Table Layout

Table Layout.

Currently the GeneralCsv and Database import require a table layout description configured by the user. Non-standard imports (plug-ins) can also require a table layout. See the third party documentation of the specific import.

There are different types of table structures supported.

- Every parameter has its own column
- Every location has its own column.
- Every qualifier has its own column.
- Every parameter/location/qualifier combination has its own column
- Single value column and a parameter, location, qualifier and unit column to describe the value in the value column.
- All values for a single month a listed in a single row, for every day of month there is separate column.
- All values for a single day listed in a single row, for every time of day there is separate column.
- Variants and combinations of the above are sometimes allowed

The column names are required for the GeneralCsv and Database import. The table name is ignored by the GeneralCsv import. Non-standard imports can ignore the column name. They expect that the order of the columns in the configuration is the same as in the file. For columns that should be ignored add a skippedColumn element in this case. When the column name is used by the import the order of the columns in the configuration has no meaning and there is no need to add a skippedColumn for the unwanted columns. For the Database import you can specify multiple tables. This has the advantage that you can reuse a database connection to import multiple tables.

A value column can have a fixed location id or every row can have a different location id when there is a location column. The same is true for the parameter id and qualifiers.

A value column can have a fixed unit or every row can have a different unit when there is a unit column.

The date/time is available in a combined data/time column or in two separated date and time columns. It is required to configure the data and time patterns when date/time is stored in a text column instead of timestamp column. The value columns can also contain a day of month or hour of day attribute. This attribute overrules the day or time provided by the date/time column. These attributes are used when all data for whole month or day is combined in a single row.

When the imported table contains grid files you have to specify the parser to parse these grids. Any grid format supported by the import module can be used.

Only one flag and comment column is allowed also when you have multiple value columns. All values in single row share the same flag and comment.

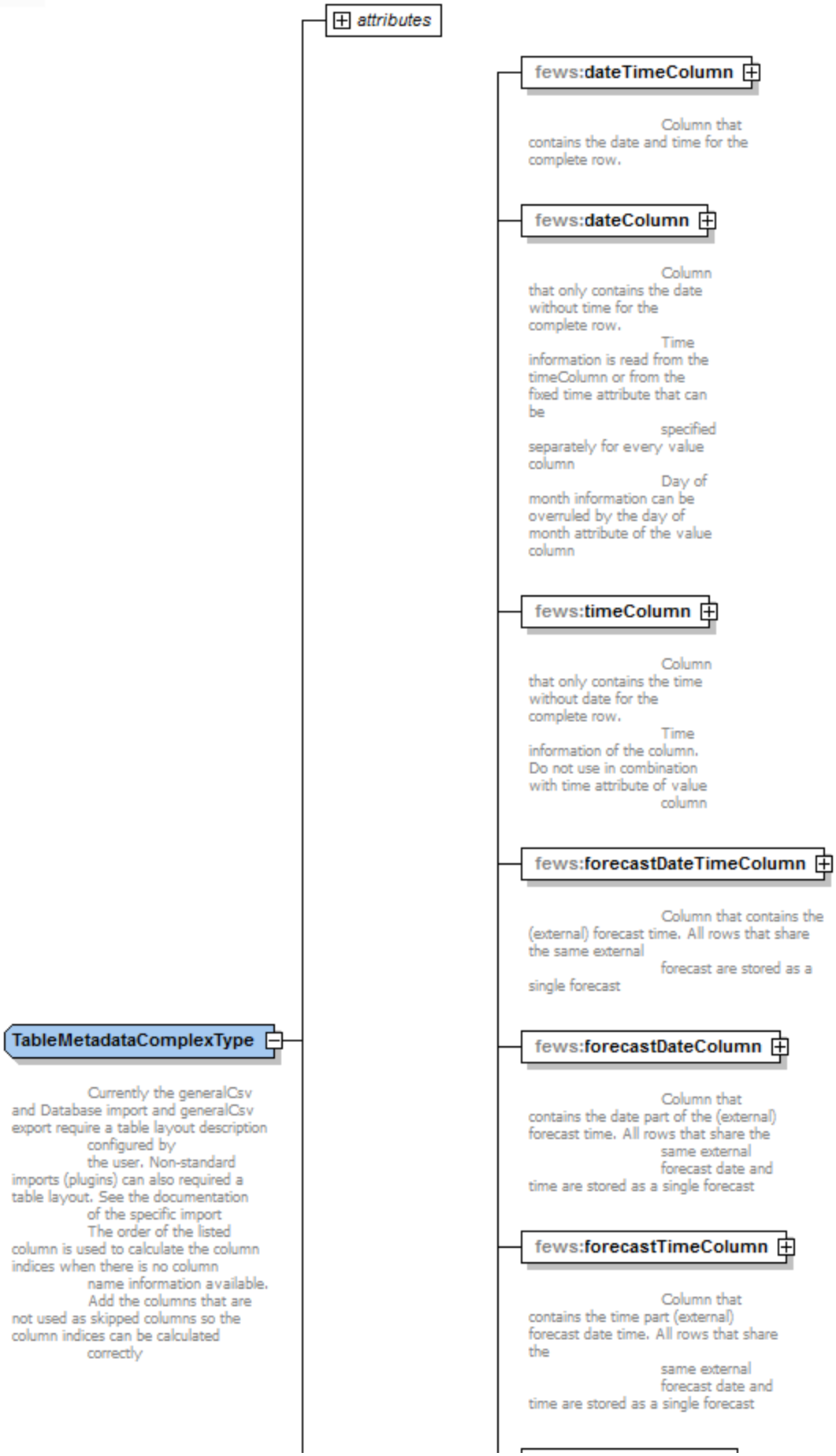
data time pattern
The date time patterns are handled by the JDK class SimpleDateFormat

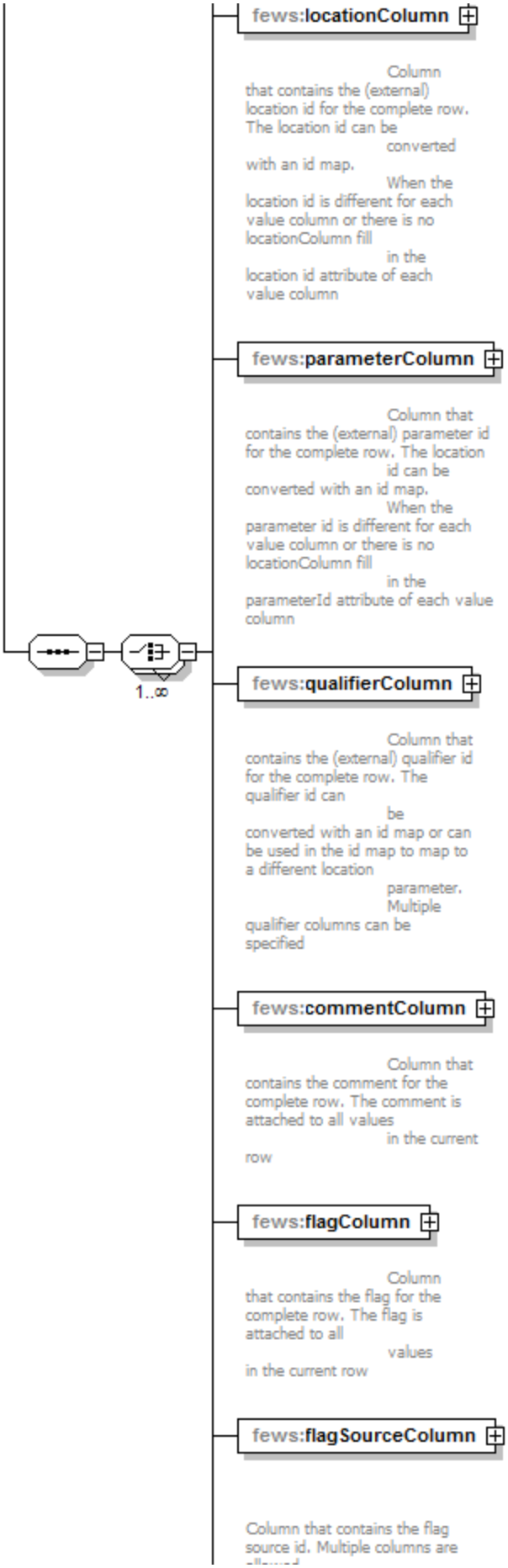
Letter	Date or Time	Component	Presentation Examples
G	Era designator	Text	AD
y	Year	Year	1996; 96
Y	Week year	Year	2009; 09
M	Month in year	Month	July; Jul; 07
w	Week in year	Number	27
W	Week in month	Number	2
D	Day in year	Number	189
d	Day in month	Number	10
F	Day of week in month	Number	2
E	Day name in week	Text	Tuesday; Tue
u	Day number of week (1 = Monday, ..., 7 = Sunday)	Number	1
a	Am/pm marker	Text	PM
H	Hour in day (0-23)	Number	0
k	Hour in day (1-24)	Number	24
K	Hour in am/pm (0-11)	Number	0
h	Hour in am/pm (1-12)	Number	12
m	Minute in hour	Number	30
s	Second in minute	Number	55
S	Millisecond	Number	978
z	Time zone	General time zone	Pacific Standard Time; PST; GMT-08:00

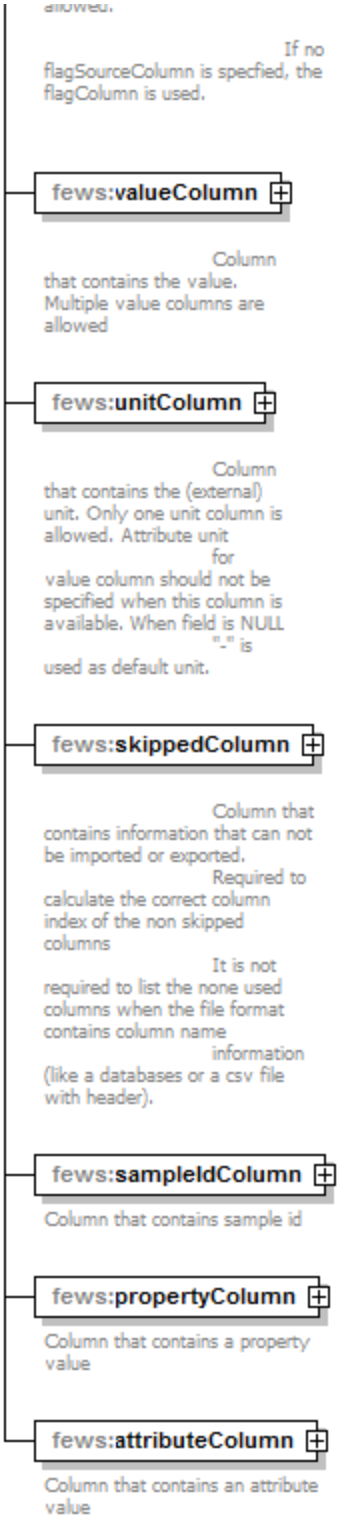
Z	Time zone	RFC 822 time zone	-0800
X	Time zone	ISO 8601 time zone	-08; -0800; -08:00

Any characters within the date-time stamp can be accommodated by including that character in the pattern, surrounded by single quotes, e.g. <dateTimeColumn name="date_obs" pattern="yyyy-MM-dd'T'HH:mm:ss'Z'"/> allows for reading a date-time column that is formatted as: 2022-03-29**T**09:30:00**Z**.

Schema


attributes
fews:dateTimeColumn
Column that contains the date and time for the complete row.
fews:dateColumn
Column that only contains the date without time for the complete row.
Time information is read from the timeColumn or from the fixed time attribute that can be
specified separately for every value column
Day of month information can be overruled by the day of month attribute of the value column
fews:timeColumn
Column that only contains the time without date for the complete row.
Time information of the column. Do not use in combination with time attribute of value
column
fews:forecastDateTimeColumn
Column that contains the (external) forecast time. All rows that share the same external
forecast are stored as a single forecast
TableMetadataComplexType
Currently the generalCsv and Database import and generalCsv export require a table layout description
configured by
the user. Non-standard imports (plugins) can also required a table layout. See the documentation
of the specific import
The order of the listed column is used to calculate the column indices when there is no column
name information available.
Add the columns that are not used as skipped columns so the column indices can be calculated
correctly
fews:forecastDateColumn
Column that contains the date part of the (external) forecast time. All rows that share the
same external
forecast date and time are stored as a single forecast
fews:forecastTimeColumn
Column that contains the time part (external) forecast date time. All rows that share the
same external
forecast date and time are stored as a single forecast


fews:locationColumn
Column that contains the (external) location id for the complete row. The location id can be converted with an id map. When the location id is different for each value column or there is no locationColumn fill in the location id attribute of each value column
fews:parameterColumn
Column that contains the (external) parameter id for the complete row. The location id can be converted with an id map. When the parameter id is different for each value column or there is no locationColumn fill in the parameterId attribute of each value column
1..∞
fews:qualifierColumn
Column that contains the (external) qualifier id for the complete row. The qualifier id can be converted with an id map or can be used in the id map to map to a different location parameter. Multiple qualifier columns can be specified
fews:commentColumn
Column that contains the comment for the complete row. The comment is attached to all values in the current row
fews:flagColumn
Column that contains the flag for the complete row. The flag is attached to all values in the current row
fews:flagSourceColumn
Column that contains the flag source id. Multiple columns are



examples

```
<table>
   <dateTimeColumn name="Time" pattern="dd-MM-yyyy HH:mm"/>
   <valueColumn unit="m" locationId="Bosscheveld" parameterId="H.meting" name="Waterstand"/>
   <valueColumn unit="min" locationId="Bosscheveld" parameterId="DT.meting" name="Pomp-1 Born"/>
</table>
```

```
<table name="rst_f">
    <dateTimeColumn name="RST_DATE"/>
    <locationColumn name="RST_CODE"/>
    <valueColumn name="RST_RAIN" unit="mm" parameterId="P.obs"/>
</table>
```

date pattern is not required because the date column is a SQL time stamp column

```
<table name="dam_f">
   <dateTimeColumn name="DAM_DATE"/>
   <valueColumn name="DAM_WLEV" locationId="Feitsui" unit="m" parameterId="H.reservoir.obs"/>
   <valueColumn name="DAM_INFLOW" locationId="Feitsui" unit="m3/s" parameterId="Q.in"/>
   <valueColumn name="DAM_OUFLOW" locationId="Feitsui" unit="m3/s" parameterId="Q.out"/>
</table>
```

```
<table name="gate_tp">
   <dateTimeColumn name="GATE_DATE"/>
   <locationColumn name="GATE_CODE"/>
   <valueColumn name="IN_LEVEL" unit="m" parameterId="H.obs.upstream"/>
   <valueColumn name="OUT_LEVEL" unit="m" parameterId="H.obs.downstream"/>
</table>
```

```
<table name="lst">
   <dateTimeColumn name="jdate"/>
   <valueColumn name="lst" unit="K" locationId="Modis_Grid" parameterId="T.landsurface" parser="AsciiGrid"/>
</table>
```

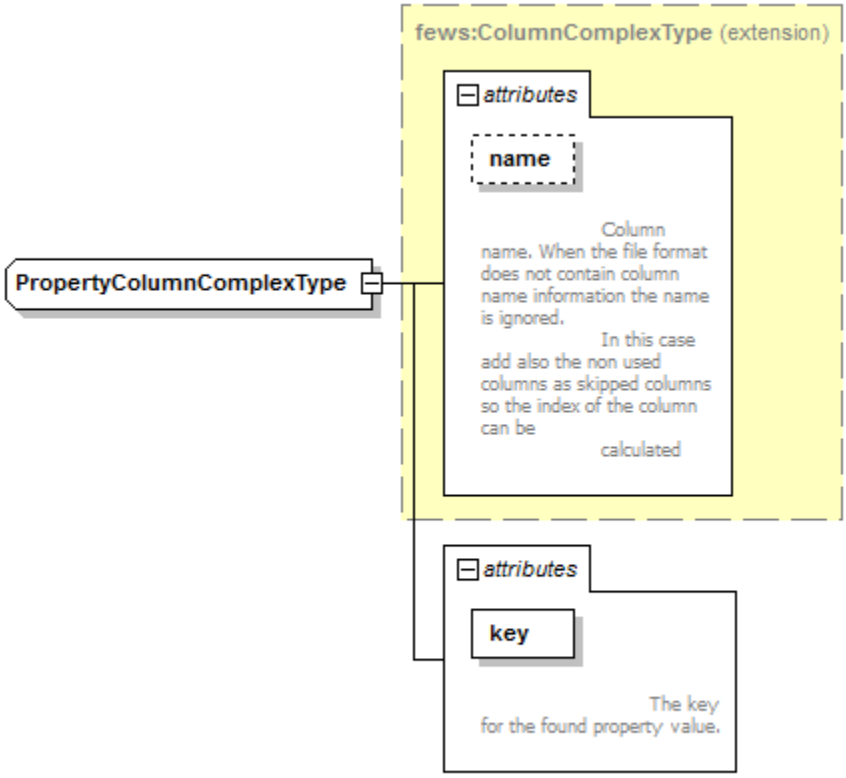
```
<table name="StageList">
  <dateTimeColumn name="InDate"/>
  <valueColumn name="Shemen" locationId="HS03" unit="m" parameterId="H.obs"/>
  <valueColumn name="Clouds" locationId="HS02" unit="m" parameterId="H.obs"/>
  <valueColumn name="High" locationId="1140H043" unit="m" parameterId="H.obs"/>
  <valueColumn name="Angle" locationId="1140H002" unit="m" parameterId="H.obs"/>
  <valueColumn name="Jade" locationId="HS01" unit="m" parameterId="H.obs"/>
  <valueColumn name="Show" locationId="1140H041" unit="m" parameterId="H.obs"/>
  <valueColumn name="Kide" locationId="HS04" unit="m" parameterId="H.obs"/>
</table>
```

An example on how to add flagSourceColumns is shown below. This is possible since FEWS version 2015.02.

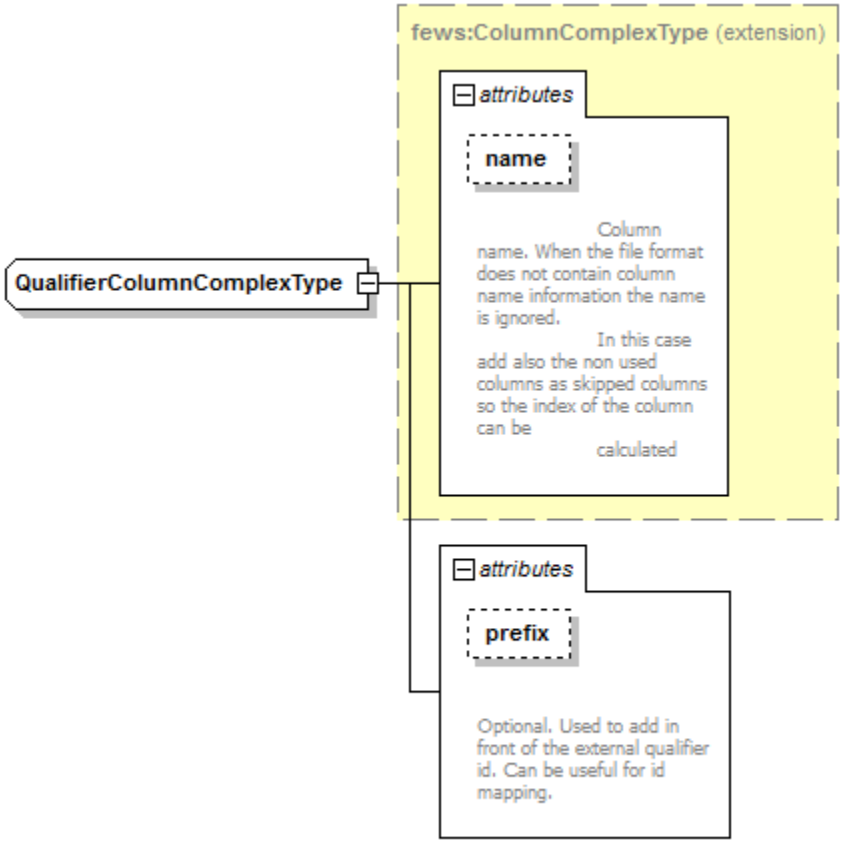
```
<table>
        <dateTimeColumn pattern="dd-MM-yy HH:mm"/>
        <locationColumn name="Location"/>
        <parameterColumn name="Parameter"/>
        <flagSourceColumn name="A" id="A"/>
        <flagSourceColumn name="B" id="B"/>
        <flagSourceColumn name="C" id="C"/>
        <flagSourceColumn name="D" id="D"/>
        <flagSourceColumn name="E" id="E"/>
        <valueColumn name="Value"/>
</table>
```

Sample, properties and qualifiers example

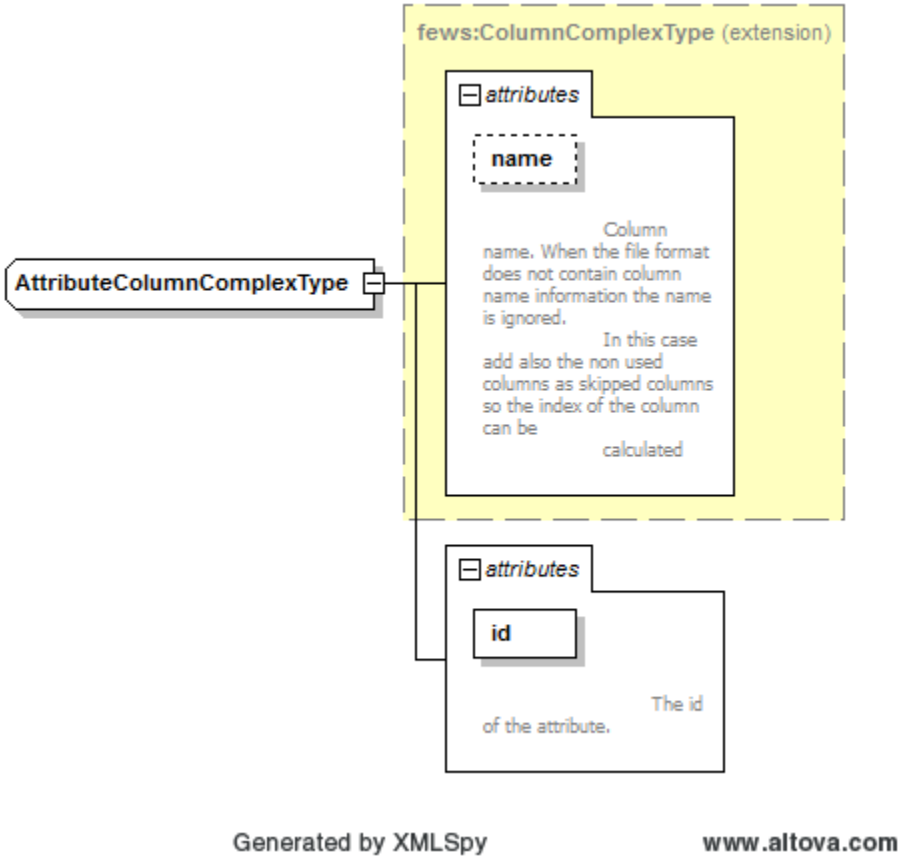
Since 2014.02 it is possible to use sampleIdColumn (normal ComplexColumnType with just name attribute), propertyColumn (with mandatory key attribute) and attribute column (with mandatory id attribute) which can only be used for export since attributes can not be imported but only configured. qualifierColumn is extended with the optional prefix attribute which will be added in front of the external qualifier id, this can be useful for id mapping.



Generated by XMLSpy www.altova.com



Generated by XMLSpy www.altova.com



Below configuration is shown which can import sample data, with properties and multiple qualifiers. In case sample data is imported the properties will be automatically used for the sample instead of the time step.

```
<table>
        <dateTimeColumn name="DATE_SMP" pattern="dd-MM-yy HH:mm"/>
        <locationColumn name="LOC_CODE"/>
        <unitColumn name="Eenheid"/>
        <parameterColumn name="PARAMETER_ID"/>
        <qualifierColumn name="TYPE" prefix="TYPE_"/>
        <qualifierColumn name="GESLACHT" prefix="GESLACHT_"/>
        <sampleIdColumn name="SMP_CODE"/>
        <propertyColumn name="EXT_REF" key="EXT_REF"/>
        <propertyColumn name="SMP_NAME" key="SMP_NAME"/>
        <propertyColumn name="BRON" key="BRON"/>
        <valueColumn name="Waarde"/>
</table>
```

Below configuration is shown which can export sample data, with properties, multiple qualifiers and attributes.

The qualifier columns use the prefix to recognise from the external id which qualifier should be put in the column. When the qualifier is found, the prefix will be stripped from the external id.

The attribute column will check the location, parameter and qualifiers for an attribute with the configured id and export the value to the column.

```
<table>
        <dateTimeColumn name="DATE_SMP" pattern="dd-MM-yy HH:mm"/>
        <locationColumn name="LOC_CODE"/>
        <unitColumn name="Eenheid"/>
        <parameterColumn name="PARAMETER_ID"/>
        <qualifierColumn name="GESLACHT" prefix="GESLACHT_"/>
        <qualifierColumn name="TYPE" prefix="TYPE_"/>
        <sampleIdColumn name="SMP_CODE"/>
        <propertyColumn name="SMP_NAME" key="SMP_NAME"/>
        <propertyColumn name="COST_CODE" key="COST_CODE"/>
        <propertyColumn name="BRON" key="BRON"/>
        <attributeColumn name="Groep" id="Groep"/>
        <valueColumn name="Waarde"/>
</table>
```