.oracle.com/javase/7/docs/api/java/util/Formatter.html>. Example: `customer id="%s"%n |name%|n |first%|s|/first%|n |last%|s|/last%|n |i/name%|n |age%|d|/age%|n |i/customer%|n`

1.80.1 Attributes

Name	Req.	Min/- Max count	Description	Values

name	x	1/1	DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name is required. Example: com.en.myPluginPackage.myPuginClass	
active		0/1	Boolean flag if this output is active/used	true false 0 1
id		0/1	DefaultParser. Identification String of this element. May be used to uniquely identify a field within the children of an Element.	

1.80.2 Nodes

Name	Req.	Min/- Max count	Description	Values
template	x	1/1	xml format template	
outputDir		0/1	Directory to put the generated files in.	
charset		0/1	Charset used for output encoding. Default: null	
sortBy- RowID		0/1	Enables sorting of RowDataDTOs by rowid. Default: false	true false 0 1
fileTemplate	x	1/1	Generic directory and filename of file to put the generated data in. Available variables: project, table, tableID, updateID, output, nodeCount, nodeNumber, outputDir, fileEnding	
footer	x	1/1	Footer of output file.	
delimiter		0/1	Delimiter char used in CSV. examples: { , ; \t}. Default: ','	
fileEnding		0/1	Ending of output files. Default:null	
header	x	1/1	Header of output file.	
bufferSize		0/1	Buffer size for Output. Buffer size is the amount of bytes cached before a write to a file occurs. Default: 0	

1.80.3 XML-Example

```

<!--Description: Simplistic template based output. Just specify a xml template string and insert
generated fields using format specifiers.
http://docs.oracle.com/javase/7/docs/api/java/util/Formatter.html. Example:<customer id="%s">\n
<name>\n          <first>%s</first>\n          <last>%s</last>\n          </name>\n
<age>%d</age>\n          </customer>\n -->
<output_pdgf.output.XMLTemplateOutput
  name="Desc: DefaultParser. (Class)Name of this element. Used to identify plugin Class. Full name
is required. Example: com.en.myPluginPackage.myPuginClass"
  active="(optional) Desc: Boolean flag if this output is active/used Allowed values: {
    true
    false
    0
    1
  }"
  id="(optional) Desc: DefaultParser. Identification String of this element. May be used to
uniquely identify a field within the children of an Element.">
<!--Required: [x] Executions(min|max): (1|1) -->
<!--Description: Generic directory and filename of file to put the generated data in. Available
variables: project, table, tableID, updateID, output, nodeCount, nodeNumber, outputDir,
fileEnding-->
<fileTemplate></fileTemplate>

```

```

<!--Required: [x] Executions(min|max): (1|1) -->
<!--Description: Footer of output file.-->
<footer</footer>
<!--Required: [x] Executions(min|max): (1|1) -->
<!--Description: Header of output file.-->
<header</header>
<!--Required: [x] Executions(min|max): (1|1) -->
<!--Description: xml format template-->
<template</template>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Buffer size for Output. Buffer size is the amount of bytes cached before a write
to a file occurs. Default: 0-->
<bufferSize</bufferSize>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Charset used for output encoding. Default: null-->
<charset</charset>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Delimiter char used in CSV. examples: { , ; | \t}. Default: ';'-->
<delimiter</delimiter>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Ending of output files. Default: null-->
<fileEnding</fileEnding>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Directory to put the generated files in.-->
<outputDir</outputDir>
<!--Required: [ ] Executions(min|max): (0|1) -->
<!--Description: Enables sorting of RowDataDTOs by rowid. Default: false-->
<sortByRowID>Allowed values: {
    true
    false
    0
    1
}
</sortByRowID>
</output_pdgf.output.XMLTemplateOutput>

```

Listing 80: XML example with all options