

# UmAquoCsvExport

## UM Aquo CSV export

This is an export that makes use of the custom export mechanism that is similar to the import mechanism described in [Custom import formats](#). The UmAquo CSV export consists of a text serializer, `UmAquoCsvTimeSeriesSerializer.java`, and a bin directory containing all dependant library files (obtain from build server).

The UmAquo standard is being managed by [Informatiehuis Water](#) and is used to exchange data with the Online Omgevingsloket. Besides the CSV format there also exists an XML version of the UmAquo standard. Informatiehuis water is responsible for maintaining the [Aquo-domaintables](#). These tables contain lists of standardised variable codes related to water management. These tables are meant to aid the exchange of data between organisations in the water-sector by means of a standard terminology.

The UmAquo CSV export assures that the export variables validate against these domaintables.

## TimeSeries CSV file format

The CSV export format is described in the document [SpecsImporterenCSV.doc](#) (which is in Dutch, as are the Domain tables). Here follows an example CSV export file:

```
Monster.identificatie;MeetPunt.identificatie;Grootheid.code;Typering.code;Parameter.groep;Parameter.code;
Parameter omschrijving;Eenheid.code;Hoedanigheid.code;Compartiment.code;Begindatum;Begintijd;Tijd_UTCoffset;
Limietsymbool;Numeriekewaarde;Alfanumeriekewaarde;Kwaliteitsoordeel.code# Exporting timeseries with header:
#528002 CONCTTE.Cl (mg/l) unknown.NVT.OW nonequidistant never main0#;528002;CONCTTE;;ChemischeStof;Cl;;mg/l;NVT;
OW;2012-02-11;11:00:00;0;;9.0;;00;528002;CONCTTE;;ChemischeStof;Cl;;mg/l;NVT;OW;2012-02-11;12:00:00;0;;8.0;;00;
528002;CONCTTE;;ChemischeStof;Cl;;mg/l;NVT;OW;2012-02-11;13:00:00;0;;8.0;;00;528002;CONCTTE;;ChemischeStof;Cl;;
mg/l;NVT;OW;2012-02-11;14:00:00;0;;8.0;;00;528002;CONCTTE;;ChemischeStof;Cl;;mg/l;NVT;OW;2012-02-11;15:00:00;0;;
-999.0;;99;528002;CONCTTE;;ChemischeStof;Cl;;mg/l;NVT;OW;2012-02-11;16:00:00;0;;-999.0;;99
&nbsp;# Exporting timeseries with header:#528002 CONCTTE.P (mg/l) unknown.NVT.OW nonequidistant never main0#;
528002;CONCTTE;;ChemischeStof;P;;mg/l;NVT;OW;2012-02-11;11:00:00;0;;9.0;;00;528002;CONCTTE;;ChemischeStof;P;;mg
/l;NVT;OW;2012-02-11;12:00:00;0;;9.0;;00;528002;CONCTTE;;ChemischeStof;P;;mg/l;NVT;OW;2012-02-11;13:00:00;0;;
9.0;;00;528002;CONCTTE;;ChemischeStof;P;;mg/l;NVT;OW;2012-02-11;14:00:00;0;;9.0;;00;528002;CONCTTE;;
ChemischeStof;P;;mg/l;NVT;OW;2012-02-11;15:00:00;0;;9.0;;00;528002;CONCTTE;;ChemischeStof;P;;mg/l;NVT;OW;2012-
02-11;16:00:00;0;;9.0;;00
&nbsp;# Exporting timeseries with header:#528001 CONCTTE.N (mg/l) unknown.NVT.OW nonequidistant never main0#;
528001;CONCTTE;;ChemischeStof;N;;mg/l;NVT;OW;2012-02-11;11:00:00;0;;9.0;;00;528001;CONCTTE;;ChemischeStof;N;;mg
/l;NVT;OW;2012-02-11;12:00:00;0;;9.0;;00;528001;CONCTTE;;ChemischeStof;N;;mg/l;NVT;OW;2012-02-11;13:00:00;0;;
9.0;;00;528001;CONCTTE;;ChemischeStof;N;;mg/l;NVT;OW;2012-02-11;14:00:00;0;;9.0;;00;528001;CONCTTE;;
ChemischeStof;N;;mg/l;NVT;OW;2012-02-11;15:00:00;0;;9.0;;00;528001;CONCTTE;;ChemischeStof;N;;mg/l;NVT;OW;2012-
02-11;16:00:00;0;;9.0;;00
&nbsp;# Exporting timeseries with header:#528001 WATDTE (m) unknown.NVT.OW nonequidistant never main0#;528001;
WATDTE;;;m;NVT;OW;2012-02-11;11:00:00;0;;1.8;;00;528001;WATDTE;;;m;NVT;OW;2012-02-11;12:00:00;0;;2;;00;
528001;WATDTE;;;m;NVT;OW;2012-02-11;13:00:00;0;;2.1;;00;528001;WATDTE;;;m;NVT;OW;2012-02-11;14:00:00;0;;
1.9;;00;528001;WATDTE;;;m;NVT;OW;2012-02-11;15:00:00;0;;1.9;;00;528001;WATDTE;;;m;NVT;OW;2012-02-11;16:00:
00;0;;1.8;;00
&nbsp;# Exporting timeseries with header:#528005 WATDTE (m) unknown.NVT.OW nonequidistant never main0#;528005;
WATDTE;;;m;NVT;OW;2012-02-11;11:00:00;0;;1.8;;00;528005;WATDTE;;;m;NVT;OW;2012-02-11;12:00:00;0;;2;;00;
528005;WATDTE;;;m;NVT;OW;2012-02-11;13:00:00;0;;2.1;;00;528005;WATDTE;;;m;NVT;OW;2012-02-11;14:00:00;0;;
1.9;;00;528005;WATDTE;;;m;NVT;OW;2012-02-11;15:00:00;0;;1.9;;00;528005;WATDTE;;;m;NVT;OW;2012-02-11;16:00:
00;0;;1.8;;00
&nbsp;# Exporting timeseries with header:#528001 T (°C) unknown.NVT.OW nonequidistant never main0#;528001;T;;;
oC;NVT;OW;2012-02-11;11:00:00;0;;1.8;;00;528001;T;;;oC;NVT;OW;2012-02-11;12:00:00;0;;2;;00;528001;T;;;oC;
NVT;OW;2012-02-11;13:00:00;0;;2.1;;00;528001;T;;;oC;NVT;OW;2012-02-11;14:00:00;0;;1.9;;00;528001;T;;;oC;NVT;
OW;2012-02-11;15:00:00;0;;1.9;;00;528001;T;;;oC;NVT;OW;2012-02-11;16:00:00;0;;1.8;;00
```

### Output CSV timeseries headers

MeetPunt.code: contains location id  
Grootheid.code: value always "DUMMY"  
Parameter.code: contains parameter id.  
Eenheid.code: contains unit code  
Hoedanigheid.code: contains qualifier 1  
Compartment.code: contains qualifier 2  
Waardebepalingsmethode.code: contains qualifier 3  
Begindatum: contains date part of timestamp  
Begintijd: contains time part of timestamp  
Limietsymbool: contains OutOfDetectionRangeFlag value  
Numeriekewaarde: contains timestep value  
Alfanumeriekewaarde: empty  
Kwaliteitsoordeel.code: contains timestep flag value  
Comment: contains timestep comment value (not part of UmAquo standard)

### Date and time formats

Format Begindatum: *yyyyMMdd*  
Format Begintijd: *HH:mm:ss*

### Syntax

Commentline prefix: '#'  
Column separator: ','  
Decimal separator: '.'

## Validation

Validation of the values in the UmAquo import files is done against the [Aquo schemas](#) . Instead of validating over the web, validation is done locally by validating against the schemas present in the resource file [UmAquo\\_schemas.jar](#) . This archive contains a subset of schemas from the Aquo schemas site. If the Aquo schemas are updated then a new version of the schemas jar must be distributed.

## Fews configuration

In order to activate the UmAquo CSV export as a FEWS export, it is required to setup a **TimeSeriesExportRun** module configuration file and an accompanying **IdMap** file and **FlagConversion** file. Also the bin directory containing all UmAquo resources must be placed in a location that can be accessed by the FEWS system.

Here is an example export module configuration file:

```

<?xml version="1.0" encoding="UTF-8"?>
<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">
    <!-- This is an example export configuration file for exporting data from as Umaquo CSV file -->
    <export>
        <general>
            <serializerClassName>nl.wldelft.webservice.umaquo.timeseriesserializers.
UmaquoCsvTimeSeriesSerializer</serializerClassName>
            <binDir>%REGION_HOME%/Modules/umaquo-bin</binDir>
            <folder>$EXPORT_FOLDER$/Umaquo/CSV</folder>
        <exportFileName>
            <name>umaquo.csv</name>
            <prefix>
                <timeZeroFormattingString>yyyyMMddHHmm</timeZeroFormattingString>
            </prefix>
        </exportFileName>
            <idMapId>IdExportUmaquo</idMapId>
        <flagConversionsId>ExportFlagConversion</flagConversionsId>
            <exportMissingValue>-999</exportMissingValue>
            <omitMissingValues>true</omitMissingValues>
        </general>
        <properties>
            <!-- Optional: comment line prefix character. Default = # -->
            <string key="COMMENT_PREFIX" value="#"/>
            <!-- Optional: column separator character. Default = ; -->
            <string key="COLUMN_SEPARATOR" value=";"/>
            <!-- Optional: decimal character. Default = . -->
            <string key="DECIMAL_SEPARATOR" value="."/>
            <!-- Optional: Regular expression for date value. Default = yyyy-MM-dd -->
            <string key="CSV_DATEPATTERN" value=";"/>
            <!-- Optional: Regular expression for time value. Default = HH:mm:ss -->
            <string key="CSV_TIMEPATTERN" value="HH:mm:ss"/>
            <!-- Optional: select Umaquo schema version 2009 or 2011. Default = 2009 -->
            <int key="SCHEMA_VERSION" value="2009"/>
            <!-- Optional: allow lenient validation. Default = false -->
            <bool key="
LENIENT" value="true"/>
            <!-- Optional: setting to export location info. Default = false -->
            <bool key="WRITELOCATIONINFO" value="true"/>
        </properties>
        <timeSeriesSet>
            <moduleInstanceId>ImportUmaquo</moduleInstanceId>
            <valueType>scalar</valueType>
            <parameterId>MyPar</parameterId>
            <locationSetId>MyLocSet</locationSetId>
            <timeSeriesType>external historical</timeSeriesType>
            <timeStep unit="nonequidistant"/>
            <relativeViewPeriod unit="hour" start="-5" end="0"/>
            <readWriteMode>add originals</readWriteMode>
        </timeSeriesSet>
    </export>
</timeSeriesExportRun>

```

Here is an example id-map file:

```

<?xml version="1.0" encoding="UTF-8"?>
<idMap 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/idMap.xsd" version="1.1">
  <!-- Parameter id mapping can be done in three ways -->
  <!-- 1st: Concatenate GrootheidCode and ParameterCode with '.' in between -->
  <map internalLocation="LOC-001" internalParameter="PAR-001" externalLocation="Meetpunt"
    externalParameter="GrootheidCode.ParameterCode" externalQualifier="EenheidCode" externalQualifier1="
HoedanigheidCode" externalQualifier2="CompartimentCode" />
  <!-- in the case of Biotaxon use ParameterName instead of ParameterCode-->
  <map internalLocation="LOC-001" internalParameter="PAR-001" externalLocation="Meetpunt"
    externalParameter="GrootheidCode.ParameterName" externalQualifier="EenheidCode" externalQualifier1="
HoedanigheidCode" externalQualifier2="CompartimentCode" />
  <!-- 2nd: If only GrootheidCode then use this as external parameter -->
  <map internalLocation="LOC-001" internalParameter="PAR-002" externalLocation="Meetpunt"
    externalParameter="GrootheidCode" externalQualifier="EenheidCode" externalQualifier1="
HoedanigheidCode" externalQualifier2="CompartimentCode" />
  <!-- 3rd: If only TyperingCode then use this as external parameter -->
  <map internalLocation="LOC-001" internalParameter="PAR-003" externalLocation="Meetpunt"
    externalParameter="TyperingCode" externalQualifier="EenheidCode" externalQualifier1="HoedanigheidCode"
externalQualifier2="CompartimentCode" />
  <!-- 4th: If WaardeBepalingsMethode used then use this as external parameter -->
  <map internalLocation="LOC-001" internalParameter="PAR-003" externalLocation="Meetpunt"
    externalParameter="TyperingCode" externalQualifier="EenheidCode" externalQualifier1="
HoedanigheidCode" externalQualifier2="CompartimentCode"
    externalQualifier3="WaardeBepalingsMethodeCode" />

</idMap>

```

Id-Mapping goes as follows:

Internal timeseries location and parameter combinations are mapped to external location, parameter and qualifier sets as follows:

- externalLocation values are written to the Meetpunt column
- externalParameter values are written to the Parameter column
- 1st externalQualifier values are written to the Eenheid column
- 2nd externalQualifier values are written to the Hoedanigheid column
- 3rd externalQualifier values are written to the Compartiment column
- 4th externalQualifier values are written to the Waardebepalingsmethode column

Here is an example flag conversion file:

```

<?xml version="1.0" encoding="UTF-8"?>
<flagConversions 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/flagConversions.xsd">
  <flagConversion>
    <inputFlag>      <value>0</value></inputFlag>
    <outputFlag>     <value>0</value></outputFlag>
  </flagConversion>
  <flagConversion>
    <inputFlag>      <value>1</value></inputFlag>
    <outputFlag>     <value>0</value></outputFlag>
  </flagConversion>
  <flagConversion>
    <inputFlag>      <value>2</value></inputFlag>
    <outputFlag>     <value>0</value></outputFlag>
  </flagConversion>
  <flagConversion>
    <inputFlag>      <value>3</value></inputFlag>
    <outputFlag>     <value>50</value></outputFlag>
  </flagConversion>
  <flagConversion>
    <inputFlag>      <value>4</value></inputFlag>
    <outputFlag>     <value>50</value></outputFlag>
  </flagConversion>
  <flagConversion>
    <inputFlag>      <value>5</value></inputFlag>
    <outputFlag>     <value>50</value></outputFlag>
  </flagConversion>
  <flagConversion>
    <inputFlag>      <value>6</value></inputFlag>
    <outputFlag>     <value>99</value></outputFlag>
  </flagConversion>
  <flagConversion>
    <inputFlag>      <value>7</value></inputFlag>
    <outputFlag>     <value>50</value></outputFlag>
  </flagConversion>
  <flagConversion>
    <inputFlag>      <value>8</value></inputFlag>
    <outputFlag>     <value>50</value></outputFlag>
  </flagConversion>
  <flagConversion>
    <inputFlag>      <value>9</value></inputFlag>
    <outputFlag>     <value>99</value></outputFlag>
  </flagConversion>
  <defaultOutputFlag><value>0</value></defaultOutputFlag>
  <missingValueFlag><value>99</value></missingValueFlag>
</flagConversions>

```

## Exporting UMaquo CSV Location info

By configuring the property *writeLocationInfo* in the *timeSeriesExportRun* file, it is also possible to export an additional CSV file containing location information.

The contents of this file is also described in the document [SpecsImporterCSV.doc](#)