SWAN adapter

SWAN pre-adapter

Model pre-adapter for running SWAN (Simulating WAves Nearshore) model from Delft-FEWS.
For information about the SWAN model see http://www.swan.tudelft.nl/

Usage: SwanPreAdapter <netcdf run file pathname relative to current working directory>

Class name: nl.deltares.swan.SwanPreAdapter

Properties

template_swan_input_file	(required)	Pathname of template swan input file to use to create swan input file. This should be either an absolute path or a path relative to the workDir specified in the netcdf run file.
swan_input_file	(required)	Pathname of swan input file to create. This should be either an absolute path or a path relative to the workDir specified in the netcdf run file.
input_wind_file_to_convert	(optional)	Convert wind (with u, v component) in **regular** grid from NetCDF to SWAN ascii format. The path to the wind NetCDF file should be specified in the value section. The conversion is done **with** grid rotation, thus in SWAN .swn file, one should specify how this wind grid should be read by adjusting 'idla' (e.g. idla=3, read the grid starting from the low-left corner). 'windy.ser' is required in SWAN .swn file. Standard names should be given to the wind parameter in NetCDF file: eastward_wind (wind_u), northward_wind (wind_v); Conversion examples are attached. windy.ser; windyfield_18071203.wnd; SWAN_wind.nc; template.swn
input_curvilinear_grid_file_to_convert_waterlevel	(optional)	Convert waterlevel in **curvelinear** grid from NetCDF to SWAN ascii format (NOHEAD version). The path to the waterlevel NetCDF file should be specified in the value section. T he conversion is done **without** grid rotation, thus in SWAN .swn file, water level grid should be read with 'idla' = 1(i.e. read the grid starting from the top-left corner, from left to right).
input_curvilinear_grid_file_to_convert_currents	(optional)	Convert currents (with u. v component) in **curvelinear** grid from NetCDF to SWAN ascii format (NOHEAD version). The ascii file is organised by timestamp (i.e. 1st timestep u, v; 2nd timestep u, v, etc). The path to the currents NetCDF file should be specified in the value section. The conversion is done **without** grid rotation, thus in SWAN .swn file, currents grid should be read by with 'idla' =1 (i.e. read the grid starting from the top-left corner, from left to right).

input_boundary_file_to_convert	(optional)	Convert wave boundaries (scalar timeseries, e.g. WaveWatch III) from NetCDF to SWAN ascii format (.bnd file). The path to the boundary NetCDF file should be specified in the value section. Three copies of the .bnd file will be created for each time series in the NetCDF file, due to the input boundary requirement of SWAN. WW3_33.75_-26.bnd; WW3_33.75_-26_2.bnd; WW3_33.75_-26_3.bnd; Beira_Wave_bnd.nc; template.swn
input_spectra_file_to_convert	(optional)	Adapter can only import data on WGS84 coordinates. Pathname of netcdf file with input spectra data that should be converted from Cartesian to WGS84. This should be either an absolute path or a path relative to the workDir specified in the netcdf run file.
stationary_times_file	(optional)	Pathname of a netcdf file that contains a parameter with times to use for the $STATIONARY_TIMES$ tag. This should be either an absolute path or a path relative to the workDir specified in the netcdf run file. This property must be present if the $STATIONARY_TIMES$ tag is used in the template swan input file.
stationary_times_parameter	(optional)	Variable name of a parameter in stationary_times_file with data, e.g. wind_speed. The times that correspond to the data for this parameter are used for the $STATIONARY_TIMES$ tag. This property must be present if the $STATIONARY_TIMES$ tag is used in the template swan input file.
input_state_file_format	(optional)	Deprecated, do not use. Format of the input state file (hotfile). Can be e.g. "NETCDF" (NetCDF format), "FREE" (Ascii format) or "UNFORMATTED" (binary format). If not specified, then NETCDF is used by default.
netcdf_files_convert_coordinates_attribute	(optional)	Property to be used when the input data for SWAN is defined on a Cartesian grid. Property replaces coordinate attributes in netcdf file from "lat lon" to "x y". Configure as "value" a list of filenames to be converted, separated a ; symbol. List of filenames should be in the input dir (next to the work dir). For these set of files, new files will be created "<name>_adapted.nc" where the coordinate attributes "lat lon" are replaced by "x y".

Notes for users

- For all files that are written by this adapter, if the file to be written already exists, then it will be overwritten.
- This program assumes that the model always runs in time zone GMT.
- This program writes log messages to a log file called swan_pre_adapter_log.txt in the workDir specified in the netcdf run file.
- This program uses the information in the specified netcdf run file as input and uses this information to do the following actions:
    1. Convert input spectra time series (optional):
        a. The netcdf file specified in the property "input_spectra_file_to_convert" will be converted to a file in SWAN ascii format with the same path and name but different extension (.BND). For example the original file input/spectra.nc is converted to input/spectra.BND. If the property "input_spectra_file_to_convert" is not specified, then this step does nothing.
        b. The follow order of boundary locations in the .BND file is important for SWAN. The pre-adapter writes the .BND file according to the locations ID's, sorted alphabetically. It is important to take this into consideration when defining the ID's of the boundary locations. Example: locations with ID's loc_1, loc_2, ..., loc_10, loc_11, etc. will be sorted in .BND as loc_1, loc_10, loc_11, ... loc_2. To have it sorted correctly, use loc_01, loc_02, ..., loc_10, loc_11, etc.
    2. Create swan input file:
       The template SWAN input file specified in the property "template_swan_input_file" will be copied to the file specified in the property "swan_input_file". In the copy the following tags will be replaced with the corresponding values:

$TSTART$	Start time of the model run.
$TSTOP$	End time of the model run.
$HOTSTART$	Either "INITIAL HOTSTART 'pathname of input state file' NETCDF" if the input state file is not empty (warm state start) or empty string if the input state file is an empty dummy file of 0 bytes length (cold state start) or empty string if there is no input state file at all (cold state start). If the property "input_state_file_format" is specified, the the value of the property "input_state_file_format" is used instead of "NETCDF" in the line above.

$STATIONARY_TIMES$	Multiple lines with time stamps for a stationary model run: COMPUTE STATIONARY <first time stamp for stationary model run> COMPUTE STATIONARY <second time stamp for stationary model run> ... COMPUTE STATIONARY <last time stamp for stationary model run>

System requirements

- This program needs Java version 8 or higher.
- This program needs the following Java libraries:
  - commons-httpclient-3.0.1.jar
  - Delft_FEWS.jar
  - Delft_Util.jar
  - log4j-1.2.14.jar
  - netcdf-4.2.jar
  - slf4j-api-1.5.6.jar
  - slf4j-log4j12-1.5.6.jar
  - TimeSeriesImport.jar

SWAN post-adapter

Model post-adapter for running SWAN (Simulating WAves Nearshore) model from Delft-FEWS.
For information about the SWAN model see http://www.swan.tudelft.nl/

Usage: SwanPostAdapter <netcdf run file pathname relative to current working directory>

Class name: nl.deltares.swan.SwanPostAdapter

Properties

output_netcdf_spectra_file_to_convert	(optional)	Path to netcdf spectra file to convert. A copy of the file will be created with <name>_adapted.nc which adds "lat" "lon" coordinate variables (according to the CF conventions) and multiplies the density variable with the scale_denstity variable.
output_ascii_grid_file_to_convert	(optional)	Convert output map file from ascii (HEADER, Layout 1) to NetCDF. The path to the map ascii file should be specified in the value section. Examples are attached. overall.asc; overall.nc
output_ascii_grid_swap_north_south	(optional)	Useful when the output SWAN grid is swapped compare to the SWAN grid definition in FEWS. The swap occurs during the file conversion from Ascii to Netcdf. Usage: <bool key=" output_ascii_grid_swap_north_south" value="true"/>, True: the output grid will be swapped north to south; False: no swap
TAB file conversion (not a keyword)	(hardcoded)	see description in the **Notes for users**. Examples are attached. overallout.nc; overallout.PNT; overallout.tab

Notes for users

- For all files that are written by this adapter, if the file to be written already exists, then it will be overwritten.
- This program assumes that the model always runs in time zone GMT.
- This program writes log messages to a log file called swan_post_adapter_log.txt in the workDir specified in the netcdf run file.
- This program uses the information in the specified netcdf run file as input and uses this information to do the following actions:
  1. Convert output scalar time series:
     All .TAB files in the workDir specified in the netcdf run file will be converted to files in netcdf format with the same path and name but different extension (.nc). For this conversion the mapping between coordinates and locationIds from the corresponding .PNT files is used. This program assumes that for a given .TAB file the corresponding .PNT file has the same path and name as the .TAB file, only a different extension. This program assumes that the locations in the .TAB file and in the corresponding .PNT file are in the same order. This program assumes that the last part between double quotes on a given line in a .PNT file is the locationId. This program assumes that the coordinates in a given .TAB file are in degrees in the WGS 1984 coordinate system. If no .TAB files are present in the specified workDir, then this step does nothing.

System requirements

- This program needs Java version 8 or higher.
- This program needs the following Java libraries:
  - commons-httpclient-3.0.1.jar
  - Delft_FEWS.jar
  - Delft_Util.jar
  - log4j-1.2.14.jar
  - netcdf-4.2.jar
  - slf4j-api-1.5.6.jar

- slf4j-log4j12-1.5.6.jar
- TimeSeriesImport.jar

GeneralAdapterRun Example Configuration

The following gives an example of how to set up the GeneralAdapterRun file for SWAN in FEWS using the SWAN model pre and post-adapters. The GeneralAdapterRun file follows the general structure as described here.

General

In this section general information regarding the module such as version number, file directories, missing values, and time zone information can be specified.

general

```
<general>
        <description>SWAN model run</description>
        <piVersion>1.8</piVersion>
        <rootDir>%REGION_HOME%/Modules/Swan/$MODEL$</rootDir>
        <workDir>%ROOT_DIR%/workDir</workDir>
        <exportDir>%ROOT_DIR%/input</exportDir>
        <exportDataSetDir>%ROOT_DIR%/</exportDataSetDir>
        <exportIdMap>IdExport_$MODEL$</exportIdMap>
        <importDir>%ROOT_DIR%/workDir</importDir>
        <importIdMap>IdImport_$MODEL$</importIdMap>
        <importUnitConversionsId>ImportUnitConversions</importUnitConversionsId>
        <dumpFileDir>$GA_DUMPFILEDIR$</dumpFileDir>
        <dumpDir>%ROOT_DIR%/diagnostics/</dumpDir>
        <diagnosticFile>%ROOT_DIR%/diagnostics/diagnostics.xml</diagnosticFile>
        <missVal>-999</missVal>
        <timeZone>
                <timeZoneName>GMT+0:00</timeZoneName>
        </timeZone>
</general>
```

Start-up activities

It may be useful to clear the model working directory of any previous runs before starting a new run.

startUpActivities

```
                <startUpActivities>
                        <purgeActivity>
                                <filter>%ROOT_DIR%/workDir/*</filter>
                        </purgeActivity>
                </startUpActivities>
```

Export activities

In this section the data to be exported from FEWS as input to the module is specified. Data to export to SWAN generally includes:

- Model state
- Model data set
- Input data (i.e. wind, water levels, wave spectra)
- Run file

The run file contains information regarding the input file names, start and stop times, and time step. Additional properties can be passed using the run file as listed above under Properties.

exportActivities

```
<exportActivities>
                        <exportStateActivity>
                                <moduleInstanceId>$MODULE_INSTANCE_ID$</moduleInstanceId>
                                <stateExportDir>%ROOT_DIR%/stateInput</stateExportDir>
                                <stateSelection>
                                        <warmState>
```

```
                                        <stateSearchPeriod unit="hour" start="-1" end="0"/>
                                </warmState>
                        </stateSelection>
                </exportStateActivity>
                <exportDataSetActivity>
                        <moduleInstanceId>Swan_$MODEL$_$METEO$</moduleInstanceId>
                </exportDataSetActivity>
                <exportNetcdfActivity>
                        <exportFile>waterlevel.nc</exportFile>
                        <timeSeriesSets>
                                <timeSeriesSet>
                                        <moduleInstanceId>D3D_flow_dcsm5_hirlam72_hc</moduleInstanceId>
                                        <valueType>grid</valueType>
                                        <parameterId>H.simulated</parameterId>
                                        <locationId>dcsm5</locationId>
                                        <timeSeriesType>simulated historical</timeSeriesType>
                                        <timeStep unit="hour"/>
                                        <relativeViewPeriod unit="hour" start="-1" startOverrulable="
true" end="0"/>
                                        <readWriteMode>read only</readWriteMode>
                                        <synchLevel>2</synchLevel>
                                </timeSeriesSet>
                        </timeSeriesSets>
                </exportNetcdfActivity>
                <exportNetcdfActivity>
                        <exportFile>$METEO$.nc</exportFile>
                        <timeSeriesSets>
                                <timeSeriesSet>
                                        <moduleInstanceId>Import_$METEO$</moduleInstanceId>
                                        <valueType>grid</valueType>
                                        <parameterId>Wind.u.simulated</parameterId>
                                        <locationId>$METEO$_regular</locationId>
                                        <timeSeriesType>external historical</timeSeriesType>
                                        <timeStep unit="hour" multiplier="1"/>
                                        <relativeViewPeriod unit="hour" start="-1" startOverrulable="
true" end="0"/>
                                        <readWriteMode>read only</readWriteMode>
                                </timeSeriesSet>
                                <timeSeriesSet>
                                        <moduleInstanceId>Import_$METEO$</moduleInstanceId>
                                        <valueType>grid</valueType>
                                        <parameterId>Wind.v.simulated</parameterId>
                                        <locationId>$METEO$_regular</locationId>
                                        <timeSeriesType>external historical</timeSeriesType>
                                        <timeStep unit="hour" multiplier="1"/>
                                        <relativeViewPeriod unit="hour" start="-1" startOverrulable="
true" end="0"/>
                                        <readWriteMode>read only</readWriteMode>
                                </timeSeriesSet>
                        </timeSeriesSets>
                </exportNetcdfActivity>
                <exportNetcdfActivity>
                        <exportFile>ecmwf_spectra.nc</exportFile>
                        <timeSeriesSets>
                                <timeSeriesSet>
                                        <moduleInstanceId>Import_ecmwf_spectra</moduleInstanceId>
                                        <valueType>scalar</valueType>
                                        <parameterId>Wave.variancedensity2D.simulated</parameterId>
                                        <domainParameterId>f</domainParameterId>
                                        <domainParameterId>dir</domainParameterId>
                                        <locationSetId>ecmwf_spectra.locations</locationSetId>
                                        <timeSeriesType>external historical</timeSeriesType>
                                        <timeStep unit="hour" multiplier="6"/>
                                        <relativeViewPeriod unit="hour" end="0"/>
                                        <readWriteMode>read only</readWriteMode>
                                </timeSeriesSet>
                        </timeSeriesSets>
                </exportNetcdfActivity>
                <exportNetcdfRunFileActivity>
                        <description>This run file is passed as argument to SwanPreAdapter</description>
                        <exportFile>%WORK_DIR%\run_info.nc</exportFile>
```

```
                                <properties>
                                        <string key="input_spectra_file_to_convert" value=".. 
\input\ecmwf_spectra.nc"/>
                                        <string key="template_swan_input_file" value="..\template\template.swn" 
/>
                                        <string key="swan_input_file" value="INPUT"/>
                                </properties>
                        </exportNetcdfRunFileActivity>
                </exportActivities>
```

Execute activities

This section calls the SWAN pre and post-adapters as well as the SWAN executable. Note: the run file must be passed as an argument to the SWAN pre and post-adapters.

executeActivities

```
<executeActivities>
                        <executeActivity>
                                <command>
                                        <className>nl.deltares.swan.SwanPreAdapter</className>
                                        <binDir>..\..\..\bin\Swan_adapter\bin</binDir>
                                </command>
                                <arguments>
                                        <argument>%WORK_DIR%\run_info.nc</argument>
                                </arguments>
                                <timeOut>99999999</timeOut>
                                <ignoreDiagnostics>true</ignoreDiagnostics>
                        </executeActivity>
                        <executeActivity>
                                <command>
                                        <executable>..\..\..\bin\Swan\swan_4091AB_3_del_win_vs2010.exe<
/executable>
                                </command>
                                <arguments>
                                        <argument>%WORK_DIR%\INPUT</argument>
                                </arguments>
                                <timeOut>99999999</timeOut>
                                <ignoreDiagnostics>true</ignoreDiagnostics>
                        </executeActivity>
                        <executeActivity>
                                <command>
                                        <className>nl.deltares.swan.SwanPostAdapter</className>
                                        <binDir>..\..\..\bin\Swan_adapter\bin</binDir>
                                </command>
                                <arguments>
                                        <argument>%WORK_DIR%\run_info.nc</argument>
                                </arguments>
                                <timeOut>99999999</timeOut>
                                <ignoreDiagnostics>true</ignoreDiagnostics>
                        </executeActivity>
                </executeActivities>
```

Import activities

In this section the data to be imported into FEWS as output from the module is specified. Data to import from SWAN generally includes:

- Model state
- Output data (i.e. wind, water levels, wave spectra)

importActivities

```
<importActivities>
                        <importStateActivity>
                                <stateFile>
                                        <importFile>%ROOT_DIR%/workDir/swan-state.nc</importFile>
                                        <relativeExportFile>swan-restart.nc</relativeExportFile>
```

```
        </stateFile>
        <expiryTime unit="day" multiplier="3"/>
</importStateActivity>
<importPiNetcdfActivity>
        <importFile>..\output\SPEC2D_P1.nc</importFile>
        <timeSeriesSets>
                <timeSeriesSet>
                        <moduleInstanceId>$MODULE_INSTANCE_ID$</moduleInstanceId>
                        <valueType>spectrum</valueType>
                        <parameterId>Wave.variancedensity2D.simulated</parameterId>
                        <domainParameterId>f</domainParameterId>
                        <domainParameterId>dir</domainParameterId>
                        <locationSetId>$MODEL$.locations</locationSetId>
                        <timeSeriesType>simulated historical</timeSeriesType>
                        <timeStep unit="hour"/>
                        <readWriteMode>add originals</readWriteMode>
                        <synchLevel>2</synchLevel>
                        <expiryTime unit="week" multiplier="1"/>
                </timeSeriesSet>
        </timeSeriesSets>
</importPiNetcdfActivity>
<importPiNetcdfActivity>
        <importFile>POINTS.nc</importFile>
        <timeSeriesSets>
                <timeSeriesSet>
                        <moduleInstanceId>$MODULE_INSTANCE_ID$</moduleInstanceId>
                        <valueType>scalar</valueType>
                        <parameterId>Wave.hm0.simulated</parameterId>
                        <locationSetId>$MODEL$.locations</locationSetId>
                        <timeSeriesType>simulated historical</timeSeriesType>
                        <timeStep unit="minute" multiplier="60"/>
                        <readWriteMode>add originals</readWriteMode>
                        <synchLevel>0</synchLevel>
                        <expiryTime unit="week" multiplier="1"/>
                </timeSeriesSet>
                <timeSeriesSet>
                        <moduleInstanceId>$MODULE_INSTANCE_ID$</moduleInstanceId>
                        <valueType>scalar</valueType>
                        <parameterId>Wave.he10.simulated</parameterId>
                        <locationSetId>$MODEL$.locations</locationSetId>
                        <timeSeriesType>simulated historical</timeSeriesType>
                        <timeStep unit="minute" multiplier="60"/>
                        <readWriteMode>add originals</readWriteMode>
                        <synchLevel>0</synchLevel>
                        <expiryTime unit="week" multiplier="1"/>
                </timeSeriesSet>
                <timeSeriesSet>
                        <moduleInstanceId>$MODULE_INSTANCE_ID$</moduleInstanceId>
                        <valueType>scalar</valueType>
                        <parameterId>Wave.period.Tm-1.0.simulated</parameterId>
                        <locationSetId>$MODEL$.locations</locationSetId>
                        <timeSeriesType>simulated historical</timeSeriesType>
                        <timeStep unit="minute" multiplier="60"/>
                        <readWriteMode>add originals</readWriteMode>
                        <synchLevel>0</synchLevel>
                        <expiryTime unit="week" multiplier="1"/>
                </timeSeriesSet>
                <timeSeriesSet>
                        <moduleInstanceId>$MODULE_INSTANCE_ID$</moduleInstanceId>
                        <valueType>scalar</valueType>
                        <parameterId>Wave.dir.E10.simulated</parameterId>
                        <locationSetId>$MODEL$.locations</locationSetId>
                        <timeSeriesType>simulated historical</timeSeriesType>
                        <timeStep unit="minute" multiplier="60"/>
                        <readWriteMode>add originals</readWriteMode>
                        <synchLevel>0</synchLevel>
                        <expiryTime unit="week" multiplier="1"/>
                </timeSeriesSet>
                <timeSeriesSet>
                        <moduleInstanceId>$MODULE_INSTANCE_ID$</moduleInstanceId>
                        <valueType>scalar</valueType>
```

```
                        <parameterId>Wave.s0bh.simulated</parameterId>
                        <locationSetId>$MODEL$.locations</locationSetId>
                        <timeSeriesType>simulated historical</timeSeriesType>
                        <timeStep unit="minute" multiplier="60"/>
                        <readWriteMode>add originals</readWriteMode>
                        <synchLevel>0</synchLevel>
                        <expiryTime unit="week" multiplier="1"/>
                </timeSeriesSet>
                <timeSeriesSet>
                        <moduleInstanceId>$MODULE_INSTANCE_ID$</moduleInstanceId>
                        <valueType>scalar</valueType>
                        <parameterId>Wave.period.Tps.simulated</parameterId>
                        <locationSetId>$MODEL$.locations</locationSetId>
                        <timeSeriesType>simulated historical</timeSeriesType>
                        <timeStep unit="minute" multiplier="60"/>
                        <readWriteMode>add originals</readWriteMode>
                        <synchLevel>0</synchLevel>
                        <expiryTime unit="week" multiplier="1"/>
                </timeSeriesSet>
                <timeSeriesSet>
                        <moduleInstanceId>$MODULE_INSTANCE_ID$</moduleInstanceId>
                        <valueType>scalar</valueType>
                        <parameterId>H.simulated</parameterId>
                        <locationSetId>$MODEL$.locations</locationSetId>
                        <timeSeriesType>simulated historical</timeSeriesType>
                        <timeStep unit="minute" multiplier="60"/>
                        <readWriteMode>add originals</readWriteMode>
                        <synchLevel>0</synchLevel>
                        <expiryTime unit="week" multiplier="1"/>
                </timeSeriesSet>
        </timeSeriesSets>
</importPiNetcdfActivity>
<importPiNetcdfActivity>
        <importFile>..\output\swan2D.nc</importFile>
        <timeSeriesSets>
                <timeSeriesSet>
                        <moduleInstanceId>$MODULE_INSTANCE_ID$</moduleInstanceId>
                        <valueType>grid</valueType>
                        <parameterId>Wave.hm0.simulated</parameterId>
                        <locationId>$MODEL$</locationId>
                        <timeSeriesType>simulated historical</timeSeriesType>
                        <timeStep unit="minute" multiplier="60"/>
                        <readWriteMode>add originals</readWriteMode>
                        <synchLevel>2</synchLevel>
                        <expiryTime unit="hour" multiplier="24"/>
                </timeSeriesSet>
                <timeSeriesSet>
                        <moduleInstanceId>$MODULE_INSTANCE_ID$</moduleInstanceId>
                        <valueType>grid</valueType>
                        <parameterId>Wave.he10.simulated</parameterId>
                        <locationId>$MODEL$</locationId>
                        <timeSeriesType>simulated historical</timeSeriesType>
                        <timeStep unit="minute" multiplier="60"/>
                        <readWriteMode>add originals</readWriteMode>
                        <synchLevel>2</synchLevel>
                        <expiryTime unit="hour" multiplier="24"/>
                </timeSeriesSet>
                <timeSeriesSet>
                        <moduleInstanceId>$MODULE_INSTANCE_ID$</moduleInstanceId>
                        <valueType>grid</valueType>
                        <parameterId>Wave.period.Tm-1.0.simulated</parameterId>
                        <locationId>$MODEL$</locationId>
                        <timeSeriesType>simulated historical</timeSeriesType>
                        <timeStep unit="minute" multiplier="60"/>
                        <readWriteMode>add originals</readWriteMode>
                        <synchLevel>2</synchLevel>
                        <expiryTime unit="hour" multiplier="24"/>
                </timeSeriesSet>
                <timeSeriesSet>
                        <moduleInstanceId>$MODULE_INSTANCE_ID$</moduleInstanceId>
                        <valueType>grid</valueType>
```

```
                                        <parameterId>Wave.period.Tps.simulated</parameterId>
                                        <locationId>$MODEL$</locationId>
                                        <timeSeriesType>simulated historical</timeSeriesType>
                                        <timeStep unit="minute" multiplier="60"/>
                                        <readWriteMode>add originals</readWriteMode>
                                        <synchLevel>2</synchLevel>
                                        <expiryTime unit="hour" multiplier="24"/>
                                </timeSeriesSet>
                                <timeSeriesSet>
                                        <moduleInstanceId>$MODULE_INSTANCE_ID$</moduleInstanceId>
                                        <valueType>grid</valueType>
                                        <parameterId>Wave.th0.simulated</parameterId>
                                        <locationId>$MODEL$</locationId>
                                        <timeSeriesType>simulated historical</timeSeriesType>
                                        <timeStep unit="minute" multiplier="60"/>
                                        <readWriteMode>add originals</readWriteMode>
                                        <synchLevel>2</synchLevel>
                                        <expiryTime unit="hour" multiplier="24"/>
                                </timeSeriesSet>
                                <timeSeriesSet>
                                        <moduleInstanceId>$MODULE_INSTANCE_ID$</moduleInstanceId>
                                        <valueType>grid</valueType>
                                        <parameterId>Wave.s0bh.simulated</parameterId>
                                        <locationId>$MODEL$</locationId>
                                        <timeSeriesType>simulated historical</timeSeriesType>
                                        <timeStep unit="minute" multiplier="60"/>
                                        <readWriteMode>add originals</readWriteMode>
                                        <synchLevel>2</synchLevel>
                                        <expiryTime unit="hour" multiplier="24"/>
                                </timeSeriesSet>
                                <timeSeriesSet>
                                        <moduleInstanceId>$MODULE_INSTANCE_ID$</moduleInstanceId>
                                        <valueType>grid</valueType>
                                        <parameterId>H.simulated</parameterId>
                                        <locationId>$MODEL$</locationId>
                                        <timeSeriesType>simulated historical</timeSeriesType>
                                        <timeStep unit="minute" multiplier="60"/>
                                        <readWriteMode>add originals</readWriteMode>
                                        <synchLevel>2</synchLevel>
                                        <expiryTime unit="hour" multiplier="24"/>
                                </timeSeriesSet>
                        </timeSeriesSets>
                </importPiNetcdfActivity>
        </importActivities>
```