

General CSV Export

Overview

Exports time series data to files in CSV format with one header line containing column headers. The export type is *generalCSV*.

This feature has been implemented in the release 2014.02 (not working with older releases). Note that the generalCSV exporter only works for configurations similar to the given examples. As such, the exporter is handling CSV files in a general way, that's why it is the *generalCSV* parser. For dedicated features dedicated parser should be used.

Comment lines are automatically recognized when a line starts with the # character and completely skipped.

Single value per line (since 2014.02)

There are different kinds of formats supported, the first one is when there is just one value per line. All column types are supported for this, so this format is very suitable when there is a lot of information about the value to be exported.

Example:

```
# comment line
DATE_SMP;LOC_CODE;Eenheid;PARAMETER_ID;PAR_REF;PROD_CODE;ANAL_CODE;TYPE;SMP_CODE;COST_CODE;Groep;Waarde
01-01-81 00:00;MBP012;-;TansleyS_PTB;Nymphoides peltata;OW;PTB;gemeten;713;ECOLDOEL;ANGIO;4.0
01-01-81 00:00;MBP012;-;TansleyS_PTB;Phragmites australis;OW;PTB;gemeten;713;ECOLDOEL;ANGIO;8.0
01-01-81 00:00;MBP012;-;TansleyS_PTB;Polygonum amphibium;OW;PTB;gemeten;713;ECOLDOEL;ANGIO;4.0
01-01-81 00:00;MBP012;-;TansleyS_PTB;Polygonum persicaria;OW;PTB;gemeten;713;ECOLDOEL;ANGIO;1.0
01-01-81 00:00;MBP044;-;TansleyS_PTB;Nymphaea alba;OW;PTB;gemeten;712;ECOLDOEL;ANGIO;8.0
01-01-81 00:00;MBP044;-;TansleyS_PTB;Phragmites australis;OW;PTB;gemeten;712;ECOLDOEL;ANGIO;8.0
01-01-81 00:00;MBP044;-;TansleyS_PTB;Polygonum amphibium;OW;PTB;gemeten;712;ECOLDOEL;ANGIO;4.0
01-01-81 00:00;MBP044;-;TansleyS_PTB;Polygonum persicaria;OW;PTB;gemeten;712;ECOLDOEL;ANGIO;1.0
```

```
<timeSeriesExportRun xmlns="http://www.wldelft.nl/fews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews http://fews.wldelft.nl/schemas/version1.0/timeSeriesExportRun.
xsd">
  <export>
    <general>
      <exportType>generalCsv</exportType>
      <folder>${EXPORT_FOLDER}/HydroBiologie</folder>
      <exportFileName>
        <name>ExportGeneralCsv.csv</name>
      </exportFileName>
      <table>
        <dateTimeColumn name="DATE_SMP" pattern="dd-MM-yy HH:mm"/>
        <locationColumn name="LOC_CODE"/>
        <unitColumn name="Eenheid"/>
        <parameterColumn name="PARAMETER_ID"/>
        <qualifierColumn name="PAR_REF" prefix="PAR_REF"/>
        <qualifierColumn name="PROD_CODE" prefix="PROD_CODE"/>
        <qualifierColumn name="ANAL_CODE" prefix="ANAL_CODE"/>
        <qualifierColumn name="TYPE" prefix="TYPE"/>
        <sampleIdColumn name="SMP_CODE"/>
        <propertyColumn name="COST_CODE" key="COST_CODE"/>
        <attributeColumn name="Groep" id="Groep"/>
        <valueColumn name="Waarde"/>
      </table>
      <idMapId>IdExport_HydroBiologie</idMapId>
      <exportMissingValueString>-9999.0</exportMissingValueString>
    </general>
    <timeSeriesSet>
      ...
    </timeSeriesSet>
    <timeSeriesSet>
      ...
    </timeSeriesSet>
  </export>
</timeSeriesExportRun>
```

For more information of the table layout configuration see [Table Layout](#)

Multiple values per line (since 2016.01)

Since 2016.01 a different format is supported: it will be possible to use multiple values per line by specifying a parameter or location by each value. The next column types are not supported in this format because it will not be clear to which value (time series) they will belong:

FORECAST_DATE_TIME, FORECAST_DATE, FORECAST_TIME, QUALIFIER, COMMENT, FLAG, UNIT, SAMPLE_ID, PROPERTY, ATTRIBUTE, FLAG_SOURCE_COLUMN, LIMIT_SYMBOL

It also does not support the use of specific location and parameters for value columns at the same time.

When using parameter specific value columns, no <parameterColumn> is allowed. Or when using location specific value columns, no <locationColumn> is allowed.

Please note that when using parameter specific value column, the internal and external parameterIds need to be the same (in the IdMapping xml file).

Example for specifying parameters:

```
<timeSeriesExportRun xmlns="http://www.wldelft.nl/fews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews http://fews.wldelft.nl/schemas/version1.0/timeSeriesExportRun.
xsd">
  <export>
    <general>
      <exportType>generalCsv</exportType>
      <folder>$EXPORT_FOLDER$/Csv</folder>
      <exportFileName>
        <name>ExportGeneralCsv.csv</name>
      </exportFileName>
      <table>
        <dateTimeColumn name="DATE_SMP" pattern="yyyyMMddHHmm" />
        <locationColumn name="LOC" />
        <valueColumn name="PAR_A" parameterId="A" />
        <valueColumn name="PAR_B" parameterId="B" />
      </table>
      <idMapId>IdExport_HydroBiologie</idMapId>
      <exportMissingValueString>-9999.0</exportMissingValueString>
      <columnSeparator> </columnSeparator>
      <decimalSeparator>.</decimalSeparator>
    </general>
    <timeSeriesSet>
      ...
    </timeSeriesSet>
    <timeSeriesSet>
      ...
    </timeSeriesSet>
  </export>
</timeSeriesExportRun>
```

```
yyyyMMddHHmm LOC PAR_A PAR_B
201604180110 loc 721.0 485.1
201604180210 loc 955.7 648.5
201604184210 loc 647.0 582.3
201604180310 loc 352.9 260.9
201604180410 loc 259.8 186.8
```

Example for specifying locations:

```

<timeSeriesExportRun xmlns="http://www.wldelft.nl/fews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews http://fews.wldelft.nl/schemas/version1.0/timeSeriesExportRun.
xsd">
  <export>
    <general>
      <exportType>generalCsv</exportType>
      <folder>$EXPORT_FOLDER$/Csv</folder>
      <exportFileName>
        <name>ExportGeneralCsv.csv</name>
      </exportFileName>
      <table>
        <dateTimeColumn name="DATE_SMP" pattern="yyyyMMddHHmm" />
        <parameterColumn name="PAR" />
        <valueColumn name="LOC_A" locationId="A" />
        <valueColumn name="LOC_B" locationId="B" />
      </table>
      <idMapId>IdExport_HydroBiologie</idMapId>
      <exportMissingValueString>-9999.0</exportMissingValueString>
      <columnSeparator> </columnSeparator>
      <decimalSeparator>.</decimalSeparator>
    </general>
    <timeSeriesSet>
      ...
    </timeSeriesSet>
    <timeSeriesSet>
      ...
    </timeSeriesSet>
  </export>
</timeSeriesExportRun>

```

```

yyyyMMddHHmm PAR LOC_A LOC_B
201604180110 par 721.0 485.1
201604180210 par 955.7 648.5
201604184210 par 647.0 582.3
201604180310 par 352.9 260.9
201604180410 par 259.8 186.8

```

Defining column separator and decimal separator

Since 2016.01 it is possible to choose from multiple column separators: comma ",", or semi-colon ";", or pipe "|" or tab "	" or space " "

When specifying a column separator it is compulsory to also specify the decimal separator as comma ",", or point "."

```

<general>
  <exportType>generalCsv</exportType>
  <folder>../junit_test_output/nl/wldelft/fews/system/plugin/dataExport
/TimeSeriesExportTest/exportGeneralCsvColumnSeparator/export</folder>
  <exportFileName>
    <name>ExportGeneralCsvColumnSeparator.csv</name>
  </exportFileName>
  <table>
    <dateTimeColumn name="DateTime" pattern="yyyy-MM-dd HH:mm" />
    <locationColumn name="Location" />
    <parameterColumn name="Parameter" />
    <qualifierColumn name="QualifierId" prefix="CAL_" />
    <qualifierColumn name="QualifierId" prefix="ILC_" />
    <attributeColumn name="QualifierName" id="qualifierName" />
    <attributeColumn name="Type" id="type" />
    <attributeColumn name="Groep" id="groep" />
    <attributeColumn name="filter" id="filter" />
    <propertyColumn name="StringProperty" key="stringKey" />
    <propertyColumn name="SecondStringProperty" key="secondStringKey" />
    <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" />
    <unitColumn name="Unit" />
  </table>
  <idMapId>GeneralCsv</idMapId>
  <exportMissingValueString>-9999.0</exportMissingValueString>
  <columnSeparator>|</columnSeparator>
  <decimalSeparator>.</decimalSeparator>
</general>

```

This enables exporting for example the format below:

```

DateTime|Location|Parameter|QualifierId|QualifierId|QualifierName|Type|Groep|filter|StringProperty|SecondStringP
roperty|A|B|C|D|E|Value|Unit
2003-02-22 00:00|chj|mxgn|CALLPRAU|ILCOCIMC|Callicorixa
praeusta|Macrofauna|INHET|1|StringValue|SecondStringValue|OK|OK|OK|OK|OK|0.0|m
2003-02-23 00:00|chj|mxgn|CALLPRAU|ILCOCIMC|Callicorixa
praeusta|Macrofauna|INHET|1|StringValue|SecondStringValue|OK|OK|OK|OK|OK|1.0|m
2003-02-24 00:00|chj|mxgn|CALLPRAU|ILCOCIMC|Callicorixa
praeusta|Macrofauna|INHET|1|StringValue|SecondStringValue|OK|OK|OK|OK|OK|2.0|m
2003-02-25 00:00|chj|mxgn|CALLPRAU|ILCOCIMC|Callicorixa
praeusta|Macrofauna|INHET|1|StringValue|SecondStringValue|OK|OK|OK|OK|OK|3.0|m
2003-02-26 00:00|chj|mxgn|CALLPRAU|ILCOCIMC|Callicorixa
praeusta|Macrofauna|INHET|1|StringValue|SecondStringValue|OK|OK|OK|OK|OK|4.0|m
2003-02-27 00:00|chj|mxgn|CALLPRAU|ILCOCIMC|Callicorixa
praeusta|Macrofauna|INHET|1|StringValue|SecondStringValue|OK|OK|OK|OK|OK|5.0|m
2003-02-28 00:00|chj|mxgn|CALLPRAU|ILCOCIMC|Callicorixa
praeusta|Macrofauna|INHET|1|StringValue|SecondStringValue|OK|OK|OK|OK|OK|6.0|m
2003-03-01 00:00|chj|mxgn|CALLPRAU|ILCOCIMC|Callicorixa
praeusta|Macrofauna|INHET|1|StringValue|SecondStringValue|OK|OK|OK|OK|OK|7.0|m

```

Separate alphanumeric and numeric values in two separate columns

Since 2017.01 it is possible to separate alphanumeric and numeric values in two separate columns

```

<general>
  <exportType>generalCsv</exportType>
  <folder>${EXPORT_DIR}</folder>
  <exportFileName>
    <name>ExportGeneralCsv.csv</name>
  </exportFileName>
  <table>
    <dateTimeColumn name="DateTime" pattern="yyyy-MM-dd HH:mm" />
    <locationColumn name="Location" />
    <parameterColumn name="Parameter" />
    <valueColumn name="ValueOnly" ignoreForEnumerationParameters="true" />
    <valueColumn name="LabelOnly" ignoreForNumericalParameters="true" />
    <unitColumn name="Unit" />
  </table>
  <idMapId>GeneralCsv</idMapId>
  <exportMissingValueString/>
  <convertValuesToEnumerationLabels>true</convertValuesToEnumerationLabels>
</general>

```

Columns not available for this export type

userColumn, yearColumn, monthColumn, dayColumn, hourColumn, minuteColumn, secondColumn

Ensemble and ensemble member columns

Since 2019.02 it is possible to export ensemble and ensemble member

```

<general>
  <exportType>generalCsv</exportType>
  <folder>${EXPORT_FOLDER}</folder>
  <exportFileName>
    <name>ExportGeneralCsv.csv</name>
  </exportFileName>
  <table>
    <dateTimeColumn name="DateTime" pattern="yyyy-MM-dd HH:mm" />
    <locationColumn name="Location" />
    <parameterColumn name="Parameter" />
    <ensembleColumn name="Ensemble" />
    <ensembleMemberColumn name="Ensemble Member" />
    <valueColumn name="Value" />
  </table>
</general>

```

Exporting separate year, month, day, hour, minute and/or second columns

While the yearColumn, monthColumn etc. are not available for this export, it is possible to export a csv with separate columns for year, month, day etc. This can be achieved by configuring a <dateTimeColumn>, <dateColumn> or <timeColumn> with pattern="yyyy", or pattern="MM" etc. You can configure several of these column elements with different patterns. An example configuration can be found below.

```

<table>
  <dateTimeColumn name="Year" pattern="yyyy" />
  <dateTimeColumn name="Month" pattern="MM" />
  <dateTimeColumn name="Day" pattern="dd" />
  <dateTimeColumn name="Hour" pattern="HH" />
  <dateTimeColumn name="Minute" pattern="mm" />
  <dateTimeColumn name="Second" pattern="ss" />
  <locationColumn name="Location" />
  <parameterColumn name="Parameter" />
  <valueColumn name="Value" />
  <unitColumn name="Unit" />
</table>

```