

UmAquoXmlExport

UM Aquo XML 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 XML export consists of a xml serializer, UmAquoXmlTimeSeriesSerializer.java, and a bin directory containing all dependant library files ([umaquo.zip](#)).

The UmAquo standard is being managed by [Informatiehuis Water](#) and is used to exchange data with the Online Omgevingsloket. Besides the Xml format there also exists a Csv 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 XML export assures that the export variables validate against these domaintables.

TimeSeries XML file format

The XML import format is described by the schema files that can be found [here](#). Here follows an example XML export file:

[201202131600umaquo.xml](#)

Expected XML elements:

MeetObject: contains location information

MonsterObject: Forms the link between the MeetObject and the timeseries

WaardeReeksTijd: Elements containing timeseries data

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 XMLfile -->
    <export>
        <general>
            <serializerClassName>nl.wldelft.webservice.umaquo.timeseriesserializers.
UmaquoXmlTimeSeriesSerializer</serializerClassName>
            <binDir>%REGION_HOME%/Modules/umaquo-bin</binDir>
            <folder>$EXPORT_FOLDER$/Umaquo/CSV</folder>
        <exportFileName>
            <name>umaquo.xml</name>
            <prefix>
                <timeZeroFormattingString>yyyyMMddHHmm</timeZeroFormattingString>
            </prefix>
        </exportFileName>
            <idMapId>IdExportUmaquo</idMapId>
        <flagConversionsId>ExportFlagConversion</flagConversionsId>
            <exportMissingValue>-999</exportMissingValue>
            <omitMissingValues>true</omitMissingValues>
        </general>
        <properties>
            <!-- Optional: select Umaquo schema version 2009 or 2011. Default = 2009 -->
            <int key="SCHEMA_VERSION" value="2009"/>
            <bool key="LENIENT" value="true" />
        </properties>
        <timeSeriesSet>
            <moduleId>ImportUmaquo</moduleId>
            <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:

For the Umaquo XML export it is important that the external location ids are mapped to a valid Umaquo format:

```
NL\.umam\.\d{2}\.\. {1,49}
```

This can be done using the **locationIdPattern** option provided by FEWS id mapping. It is also possible to export location in a different format by setting the export property **LENIENT** to **true**.

```

<?xml version="1.0" encoding="UTF-8"?>
<idMap version="1.1" 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">
  <!--external: Grootheden-->
  <!--externalQualifier: eenheid-->
  <!--externalQualifier1: Hoedanigheid-->
  <!--externalQualifier2: Compartment-->
  <!--externalQualifier3: Waardebepalingsmethode-->
  <parameter internal="NEERSG" external="NEERSG" externalQualifier="mm" externalQualifier1="NVT"
externalQualifier2="LT"/>
  <parameter internal="VERDPG" external="VERDPG" externalQualifier="mm" externalQualifier1="NVT"
externalQualifier2="LT"/>
  <parameter internal="WATHTE" external="WATHTE" externalQualifier="m" externalQualifier1="NVT"
externalQualifier2="OW"/>
  <parameter internal="WATDTE" external="WATDTE" externalQualifier="m" externalQualifier1="NVT"
externalQualifier2="OW"/>
  <parameter internal="Q" external="Q" externalQualifier="m3/d" externalQualifier1="NVT"
externalQualifier2="OW"/>
  <parameter internal="Ntot" external="CONCTTE.Ntot" externalQualifier="mg/l" externalQualifier1="NVT"
externalQualifier2="OW"/>
  <parameter internal="Ptot" external="CONCTTE.Ptot" externalQualifier="mg/l" externalQualifier1="NVT"
externalQualifier2="OW"/>
  <parameter internal="Ptot" external="CONCTTE.Ptot" externalQualifier="mg/l" externalQualifier1="NVT"
externalQualifier2="OW" externalQualifier3="I10048.91"/>
  <locationIdPattern internalLocationSet="Meetnet" internalLocationPattern="*" externalLocationPattern="
NL.umam.00.*" />
</idMap>

```

Here is an example flag conversion file:

[please see UmAquoCsvExport](#)