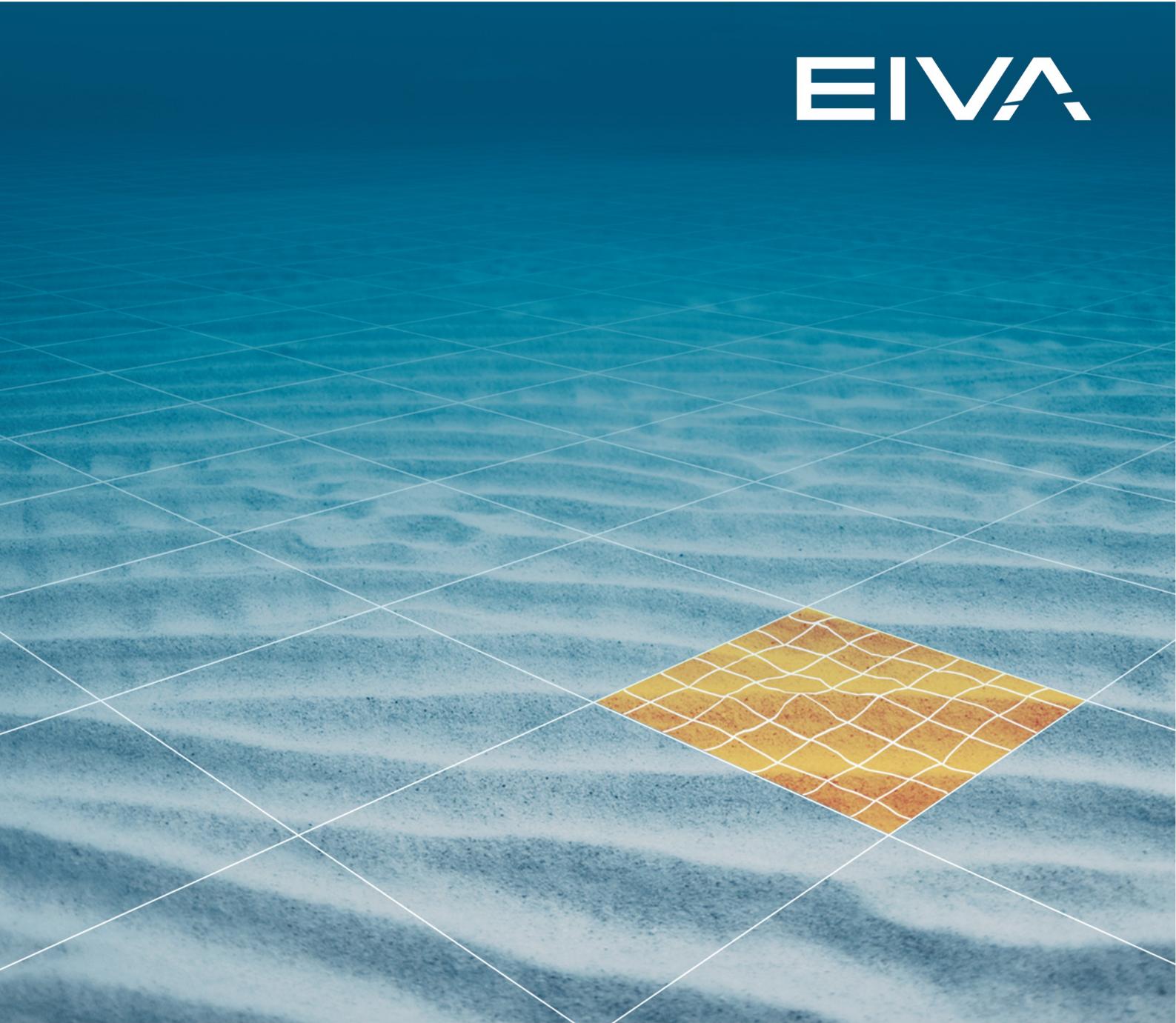


EIVA



# WFM User Guide

Last update: 02/05/2025  
Version: 4.10

# Contents

<b>1 Introduction .....</b>	<b>7</b>
1.1 Getting started .....	8
1.2 Change log .....	15
1.3 How does a Workflow.xml file look like? .....	16
1.4 Workflow_Manager_Demo_Project.xml .....	19
1.5 WorkFlow_NaviPac .....	23
<b>2 Workflow Manager Tasks .....</b>	<b>26</b>
2.1 Calculate Expression .....	26
2.2 Compare Property Lists .....	26
2.3 Copy Directory .....	27
2.4 Copy File .....	27
2.5 Create Directory .....	27
2.6 DateTime .....	27
2.7 Delete Directory .....	28
2.8 Delete File .....	28
2.9 Equal .....	28
2.10 External Application .....	28
2.11 File size .....	28
2.12 File watcher .....	29
2.13 Geodesy Settings .....	29
2.14 Group .....	30
2.15 Input Task .....	30
2.16 If Task .....	31
2.17 Licence Check .....	32
2.18 Move file .....	32
2.19 NaviEdit subfolders .....	32
2.20 OnError attribute .....	32
2.21 Overwrite Properties .....	33
2.22 Property Iterator .....	33
2.23 Release lock .....	33
2.24 Regular expression .....	33
2.25 Read file .....	34
2.26 SetPropertyTask .....	34

· 2.27 Set title .....	35
· 2.28 Set Max .....	35
· 2.29 Test .....	35
· 2.30 Time Interval Task .....	35
· 2.31 Split variables .....	36
· 2.32 Spawn children .....	36
· 2.33 Subdirectory watcher .....	36
· 2.34 Wait for User .....	36
· 2.35 Wait for lock .....	36
· 2.36 Wait Time .....	36
· 2.37 WriteToFileTask .....	37
· 2.38 Trim file .....	37
· 2.39 Time Interval Task .....	37
<b>3 NaviEdit Tasks .....</b>	<b>38</b>
· 3.1 All Import .....	38
· 3.2 Batch Smooth .....	38
3.2.1 BatchJob example for Bathy Recalculation .....	39
· 3.3 Batch Job in general .....	40
· 3.4 Bathy Smooth .....	41
· 3.5 Block Settings .....	42
· 3.6 BlockSettingsBlockStartTime .....	46
· 3.7 Export Ascii .....	46
· 3.8 ExportETR .....	51
· 3.9 ExportGRAD .....	52
· 3.10 ExportGSF2Bytes .....	53
· 3.11 Export GPS Tide .....	53
· 3.12 Export GSF .....	54
· 3.13 Export LAS/LAZ .....	55
· 3.14 Export NED .....	56
· 3.15 Export NavInXTF .....	57
· 3.16 Export Magnetic Field .....	57
· 3.17 Export Surface Pressure .....	58
· 3.18 Export Visual Soft Navigation .....	59
· 3.19 Export Report .....	60
· 3.20 Find Subfolders .....	60
· 3.21 Find Blocks .....	61
· 3.22 Find NaviPac Blocks .....	61
· 3.23 Get Parent Folder Names .....	61

3.24	Get Block File Name .....	62
3.25	Get KP Value .....	62
3.26	Hugin MissionPlan Import .....	63
3.27	Import CTD .....	63
3.28	Import Pressure .....	64
3.29	Import Runline .....	64
3.30	Import SVP .....	65
3.31	Import Tide .....	65
3.32	Las Import .....	66
3.33	Link Blocks .....	68
3.34	Link Block Tide .....	69
3.35	Link Block Surface Pressure .....	70
3.36	Merge Data Files .....	70
3.37	Merge Pipetrackers .....	72
3.38	Merge NavLab Calendar .....	72
3.39	Merge NavLab Classic .....	73
3.40	MergeTerraPosNMA .....	73
3.41	Merge Delayed Heave .....	74
3.42	Merge Applanix SBET in WFM .....	75
3.43	Move Block .....	76
3.44	Multibeam Import .....	76
3.45	NaviEdit Util Task .....	80
3.46	NaviPac Interpreter .....	80
3.47	NavLab Import .....	81
3.48	Pipetracker Merge .....	81
3.49	Recalc Bathy .....	81
3.50	Recalc KP .....	82
3.51	Reson S7K Import .....	83
3.52	Recalculation of GPS Height .....	83
3.53	SBD Import .....	84
3.54	SBD Import task .....	86
3.55	Set Geodesy .....	87
3.56	More on the Set geodesy task .....	88
3.57	Set Block Quality Threshold .....	90
3.58	SetMounting .....	90
3.59	Set window .....	91
3.60	Smooth Surface Pressure .....	92
3.61	Smooth Tide .....	92

..	3.62 Turbidity Merge .....	93
..	3.63 Wait for previous NaviEdit block .....	94
..	3.64 Yaw Stabilization Warning .....	94
<b>4</b>	<b>NaviModel Tasks .....</b>	<b>95</b>
..	4.1 Automatic Flagging .....	95
..	4.2 Camera Eventing .....	95
..	4.3 Cell Stdev .....	96
..	4.4 Compare Event Collections .....	96
..	4.5 Corridor Coverage .....	96
..	4.6 Create DTM .....	97
..	4.7 Depth Tiff .....	99
..	4.8 DTMinintensity2raster .....	99
..	4.9 DTMsurface2singlebandraster .....	100
..	4.10 EC-3D Combined .....	101
..	4.11 EC-3D cleaning .....	104
..	4.12 Export Grid .....	106
..	4.13 Exponential Smoothing .....	107
..	4.14 Export points .....	107
..	4.15 Extract Event Frames .....	108
..	4.16 ExportEventCollection .....	108
..	4.17 ExportTrack .....	108
..	4.18 Find Gaps .....	109
..	4.19 Find Rocks .....	109
..	4.20 Level Check .....	110
..	4.21 NMCLI .....	110
..	4.22 Merge Intensity .....	111
..	4.23 New NaviModel Project .....	111
..	4.24 Reject Fraction .....	112
..	4.25 Runline to Shape .....	113
..	4.26 ShadedRelief .....	113
..	4.27 ScanScore Scalgo .....	113
..	4.28 Track2Shape .....	114
..	4.29 WriteBack .....	114
<b>..5</b>	<b>FAQ .....</b>	<b>115</b>
..	5.1 FAQ .....	115

**Index** ..... 0

# 1 Introduction

The **Workflow Manager** is a configurable workflow process automation tool. It allows you to automate the small steps you normally go through when processing subsea data.

The software is designed for automatic data processing of massive amounts of subsea sensor data and comes in these variants:

## 1. Workflow Manager – Desktop

Is available for free when purchasing **NaviEdit Pro** and **NaviModel Producer** licences and is limited to being able to run one NaviEdit and one NaviModel task in parallel

## 2. Workflow Manager – Server

Can run an unlimited number of NaviEdit and NaviModel workflow steps (the NaviSuite products dedicated to navigation/positioning and data modelling/visualisation respectively) in parallel, typically 30-100.

The result is a significant parallelisation of processing for large data sets, and significant savings in terms of man-hours spent on data processing.

It can run without NaviEdit/NaviModel GUI.

A Workflow Manager licence looks like this:

```
licence=4130,10,5  
# NaviEdit; NaviModel Producer [PIPE]; WFM Server;
```

It is possible to install each package on separate PCs that are on one network.

A licence with parallel processes without user interface (Command Line Interface only) looks like this:

```
licence= 4096,10,0  
# WFM Server;
```

\*S-CAN functionality can be added as required.

To purchase of a licence, please contact the EIVA Sales department - or use the EIVA Webshop.

For support questions, please contact [support@eiva.com](mailto:support@eiva.com).

## 1.1 Getting started

This **Workflow Manager version 4.10** was tested with these software versions:

- WorkFlowManager\_4.10
- NaviEdit\_9.0
- NaviModelProducer\_4.10.2

You can download the test data set (Workflow Manager demo project (NaviSuite Kuda) – 4.8) from the EIVA Download Site/Workflow Manager.

Getting started:

- Install the three software packages, NaviModel Producer, NaviEdit and Workflow Manager locally on one computer.
- You need at least a NaviEdit and a NaviModel licence. Check in the **EIVA Licence Activator** from eg the **NaviEdit** menu bar / Help tab.
- You could test your EIVA licences by loading an SBD file into **NaviEdit**, opening **NaviModel** and import the file and clean out some data.  
If this can be done, your EIVA licences are correct and you can proceed with running the **Workflow Manager**.
- Create a data folder on your computer called C:\Data
- Copy the **Workflow Manager demo project (NaviSuite Kuda) – 4.10.zip** file to the **C:\Data** folder
- Unzip/extract it to the same location, C:\Data\Workflow\_Manager\_Demo\_Project
- This is the expected folder structure:

This PC > Local Disk (C:) > Data > Workflow\_Manager\_Demo\_Project

Name	Size	Type
Logs		File folder
output		File folder
WatchDir		File folder
Workflow_Manager_Demo_Project_with_error	10 KB	XML Document
Workflow_Manager_Demo_Project_without_error	9 KB	XML Document

Figure 1: Unzipped test data

- Start a text editor and open the **Workflow\_Manager\_Demo\_Project\_without\_error.xml** from C:\Data\Workflow\_Manager\_Demo\_Project folder.  
(If SQL server is installed locally, ignore this: Change the PC name in the 7th line. Here you need to enter the Device name from the Windows Control Panel / System. Localhost is also possible.)
- Start **NaviEdit** and create a new database with the name **WFM**

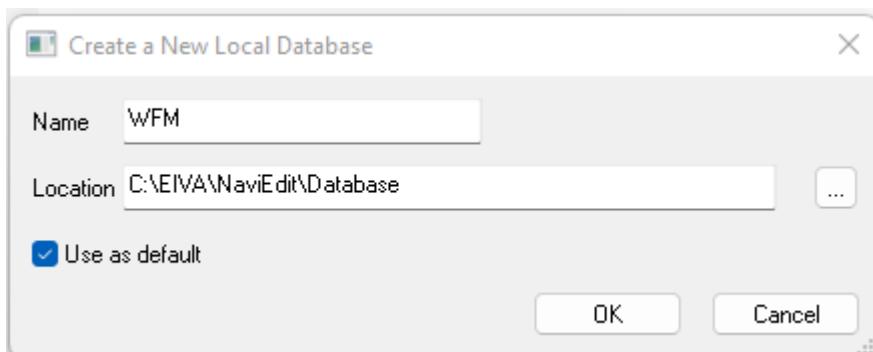


Figure 2: Creating the database for the test project

- Start the Workflow Manager from the desktop shortcut



- Go to the **Open Workflow** option and select the **Workflow\_Manager\_Demo\_Project\_without\_error.xml** from the folder C:\Data\Workflow\_Manager\_Demo\_Project

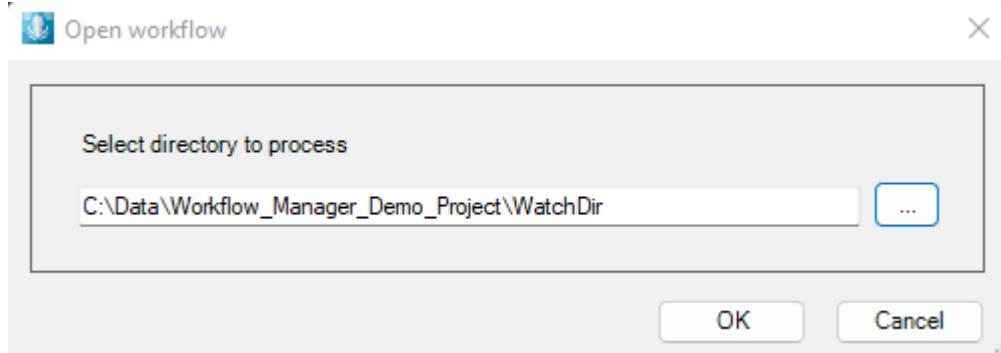


Figure 3: WatchDir path

- Point to the data folder called WatchDir. If a new file is added to this folder, the Workflow Manager will start automatically to process it.
- Trouble shooting: If you get the cryptic error mentioning V6, you need open NaviEdit and create or switch to the database called WFM.
- Now the Workflow Manager starts importing SBD files from the WatchDir folder

> This PC > Local Disk (C:) > Data > Workflow\_Manager\_Demo\_Project >

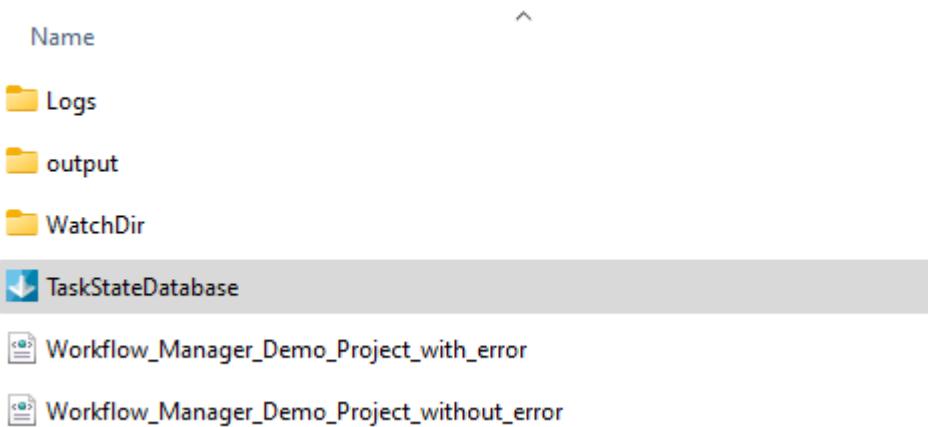


Figure 4: Workflow Manager is started, and TaskStateDatabase.db is created.automatically

- At the bottom right corner the name of the.xml file is shown.

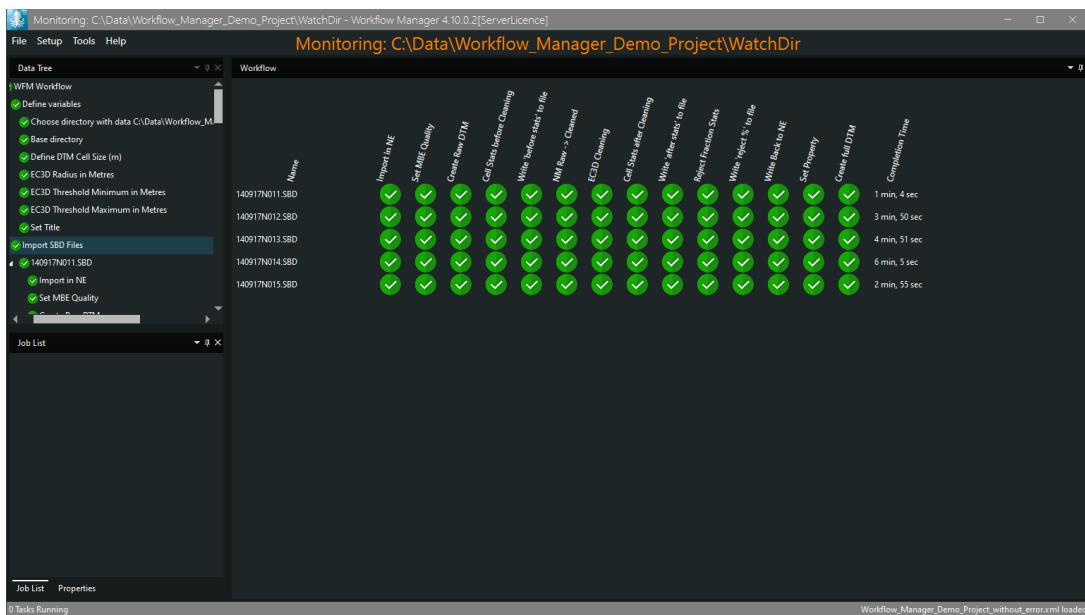


Figure 4: Workflow Manager in progress, the import process ran successfully.

- Clean all folders and run the Workflow\_Manager\_Demo\_Project\_with\_error.xml

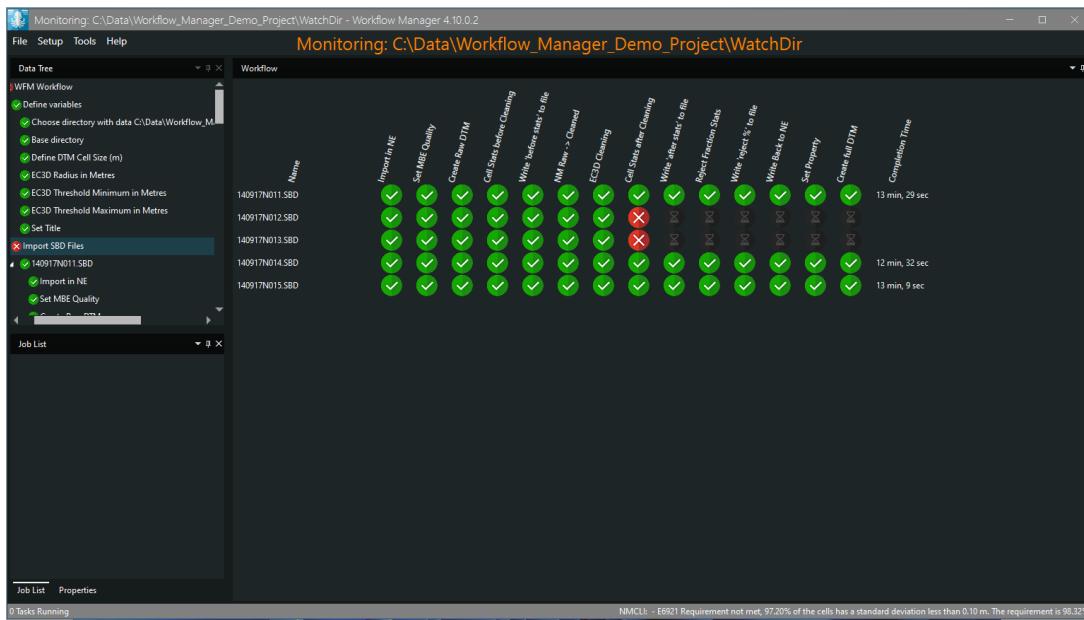


Figure 5: Workflow Manager: Two SBD files return errors.

- If a task fails, it will show in red. Right-click on the red circle to check the error message. The error message indicates which process failed, and why it failed. You can then – if needed – modify the settings in the .xml file.
- See the example below:

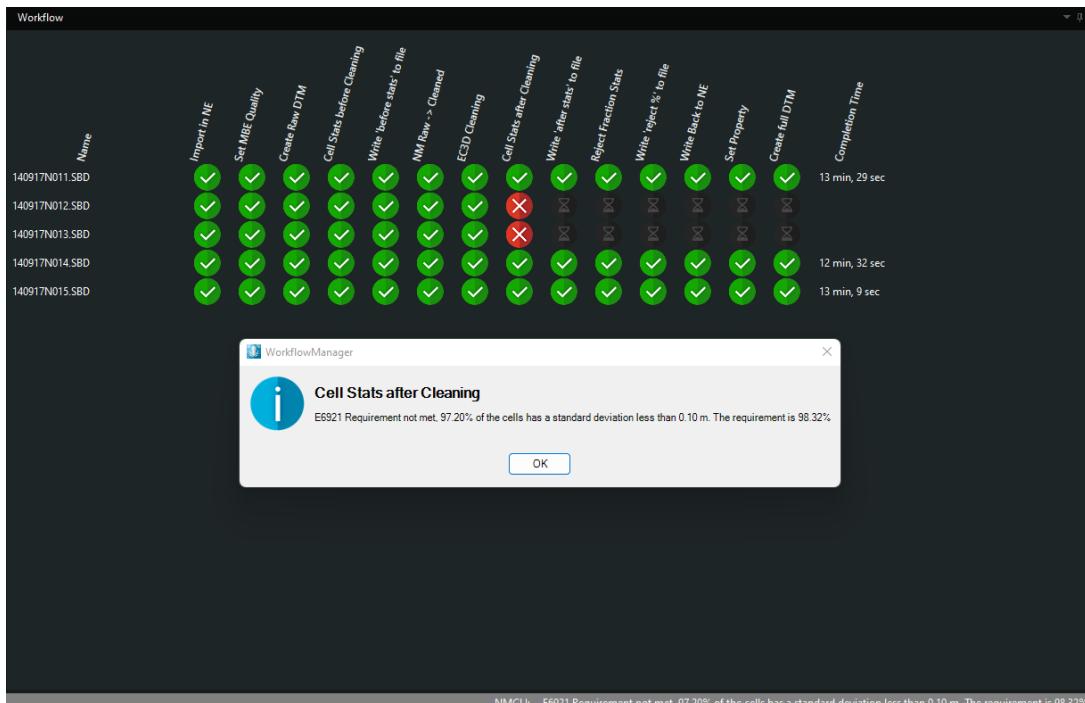


Figure 6: Example of a failing process

- Check which task fails, in this example the 8th task fails. Open the XML and find the process order (po) 8.  
Here you can change the value of the requirement.
- When changing the xml, the Workflow Manager has to be restarted.
- A process can be restarted by clicking on the red circle.  
The number 2 will then be shown, indicating that the task has been started twice

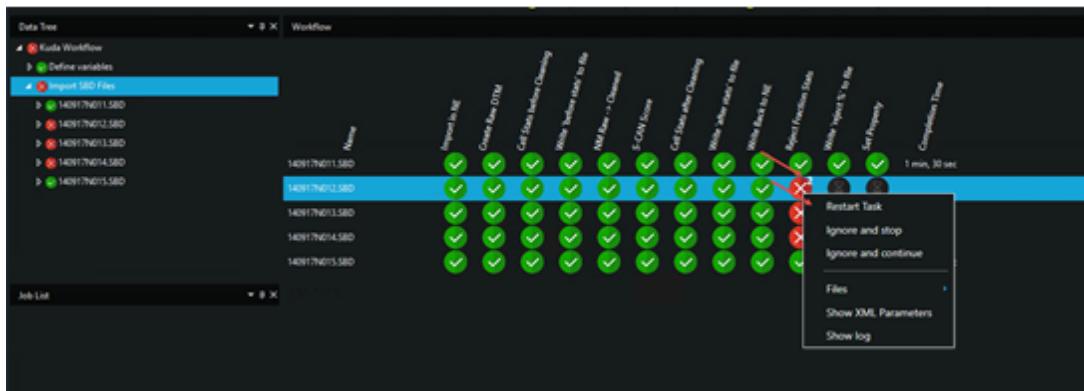


Figure 7: Restart Task and visualisation of running a task twice

- Note, after each WFM run, the TaskStateDatabase.db needs to be deleted  
It is also good to clear the output folder for a clearer overview.  
Note, that open \*.cli (CommandLine) processes should be closed, which can be done in the TaskManager.

#### How to start the WorkFlow Manager from scratch?

To start a workflow from scratch do the following steps, otherwise the WorkFlowManager will start from where it was

- Find the TaskStateDatabase.db and delete it, eg C:\Data\
- Consider cleaning the NaviEdit database.
- Delete data from the C:\Data\output folder.
- Open the TaskManager and close any \*.cli (NaviModel command line interface) processes.

- Warnings. The Import starts and hits the first wait for user interaction.

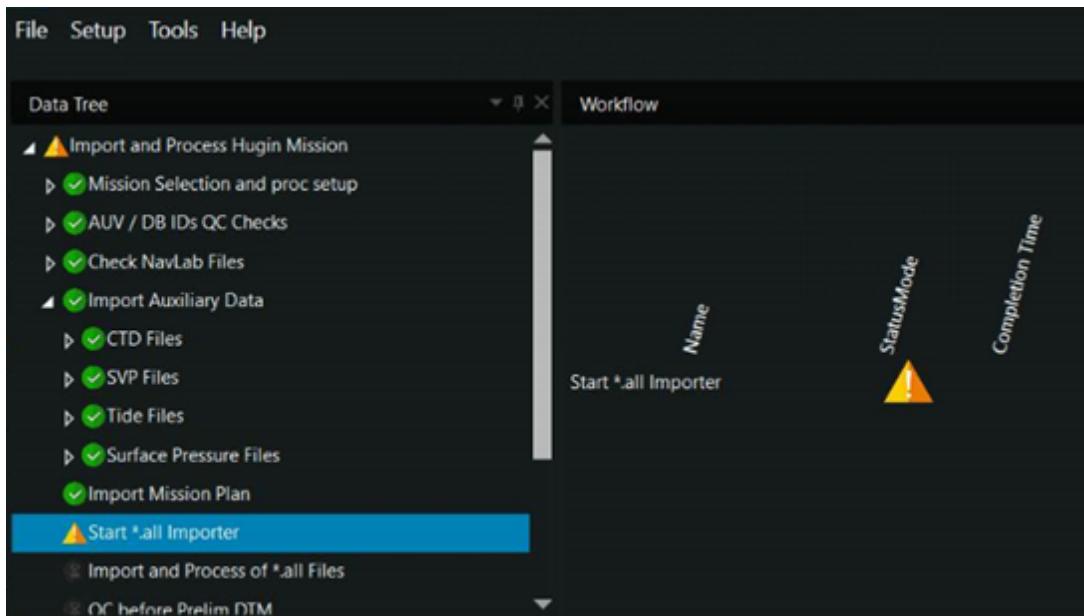


Figure 8: Import started

In the **WorkflowManager** setup you should set the max number of simultaneous tasks to the number of computer cores, minus one or less.

## 1.2 Change log

30.04.2024: Added example to the IfTask. SBDImportTask has two new attributes (`importGPSTide="no" linkGPSTide="no"`). Added export type to Export GSF task. Added SetMounting to the BlockSettings task. Improved CorridorCoverage task. Added ExportGSF2Bytes task. Updated the `Workflow_Manager_Demo_Project.zip`. Added MultibeamImport xml. Added SetMounting task. Added ExportGRAD task.

12.07.2023: Reviewed ShadedRelief, added Find Rocks

25.03.2023: Added GetBlockKP and BlockSettingsBlockStartTime tasks. Reviewed chapter 'How does a `Workflow.xml` file look like'

02.03.2023: Update of NewNaviModelproject, CreateDTM, SetProperty task, added ExportReport, added IfTask, added TimeIntervalTask, S-CAN Scalgo Score task, Link block tide.

19.01.2023: Added Recalculate bathy, NewNaviModelproject

09.11.2022: Added Time Interval Task

16.08.2022: Updated ShadedRelief, DepthTiff, ec3d, ecdcombined.

12.08.2022: Updated Writeback, ScanScore, Export Grid, Cell Stdev, NMCLI, Corridor coverage.

11.08.2022: Updated Shaded Relief and Create DTM to the new xml style, not backwards compatible.

02.11.2021: Added xml code for EC-3D Combined, Get Parent Folder Names

18.05.2021: Added xml code for Yaw-Stabilization-Warning

14.05.2021: Added xml code for Export LAS, Export accepted points, GEO Tiff

## 1.3 How does a Workflow.xml file look like?

Abbreviations:

po = Process order

level="2" = Used to define to which depth the task is valid in the nested XML data structure.

Cli = Command Line Interface

The Workflow Manager.xml file consists of **EIVA/Group task** and **software specific tasks**.

### **EIVA/Group task or Workflow Manager task:**

It is used for the entire workflow (global), such as log file paths, database locations, server settings and geodesy.

It is outside NaviEdit/NaviModel tasks.

Example:

```
<Setup StateDatabaseLocation="c:\Data\TaskStateDatabase.db">
  <Log TimingLog="C:\Data\Logs\Timing.csv"/>
  <Log ErrorLog="C:\Data\Logs\Error.csv"/>
  <Log ErrorLog="C:\Data\Logs\InfoLogFile.txt"/>

  <GroupTask name="DB IDs QC Checks" po="2">
    <WaitForInputTask po="1" input="{Directory}" />
    <RegExTask po="2" name="Find Mission Name" Input="{Directory.Filename}" Expression="(.)" GroupName="MissionName" />
    <!-- Extract mission name -->
    <SetPropertyTask po="3" name="Define Mission Name variable" input="{MissionName}" output="MissionName" />
  </GroupTask>
</Setup>
```

### **NaviEdit and NaviModel tasks:**

NaviEdit task starts with **NaviEditTask**

NaviModel task starts with **NMCLITask** (NaviModel Command-line interface)

A Workflow.xml would typically look like this:

```

<OnlineWorkFlowManager>
  1 <xml version="1.0">
  2 <Setup>
    3 <log TimingLog>"c:\EivaLogs\Timing.csv"/>
    4 <log ErrorLog>"c:\EivaLogs\Error.csv"/>
    5 <ConnectionString><add name="NaviEdit" providerName="System.Data.SqlClient" connectionString="Data Source=TESTPC5\SQLEXPRESS;Initial Catalog=NaviEdit;User ID=sa;Password=123456;"/>
    6 <SqlServerName>TESTPC5\SQLEXPRESS</SqlServerName>
    7 <DatabaseName>SimpleOnline</DatabaseName>
    8 <Geodesy>...</Geodesy>
  9 <GroupTask po="1" name="Data Selection and setup">
  10   <InputTask po="1" name="Choose Data Folder" output="Directory" AskForInput="true"/>
  11   <GetPropertyTask po="2" name="Define Data Directory" input="Directory" output="Directory" level="2"/>
  12   <GetPropertyTask po="3" name="Define Cleaning Path" input="Directory"\..\..\..\..\Process\SE\2-Data_Process\NaviModel\3.DTM\3.Cleaning_DTM\2.Line 2 East\UTM04N\2099\> output="CleaningDirectory" level="2"/>
  13   <GetPropertyTask po="4" name="Define Block Output Path" input="Directory"\..\..\..\..\Process\SE\Work\CleanedBlockIDs.txt" output="BlockOutputFile" level="2"/>
  14   <GetPropertyTask po="5" name="Define DTM Cell Size (m)" input="2" output="DtmCellsize" level="2"/>
  15   <GetPropertyTask po="6" name="Define S-CAN Score threshold (cm)" input="10" output="ScoreThreshold" level="2"/>      <!-- S-CAN threshold is in centimeters -->
  16 </GroupTask>
  17 <FileWatcherTask po="2" name="Import and Process of SBD Files" Filter="*.sbd" location="Directory" output="SBDfile" Timeout="525960">
  18   <GroupTask name="[$BDfile.Filename]">>
  19     <SBDImportTask po="1" name="Import File" output="SBDblockId" logKeyword="[$BDfile.Filename]>
  20       <NavEditTask version="1.0">
  21         <Input xml="[$SqlServer]" />
  22         <Input xml="[$DatabaseName]" />
  23         <Input xml="[$Geodesy]" />
  24       </NavEditTask>
  25     </SBDImportTask>
  26   </GroupTask>
  27   <CreateDTMTask po="1" name="Create Raw DTM" logKeyword="[$BDfile.Filename]">>
  28     <CreateDTM version="1.0">
  29       <Model path="(CleaningDirectory)Block_[$BDfile.NameWithoutExtension].db" cellsize="[$DtmCellsize]" OverwriteExistingFile="true" CreateMinimumSurface="true" CreateMaximumSurface="true" ReadIntensity="false"/>
  30       <Database>
  31         <Input xml="[$SqlServer]" />
  32         <Input xml="[$DatabaseName]" />
  33         <BlockIds>
  34           <SBDblockId>
  35             <@BlockId>
  36             </BlockId>
  37           </SBDblockId>
  38         </BlockIds>
  39       </Database>
  40     </CreateDTM>
  41   </CreateDTMTask>
  42   <ScanScoreTask po="4" name="Cleaning: S-CAN Score" logKeyword="[$BDfile.Filename]">>
  43     <ScanScore version="1.4">
  44       <Input path="[(CleaningDirectory)Block_[$BDfile.NameWithoutExtension].db]" threshold="[$ScoreThreshold]" tempFolder="c:\temp\scalgotemp" deletePatchFile="yes"/>
  45     </ScanScore>
  46   </ScanScoreTask>
  47   <WaitForUserTask po="5" name="QC Check Before Writeback" caption="QC Check of DTM, before writing the data back to NaviEdit" />
  48   <WritebackTask po="6" name="Write Back Edits to NE" logKeyword="[$BDfile.Filename]">>
  49     <Writeback>
  50       <Input path="[(CleaningDirectory)Block_[$BDfile.NameWithoutExtension].db]" />
  51     </Writeback>
  52   </WritebackTask>
  53   <WaitForLockTask po="7" name="Lock File" keyword="filelock"/>
  54   <DeleteFileTask po="8" name="Write Block ID's to file" input="[$SBDblockId]" file="[(BlockOutputFile)]" append="true"/>
  55   <DeleteFileTask po="9" name="Release Lock" keyword="filelock"/>
  56   <DeleteDTMTask po="10" name="Delete Cleaned DTM" file="[(CleaningDirectory)Block_[$BDfile.NameWithoutExtension].db]" logKeyword="[$BDfile.Filename]"/>
  57 </GroupTask>
  58 </FileWatcherTask>
  59 </GroupTask>
  60 </Setup>

```

Figure WorkFlow.xml example with explanation

1. **StateDataBase** is a small database which remember what task have completed, so it's possible to start up where you left off.

In this example, a file called **TaskStateDatabase.db** will be created next to the WorkFlowManager.exe file.

You might want to change this to another location.

Just write the full path to the where you want the database placed on disk. If you want to start from fresh, just delete the TaskStateDatabase.db file before you start the WorkFlowManager again.

TimingLog and ErrorLog should maybe also have their location changed. Timing is used to time every single task which have finished successfully.

The ErrorLog will show what errors you might encounter.

2. This is the login to your NaviEdit database. Instead of **TESTPC5** write the name of your SQL Server and **DataBaseName** should be the name of your database.
3. A few different variables are setup in the start, which make it easier for changing paths and such
  - a. **InputTask** – Ask for a directory and save chosen directory in the variable called **Directory**.

- b. **SetPropertyTask** – Takes a input, and define a output for everyone to use.
- i. Input = Input
  - ii. Output = name of variable to write output to
  - iii. Level = how many levels up in the tree should the output be written to.

1 will be writing the data to the parent of this task  
 2 will be grand parent of this task and so on.

In this case 2 will be the task called: <GroupTask name="Online Workflow"> which is the root of the workflow, so everyone will have access to the data here.

4. Create a **FileWatcher** and look for new files in the selected directory
  - a. The Filewatcher first looks for files in the selected directory and processes those.
  - b. If a timeout have been defined like in this case, it will keep monitoring the directory for new files matching the filter until the timeout runs out.
    - i. Timeout is in minutes, in this case  $365 \times 24 \times 60 = 1$  year
5. After a file have been imported, a DTM will be created
6. The DTM will be cleaned using a Threshold of 10 (Defined in variables in 3)
7. Wait for user to allow WFM to continue processing the file
8. Write cleaned data back to NaviEdit
9. Write the cleaned blocks to a txt file. So the offline workflow knows what blocks have been cleaned and is ready for further processing.
10. Finishing by deleting the Cleaned DTM.

Line 2	<pre>&lt;Setup StateDatabaseLocation="TaskStateDatabase.db"&gt;</pre>	<p>Here you can define where the TaskStateDatabase_Kuda.db is created. The StateDataBase is a small database which remember what task have completed, so it's possible to start up where you left off.</p> <p>In the example line above, the file called TaskStateDatabase.db will be created next to the WorkFlowManager exe file, (C:\EIVA\WorkFlowManager). If you want to change to another location, write the full path to the where you want the database placed on disk.</p>
--------	---	--

1.4 Workflow Manager Demo Project.xml

```
<?xml version="1.0"?>
<Setup StateDatabaseLocation="C:\Data\Workflow_Manager_Demo_Project\TaskStateDatabase.db">
    <Log TimingLog="c:\Data\Workflow_Manager_Demo_Project\Logs\Timing.csv"/>
    <Log ErrorLog="c:\Data\Workflow_Manager_Demo_Project\Logs\Error.csv"/>
    <Log InfoLogFile="c:\Data\Workflow_Manager_Demo_Project\Logs\InfoLogFile.txt"/>
    <GroupTask name="WFM Workflow">
        <SqlServer>localhost</SqlServer>
        <DatabaseName>WFM</DatabaseName>

        <GroupTask po="1" name="Define variables">
            <InputTask po="1" name="Choose directory with data C:\Data\Workflow_Manager_Demo_Project\TaskStateDatabase.db" />
            < SetPropertyTask po="2" name="Base directory" input="C:\Data\Workflow_Manager_Demo_Project\TaskStateDatabase.db" />
            < SetPropertyTask po="2" name="Define DTM Cell Size (m)" input="0.1" output="DTMcellsize" />
            < SetPropertyTask po="2" name="EC3D Radius in Metres" input="0.50" output="RadiusM" />
            < SetPropertyTask po="2" name="EC3D Threshold Minimum in Metres" input="0" output="ThresholdMin" />
            < SetPropertyTask po="2" name="EC3D Threshold Maximum in Metres" input="0.6" output="ThresholdMax" />
        </GroupTask>
    </GroupTask>
</Setup>
```

```

< SetPropertyTask po="2" name="Define threshold number" input="12" output="NumberThreshold" />
< SetTitleTask po="3" name="Set Title" title="Monitoring: {BaseDirectory}\WatchDir" size="1000" />
</GroupTask>

< FileWatcherTask po="2" name="Import SBD Files" Filter="*.sbd" location="{BaseDirectory}\WatchDir" />
< GroupTask name="{SBDFile.Filename}">

    <!-- Comment on IfTask. The IfTask uses regex to find the last digits of the SBD file name -->
    < RegExpTask po="1" name="Extract number" logKeyword="{SBDFile.Filename}" Input="{SBDFile.Filename}" />

    < IfTask po="2" name="If number less than {NumberThreshold}" input1="{Number}" operator="lessThan" />
        < GroupTask po="1" name="If">
            < OverwritePropertyTask po="1" name="Set NaviEdit folder to first area" input="1" />
        </GroupTask>
        < GroupTask po="2" name="Else">
            < OverwritePropertyTask po="1" name="Set NaviEdit folder to second area" input="2" />
        </GroupTask>
    </IfTask>

    < SbdImportTask po="3" name="Import in NE" Output="SBDBlockId" logKeyword="{SBDFile.Filename}" />
        < NaviEditTask version="1.0">
            < Input xml="{SqlServer}" />
            < Input xml="{DatabaseName}" />
            < Geodesy useSbdGeodesy="yes" />
            < SbdInterpreter version="1.0">
                < Destination matchFolder="yes" importSVP="no" linkSVP="no">
                    < Folder>\{SBDFolder}</Folder>
                </Destination>
                < Source>
                    < fileList>
                        {SBDFile}
                    </fileList>
                </Source>
                < Filters>
                    <!-- Note when the setDeletedFlag is set to yes, the beams below the threshold will be deleted -->
                    < MBEQuality setDeletedFlag="no" qualityThreshold="12" />
                </Filters>
                < Options disableBathy="no" useOnlineSV="no" onlineSVStart="-1.1" onlineSVEnd="1.1" />
            </SbdInterpreter>
        </NaviEditTask>
    </SbdImportTask>
    < BlockSettingsTask po="4" name="Set MBE Quality" logKeyword="{SBDFile.Filename}" />
        < NaviEditTask version="1.0">
            < SqlServer>{SqlServer}</SqlServer>
            < DatabaseName>{DatabaseName}</DatabaseName>
            <!--< Input xml="{SqlServer}" />
            < Input xml="{DatabaseName}" />-->
            < BlockSettings version="1.0">
                < BlockId>

```

```

        {SBDBlockId}
    </BlockId>
    <Options skipScansWithoutMotion="yes" disableBathy="yes" useHeave="no" indu...
    <QualityThreshold quality="10"/>
    <Windows> <!-- All units in meters -->
        <Easting min="-200000" max="900000"></Easting>
        <Northing min="-50" max="50"></Northing>
        <DepthRelRefPoint min="-2000" max="2000"></DepthRelRefPoint>
        <XDistRelRefPoint min="-500" max="500"></XDistRelRefPoint>
    </Windows>
    <BlockStartTime>
        <Offset hours="-2" minutes="0" />
        <UTCOffset hours="3" minutes="30" /> <!-- UTC Offset is optional -->
    </BlockStartTime>
    </BlockSettings>
</NaviEditTask>
</BlockSettingsTask>

<NMCLITask po="5" name="Create Raw DTM" logKeyword="{SBDFile.Filename}" output=" " >
    <Command name="CreateDTM">
        <Files></Files>
        <Model>
            <CellSize>{DTMcellsize}</CellSize>
            <Path>{BaseDirectory}\Output\Block_{SBDFile.Filename}_Raw.db</Path>
            <CreateMinimumSurface>no</CreateMinimumSurface>
            <CreateMaximumSurface>yes</CreateMaximumSurface>
            <CreateMedianSurface>no</CreateMedianSurface>
            <CreateStdSurface>yes</CreateStdSurface>
            <CreateRejectedCountSurface>yes</CreateRejectedCountSurface>
            <CreateDeletedCountSurface>yes</CreateDeletedCountSurface>
            <CreateFileCountSurface>yes</CreateFileCountSurface>
            <CreateBackscatterSurface>yes</CreateBackscatterSurface>
            <CreateQualitySurface>yes</CreateQualitySurface>
            <OverwriteExistingFile>yes</OverwriteExistingFile>
            <ReadIntensity>no</ReadIntensity>
            <CellSizeIntensity>0.5 m</CellSizeIntensity>
        </Model>
        <Database>
            <Input xml="{SqlServer}" />
            <Input xml="{DatabaseName}" />
            <BlockIds>
                {SBDBlockId}
            </BlockIds>
        </Database>
    </Command>
</NMCLITask>

<!-- Gather stats prior to cleaning -->
<NMCLITask po="6" name="Cell Stats before Cleaning" output="CellStdevBeforeCleaning" >
    <Command name="CellStdev">
        <Input>
            <dbfile>{BaseDirectory}\Output\Block_{SBDFile.Filename}_Raw.db</dbfile>
            <RequiredPercentageOfCells_pct>80</RequiredPercentageOfCells_pct>
            <StandardDeviationLimit_m>0.1</StandardDeviationLimit_m>
        </Input>
    </Command>

```



```

<CreateStdSurface>yes</CreateStdSurface>
<CreateRejectedCountSurface>yes</CreateRejectedCountSurface>
<CreateDeletedCountSurface>yes</CreateDeletedCountSurface>
<CreateFileCountSurface>yes</CreateFileCountSurface>
<CreateBackscatterSurface>yes</CreateBackscatterSurface>
<CreateQualitySurface>yes</CreateQualitySurface>
<OverwriteExistingFile>no</OverwriteExistingFile>
<ReadIntensity>no</ReadIntensity>
<CellSizeIntensity>0.5 m</CellSizeIntensity>
</Model>
<Database>
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />
    <BlockIds>
        {SBDBlockID}
    </BlockIds>
</Database>
</Command>
</NMCLITask>

</GroupTask>
</FileWatcherTask>

</GroupTask>
</Setup>

```

## 1.5 WorkFlow\_NaviPac

Description: This is an example workflow.

```

<?xml version="1.0"?>
<Setup
StateDatabaseLocation="\Data\Workflow_Manager_Demo_Project\TaskStateDatabase.
db">
    <Log TimingLog="c:\Eiva\Logs\Timing_NPD.csv"/>
    <Log ErrorLog="c:\Eiva\Logs\Error_NPD.csv"/>
    <Log InfoLogFile="c:\Eiva\Logs\InfoLogFile.txt"/>
    <GroupTask name="Import and Process NaviPac NPD files">
        <SqlServer>testpc5</SqlServer>
        <DatabaseName>WFM</DatabaseName>
        <!-- <Geodesy>.. <Geodesy> -->
        <SetTitleTask title="This Workflow is for NaviPac files" name="Set Title"
po="0"/>
        <LicenseCheckTask name="License Check" po="0" License2="NaviEdit"/>
        < SetPropertyTask name="Set AUV ID" po="0" level="1" output="AuvID"
input="48"/>
        <InputTask name="Choose NaviPac File Folder" po="1" output="Directory"
AskForInput="true"/>
        <FileWatcherTask name="NPD Files" po="2" output="NaviPacFile"
location="{Directory}" Filter="*.npd">
            <GroupTask name="{NaviPacFile.Filename}">
                <NaviPacInterpreterTask po="1" output="NaviPacBlockID">
                    <NaviEditTask version="1.0">
                        <Input xml="{SqlServer}" />

```

```

<Input xml="{DatabaseName}"/>
<Input xml="{Geodesy}"/>
  <NaviPacInterpreter version="1.0" useNpdGeodesy="yes">
    <Destination>
      <Folder>\NaviPac\SubFolder</Folder>
    </Destination>
    <Source>
      <FileList>{NaviPacFile}</FileList>
    </Source>
    <GPSTide calculate="yes">
      <GeoidHeight geoidSelection="0">
        <!-- 0 = Fixed height,
              1 = Geoid file -->
        <FixedHeight additionalHeight="0.0" height="42.0"/>
      <GeoidFile>c:\EIVA\NaviPac\Setup\Geoide.bin</GeoidFile>
    </GeoidHeight>
    <SpeedCorrection correctionType="0">
      <!-- 0 = No correction,
          1 = Corr function,
          2 = Corr file -->
      <CorrectionFunction v0="3.3" v1="2.2" v2="1.1"/>
      <CorrectionFile>c:
\NaviPac\Setup\Squat.txt</CorrectionFile>
    </SpeedCorrection>
    <DynamicDraft scaleToMeter="1.0" singleBeamSeq="0" apply="no"/>
  </GPSTide>
  </NaviPacInterpreter>
  </NaviEditTask>
</NaviPacInterpreterTask>
<SetPropertyTask name="Collect all NaviPac Block IDs" po="2"
level="3" output="AllNaviPacBlockID" input="{NaviPacBlockID}"/>
</GroupTask>
</FileWatcherTask>

<ExportGPSTideTask name="Export GPS Tide" po="3">
<NaviEditTask version="1.0">
<Input xml="{SqlServer}"/>
<Input xml="{DatabaseName}"/>
  <Export type="gpstide">
    <BlockId>{AllNaviPacBlockID}</BlockId>
    <Output>
      <!--OutputMethod // PATH = 0, APPENDTOFILE = 1ReduceMethod // ALL = 0, INTERVAL = 1Sel
      <Location sortCombined="yes" combineFiles="yes" prefixKPRange="no"
outputMethod="0">
        <Path>c:\temp\GPSTide</Path>
        <AppendPath/>
      </Location>
      <DbFolderUsage useAsFilename="no" prefixFileName="no"
createSubdir="no"/>
        <Options interval="10.0" selectionMethod="0" reduceMethod="0"
sortKP="no" writeHeader="no"/>
        <!-- interval unit meter-->
    </Output>
    <Units depthScale="1.0" posScale="1.0"/>
      <!-- depth scale is positive down -->
  <Settings>

```

```
<Selection singleBeamSeq="2"/>
    <!-- This is the zero based index of the single beam channel -->
</Settings>
</Export>
</NaviEditTask>
</ExportGPSTideTask>

<ExportSurfacePressureTask name="Export Surface Pressure" po="3">
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}" />
        <Input xml="{DatabaseName}" />
        <Export type="single">
            <BlockId>{AllNaviPacBlockID}</BlockId>
            <Output>
                <!-- OutputMethod // PATH = 0, APPENDTOFILE = 1ReduceMethod // ALL = 0, INTERVAL = 1Second -->
                <Location sortCombined="yes" combineFiles="yes" prefixKPRange="no"
outputMethod="0">
                    <Path>c:\NaviPac\SurfacePressure</Path>
                    <AppendPath/>
                    </Location>
                    <DbFolderUsage useAsFilename="no" prefixFileName="no"
createSubdir="no"/>
                    <Options interval="10.0" selectionMethod="0" reduceMethod="0"
sortKP="no" writeHeader="no"/>
                </Output>
                <!-- depth scale is positive down -->
                <Settings addName="no" decimals="5" scale="0.001">
                    <Selection singleBeamSeq="0"/>
                    <!-- This is the zero based index of the single beam channel -->
                </Settings>
            </Export>
        </NaviEditTask>
    </ExportSurfacePressureTask>
</GroupTask>
</Setup>
```

## 2 Workflow Manager Tasks

### 2.1 Calculate Expression

Description: Calculate an formula and output the result.

OnError attribute: This attribute is optional attribute. Usage: Action to take if the task fails: Default, IgnoreAndContinue, Restart, RestartParent. If present, will overrule "Auto restart failed task".

```
<CalculateExpressionTask po="" name="" logKeyword="" Expression="" Output="" OnError="/" />
```

### 2.2 Compare Property Lists

Description: This Task can compare two lists of space separated properties. It could typically be list of block ids.

The following compare options are available:

- 'Intersection' (elements exist in both lists)
- 'Union' (elements exist in either of the lists)
- 'Difference' (elements exist only in the first list)

Example:

```
<?xml version="1.0"?>

<Setup StateDatabaseLocation="e:\WFM\TaskStateDatabase_CompareBlockIds.db">
    <Log TimingLog="c:\Eiva\Logs\Timing.csv"/>
    <Log ErrorLog="c:\Eiva\Logs>Error.csv"/>
    <Log InfoLogFile="c:\Eiva\Logs\InfoLogFile.txt" />

    <GroupTask name="Test">
        <GroupTask po="2" name="Define Variables">
            < SetPropertyTask po="1" name="Set List 1" input="1 2 3 4 5"
output="LongList" level="2" />
            < SetPropertyTask po="1" name="Set List 2" input="4 3"
output="ShortList" level="2" />
        </GroupTask>

        <ComparePropertyListsTask po="2" name="Compare Lists" list1="{LongList}"
list2="{ShortList}" operation="intersection" output="ResultList"/>

        <WriteToFileTask po="5" name="Write List 1" input="{LongList}" file="e:
\WFM\Output\LongList.txt" overwrite="true"/>
        <WriteToFileTask po="5" name="Write List 2" input="{ShortList}" file="e:
\WFM\Output\ShortList.txt" overwrite="true"/>
        <WriteToFileTask po="5" name="Write result" input="{ResultList}" file="e:
\WFM\Output\Intersection.txt" overwrite="true"/>
    </GroupTask>
</Setup>
```

```

    <ComparePropertyListsTask po="6" name="Compare Lists" list1="{LongList}"
list2="{ShortList}" operation="union" output="ResultList"/>
    <WriteToFileTask po="7" name="Write result" input="{ResultList}" file="e:
\WFM\Output\Union.txt" overwrite="true"/>

    <ComparePropertyListsTask po="8" name="Compare Lists" list1="{LongList}"
list2="{ShortList}" operation="difference" output="ResultList"/>
    <WriteToFileTask po="9" name="Write result" input="{ResultList}" file="e:
\WFM\Output\Difference.txt" overwrite="true"/>

</GroupTask>
</Setup>
```

## 2.3 Copy Directory

Description: This command can be used to copy a directory from **Source** to **Destination** with all content files and sub directories.

Possibly overwrites existing files in destination directory.

```
<CopyDirectoryTask po=" " name=" " logKeyword=" " Source=" " Destination=" "
OnError=" "/>
```

## 2.4 Copy File

Description: Copy file from **Source** to **Destination**.

```
<CopyFileTask po=" " name=" " logKeyword=" " Source=" " Destination=" "
Overwrite=" "/>
```

## 2.5 Create Directory

Description: Create directory at path. Creates missing parent directories if recursive is true.

```
<.CreateDirectoryTask po=" " name=" " logKeyword=" " Path=" " Recursive=" "/>
```

## 2.6 DateTime

Description:

Manipulate a DateTime variable, by inputting a DateTime string (in the format specified) and adding the amount of minutes provided.

Leave Input empty to get the Time of execution in output. Set Input to "UTC" to get Time of execution in UTC.

Input: Input string with the DateTime, Leave out to get the Time of execution. Use UTC to get Time of execution in UTC.

Output: Variable to write the new DateTime string to, same format as input format will be used.

Format: Format of the input and output string, e.g. yyyy-MM-dd HH:mm:ss.fff.  
ExtraMinutes: Extra minutes to add to the input DateTime. Negative number is OK. Do not use if Input is empty or "UTC".

```
<DateTimeTask po="" name="" logKeyword="" OnError="" Input="" Output="" Format="" ExtraMi
```

## 2.7 Delete Directory

Description: Deletes a directory and its contents. Use with caution.

```
<DeleteDirectoryTask po="" name="" logKeyword="" OnError="" Directory="" />
```

## 2.8 Delete File

Description: Delete a specified file.

```
<DeleteFileTask po="" name="" logKeyword="" OnError="" File="" />
```

## 2.9 Equal

Description: Checks if two inputs are identical.

```
<EqualTask po="" name="" logKeyword="" OnError="" Input1="" Input2="" />
```

## 2.10 External Application

Description: Start an external application with specified parameters.

```
<ExternalTask po="" name="" logKeyword="" OnError="" Location="" Parameter="" />
```

## 2.11 File size

Description: Verify file size in KB (1024 bytes).

```
<FileSizeTask po="" name="" logKeyword="" File="" ExpectedSize="" />
```

Example:

```
<FileSizeTask po="1" name="Ignore empty files" logKeyword="{SBDFile.Filename}" File="{SBDFile}">
  <SbdImportTask po="2" name="Import in NE" Output="SBDBlockId" logKeyword="{SBDFile.Filename}">
    ...
  </SbdImportTask>
```

## 2.12 File watcher

Description: Monitor a directory for files

```
<FileWatcherTask po="" name="" logKeyword="" Output="" Location="" Filter="" Timeout="" UserFilter="" IncludeSubDir="" />
```

Timeout is measured in minutes. Default is 0, that means no timeout, it runs once, handles all the files it finds, and then stops.

When the time expires, the FileWatcher stops, just as it does when no timeout has been set.

## 2.13 Geodesy Settings

Description: The Geodesy is split into two parts, <Source> and <Destination>.

In this example the source geodesy WGS84, UTM zone 31 N is changed to UTM zone 32 N.

Example 1:

```
<Input xml="{Geodesy}" />
```

Example 2:

```
<Geodesy useSbdGeodesy="yes">
</Geodesy>
```

Example 3:

```
<Geodesy>
  <Source>
    <Ellipsoid name="WGS 84" epsg="7030">
      <!-- eiva does not use the ellipsoid epsgs number, but indirectly by use of
the projection epsgs number -->
      <SemiMajorAxis>6378137.0</SemiMajorAxis>
      <InverseFlattening>298.257223563</InverseFlattening>
    </Ellipsoid>
    <Projection epsg="32632" name="UTM (north)">
      <!-- epsg 32632 requires/implies wgs 84 ellipsoid -->
      <EivaType>5</EivaType>
      <!-- integer PROJPAC.h UTM N = 5 -->
      <FalseEasting>500000</FalseEasting>
      <FalseNorthing>0</FalseNorthing>
      <FirstParallel>0</FirstParallel>
      <SecondParallel/>
      <Scale>0.9996</Scale>
      <OrigLatitude>0</OrigLatitude>
      <OrigLongitude>3</OrigLongitude>
      <Zone>31</Zone>
    </Projection>
  </Source>
  <Destination>
    <Ellipsoid name="WGS 84" epsg="7030">
      <!-- eiva does not use the ellipsoid epsgs number,
```

```

        but indirectly by use of the projection epsgs number -->
<SemiMajorAxis>6378137.0</SemiMajorAxis>
<InverseFlattening>298.257223563</InverseFlattening>
</Ellipsoid>
<Projection name="UTM (north)" epsg="32632">
    <!-- epsg 32632 requires/implies wgs 84 ellipsoid -->
<EivaType>5</EivaType>
    <!-- integer PROJPAC.h UTM N = 5 -->
<FalseEasting>500000</FalseEasting>
<FalseNorthing>0</FalseNorthing>
<FirstParallel>0</FirstParallel>
<SecondParallel/>
<Scale>0.9996</Scale>
<OrigLatitude>0</OrigLatitude>
<OrigLongitude>9</OrigLongitude>
<Zone>32</Zone>
</Projection>
</Destination>
</Geodesy>

```

## 2.14 Group

Description:

GroupTask

## 2.15 Input Task

Description:

```

<?xml version="1.0"?>
<Setup StateDatabaseLocation="TaskStateDatabase.db">
    <!-- Creates a BaseDirectory called ImportSBD, holding the
TaskStateDatabase.db -->
    <Log ErrorLog="c:\Data\Logs\Error.csv"/>
    <!-- Creates a folder for error logs -->
    <GroupTask name="Online Workflow">
        <SqlServer>testpc2</SqlServer>
        <!-- Computer name running the SQL server -->
        <DatabaseName>WFM</DatabaseName>
        <!-- NaviEdit database name, manually created -->

        <GroupTask po="1" name="Data selection and setup">
            <InputTask po="1" name="Choose data folder" output="BaseDirectory"
AskForInput="true"/>
            < SetPropertyTask po="2" name="Define data directory"
input="{BaseDirectory}" output="BaseDirectory" level="2"/>
        </GroupTask>

        <FileWatcherTask po="2" name="Import SBD files" Filter="*.sbd"
location="{BaseDirectory}" output="SBDFile" Timeout="60000">

```

```

<SbdImportTask po="1" name="Import SBD files in NE" Output="SBDBlockId"
logKeyword="{SBDFile.Filename}">
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}" />
        <Input xml="{DatabaseName}" />
        <Geodesy useSbdGeodesy="yes">
            </Geodesy>
        <SbdInterpreter version="1.0">
            <Destination matchFolder="yes" importSVP="no" linkSVP="no">
                <Folder>\SBD</Folder>
            </Destination>
            <Source>
                <FileList>
                    {SBDFile}
                </FileList>
            </Source>
            <Filters>
                <!-- Note when the setDeletedFlag is set to yes, the beams below
the threshold are marked as deleted/rejected. When the quality threshold flag
is set in the HeaderEditor, the beams are not marked as deleted, but marked
as filtered in NaviModel. Beams marked as deleted/rejected can be manually
accepted in NaviModel -->
                <MBEQuality setDeletedFlag="yes" qualityThreshold="12"/>
            </Filters>
            <Options disableBathy="no" useOnlineSV="no" onlineSVStart="-1.1"
onlineSVEnd="15.5" scanReductionImportEvery="1" />
        </SbdInterpreter>
        </NaviEditTask>
    </SbdImportTask>
</FileWatcherTask>
</GroupTask>
</Setup>

```

## 2.16 If Task

Description: WFM task.

Performs the specified operation (equal, not\_equal, greater\_than, less\_than, boolean, not\_empty, contains) on the input(s).

If the result is true, the first sub-group task will be executed.

Else, the second (optional) sub-group task will be executed.

The second input is optional, depending on the type of operation to perform.

```

<IfTask name="Check if true" po="1" logKeyword="" OnError="" input1="42"
operation="equal" input2="42" type="integerValue" >
<GroupTask name="ToBePerformedIfTrue">
<!-- Subtasks here -->
</GroupTask>
<GroupTask name="ToBePerformedIfFalse">
<!-- Subtasks here -->
</GroupTask>
</IfTask>

```

Example:

```
<!-- Comment on IfTask. The IfTask uses regex to find the last digits of the
SBD file, eg 011.
    Then copies the SBD file if the file name number is smaller then the
defined treshold number into folder Area1, else into folder Area2.
    Here the level must be kept.-->
<RegExpTask po="1" name="Extract number" logKeyword="{SBDFile.Filename}"
Input="{SBDFile.NameWithoutExtension}" Expression="\d+$" Group0="Number"/>

<IfTask po=" " name="If number less than {NumberThreshold}" input1="{Number}"
operation="less_than" input2="{NumberThreshold}">
    <GroupTask po="1" name="If">
        <OverwritePropertyTask po="1" name="Set NaviEdit folder to first area"
input="Area1" output="SBDFolder" level="2"/>
    </GroupTask>
    <GroupTask po="2" name="Else">
        <OverwritePropertyTask po="1" name="Set NaviEdit folder to second area"
input="Area2" output="SBDFolder" level="2"/>
    </GroupTask>
</IfTask>
```

## 2.17 Licence Check

Description: The license check warns the user in early stage, if no EIVA dongle is available.

```
<LicenseCheckTask name="License Check" po="0" License2="NaviEdit"/>
```

## 2.18 Move file

Description:

```
<MoveFileTask po=" " name=" " logKeyword=" " output=" "/>
```

## 2.19 NaviEdit subfolders

Description:

```
<NaviEditSubFoldersTask po=" " name=" " logKeyword=" " output=" "/>
```

## 2.20 OnError attribute

OnError attribute: This attribute is optional.

Usage: Action to take if the task fails: Default, IgnoreAndContinue, Restart, RestartParent. If present, will overrule "Auto restart failed task".

## 2.21 Overwrite Properties

Description: Read a number of inputs and overwrite them to a matching number of variables. It is possible to use only Input and Output, or to use numbered inputs and outputs, or a combination.

```
<OverwritePropertyTask po="1" name="Overwrite Property"
logKeyword="{BlockId}" input="{BlockId}" output="ProcessedFiles" Input0="0"
Output0="0" input1="1" output1="1" input2="2" output2="2" input3="3"
output3="3" input4="4" output4="4" input5="5" output5="5" Input6="6"
Output6="6" input7="7" output7="7" input8="8" output8="8" input9="9"
output9="9" level="3" />
```

## 2.22 Property Iterator

Description:

```
<PropertyIteratorTask po=" " name="Property Iterator" logKeyword=" "
Output="BlockId" Property="ProcessedFiles"/>
  <GroupTask name="BlockId: {BlockId}">
    <!-- Insert sub tasks here. -->
  </GroupTask>
</PropertyIteratorTask>
```

## 2.23 Release lock

Description:

```
<ReleaseLockTask po=" " name="Release lock" logKeyword=" " output=" "/>
```

## 2.24 Regular expression

Description:

```
<RegExpTask po=" " name=" " logKeyword=" " output=" "/>
```

Example:

```
<RegExpTask po="2" name="Find Mission Name" Input="{Directory.Filename}" Expression=".+" Group0="0" Group1="1" Group2="2" Group3="3" Group4="4" Group5="5" Group6="6" Group7="7" Group8="8" Group9="9" />
```

Another example:

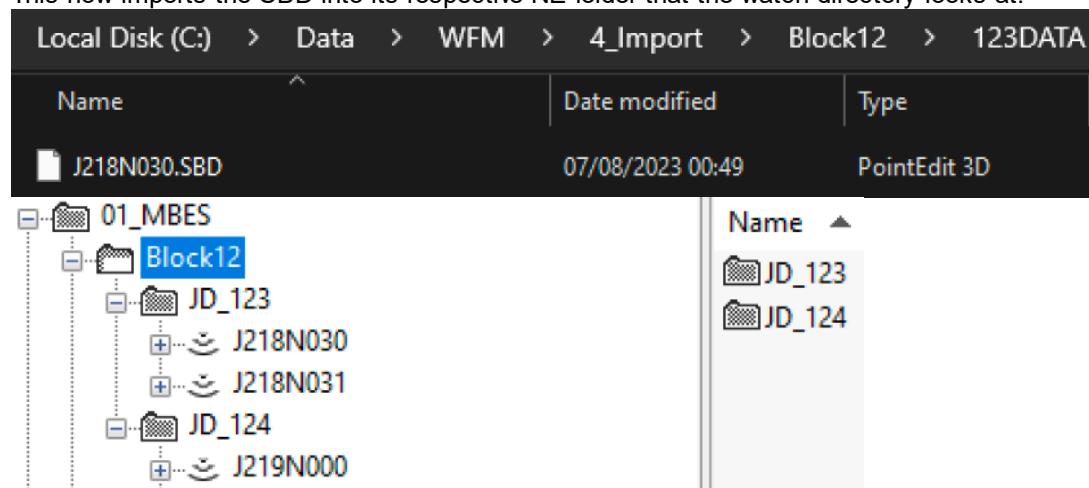
```
<!-- Task to extract SurveyAreaID and Julian_Day from the folder path -->
  <RegExpTask name="Extract NE BlockID and Julian_Day" po=" " input="{SBDFile}"
    Expression="\\(([^\n]+)\\)([0-9][0-9])DATA"
    Group0="zero" Group1="BlockID" Group2="JulianDay" SaveState="false"/>

  <!-- Import the SBD files into the database -->
  <SbdImportTask po="2" name="Import in NE" Output="SBDBlockId" logKeyword="{SBDFile.File}"
    <NaviEditTask version="1.0">
      <Input xml="{SqlServer}" />
```

```
<Input xml="{DatabaseName}" />
<input xml="{Geodesy}" />
<SbdInterpreter version="1.0">
    <Destination matchFolder="yes" importSVP="no" linkSVP="no">
        <Folder>\01_MBES\{BlockID}\JD_{JulianDay}</Folder>
    </Destination>
    <Source>
        <FileList>
            {SBDFile}
        </FileList>
    </Source>
    <Filters>
        <MBEQuality setDeletedFlag="no" qualityThreshold="12"/>
    </Filters>
    <Options disableBathy="no" useOnlineSV="yes" onlineSVStart="-1.1" onlineSVEnd="1.1" />
</SbdInterpreter>
</NaviEditTask>
</SbdImportTask>
```

## Result:

This now imports the SBD into its respective NE folder that the watch directory looks at.



## 2.25 Read file

Description:

```
<ReadFileTask po="" name="" logKeyword="" output="" />
```

## 2.26 SetPropertyTask

Description: Read an input and add it to an existing or new variable.

```
< SetPropertyTask po="" name="" logKeyword="" OnError="" Input="" Output="" Separator="/" />
```

## 2.27 Set title

Write a text in the top of the screen, to make sure the user fast can see what workflow is running and on what data.

Also sets the window title to this text.

```
<SetTitleTask po="" name="" logKeyword="" OnError="" Title="" Size="" Red="" Green="" B>
```

## 2.28 Set Max

Set Maximum number of concurrent tasks to input.

If Save= False, the value is only for the remainder of this workflow.

If Save=True, value is saved in Settings for future use.

Value must be greater than 0.

```
<SetMaxTasksTask po="1" name="Set Max Tasks" OnError="" logKeyword="maxTasks" Input="10" Save="false"/>
```

## 2.29 Test

Description:

```
<TestTask po="" name="" logKeyword="" output="" />
```

## 2.30 Time Interval Task

Executes all child tasks at the given interval (in minutes).

If timeout is > 0, task will stop after this timeout (in minutes).

The run number will be written to the property specified in output.

Timeout: Specify a Timeout (in minutes) for this task. If Timeout is set, the Task will stop when the timeout runs out. If timeout is not set, or is 0, this task will run indefinitely.

Interval: Specify the desired Interval (in minutes) for execution of child tasks.

Output: Property to write run number to.

```
<?xml version="1.0" encoding="utf-8"?>
<TimeIntervalTask name="TimeIntervalTask" po="1" interval="1" timeout="10" output="number" >
    <GroupTask name="Child grouptask number {number}">
        <!-- Subtasks here -->
    </GroupTask>
</TimeIntervalTask>
```

## 2.31 Split variables

Description: Split input variable into many output variables. Use SetAsProperty to overwrite variables as external variables (same as OverwriteProperty).

```
<SplitVariablesTask po="" name="" logKeyword="" OnError="" Input="" Seperator="" Output0="" Output1="" Output2="" Output3="" Output4="" Output5="" Output6="" Output7="" Output8="" Output9="" SetAsProperty="" Level="/" />
```

## 2.32 Spawn children

Description:

```
<SpawnChildrenTask po="" name="Spawn children" logKeyword="" output="/" />
```

## 2.33 Subdirectory watcher

Description:

```
<SubdirectoryWatcherTask po="" name="" logKeyword="" output="/" />
```

## 2.34 Wait for User

Description: Wait for the user to continue.

```
<WaitForUserTask po="" name="" logKeyword="" Text="" Caption="" WaitKey="/" />
```

## 2.35 Wait for lock

Description:

```
<WaitForLockTask po="" name="" logKeyword="" output="/" />
```

## 2.36 Wait Time

Description: Waits for the specified milliseconds.

```
<WaitTask po="" name="" logKeyword="" WaitTime="/" />
```

## 2.37 WriteToFileTask

Write the content of a variable to a file.

If the variable is a property, the contents of the property will be written to the file (could for example be a list of Block IDs).

Will keep trying to get write access to file for up to 5 seconds, if file is locked.

```
<WriteToFileTask po="" name="" logKeyword="" File="" Input="" Overwrite="" Append="" />
```

## 2.38 Trim file

Description:

```
<TrimFileTask po="" name="" logKeyword="" output="" />
```

## 2.39 Time Interval Task

Executes all child tasks at the given interval (in minutes).

If timeout is > 0, task will stop after this timeout (in minutes).

The run number will be written to the property specified in output.

Timeout: Specify a Timeout (in minutes) for this task. If Timeout is set, the Task will stop when the timeout runs out. If timeout is not set, or is 0, this task will run indefinitely.

Interval: Specify the desired Interval (in minutes) for execution of child tasks.

Output: Property to write run number to.

```
<?xml version="1.0" encoding="utf-8"?>
<TimeIntervalTask name="TimeIntervalTask" po="1" interval="1" timeout="10" output="number" >
    <GroupTask name="Child grouptask number {number}">
        <!-- Subtasks here -->
    </GroupTask>
</TimeIntervalTask>
```

## 3 NaviEdit Tasks

### 3.1 All Import

Description: \*.ALL files

```
<AllImportTask po="" name="" logKeyword="" output="" OnError="" >
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />
    <Input xml="{Geodesy}" />
      <EmInterpreter version="1.0" emType="3" datagram="1">
        <!-- (emType 0 = EM121A,
              1 = EM1000,
              2 = EM3000HUGIN2001,
              3 = EMSERIES,
              4 = KMALL
              *** (datagram 0 = 'Raw Range/Bearing', 1 = 'XYZ')
        -->
    <Destination matchfolder="no" importsdp="no" linksdp="no">
      <Folder>
        \All
      </Folder>
    </Destination>
    <Source>
      <FileList>
        d:\WFM\AUV\EM2040-0139-A1L74-20170802-034350.all
        <!-- c:\Data\EM2040-0040-A1L81-20170731-182934.all
            c:\Data\EM2040-0041-A1L81-20170731-184434.all
        -->
      </FileList>
    </Source>
    <Filters>
      <MBEQuality setDeletedFlag="yes" qualityThreshold="0"/>
    </Filters>
    <Options disableBathy="no" useOnlineSV="yes" onlineSVStart="-1.0"
onlineSVEnd="15.0" scanReductionImportEvery="1"
skipScansWithoutMotion="true"/>
      </EmInterpreter>
    </NaviEditTask>
  </AllImportTask>
```

### 3.2 Batch Smooth

Description: Smooth Tide or Surface pressure.

```
<BatchSmoothTask po="" name="" logKeyword="" >
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />
      <BatchProcess>
        <BlockId>5814</BlockId>
```

```

<!-- Multiple allowed, but smoothed as single blocks -->
<batchjob>
    <edititem action="Smooth points" target="Tide[Active]">
        <sensortype>15</sensortype>
        <actionid>40100</actionid>
        <selection>0</selection>
        <index>0</index>
        <smooth>
            <wavelength>3600</wavelength> <!-- seconds -->
        </smooth>
    </edititem>
</batchjob>
</BatchProcess>
</NaviEditTask>
</BatchSmoothTask>

```

### 3.2.1 BatchJob example for Bathy Recalculation

Description: This is an example recalculation of bathy sensor via the BatchJob functionality.

```

<BatchSmoothTask po=" " name=" " logKeyword=" ">
    <NaviEditTask version="1.0">
        <SqlServer>testpc5</SqlServer> <!-- Could also be "{SqlServer}" -->
        <DatabaseName>WFM</DatabaseName> <!-- Could also be "DatabaseName" -->
        <BatchProcess>
            <BlockId>158 197</BlockId>
            <!-- Multiple allowed, could also be {AllBlockId}-->
            <batchjob>
                <edititem target="Bathy[Active]" action="Recalculate Bathy (ocean)">
                    <sensortype>4</sensortype>
                    <actionid>34002</actionid>
                    <!-- recalc bathy task id -->
                    <selection>0</selection>
                    <!-- 0 = ACTIVE_SENSOR,
                        1 = ALL_SENSORS_OR_CHANNEL_OF_TIS_TYPE,
                        2 = THIS_SENSOR_INDEX -->
                    <index>0</index>
                    <recalcbathyoceangocean>
                        <gravity>9.80173664581538</gravity>
                        <surfpres>1.01325</surfpres>
                        <waterdensity>0</waterdensity>
                        <calctype>3</calctype>
                            <!-- 0 = SIMPLE (DEPTH CORRECTION),
                                1 = SIMPLE (CONST DENSITY),
                                2 = SAUNDERS AND FOFONOFF,
                                3 = UNESCO (REQ. CTD) -->
                        <selectpressure>1</selectpressure>
                            <!-- 0 = FixedPressure,
                                1 = FromAPressureBlock,
                                2 = ABathySeqFromThisBlock -->
                        <refpressure>0</refpressure>
                        <pressureid>155</pressureid> <!-- Could also be {PressureBlock}
                    <-->
                    </recalcbathyoceangocean>
                </edititem>
            </batchjob>
        </BatchProcess>
    </NaviEditTask>
</BatchSmoothTask>

```

```

        </edititem>
        </batchjob>
    </BatchProcess>
    </NaviEditTask>
</BatchSmoothTask>

```

### 3.3 Batch Job in general

Description: The BatchJobTask can be used to call all the different batch jobs as known from the JobPlanner, Batch Job action.

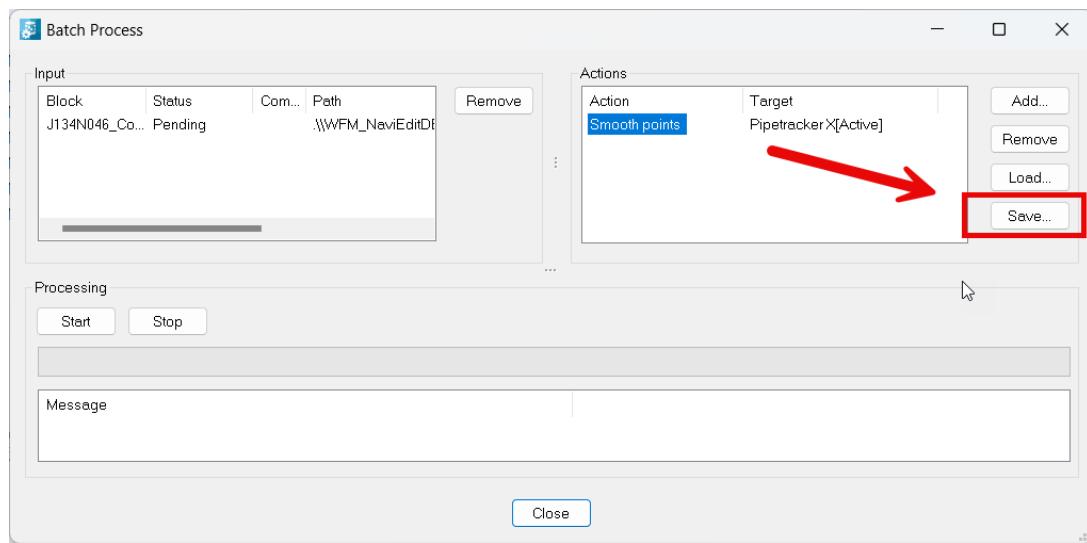
Use this skeleton below and replace the <batchjob> element with the out of the batch job as saved from the JobPlanner.

```

<BatchJobTask po="1" name="My BatchJob">
    <NaviEditTask version="1.0">
        <SqlServer>localhost</SqlServer>
            <DatabaseName>WFM_NaviEditDB</DatabaseName>
            <BatchProcess>
                <BlockId>42</BlockId>
                <batchjob>
                </batchjob>
            </BatchProcess>
    </NaviEditTask>
</BatchJobTask>

```

From the JobPlanner, Batch Job menu create the batch job and save it.



Resulting file:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<batchjob>
  <edititem action="Smooth points" target="Pipetracker X[Active]">
    <sensortype>10</sensortype>
    <actionid>40100</actionid>
    <selection>0</selection>
    <index>0</index>
    <smooth>
      <wavelength>20</wavelength>
    </smooth>
  </edititem>
</batchjob>

```

Ignore the first xml version line and replace the batchjob element with the saved output:

```

<BatchJobTask po="1" name="My BatchJob">
  <NaviEditTask version="1.0">
    <SqlServer>localhost</SqlServer>
    <DatabaseName>WFM_NaviEditDB</DatabaseName>
    <BatchProcess>
      <BlockId>42</BlockId>
      <batchjob>
        <edititem action="Smooth points" target="Pipetracker X[Active]">
          <sensortype>10</sensortype>
          <actionid>40100</actionid>
          <selection>0</selection>
          <index>0</index>
          <smooth>
            <wavelength>20</wavelength>
          </smooth>
        </edititem>
      </batchjob>
    </BatchProcess>
  </NaviEditTask>
</BatchJobTask>

```

Most likely you will also replace the hardcoded block id 42 with a property like {SbdBlockId} and the database name with <Input xml="{DatabaseName}" />

### 3.4 Bathy Smooth

Description: Smooth Tide or Surface pressure.

```

<BatchSmoothTask po="" name="" logKeyword="">
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />
    <BatchProcess>
      <BlockId>5814</BlockId>
      <!-- Multiple allowed, but smoothed as single blocks -->

```

```

<batchjob>
  <edititem action="Smooth points" target="Tide[Active]">
    <sensortype>15</sensortype>
    <actionid>40100</actionid>
    <selection>0</selection>
    <index>0</index>
    <smooth>
      <wavelength>3600</wavelength> <!-- seconds -->
    </smooth>
  </edititem>
</batchjob>
</BatchProcess>
</NaviEditTask>
</BatchSmoothTask>

```

## 3.5 Block Settings

Description: Block Settings refer to the block in NaviEdit.

Please note:

- All tasks have the outer element type BlockSettingsTask.
- The QualityThreshold, Options, Window and BlockStartTime elements can either all be present inside a single BlockSettingsTask as above, or you can use only one of them in individual BlockSettingsTasks.

Example - Move a NaviEdit block to NaviEdit database folder (and possibly create it if missing)

```

<BlockSettingsTask po="3" name="Move block" logKeyword="{AllFile.Filename}" OnError=" ">
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />
    <MoveBlock>
      <BlockId>
        {AllBlockId}
      </BlockId>
      <DestinationFolder createMissingFolder="yes">
        <!-- Use backslash notation and start from the root like:
        \FirstFolder\SecondFolder -->
        \MovedToHere
      </DestinationFolder>
    </MoveBlock>
  </NaviEditTask>
</BlockSettingsTask>

```

Example - Abort if yaw stabilization is active

```

<BlockSettingsTask po="4" name="Test for Yaw Stabilization"
logKeyword="{AllFile.Filename}">
  <NaviEditTask version="1.0">

```

```

<Input xml="{SqlServer}" />
<Input xml="{DatabaseName}" />
<AbortIf yawStabilizationActive="yes">
  <BlockId>
    {AllBlockId}
  </BlockId>
</AbortIf>
</NaviEditTask>
</BlockSettingsTask>

```

Example - Set the Skip scans without motion check box

```

<BlockSettingsTask po="5" name="Set MBE Quality"
logKeyword="{AllFile.Filename}">
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />
    <BlockSettings version="1.0">
      <BlockId>
        {AllBlockId}
      </BlockId>
      <QualityThreshold quality="14"/>
      <Options skipScansWithoutMotion="yes" />
    </BlockSettings>
  </NaviEditTask>
</BlockSettingsTask>

```

Example - Disabling bathy and heave

```

<BlockSettingsTask po="5" name="Set MBE Quality"
logKeyword="{AllFile.Filename}">
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />
    <BlockSettings version="1.0">
      <BlockId>
        {AllBlockId}
      </BlockId>
      <QualityThreshold quality="14"/>
      <Options skipScansWithoutMotion="yes" disableBathy="no" useHeave="no"
inducedHeave="no" surfaceBased="yes" />
    </BlockSettings>
  </NaviEditTask>
</BlockSettingsTask>

```

Example - Setting the Gyro Parameters to 'Apply meridian convergence':

```

<BlockSettingsTask po="2" name="Apply meridian convergence"
logKeyword="BlockSettings">
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />

```

```

<Input xml="{DatabaseName}" />
<BlockSettings version="1.0">
    <BlockId>
        {BlockId}
    </BlockId>
    <!-- gyroParameters: 0: Grid Heading, 1: Apply Meridian Convergence, 2:
Calculate Heading -->
    <Options gyroParameters="1" calcHeadingGain="0.4" />
</BlockSettings>
</NaviEditTask>
</BlockSettingsTask>

```

Example - Set the Window parameters as found in the HeaderEditor and changing the Start Time of the block

```

<BlockSettingsTask po="6" name="Several Block Settings"
logKeyword="{AllFile.Filename}">
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}" />
        <Input xml="{DatabaseName}" />
        <BlockSettings version="1.0">
            <BlockId>
                {AllBlockId}
            </BlockId>
            <QualityThreshold quality="14"/>
            <Options skipScansWithoutMotion="yes" />
            <Windows>
                <!-- All units in meters -->
                <Easting min="-200000" max="900000"></Easting>
                <Northing min="-50" max="50"></Northing>
                <DepthRelRefPoint min="-2" max="200"></DepthRelRefPoint>
                <XDistRelRefPoint min="-50" max="50"></XDistRelRefPoint>
            </Windows>
            <BlockStartTime>
                <!-- NaviEdit_8.7-84580: Added BlockStartTime element to change time in blocks with 2 hours offset -->
                <Offset hours="-2" minutes="0" />
                <UTCOffset hours="3" minutes="30" /> <!-- UTCOffset is optional -->
            </BlockStartTime>
        </BlockSettings>
    </NaviEditTask>
</BlockSettingsTask>

```

Example - Change the active motion sensor

```

<BlockSettingsTask po="3" name="Change motion sensor"
logKeyword="{SBDFFile.Filename}">
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}" />
        <Input xml="{DatabaseName}" />
        <SetActiveSensors motion="1">
            <BlockId>{SBDBlockId}</BlockId>
        </SetActiveSensors>

```

```

    </NaviEditTask>
</BlockSettingsTask>
```

#### Example - Change/Set the Mounting

```

<BlockSettingsTask po="4" name="Change mounting" logKeyword="{SBDFFile.Filename}">
    <NaviEditTask version="1.0">
        <SqlServer>localhost</SqlServer>
        <DatabaseName>NaviEdit86_RC7</DatabaseName>
        <SetMounting>
            <BlockId>{SBDBlockId}</BlockId>
            <!-- type:
            0 Absolute
            1 Bathy
            2 Motion
            3 Pipetracker
            4 Doppler Log
            5 Gyro
            6 Position
            7 Remote positon
            8 Waterline
            9 Gps Antenna
            10 User defined offset
            12 Auxilary
            13 Relative Position
            15 Laser
            16 Echosounder
            17 Echosounder Rx
            18 Sidescan
            20 Combined position
            256 Single data
            -->

            <!-- Position -->
            <Mounting type="6" seq="1">
                <!-- In meters -->
                <Offset x="1.11" y="2.22" z="3.33" />
            </Mounting>

            <!-- Several Mountings can be specified-->
            <!-- Echo sounder-->
            <Mounting type="16" seq="0">
                <!-- In meters -->
                <Offset x="3.321" y="2.222" z="1.123" />
                <!-- In degrees -->
                <Angle roll="-21.1" pitch="0.456" heading="-0.111"/>
                <Multibeam isRollCompensated="no" isPitchCompensated="no" />
            </Mounting>

            <!-- Single beam, example with sequence 2 -->
            <Mounting type="256" seq="2">
                <!-- In meters -->
                <Offset x="0.11" y="0.22" z="3.33" />
                <!-- In degrees -->
                <Angle roll="0.0" pitch="0.0" heading="0.0"/>
                <Singlebeam useRawSingleData="no" />
            </Mounting>
```

```

    </SetMounting>
    </NaviEditTask>
</BlockSettingsTask>
```

## 3.6 BlockSettingsBlockStartTime

Description: The tsk allows you to change the Block Start time.

```

<BlockSettingsTask po="" name="Change Block start time" logKeyword="{AllFile.Filename}" OnError="Stop"
    <NaviEditTask version="1.0">
        <SqlServer>localhost</SqlServer>
        <DatabaseName>WFM</DatabaseName>
        <BlockSettings version="1.0">
            <BlockId>
                1680
            </BlockId>
            <BlockStartTime>
                <Offset hours="-2" minutes="0" />
            </BlockStartTime>
        </BlockSettings>
    </NaviEditTask>
</BlockSettingsTask>
```

## 3.7 Export Ascii

Description: Exports to Ascii. Use type to select type of export

The following types can be used

- xyz\_kpdoldepth
- xyz\_etrack
- xyz\_bpl
- xyp\_simple
- xyp\_ptr
- xyposcustom
- xytriton
- xycoda
- xycodanew
- xytimekykpf
- xytimegc
- xyvsoftvev
- latlongmotion
- xynavinxtf
- bin\_ned
- bin\_fau
- bin\_las

- navlab
- navedit
- gpstide
- single
- innonav

Example one:

```

<ExportAsciiTask po="" name="" logKeyword="" OnError="" >
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}"/>
        <Input xml="{DatabaseName}"/>
        <Input xml="{Geodesy}"/>
        <Export type="xynavinxtf">
            <BlockId>
                62
            </BlockId>
            <Output>
                <!-- OutputMethod // 0 = PATH, 1 = APPENDTOFILE
                    ReduceMethod // 0 = ALL, 1 = INTERVAL
                    SelectionMethod // 0 = CLOSEST, 1 = MINIMUM, 2 = MAXIMUM
                -->
                <Location outputMethod="0" prefixKPRange="no" combineFiles="no"
sortCombined="no">
                    <Path>C:\Temp</Path>
                    <AppendPath></AppendPath>
                </Location>
                <DbFolderUsage createSubdir="no" prefixFileName="no"
useAsFilename="no" />
                <Options writeHeader="yes" sortKP="no" reduceMethod="0"
selectionMethod="0" interval="1.0" />
                <!-- interval unit meter-->
            </Output>
            <Units posScale="1.0" depthScale="1.0" />
            <!-- depth scale is positive down -->
            <Settings>
                <Selection object="-1" />
                <!-- object -1 CRP -->
                <Singlebeam interpolateDepth="yes" maxInterpolDist="-1" />
                <!-- maxInterpolDist -1 Auto -->
                <Multibeam averagingWindow="1.0" />
                <!-- averaging windows i meter -->
                <NoOfDecimals position="3" depth="3" />
                <Options downsampleFactor="0" downsampleMinDist="0.0"
timeFormat="0" />
                <!-- timeFormat 0 = Default -->
            </Settings>
        </Export>
    </NaviEditTask>
</ExportAsciiTask>
```

Example two:

### Example three:

```
<ExportAsciiTask po="2" name="Export Etrack" logKeyword="EtrackExport">
  <NaviEditTask version="1.0">
```

```

<SqlServer>localhost</SqlServer>
<DatabaseName>WFM_Database</DatabaseName>
<Export type="xyz_etrack">
  <BlockId>42</BlockId>
  <Output>
    <!--
      OutputMethod           // PATH = 0, APPENDTOFILE = 1
      ReduceMethod          // ALL = 0, INTERVAL = 1
      SelectionMethod       // CLOSEST = 0, MINIMUM = 1, MAXIMUM = 2
    -->
    <Location outputMethod="0" prefixKPRange="no" combineFiles="no"
    sortCombined="no">
      <Path>c:\temp\export</Path>
      <AppendPath></AppendPath>
    </Location>
    <DbFolderUsage createSubdir="no" prefixFileName="no"
    useAsFilename="no" />
      <Options writeHeader="yes" sortKP="no" reduceMethod="0"
    selectionMethod="0" interval="1.0" />
        <!-- interval unit meter-->
    </Output>
    <Units posScale="1.0" depthScale="1.0" />
    <!-- depth scale is positive down -->
    <Settings>
      <Selection object="0" />
      <!-- object:
        -2 = depth channel
        -1 = CRP
        0 = User Selected specify in UserOffset element below
        n = UserOffset Id in the NaviEdit database
      -->
      <UserOffset x="1" y="2" z="3" name="MyOffset" /> <!-- Only needed for
object="0" -->
        <Singlebeam interpolateDepth="yes" maxInterpolDist="-1" />
        <!-- maxInterpolDist -1 Auto -->
        <Multibeam averagingWindow="1.0" />
        <!-- averaging windows i meter -->
        <NoOfDecimals position="3" depth="3" />
        <Options downsampleFactor="0" downsampleMinDist="0.0"
timeFormat="0" />
            <!-- timeFormat 0 = Default -->
      </Settings>
    </Export>
  </NaviEditTask>
</ExportAsciiTask>

```

Example four the Pipetracer (PTR) export:

```

<ExportAsciiTask po="2" name="Export pipetracker" logKeyword="PTRExport">
  <NaviEditTask version="1.0">
    <SqlServer>localhost</SqlServer>
    <DatabaseName>NaviEditDB</DatabaseName>
    <Export type="xyp_ptr">
      <BlockId>
        45
      </BlockId>

```

```

<Output>
  <!--
    APPENDTOFILE = 1           OutputMethod          // PATH = 0,
    INTERVAL = 1               ReduceMethod         // ALL = 0,
    MINIMUM = 1, MAXIMUM = 2   SelectionMethod      // CLOSEST = 0,
  -->
  <Location outputMethod="0" prefixKPRange="no" combineFiles="no"
  sortCombined="no">
    <Path>c:\temp\export</Path>
    <AppendPath></AppendPath>
  </Location>
  <DbFolderUsage createSubdir="no" prefixFileName="no"
  useAsFilename="no" />
    <Options writeHeader="yes" sortKP="no" reduceMethod="0"
  selectionMethod="0" interval="1.0" />
    <!-- interval unit meter-->
  </Output>
  <Units posScale="1.0" depthScale="1.0" />
  <!-- depth scale is positive down -->
</Export>
</NaviEditTask>
</ExportAsciiTask>

```

Example five the Pipetracker PIP export:

```

<ExportAsciiTask po="2" name="Export PIP" logKeyword="PIPEExport">
  <NaviEditTask version="1.0">
    <SqlServer>localhost</SqlServer>
    <DatabaseName>NaviEditDB</DatabaseName>
    <Export type="xyp_simple">
      <BlockId>
        {SBDBlockId}
      </BlockId>
      <Output>
        <Location outputMethod="0" prefixKPRange="no" combineFiles="no"
  sortCombined="no">
          <Path>c:\temp\export</Path>
          <AppendPath></AppendPath>
        </Location>
        <DbFolderUsage createSubdir="no" prefixFileName="no"
  useAsFilename="no" />
          <Options writeHeader="yes" sortKP="no" reduceMethod="0"
  selectionMethod="0" interval="1.0" />
        </Output>
        <Units posScale="1.0" depthScale="1.0" />
        <Settings>
          <!-- type = 0: pipe, type = 1: bathy -->
          <Pipe type="0" onlyExportGoodValues="true">
            <!-- The bathy element is only needed if Pipe type 1 is used. Use
the zero based sequence number as seen in the HeaderEditor as the seq
attribute -->
            <Bathy seq="1"/>

```

```

        </Pipe>
        </Settings>
    </Export>
    </NaviEditTask>
</ExportAsciiTask>
```

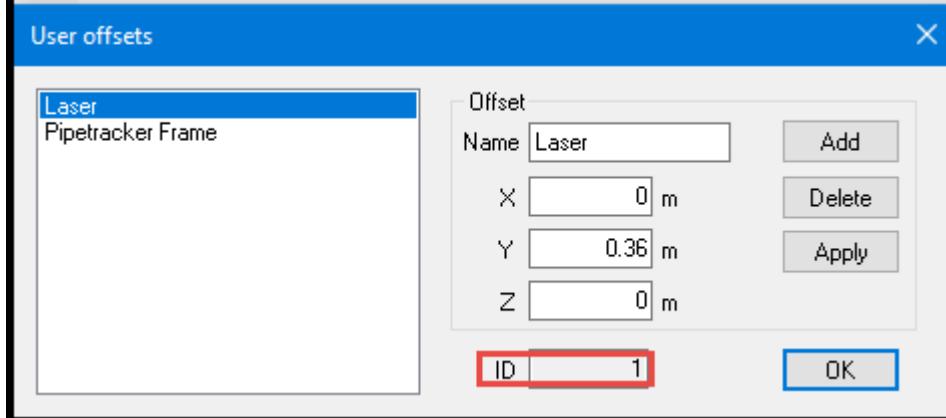
## 3.8 ExportETR

Description:

```

<!-- 7 Export etr file in NE -->
<ExportAsciiTask po="" name="Export etr" logKeyword="" >
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}" />
        <Input xml="{DatabaseName}" />
        <Export type="xyz_etrack">
            <BlockId>{SBDBlockId}</BlockId>
            <Output>
                <!--
                    OutputMethod          // PATH = 0, APPENDTOFILE = 1
                    ReduceMethod         // ALL = 0, INTERVAL = 1
                    SelectionMethod      // CLOSEST = 0, MINIMUM = 1, MAXIMUM = 2
                -->
                <Location outputMethod="0" prefixKPRange="no" combineFiles="no" sortCombined="no">
                    <Path>{BaseDirectory}\output\etr_export</Path>
                    <AppendPath></AppendPath>
                </Location>
                <DbFolderUsage createSubdir="no" prefixFileName="no" useAsFilename="no" />
                <Options writeHeader="yes" sortKP="no" reduceMethod="0" selectionMethod="0" interval=""
                    <!-- interval unit meter-->
                </Output>
                <Units posScale="1.0" depthScale="1.0" />
                <!-- depth scale is positive down -->
                <Settings>
                    <Selection object="n" />
                    <!-- object:
                        -2 = depth channel
                        -1 = CRP
                        0 = User Selected specify in UserOffset element below
                        n = UserOffset Id in the NaviEdit database
                    -->
                    <UserOffset x="1" y="2" z="3" name="MyOffset" /> <!-- Only needed for object="0" -->
                    <Singlebeam interpolateDepth="yes" maxInterpolDist="-1" />
                    <!-- maxInterpolDist -1 Auto -->
                    <Multibeam averagingWindow="1.0" />
                    <!-- averaging windows i meter -->
                    <NoOfDecimals position="3" depth="3" />
                    <Options downsampleFactor="0" downsampleMinDist="0.0" timeFormat="0" />
                    <!-- timeFormat 0 = Default -->
                </Settings>
            </Export>
        </NaviEditTask>
    </ExportAsciiTask>
```

Example for the Useroffset in NaviEdit option n = UserOffset Id in the NaviEdit database:



### 3.9 ExportGRAD

Example:

```
<ExportAsciiTask po="" name="ExportGRAD" logKeyword="" >
  <NaviEditTask version="1.0">
    <SqlServer>localhost</SqlServer>
    <DatabaseName>WFM_Kuda</DatabaseName>
    <Export type="grad">
      <BlockId>
        2045
      </BlockId>
      <Output>
        <Location outputMethod="0" prefixKPRange="no" combineFiles="no" sortCombined="no">
          <Path>C:\temp\GRAD</Path>
          <AppendPath></AppendPath>
        </Location>
        <DbFolderUsage createSubdir="no" prefixFileName="no" useAsfilename="no" />
        <Options writeHeader="yes" sortKP="no" reduceMethod="0" selectionMethod="0" interval="1.0" >
          <!-- interval unit meter-->
        </Options>
        <Units posScale="1.0" depthScale="1.0" />
        <!-- depth scale is positive down -->
      <Settings>
        <Grad>
          <Magnetometer count="12"> <!-- Auxiliary -->
            <Sequences>0 1 2 3 4 5 6 7 8 9 10 11</Sequences>
          </Magnetometer>
          <Gradiometer count="12"> <!-- Singlebeam -->
            <Sequences>3 4 5 6 7 8 9 10 11 12 13 14</Sequences>
          </Gradiometer>
          <Altimeter count="3"> <!-- Singlebeam -->
            <Sequences>0 1 2</Sequences>
          </Altimeter>
        </Grad>
      </Settings>
    </Export>
```

```

    </NaviEditTask>
</ExportAsciiTask>
```

## 3.10 ExportGSF2Bytes

Description:

```

<ExportAsciiTask po="" name="ExportGSF2Bytes" logKeyword="" >
<NaviEditTask version="1.0">
    <SqlServer>localhost</SqlServer>
    <DatabaseName>WFM</DatabaseName>
    <Export type="gsf">
        <!--
            3158 shallow water
            3159 deep water
        -->
        <BlockId>
            3159
        </BlockId>
        <Output>
            <!--
                OutputMethod // PATH = 0, APPENDTOFILE = 1
                ReduceMethod // ALL = 0, INTERVAL = 1
                SelectionMethod // CLOSEST = 0, MINIMUM = 1, MAXIMUM = 2
            -->
            <Location outputMethod="0" prefixKPRange="no" combineFiles="no" sortCombined="no">
                <Path>c:\temp</Path>
                <AppendPath></AppendPath>
            </Location>
            <DbFolderUsage createSubdir="yes" prefixFileName="no" useAsFilename="no" />
            <Options writeHeader="yes" sortKP="no" reduceMethod="0" selectionMethod="0" interval="1.0" />
        </Output>
        <Units posScale="1.0" depthScale="1.0" />
        <!-- depth scale is positive down -->
        <!-- 4 byte output is default. output2byte is needed for compatibility with some 3rd party p
        <!-- maxDepth and maxAcrossDepth are only used in for 2 byte output, to fine tune the preci
        <Settings output2Byte="no" maxDepth="800" maxAcrossTrack="1500"/>
    </Export>
</NaviEditTask>
</ExportAsciiTask>
```

## 3.11 Export GPS Tide

Description:

```

<ExportAsciiTask po="" name="Export gps tide" logKeyword="" output="" OnError="" >
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}" />
        <Input xml="{DatabaseName}" />
        <Export type="gpstide">
            <BlockId>
                57
                58
            </BlockId>
        </Export>
    </NaviEditTask>
</ExportAsciiTask>
```

```

</BlockId>
<Output>
<!-- OutputMethod    // 0 = PATH, 1 = APPENDTOFILE
    ReduceMethod   // 0 = ALL, 1 = INTERVAL
    SelectionMethod // 0 = CLOSEST, 1 = MINIMUM, 2 = MAXIMUM
-->
<Location sortCombined="yes" combineFiles="yes" prefixKPRange="no" outputMethod="0">
    <Path>c:\temp\GPSTide</Path>
    <FileName>asdf.tde</FileName>
    <!-- If empty GUI defaults are used.
        Should be empty if combineFiles is no (false) (and more than one block is exported)
    -->
    <AppendPath><AppendPath/>
</Location>
<DbFolderUsage createSubdir="no" prefixFileName="no" useAsFilename="no" />
<Options interval="10.0" selectionMethod="0" reduceMethod="0" sortKP="no" writeHeader="no" />
    <!-- interval unit meter -->
</Output>
<Units depthScale="1.0" posScale="1.0" />
    <!-- depth scale is positive down -->
<Settings>
    <Selection singleBeamSeq="5" />
        <!-- This is the zero based index of the single beam channel -->
    </Settings>
</Export>
</NaviEditTask>
</ExportAsciiTask>

```

Old option:

```

<ExportGPSTideTask po="" name="Export gps tide" logKeyword="" output="" OnError="" />

```

## 3.12 Export GSF

Description:

```

<ExportAsciiTask po="" name="Export GSF" logKeyword="{SBDFile.Filename}" OnError="" />
    <NaviEditTask version="1.0">
        <SqlServer>localhost</SqlServer>
        <DatabaseName>MyNaviEditDB</DatabaseName>
        <Export type="gsf">
            <BlockId>
                {SBDBlockId}
            </BlockId>
            <Output>
                <!--
                    OutputMethod // PATH = 0, APPENDTOFILE = 1
                    ReduceMethod // ALL = 0, INTERVAL = 1
                    SelectionMethod // CLOSEST = 0, MINIMUM = 1, MAXIMUM = 2
                -->
                <Location outputMethod="0" prefixKPRange="no" combineFiles="no" sortCombined="no">
                    <Path>{BaseDirectory}\Output</Path>

```

```

        <AppendPath></AppendPath>
    </Location>
    <DbFolderUsage createSubdir="no" prefixFileName="no"
useAsFilename="no" />
        <Options writeHeader="yes" sortKP="no" reduceMethod="0"
selectionMethod="0" interval="1.0" />
            <!-- interval unit meter-->
        </Output>
        <Units posScale="1.0" depthScale="1.0" />
            <!-- depth scale is positive down -->
    </Export>
</NaviEditTask>
</ExportAsciiTask>

```

### 3.13 Export LAS/LAZ

Description: The XML shows you how to export LAS data.

By setting the combineFiles attributes to 'yes', the export combines several blocks to one rather than exporting file by file.

```

<ExportAsciiTask po=" " name="Export" logKeyword="{SBDFFile.Filename}" OnError=" "
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}"/>
        <Input xml="{DatabaseName}"/>
        <Export type="bin_las">
            <BlockId>
                {SBDBlockId}
            </BlockId>
            <Output>
                <!--
                    OutputMethod                                // PATH = 0, APPENDTOFILE = 1
                    ReduceMethod                               // ALL = 0, INTERVAL = 1
                    SelectionMethod                          // CLOSEST = 0, MINIMUM = 1, MAXIMUM = 2
                -->
                <Location outputMethod="0" prefixKPRange="no" combineFiles="no"
sortCombined="no">
                    <Path>{BaseDirectory}\Output</Path>
                    <AppendPath></AppendPath>
                </Location>
                <DbFolderUsage createSubdir="no" prefixFileName="no"
useAsFilename="no" />
                    <Options writeHeader="yes" sortKP="no" reduceMethod="0"
selectionMethod="0" interval="1.0" />
                        <!-- interval unit meter-->
                    </Output>
                    <Units posScale="1.0" depthScale="1.0" />
                        <!-- depth scale is positive down -->
                <Settings>
                    <!-- lasFormat: 1 = 1.1, 2 = 1.2, 3 = 1.3, 4 = 1.4 -->
                    <LAS enableCompression="yes" positiveUp="false"
applyLeapSeconds="yes" lasFormat="4" />
                </Settings>
            </Export>
        </NaviEditTask>

```

```
</ExportAsciiTask>
```

## 3.14 Export NED

Description:

```
<ExportAsciiTask po="" name="" logKeyword="" output="" >
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}" />
        <Input xml="{DatabaseName}" />
        <Input xml="{Geodesy}" />
        <Export type="bin_ned">
            <BlockId>12</BlockId>
            <Output>
                <!-- OutputMethod      // 0 = PATH, 1 = APPENDTOFILE
                    ReduceMethod    // 0 = ALL, 1 = INTERVAL
                    SelectionMethod // 0 = CLOSEST, 1 = MINIMUM, 2 = MAXIMUM
                -->
                <Location outputMethod="0" sortCombined="no" combineFiles="no"
prefixKPRange="no">
                    <Path>c:\Data\Output\NED_export</Path>
                    <AppendPath/>
                </Location>
                <DbFolderUsage useAsFilename="no" prefixFileName="no"
createSubdir="no"/>
                    <Options interval="1.0" selectionMethod="0" reduceMethod="0"
sortKP="no" writeHeader="yes"/>
                        <!-- interval unit meter-->
                    </Output>
                    <Units depthScale="1.0" posScale="1.0"/>
                        <!-- depth scale is positive down -->
                    <Settings>
                        <Output TPEType="-1"/>
                        <!-- TPEType: -1 = Depth,
                            0 = X,
                            1 = Y,
                            2 = Z,
                            3 = Pos,
                            4 = FeatureSize
                        -->
                    </Settings>
                </Export>
            </NaviEditTask>
        </ExportAsciiTask>
```

Old option:

```
<ExportNETask po="" name="" logKeyword="" output="" >
```

## 3.15 Export NavInXTF

Description: Export NavInXTF

```

<ExportAsciiTask po="" name="" logKeyword="" output="" >
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />
    <Export type="xynavinxtf">
      <BlockId>62</BlockId>
      <Output>
        <!-- OutputMethod // PATH = 0, APPENDTOFILE = 1
            ReduceMethod // ALL = 0, INTERVAL = 1
            SelectionMethod // CLOSEST = 0, MINIMUM = 1, MAXIMUM = 2
        -->
        <Location outputMethod="0" prefixKPRange="no" combineFiles="no"
        sortCombined="no">
          <Path>C:\Temp</Path>
          <AppendPath></AppendPath>
        </Location>
        <DbFolderUsage createSubdir="no" prefixFileName="no"
        useAsFilename="no" />
        <Options writeHeader="yes" sortKP="no" reduceMethod="0"
        selectionMethod="0" interval="1.0" />
        <!-- interval unit meter-->
      </Output>
      <Units posScale="1.0" depthScale="1.0" />
      <!-- depth scale is positive down -->
      <Settings>
        <Selection object="-1" />
        <!-- object -1 CRP -->
        <Singlebeam interpolateDepth="yes" maxInterpolDist="-1" />
        <!-- maxInterpolDist -1 Auto -->
        <Multibeam averagingWindow="1.0" />
        <!-- averaging windows i meter -->
        <NoOfDecimals position="3" depth="3" />
        <Options downsampleFactor="0" downsampleMinDist="0.0" timeFormat="0" />
        <!-- timeFormat 0 = Default -->
      </Settings>
    </Export>
  </NaviEditTask>
</ExportAsciiTask>

```

## 3.16 Export Magnetic Field

Example

```

<ExportAsciiTask po="" name="Export Magnetic Field"
logKeyword="{SBDFile.Filename}" OnError=" " >
  <NaviEditTask version="1.0">
    <SqlServer>localhost</SqlServer>
    <DatabaseName>NaviEditDB</DatabaseName>
    <Export type="magneticfield">

```

```

<BlockId>
    {SBDBlockId}
</BlockId>
<Output>
    <!--
        OutputMethod // PATH = 0, APPENDTOFILE = 1
        ReduceMethod // ALL = 0, INTERVAL = 1
        SelectionMethod // CLOSEST = 0, MINIMUM = 1, MAXIMUM = 2
    -->
    <Location outputMethod="0" prefixKPRange="no" combineFiles="yes"
sortCombined="no">
        <Path>c:\temp\Mag</Path>
        <AppendPath></AppendPath>
    </Location>
    <DbFolderUsage createSubdir="no" prefixFileName="no"
useAsFilename="no" />
    <!-- interval unit meter-->
    <Options writeHeader="no" sortKP="no" reduceMethod="0"
selectionMethod="0" interval="10.0" />
    </Output>
    <Units posScale="1.0" depthScale="1.0" /> <!-- depth scale is positive
down -->
    <Settings>
        <ExportSettings outputAll="false" outputFreq="10"
addExtremeEasting="false" trackId="Track1"/>
        <FlightHeight useAltimeter="true" useMbe="false"/> <!-- Exact one of
these must be true -->
        <Altimeter seq="1" useRaw="false"/> <!-- seq is the zero based index
of the altimeter single beam channel -->
        <Multibeam width="1.0"/> <!-- Width of the multibeam swath in
meters -->
    </Settings>
    </Export>
</NaviEditTask>
</ExportAsciiTask>

```

## 3.17 Export Surface Pressure

Description:

```

<ExportAsciiTask po="" name="" logKeyword="" output="" >
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}" />
        <Input xml="{DatabaseName}" />
        <Input xml="{Geodesy}" />
        <Export type="single">
            <BlockId>1 2 3</BlockId>
            <Output>
                <!-- OutputMethod // 0 = PATH, 1 = APPENDTOFILE
                    ReduceMethod // 0 = ALL, 1 = INTERVAL
                    SelectionMethod // 0 = CLOSEST, 1 = MINIMUM, 2 = MAXIMUM
                -->
                <Location sortCombined="yes" combineFiles="yes" prefixKPRange="no"
outputMethod="0">

```

```

<Path>c:\Data\Output\SurfacePressure</Path>
<FileName>asdf.txt</FileName>
<!-- if empty GUI defaults are used. Should be empty if
combineFiles is no (false) (and more than one block is exported --&gt;
    &lt;AppendPath/&gt;
&lt;/Location&gt;
&lt;DbFolderUsage useAsFilename="no" prefixFileName="no"
createSubdir="no"/&gt;
    &lt;Options interval="10.0" selectionMethod="0" reduceMethod="0"
sortKP="no" writeHeader="no"/&gt;
    &lt;/Output&gt;
    <!-- depth scale is positive down --&gt;
    &lt;Settings addName="no" decimals="5" scale="0.001"&gt;
        &lt;Selection singleBeamSeq="3"/&gt;
        <!-- This is the zero based index of the single beam channel --&gt;
    &lt;/Settings&gt;
    &lt;/Export&gt;
&lt;/NaviEditTask&gt;
&lt;/ExportAsciiTask&gt;
</pre>

```

Old option:

```
<ExportSurfacePressureTask po="" name="" logKeyword="" output="">
```

## 3.18 Export Visual Soft Navigation

Description:

```

<ExportAsciiTask po="" name="" logKeyword="" output="">
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}" />
        <Input xml="{DatabaseName}" />
        <Export type="xyvsoftvev">
            <BlockId>157 158</BlockId>
            <Output>
                <!-- OutputMethod // 0 = PATH, 1 = APPENDTOFILE
                    ReduceMethod // 0 = ALL, 1 = INTERVAL
                    SelectionMethod // 0 = CLOSEST, 1 = MINIMUM, 2 = MAXIMUM
                --&gt;
                &lt;Location outputMethod="0" prefixKPRange="no" combineFiles="no"
sortCombined="no"&gt;
                    &lt;Path&gt;c:\Data\Output\VSoftExport&lt;/Path&gt;
                    &lt;AppendPath/&gt;
                &lt;/Location&gt;
                &lt;DbFolderUsage createSubdir="no" prefixFileName="no"
useAsfilename="no"/&gt;
                    &lt;Options writeHeader="yes" selectionMethod="0" reduceMethod="0"
sortKP="yes" interval="10.0"/&gt;
                    <!-- interval unit meter--&gt;
                &lt;/Output&gt;
                &lt;Units depthScale="1.0" posScale="1.0"/&gt;
                    <!-- depth scale is positive down --&gt;
                &lt;Settings&gt;
                    &lt;Selection object="-1"/&gt;
                    <!-- object -1 CRP --&gt;
                    &lt;Singlebeam maxInterpolDist="-1" interpolateDepth="yes"/&gt;
</pre>

```

```

<!-- maxInterpolDist -1 Auto -->
<Multibeam averagingWindow="1.0"/>
  <!-- averaging windows in meter -->
  <Options downsampleMinDist="0.0" downsampleFactor="20"/>
    <!-- downsampleMinDist 0 disabled,
        downsampleFactor 0 disabled
    -->
  </Settings>
</Export>
</NaviEditTask>
</ExportAsciiTask>

```

Old option:

```
<ExportVisualSoftNavigationTask po="" name="" logKeyword="" output="">
```

## 3.19 Export Report

Description: Exports report of a NaviEdit block.

It is not a data export, but an export of the HeaderEditor text report.

```

<!-- Export report -->
<BlockSettingsTask po="" name="Export NaviEdit Report" logKeyword="{AllFile.Filename}" OnError="OnReport" version="1.0">
  <NaviEditTask version="1.0">
    <SqlServer>localhost</SqlServer>
    <DatabaseName>WFM</DatabaseName>
    <ExportReport>
      <BlockId>
        {AllBlockId}
      </BlockId>
      <ReportFile>
        {BaseDirectory}\Output\Block_{AllFile.Filename}_Report.txt
      </ReportFile>
    </ExportReport>
  </NaviEditTask>
</BlockSettingsTask>

```

## 3.20 Find Subfolders

Description:

```

<NaviEditSubFoldersTask po="2" name="Find NaviEdit sub folders" logKeyword=""
" output="NERootFolder" OnError=" ">
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />
    <Input xml="{Geodesy}" />
      <FindSubFolders version="1.0">
        <DbFolder startFromRoot="false">\48</DbFolder>
      </FindSubFolders>
    </NaviEditTask>
  </NaviEditSubFoldersTask>

```

## 3.21 Find Blocks

Description:

```
<!-- Find blocks -->
<FindBlocksTask po="" name="" logkeyword="" output="" >
  <NaviEditTask version="1.0">
    <SqlServer>localhost</SqlServer>
    <DatabaseName>WFM</DatabaseName>
    <!-- Use an empty dbPath if the whole database should be searched and set
includeSubFolders to true, note the backslash notation \NaviPac\SomeSubFolder
-->
    <FindBlocks version>1.0</FindBlocks version>
    <blockType>256</blockType>
    <ignoreMbeBlocks>no</ignoreMbeBlocks>
    <firstTime>2007-10-28 20:00:00</firstTime>
    <lastTime>2027-10-28 22:00:00</lastTime>
    <dbPath>\SubFolder</dbPath>
    <includeSubFolders>false</includeSubFolders>
  </NaviEditTask>
</FindBlocksTask>
```

## 3.22 Find NaviPac Blocks

Description:

```
<!-- Find NaviPac blocks in folder -->
<FindNaviPacBlocksTask po="" name="Find blocks in folder" logKeyword="" 
output="MyCollectedBlocks">
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />
    <FindBlocksInFolder version="1.0">
      <DbFolder>\48</DbFolder>
    </FindBlocksInFolder>
  </NaviEditTask>
</FindNaviPacBlocksTask>
```

## 3.23 Get Parent Folder Names

Description: Extract folder names from path. Specify either folder levels or indexes to get these folders. Use optional DirectorySeperator for output.

Examples:

```
Input path=C:\folder1\folder2\folder3\folder4\
Folderlevel=-3 -> Output=C:\folder1\
Folderlevel=2, directoryseperator=/ -> Output=/folder3/folder4/
Index=0 2 -> Output=\folder2\folder4\

<GetParentFolderNamesTask po="1" name="Extract folder name"
logKeyword="FolderName" OnErrorAction="" path="C:"
```

```

\folder1\folder2\folder3\folder4\" output="folders" folderlevels="2"
directoryseparator="/" />
<GetParentFolderNamesTask po="2" name="Extract folder name"
logKeyword="FolderName" OnErrorAction="" path="C:
\folder1\folder2\folder3\folder4\" output="folders" folderlevels="-3" />
<GetParentFolderNamesTask po="3" name="Extract folder name"
logKeyword="FolderName" OnErrorAction="" path="C:
\folder1\folder2\folder3\folder4\" output="folders" indexes="0 2" />

<GroupTask po="1" name="Define variables">
  <InputTask po="1" name="Choose base directory" output="BaseDirectory"
AskForInput="true"/>
  <SetPropertyTask po="2" name="Base directory" input="{BaseDirectory}"
output="BaseDirectory" level="2"/>
  ...
  <GetParentFolderNamesTask po="4" name="Get parent folder" logKeyword="Test
name" path="{BaseDirectory}" output="DBAdress2" folderlevels="-1"/>
  <SetPropertyTask po="5" name="Get parent 2" input="{DBAdress2}2 - processed
data" output="ParentFolder" level="2"/>
</GroupTask>

```

## 3.24 Get Block File Name

Description: Get the original full file name - including full path - of a block in the NaviEdit database by specifying the block id.

Example:

```

<!-- Get block file name -->
<GetBlockFileName po=" " name="Get Block File Name" logKeyword=" " output=" " >
  <NaviEditTask version="1.0">
    <SqlServer>HKP-PREC7540</SqlServer>
    <DatabaseName>WFM_Matt</DatabaseName>
    <GetBlockFileName>
      <BlockId>170</BlockId>
    </GetBlockFileName>
  </NaviEditTask>
</GetBlockFileName>

```

## 3.25 Get KP Value

Description: Get the KP value of the first or last position for the active navigation sensor.  
 Possible values of the 'Which' attribute: start, end.

Can be used as property in a file name.

Example:

```

<!-- Get KP value -->
<GetKPValue po=" " name="KP Start" Which="start" Output="KPStart"
logKeyword="{SBDFFile.Filename}" level="1">
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />

```

```

<GetBlockKP>
    <BlockId>{SBDBlockId}</BlockId>
</GetBlockKP>
</NaviEditTask>
</GetKPValue>

```

## 3.26 Hugin MissionPlan Import

Description: This example shows you how to import a Hugin mission plan.

```

<HuginMissionPlanImportTask po="" name="" logKeyword="" output="">
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}" />
        <Input xml="{DatabaseName}" />
        <Input xml="{Geodesy}" />
        <HuginImport version="1.0" loadRealRunlines="yes" splitRunline="yes"
askOverwrite="no">
            <DbFolder createSubFolders="yes">
                <DbPath>\48\XXX</DbPath>
            </DbFolder>
            <FileList>c:\Data\mission.mp </FileList>
            <RunlineFolder>c:\Data\RunlineAndMissionPlan</RunlineFolder>
        </HuginImport>
    </NaviEditTask>
</HuginMissionPlanImportTask>

```

## 3.27 Import CTD

Description: This function allows you to import a ASCII file containing CTD data.

```

<!-- Import CTD -->
<CTDImportTask po="" name="" logKeyword="" output="">
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}" />
        <Input xml="{DatabaseName}" />
        <Input xml="{Geodesy}" />
        <AsciiImport version="1.0" template="HuginCTD" type="CTD">
            <Source>
                <FileList>c:\Data\CTD\ctd.txt</FileList>
            </Source>
            <Destination appendName="029" addProjDiveNo="yes" layerInterval="1.0"
reduceLayers="yes" extendLength="400" extendProfile="yes" deepSplit="yes">
                <!-- Note backslash notation -->
                <!-- Split at deepest point -->
                \MyFolder\CTD
                <!-- NE Folder - Note backslash notation -->
            </Destination>
        </AsciiImport>
    </NaviEditTask>
</CTDImportTask>

```

**Note:**

```
<SelectionPolicy>1</SelectionPolicy> <!-- 0 = CTD_SELECT_BY_TIME, 1 = CTD_SELECT_BY_DISTANCE, 2
<ProfileUsage>0</ProfileUsage> <!-- 0 = CTD_USE_NEAREST, 1 = CTD_USE_INTERPOLATED -->
<ProfileDistance>10</ProfileDistance>
```

## 3.28 Import Pressure

Description: This function allows you to import a ASCII file containing pressure data.

```
<!-- Import pressure -->
<PressureImportTask po="" name="" logKeyword="" output="" >
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />
    <Input xml="{Geodesy}" />
    <AsciiImport version="1.0" template="SurfacePressure2" type="pressure">
      <Source>
        <FileList>c:\Data\Aanderaa.txt</FileList>
      </Source>
      <Destination appendName="" addProjDiveNo="no" deepSplit="yes">
        <!-- Note backslash notation -->
        \MyFolder\SurfacePressure
      </Destination>
    </AsciiImport>
  </NaviEditTask>
</PressureImportTask>
```

## 3.29 Import Runline

Description: Import Runline. It is important to keep the extension of the runline file name.

```
<!-- Import Runline -->
<ImportRunlineTask po="" name="Import Runline" logKeyword="" output="" >
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />
    <Input xml="{Geodesy}" />
    <ImportRunline version="1.0">
      <DbFolder>
        \Testroot\Testchild
      </DbFolder>
      <RunlineFile>
        C:\Data\runline.rln
      </RunlineFile>
      <Name>
        myRunline.rln
      </Name>
    </ImportRunline>
  </NaviEditTask>
</ImportRunlineTask>
```

## 3.30 Import SVP

Description: This function allows you to import a ASCII file containing Sound Velocity data.

```
<!-- Import SVP -->
<SVPIimportTask po="" name="" logKeyword="" output="" >
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}"/>
    <Input xml="{DatabaseName}"/>
    <Input xml="{Geodesy}"/>
      <AsciiImport version="1.0" template="Hugin" type="SVP">
        <Source>
          <FileList>c:\Data\VesselCTD.txt</FileList>
        </Source>
        <Destination appendName="029" addProjDiveNo="yes" layerInterval="1.0"
reduceLayers="yes" deepSplit="yes">
          <!-- Note backslash notation -->
          \MyFolder\SVP
        </Destination>
      </AsciiImport>
    </NaviEditTask>
  </SVPIimportTask>
```

## 3.31 Import Tide

Description: This function allows you to import a ASCII file containing tide data.  
ASCII tide import -> smoothing tide.

Option 1:

```
<TideImportTask po="" name="" logKeyword="" output="TideBlocks">
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}"/>
    <Input xml="{DatabaseName}"/>
    <Input xml="{Geodesy}"/>
      <AsciiImport version="1.0" template="GPS Tide" type="tide">
        <Source>
          <FileList>
            {TideFile}
          </FileList>
        </Source>
        <Destination deepSplit="yes" appendName="" addProjDiveNo="no">
          \GPSTide <!-- NE Folder - Note backslash notation -->
        </Destination>
      </AsciiImport>
    </NaviEditTask>
  </TideImportTask>
```

Option 2:

Note the batchjob-element is exactly as a saved batch job, so if you want to change any of the parameters try to make a batch job and save it.

```
<BatchSmoothTask po="" name="Smooth Tide">
```

```

<NaviEditTask version="1.0">
<Input xml="{SqlServer}" />
<Input xml="{DatabaseName}" />
<BatchProcess>
  <BlockId> <!-- Multiple allowed, but smoothed as single blocks -->
    {TideBlocks}
  </BlockId>
<batchjob>
  <edititem action="Smooth points" target="Tide[Active]">
    <sensortype>15</sensortype> <!-- Tide -->
    <actionid>40100</actionid> <!-- Smooth -->
    <selection>0</selection> <!-- Active = 0, All = 1, Index = 2 -->
    <index>0</index>
    <smooth>
      <wavelength>3600</wavelength> <!-- seconds -->
    </smooth>
  </edititem>
</batchjob>
</BatchProcess>
</NaviEditTask>
</BatchSmoothTask>

```

### 3.32 Las Import

Description: The LAS (LASer) format is a file format designed for the interchange and archiving of lidar point cloud data.

It is an open, binary format specified by the American Society for Photogrammetry and Remote Sensing (ASPRS).

```

<NaviEditTask version="1.0">
<SqlServer>localhost</SqlServer>
<DatabaseName>WFM</DatabaseName>
<Geodesy>
  <Source>
    <Ellipsoid epsg="7030" name="WGS 84">
      <!-- eiva does not use the ellipsoid epsg number, but indirectly
by use of the projection epsg number-->
      <SemiMajorAxis>6378137.0</SemiMajorAxis>
      <InverseFlattening>298.257223563</InverseFlattening>
    </Ellipsoid>
    <Projection epsg="32629" name="UTM (north)">
      <!-- epsg 32629 requires/implies wgs 84 ellipsoid -->
      <EivaType>5</EivaType>
      <!-- integer PROJPAC.h UTM N = 5 -->
      <FalseEasting>500000</FalseEasting>
      <FalseNorthing>0</FalseNorthing>
      <FirstParallel>0</FirstParallel>
      <SecondParallel></SecondParallel>
      <Scale>0.9996</Scale>
      <OrigLatitude>0</OrigLatitude>
      <OrigLongitude>-9</OrigLongitude>
      <Zone>29</Zone>
    </Projection>
  </Source>

```

```

<Destination>
    <Ellipsoid epsg="7030" name="WGS 84">
        <!-- eiva does not use the ellipsoid epsgs number, but indirectly
by use of the projection epsgs number-->
        <SemiMajorAxis>6378137.0</SemiMajorAxis>
        <InverseFlattening>298.257223563</InverseFlattening>
    </Ellipsoid>
    <Projection epsg="32629" name="UTM (north)">
        <!-- epsg 32629 requires/implies wgs 84 ellipsoid -->
        <EivaType>5</EivaType>
        <!-- integer PROJPAC.h UTM N = 5 -->
        <FalseEasting>500000</FalseEasting>
        <FalseNorthing>0</FalseNorthing>
        <FirstParallel>0</FirstParallel>
        <SecondParallel></SecondParallel>
        <Scale>0.9996</Scale>
        <OrigLatitude>0</OrigLatitude>
        <OrigLongitude>-9</OrigLongitude>
        <Zone>29</Zone>
    </Projection>
</Destination>
<Source></Source>
<Shift method="0" name="No shift"></Shift>      <!-- name is not in use
-->
</Geodesy>
<LasInterpreter version="1.0">
    <Destination matchFolder="no" importSVP="no" linkSVP="no">
        <Folder>
            \LAS
        </Folder>
    </Destination>
    <Source>
        <FileList>
            d:\Data\20210713.las
        </FileList>
    </Source>
    <Filters>
        <!-- Note when the setDeletedFlag is set to yes, the beams below the
threshold are marked as deleted/rejected.
        When the quality threshold flag is set in the HeaderEditor, the
beams are not marked as deleted, but marked as filtered in NaviModel.
        Beams marked as deleted/rejected can be manually accepted in
NaviModel -->
        <MBEQuality setDeletedFlag="yes" qualityThreshold="0"/>
    </Filters>
    <Options disableBathy="no" useOnlineSV="no" onlineSVStart="-1.1"
onlineSEnd="15.5" scanReductionImportEvery="1"
skipScansWithoutMotion="false" />
        <Settings invertZ="no"/>
    </LasInterpreter>
</NaviEditTask>

```

Two commands can be used:

Task Description

**Tasks**

- Input Task
- Las Import
- Level Check
- License Check
- Link Blocks
- Merge Data Files
- Merge Pipetrackers
- Multibeam Import
- NaviEdit Util Tasks
- NaviEditSubFolders
- NaviPac Interpreter
- Navlab Import
- NMCLI
- Overwrite Property
- Pipetracker Merge

**Las Import**

**Description**

Import \*.LAS files

**XML Example**

```
<LasImportTask po="" name="" logKeyword="" OnError="" output="">
    <NaviEditTask version="1.0">
        <SqlServer>localhost</SqlServer>
        <DatabaseName>WFM_Kuda</DatabaseName>
        <Geodesy>
```

Task Description

**Tasks**

- Merge Pipetrackers
- Multibeam Import
- NaviEdit Util Tasks
- NaviEditSubFolders
- NaviPac Interpreter
- Navlab Import
- NMCLI
- Overwrite Property
- Pipetracker Merge
- Property Iterator
- Read File
- Recalc Bathy
- Recalc KP

**Multibeam Import**

**Description**

Import multibeam files. Use filetype to specify type of file to import, see examples below for possible types.

**XML Example**

```
<MultibeamImportTask po="" name="" logKeyword="" OnError="" output="" filetype="">
    <NaviEditTask>
        <SqlServer>localhost</SqlServer>
        <DatabaseName>WFM_Kuda</DatabaseName>
        <Geodesy>
        <Filetype>The filetype to import, options are sbd, em, gsf, las and s7k
```

### 3.33 Link Blocks

Description:

```
<!-- Link block -->
<LinkTask po="" name="Link block" logKeyword="" output="">
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}" />
        <Input xml="{DatabaseName}" />
        <LinkBlock version="1.0">
            <Parent>      <!-- Assuming SurveyType -->
                <BlockId>
                    158
                    197
                </BlockId>   <!-- Multiple allowed -->
            </Parent>      <!-- Allow child types: SVP, CTD, Tide, SurfacePressure
-->
```

```

<Child type="SVP">
    <!-- Specify either BlockId or database Folder path, Folder is not
possible for SurfacePressure-->
    <BlockId>      <!-- Multiple allowed-->
        3 6
    </BlockId>
    <!-- Note backslash notation -->
    <!-- <Folder>SVP\Subfolder</Folder> -->
</Child>
<Settings>
    <SelectionPolicy>2</SelectionPolicy>      <!-- 0 = CTD_SELECT_BY_TIME,
1 = CTD_SELECT_BY_DISTANCE,2 = CTD_SELECT_BY_KP -->
    <ProfileUsage>0</ProfileUsage>      <!-- 0 = CTD_USE_NEAREST, 1 =
CTD_USE_INTERPOLATED -->
        <ProfileDistance>
            33
        </ProfileDistance>
    </Settings>
</LinkBlock>
</NaviEditTask>
</LinkTask>

```

### 3.34 Link Block Tide

Description: Link tide file to block (SBD file).

```

<!-- Link block tide -->
<LinkTask po=" " name="Link block tide" logKeyword=" " output=" " OnError=" " >
    <NaviEditTask version="1.0">
        <SqlServer>localhost</SqlServer>
        <DatabaseName>WFM</DatabaseName>
        <LinkBlock version="1.0">
            <Parent> <!-- Assuming SurveyType -->
            <BlockId>154</BlockId> <!-- Multiple allowed -->
        </Parent>
        <!-- Allow child types: SVP, CTD, Tide, SurfacePressure -->
        <Child type="Tide">
            <BlockId>{TideBlocks}</BlockId> <!-- Multiple allowed-->
        </Child>
        <Settings>
            <ProfileUsage>1</ProfileUsage>
            <!-- 0 = Use nearest profile, 1 = Use interpolate -->
        </Settings>
        <SurfacePressure useCorrection="yes" inverseCorrection="yes">
            <!-- inverse correction is for GNSS Tide to predicted tide -->
            <UsageType>PressureBlock</UsageType>
            <!-- Fixed, PressureBlock, ThisBlock -->
            <Fixed>1.0111</Fixed>
            <Reference>0.0</Reference>
            <Sequence>0</Sequence>
        </SurfacePressure>
    </LinkBlock>
</NaviEditTask>
</LinkTask>

```

### 3.35 Link Block Surface Pressure

Description:

```
<LinkTask po="" name="Link block surface pressure" logKeyword="" output="">
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />
    <LinkBlock version="1.0">
      <Parent> <!-- Assuming SurveyType -->
        <BlockId>
          512 513
        </BlockId> <!-- Multiple allowed -->
      </Parent> <!-- Allow child types: SVP, CTD, Tide, SurfacePressure -->
      <Child type="SurfacePressure">
        <BlockId>
          155
        </BlockId> <!-- Only one allowed-->
      </Child>
    </LinkBlock>
  </NaviEditTask>
</LinkTask>
```

### 3.36 Merge Data Files

Description:

The navLab files are first found by two file watchers like this:

```
<FileWatcherTask po="2" name="NavLab Position" Filter="position_smooth.txt"
location="c:\myData\" output="PositionSmooth">
  < SetPropertyTask po="1" input="{PositionSmooth}" output="PositionSmooth"
level="3"/>
</FileWatcherTask>

<FileWatcherTask po="2" name="NavLab Attitude" Filter="attitude_smooth.txt"
location=" c:\myData\" output="AttitudeSmooth">
  < SetPropertyTask po="1" input="{AttitudeSmooth}" output="AttitudeSmooth"
level="3"/>
</FileWatcherTask>
```

And then merged into the All files:

```
<MergeFileTask po="4" name="Merge NavLab Pos + Att"
logKeyword="{AllFile.Filename}" output="MergedFiles">
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />
    <Input xml="{Geodesy}" />
    <Merge version="1.0">
      <Destination>
        <BlockIds>
```

```
        {AllBlockId}
    </BlockIds>
</Destination>
<Source inputType="1" name="NavLab" addNew="yes" doGapCheck="yes"
maxGapSec="2.0">
    <!-- inputType 1 = NAVLAB, 2 = NAVLAB_CRP_CALENDAR_FORMAT, 6 =
NAVIEDIT -->
    <Position do="yes">
        <FileList>
            {PositionSmooth}
        </FileList>
    </Position>
    <Attitude do="no">
        <FileList>
            {AttitudeSmooth}
        </FileList>
    </Attitude>
    <Bathy do="no" usePositionFile="yes">
        </Bathy>
    </Source>
    </Merge>
</NaviEditTask>
</MergeFileTask>

<MergeFileTask po="5" name="Merge NavLab Pressure"
logKeyword="{AllFile.Filename}">
    <NaviEditTask version="1.0">,
        <Input xml="{SqlServer}" />
        <Input xml="{DatabaseName}" />
        <Input xml="{Geodesy}" />
        <Merge version="1.0">
            <Destination>
                <BlockIds>
                    {AllBlockId}
                </BlockIds>
            </Destination>
            <Source inputType="2" name="NavLabCRPCal" addNew="yes" doGapCheck="yes"
maxGapSec="2.0">
                <!-- inputType 1 = NAVLAB, 2 = NAVLAB_CRP_CALENDAR_FORMAT, 6 =
NAVIEDIT -->
                <Position do="no">
                    <FileList>
                    </FileList>
                </Position>
                <Attitude do="no">
                    <FileList>
                    </FileList>
                </Attitude>
                <Bathy do="yes" usePositionFile="no">
                    <FileList>
                        {PressureSmooth}
                    </FileList>
                </Bathy>
            </Source>
            </Merge>
```

```

    </NaviEditTask>
</MergeFileTask>
```

### 3.37 Merge Pipetrackers

Description: Merge Pipetracker Ascii file.

```

<!-- Merge Pietracker -->
<MergePipetrackerTask po="" name="Combine Pipetrackers"
logKeyword="{AllFile.Filename}">
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />
    <Input xml="{Geodesy}" />
      <CombinePipetrackers version="1.0">
        <BlockIds>3</BlockIds> <!-- Multiple allowed -->
      </CombinePipetrackers>
    </NaviEditTask>
  </MergePipetrackerTask>
```

### 3.38 Merge Navlab Calendar

Description:

```

<!-- Merge Navlab Calendar -->
<MergeFileTask po="" name="Merge NavLab" logKeyword="" output="">
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />
    <Input xml="{Geodesy}" />
    <Merge version="1.0">
      <Destination>
        <!-- either block id set or folder id/path (set) -->
        <!-- first version handles block ids only-->
        <BlockIds>157 158 159</BlockIds>
      </Destination>
      <!-- EBlockMergeWizard.h enum InputTypeNavFile = 0, NavLab,
NavLabCRPCal, NMA, PosMV, SeatexRealHeave, NaviEdit, IxSeaDelphINS, Applanix,
Trimble, Allseas, Standard,HorizontalCTDHugin3000, HorizontalCTDHugin1000,
Subsea7AIV, CathxMbe, Robotics2G, PsonNavinputtype is the important one, name
is just passed as a service -->
      <Source name="NavLabCRPCal" addNew="yes" inputType="2" doGapCheck="yes"
maxGapSec="2.0">
        <Position do="no">
          <!-- possibly add an creating missing sensor -->
          <FileList>c:
\Data\Navlab\position_smooth_calender_format.txt</FileList>
        </Position>
        <Attitude do="no">
          <FileList>c:
\Data\Navlab\attitude_smooth_calender_format.txt</FileList>
        </Attitude>
```

```

<Bathy do="yes" usePositionFile="no">
    <Mounting z="0.33" y="-0.22" x="-0.11"/>
    <FileList>c:\Data\SmoothedPressure.txt</FileList>
</Bathy>
</Source>
</Merge>
</NaviEditTask>
</MergeFileTask>

```

### 3.39 Merge Navlab Classic

Description:

```

<MergeFileTask po=" " name="Merge NavLab" logKeyword="" output="">
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}" />
        <Input xml="{DatabaseName}" />
        <Input xml="{Geodesy}" />
        <Merge version="1.0">
            <Destination>
                <!-- either block id set or folder d/path (set) -->
                <!-- first version handles block ids only-->
                <BlockIds>157 158 159</BlockIds>
            </Destination>
            <!-- EBlockMergeWizard.h enum InputTypeNavFile = 0, NavLab,
NavLabCRPCal, NMA, PosMV, SeatexRealHeave, NaviEdit, IxSeaDelphINS, Applanix,
Trimble, Allseas, Standard,HorizontalCTDHugin3000, HorizontalCTDHugin1000,
Subsea7AIV, CathxMbe, Robotics2G, PsonNavinputtype is the important one, name
is just passed as a service -->
            <Source name="NavLab" addNew="yes" inputType="1" doGapCheck="yes"
maxGapSec="2.0">
                <Position do="yes">
                    <!-- possibly add an creating missing sensor -->
                    <FileList>c:\Data\position_smooth.txt</FileList>
                </Position>
                <Attitude do="yes">
                    <Mounting z="6.6" y="5.5" x="4.4"/>
                    <FileList>c:\Data\attitude_smooth.txt </FileList>
                </Attitude>
                <Bathy do="no" usePositionFile="yes"> </Bathy>
            </Source>
        </Merge>
    </NaviEditTask>
</MergeFileTask>

```

### 3.40 MergeTerraPosNMA

Description:

```

<!-- MergeTerraPos NMA -->
<MergeFileTask po=" " name="Merge terrapos NMA" logKeyword=" "
output="MergedFiles">

```

```

<NaviEditTask version="1.0">
  <Input xml="{SqlServer}"/>
  <Input xml="{DatabaseName}"/>
  <Input xml="{Geodesy}"/>
  <Merge version="1.0">
    <Destination>
      <BlockIds>1078</BlockIds>
    </Destination>
    <Source name="NMA" addNew="yes" inputType="3" doGapCheck="yes"
maxGapSec="2.0">
      <!-- inputType 1 = NAVLAB,
          2 = NAVLAB_CRP_CALENDAR_FORMAT,
          3 = NMA,
          6 = NAVIEDIT -->
      <Position do="yes">
        <Mounting z="3" y="2" x="1"/>
        <FileList>d:\Data\MyTerraPosFile.nma </FileList>
      </Position>
    </Source>
  </Merge>
</NaviEditTask>
</MergeFileTask>

```

### 3.41 Merge Delayed Heave

Description: NaviEdit task

```

<!-- Merge Delayed Heave -->
<MergeFileTask po=" " name="Merge delayed heave" logKeyword=" "
output="MergedFiles">
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}"/>
    <Input xml="{DatabaseName}"/>
    <Input xml="{Geodesy}"/>
    <Merge version="1.0">
      <Destination>
        <BlockIds>32</BlockIds>
      </Destination>
      <Source name="PosMVDelayedHeave" addNew="yes" inputType="4"
doGapCheck="yes" maxGapSec="2.0">
        <LocalTimeBias min="0" hours="0"/>
        <!-- inputType
            1 = NAVLAB,
            2 = NAVLAB_CRP_CALENDAR_FORMAT,
            6 = NAVIEDIT -->
        <!-- EBlockMergeWizard.h enum InputTypeNavFile = 0, NavLab,
NavLabCRPCal, NMA, PosMV, SeatexRealHeave, NaviEdit, IxSeaDelphINS,
Applanix, Trimble, Allseas, Standard,HorizontalCTDHugin3000,
HorizontalCTDHugin1000, Subsea7AIV, CathxMbe, Robotics2G, PsonNavinputtype is
the important one, name is just passed as a service -->
        <Position do="no">
          <FileList/>
        </Position>
        <Attitude do="yes">

```

```

    <FileList>c:\CTD_TIDE_HEAVE\250419.111</FileList>
</Attitude>
<Bathy do="no" usePositionFile="no">
    <FileList/>
</Bathy>
</Source>
</Merge>
</NaviEditTask>
</MergeFileTask>

```

### 3.42 Merge Applanix SBET in WFM

Please find below a piece of code to insert in the XML configuration file to merge an Applanix SBET file.

The important part is to set the inputType to 8.

```

<MergeFileTask po="4" name="Merge Applanix" logKeyword="{SbdFile.Filename}"
output="MergedFiles">
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}"/>
        <Input xml="{DatabaseName}"/>
        <Input xml="{Geodesy}"/>
        <Destination>
            <!-- either block id set or folder id/path (set) -->
            <!-- first version handles block ids only-->
            <BlockIds>
                {SBDBlockId}
            </BlockIds>
        </Destination>
        <!-- InputType NavFile = 0, NavLab = 1, NavLabCRPCal = 2, NMA = 3,
PosMV = 4, SeatexRealHeave = 5, NaviEdit = 6, IxSeaDelphINS = 7, Applanix =
8, Trimble = 9, Allseas= 10, Standard = 11, HorizontalCTDHugin3000 = 12,
HorizontalCTDHugin1000 = 13, Subsea7AIV = 14, CathxMbe = 15, Robotics2G = 16,
PsonNav = 17 inputtype is the important one, name is just passed as a service
-->
        <Source inputType="8" name="ApplanixSBET" addNew="yes" doGapCheck="yes"
maxGapSec="2.0">
            <Position do="yes"> <!-- possibly add and creating missing sensor -->
                <FileList>
                    d:\Data\ApplanixPosPacSbet\SBET_140455_1539.out
                </FileList>
            </Position>
        </Source>
    </NaviEditTask>
</MergeFileTask>

```

## 3.43 Move Block

Description:

```
<!-- Move block -->
<BlockSettingsTask po="" name="Move Block" logKeyword="" output="" >
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />
    <MoveBlock>
      <BlockId>19</BlockId>
      <DestinationFolder createMissingFolder="yes">
        <!-- Use backslash notation and start from the root like: \FirstFolder\SecondFolder -->
        \Test
      </DestinationFolder>
    </MoveBlock>
  </NaviEditTask>
</BlockSettingsTask>
```

## 3.44 Multibeam Import

Description: Imports different echosounder data formats.

```
<!-- Multibeam import -->
<MultibeamImportTask po="" filetype="SBD" name="Move Blocks" logKeyword="" output="" >
  <NaviEditTask version="1.0">
    <SqlServer>localhost</SqlServer>
    <DatabaseName>NaviEditDB</DatabaseName>
    <Geodesy>
      <Source>
        <Ellipsoid epsg="7030" name="WGS 84">
          <!-- eiva does not use the ellipsoid epsgs number, but indirectly by use of the proj -->
          <SemiMajorAxis>6378137.0</SemiMajorAxis>
          <InverseFlattening>298.257223563</InverseFlattening>
        </Ellipsoid>
        <Projection epsg="32629" name="UTM (north)">
          <!-- epsg 32629 requires/implies wgs 84 ellipsoid -->
          <EivaType>5</EivaType>
          <!-- integer PROJPAC.h UTM N = 5 -->
          <FalseEasting>500000</FalseEasting>
          <FalseNorthing>0</FalseNorthing>
          <FirstParallel>0</FirstParallel>
          <SecondParallel></SecondParallel>
          <Scale>0.9996</Scale>
          <OrigLatitude>0</OrigLatitude>
          <OrigLongitude>-9</OrigLongitude>
          <Zone>29</Zone>
        </Projection>
      </Source>
      <Destination>
        <Ellipsoid epsg="7030" name="WGS 84">
          <!-- eiva does not use the ellipsoid epsgs number, but indirectly by use of the proj -->
          <SemiMajorAxis>6378137.0</SemiMajorAxis>
        </Ellipsoid>
      </Destination>
    </Geodesy>
  </NaviEditTask>
</MultibeamImportTask>
```

```

        <InverseFlattening>298.257223563</InverseFlattening>
    </Ellipsoid>
    <Projection epsg="32629" name="UTM (north)">
        <!-- epsg 32629 requires/implies wgs 84 ellipsoid -->
        <EivaType>5</EivaType>
        <!-- integer PROJPAC.h UTM N = 5 -->
        <FalseEasting>500000</FalseEasting>
        <FalseNorthing>0</FalseNorthing>
        <FirstParallel>0</FirstParallel>
        <SecondParallel></SecondParallel>
        <Scale>0.9996</Scale>
        <OrigLatitude>0</OrigLatitude>
        <OrigLongitude>-9</OrigLongitude>
        <Zone>29</Zone>
    </Projection>
</Destination>
<Source></Source>
<Shift method="0" name="No shift"></Shift>      <!-- name is not in use -->
</Geodesy>

<!-- filetype="em" -->
<EmInterpreter version="1.0" emType="3" datagram="1">      <!-- (emType EM121A = 0, EM1000 =
    <Destination matchFolder="no" importSVP="no" linkSVP="no">
        <Folder>
            \All
        </Folder>
    </Destination>
    <Source>
        <FileList>
            {AllFile}
        </FileList>
    </Source>
    <Filters>
        <!-- Note when the setDeletedFlag is set to yes, the beams below the threshold are marked as deleted/rejected. When the quality threshold flag is set in the HeaderEditor, the beams are not marked as deleted/rejected. Beams marked as deleted/rejected can be manually accepted in NaviModel -->
        <MBEQuality setDeletedFlag="yes" qualityThreshold="0"/>
    </Filters>
    <Options disableBathy="no" useOnlineSV="no" onlineSVStart="-1.1" onlineSVEnd="15.5" scanRe
</EmInterpreter>

<!-- filetype="sbd" -->
<SbdInterpreter version="1.0">
    <Destination importSVP="no" linkSVP="no" linkGPSTide="no">
        <Folder>
            \Multibeam
        </Folder>
    </Destination>
    <Source>
        <FileList>
            {SBDFile}
        </FileList>
    </Source>
    <Filters>
        <!-- Note when the setDeletedFlag is set to yes, the beams below the threshold are marked as deleted/rejected. When the quality threshold flag is set in the HeaderEditor, the beams are not marked as deleted/rejected. -->
    </Filters>
</SbdInterpreter>

```

```
    Beams marked as deleted/rejected can be manually accepted in NaviModel -->
    <MBEQuality setDeletedFlag="yes" qualityThreshold="0"/>
</Filters>
<Options disableBathy="no" useOnlineSV="no" onlineSVStart="-1.1" onlineSVEnd="15.5" scanRe
</SbdInterpreter>

<!-- filetype="las" -->
<LasInterpreter version="1.0">
    <Destination matchFolder="no" importSVP="no" linkSVP="no">
        <Folder>
            \LAS
        </Folder>
    </Destination>
    <Source>
        <FileList>
            {LasFile}
        </FileList>
    </Source>
    <Filters>
        <!-- Note when the setDeletedFlag is set to yes, the beams below the threshold are mark
        When the quality threshold flag is set in the HeaderEditor, the beams are not marked as
        Beams marked as deleted/rejected can be manually accepted in NaviModel -->
        <MBEQuality setDeletedFlag="yes" qualityThreshold="0"/>
    </Filters>
    <Options disableBathy="no" useOnlineSV="no" onlineSVStart="-1.1" onlineSVEnd="15.5" scanRe
</LasInterpreter>

<!-- filetype="s7k" -->
<ResonS7KInterpreter version="1.0">
    <Destination matchFolder="no" importSVP="no" linkSVP="no">
        <Folder>
            \S7K
        </Folder>
    </Destination>
    <Source>
        <FileList>
            {S7KFile}
        </FileList>
    </Source>
    <Filters>
        <!-- Note when the setDeletedFlag is set to yes, the beams below the threshold are mark
        When the quality threshold flag is set in the HeaderEditor, the beams are not marked as
        Beams marked as deleted/rejected can be manually accepted in NaviModel -->
        <MBEQuality setDeletedFlag="yes" qualityThreshold="0"/>
    </Filters>
    <Options disableBathy="no" useOnlineSV="no" onlineSVStart="-1.1" onlineSVEnd="15.5" scanRe
</ResonS7KInterpreter>

<!-- filetype="gsf" -->
<GsfInterpreter version="1.0">
    <Destination matchFolder="no" importSVP="no" linkSVP="no">
        <Folder>
            \GSF
        </Folder>
    </Destination>
    <Source>
```

```
<FileList>
{GSFFfile}
</FileList>
</Source>
<Filters>
<!-- Note when the setDeletedFlag is set to yes, the beams below the threshold are marked as deleted/rejected. When the quality threshold flag is set in the HeaderEditor, the beams are not marked as deleted/rejected. Beams marked as deleted/rejected can be manually accepted in NaviModel --&gt;
&lt;MBEQuality setDeletedFlag="yes" qualityThreshold="0"/&gt;
&lt;/Filters&gt;
&lt;Options disableBathy="no" useOnlineSV="no" onlineSVStart="-1.1" onlineSVEnd="15.5" scanRe
&lt;/GsfInterpreter&gt;

&lt;!-- filetype="jsf" --&gt;
&lt;JsfInterpreter version="1.0"&gt;
&lt;Destination matchFolder="no" importSVP="no" linkSVP="no"&gt;
&lt;Folder&gt;
\GSF
&lt;/Folder&gt;
&lt;/Destination&gt;
&lt;Source&gt;
&lt;FileList&gt;
c:\MyFiles\MySurveyFile.jsf
&lt;/FileList&gt;
&lt;/Source&gt;
&lt;Filters&gt;
<!-- Note when the setDeletedFlag is set to yes, the beams below the threshold are marked as deleted/rejected. When the quality threshold flag is set in the HeaderEditor, the beams are not marked as deleted/rejected. Beams marked as deleted/rejected can be manually accepted in NaviModel --&gt;
&lt;MBEQuality setDeletedFlag="yes" qualityThreshold="0"/&gt;
&lt;/Filters&gt;
&lt;Options disableBathy="no" useOnlineSV="no" onlineSVStart="-1.1" onlineSVEnd="15.5" scanRe
&lt;/JsfInterpreter&gt;

&lt;!-- filetype="Imagenex" --&gt;
&lt;ImagenexInterpreter version="1.0"&gt;
&lt;Destination matchFolder="no" importSVP="no" linkSVP="no"&gt;
&lt;Folder&gt;
\GSF
&lt;/Folder&gt;
&lt;/Destination&gt;
&lt;Source&gt;
&lt;FileList&gt;
c:\MyFiles\MySurveyFile.83p
&lt;/FileList&gt;
&lt;/Source&gt;
&lt;Filters&gt;
<!-- Note when the setDeletedFlag is set to yes, the beams below the threshold are marked as deleted/rejected. When the quality threshold flag is set in the HeaderEditor, the beams are not marked as deleted/rejected. Beams marked as deleted/rejected can be manually accepted in NaviModel --&gt;
&lt;MBEQuality setDeletedFlag="yes" qualityThreshold="0"/&gt;
&lt;/Filters&gt;
&lt;Options disableBathy="no" useOnlineSV="no" onlineSVStart="-1.1" onlineSVEnd="15.5" scanRe
&lt;/ImagenexInterpreter&gt;
&lt;/NaviEditTask&gt;</pre>
```

```
</MultibeamImportTask>
```

### 3.45 NaviEdit Util Task

```
<!-- NaviEdit Util Task -->
<NaviEditUtilTask po="" name=" NaviEdit Util Task" Output="" logKeyword="{AllFile.Filename}">
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}"/>
        <Input xml="{DatabaseName}"/>
        <GapHandling version="1.0" MaxSecPreviousBlockEnd="10" MaxSecPreviousBlockStart="600">
            <BlockIds>
                {AllBlockID}
            </BlockIds>
        </GapHandling>>
    </NaviEditTask>
</NaviEditUtilTask>
```

### 3.46 NaviPac Interpreter

Description:

```
<!-- NaviPac Interpreter -->
<NaviPacInterpreterTask po="" name="" logKeyword="" output="">
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}"/>
        <Input xml="{DatabaseName}"/>
        <Input xml="{Geodesy}"/>
        <NaviPacInterpreter version="1.0" useNpdGeodesy="yes">
            <Destination>
                <Folder>
                    \Data
                </Folder>
            </Destination>
            <Source>
                <FileList>c:\Data\20200731_183421_S.NPD</FileList>
            </Source>
            <GPSTide calculate="yes">
                <GeoidHeight geoidSelection="1">
                    <!-- 0 = Fixed height,
                        1 = Geoid file -->
                    <FixedHeight additionalHeight="0.0" height="57.0"/>
                    <GeoidFile>c:\EIVA\NaviPac\Setup\Geoide.bin</GeoidFile>
                </GeoidHeight>
                <SpeedCorrection correctionType="0">
                    <!-- 0 = No correction,
                        1 = Corr function,
                        2 = Corr file
                    -->
                    <CorrectionFunction v0="3.3" v1="2.2" v2="1.1"/>
                    <CorrectionFile>
                        c:\Data\NaviEdit\10_Gpstide\squat.txt
                    </CorrectionFile>
                </SpeedCorrection>
            </GPSTide>
        </NaviPacInterpreter>
    </NaviEditTask>
</NaviPacInterpreterTask>
```

```

        </CorrectionFile>
    </SpeedCorrection>
    <DynamicDraft apply="no" scaleToMeter="1.0" singleBeamSeq="2" />
    <!-- 2 = singleBeamSeq = 2 TODO -->
</GPSTide>
</NaviPacInterpreter>
</NaviEditTask>
</NaviPacInterpreterTask>
```

## 3.47 Navlab Import

Deprecated. Use the MergeFileTask instead.

Used to merge Navlab data into an existing survey block like found in the JobPlanner, Tools, Merge Data Files menu.

## 3.48 Pipetracker Merge

Description:

```

<!-- Pipetracker Merge -->
<PipetrackerMergeTask po="" name="Merge Deep Learning PipeTracker"
logKeyword="{AllFile.Filename}">
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}" />
        <Input xml="{DatabaseName}" />
        <Input xml="{Geodesy}" />
            <CombinePipetrackers version="1.0">
                <BlockIds>3</BlockIds> <!-- Multiple allowed -->
            </CombinePipetrackers>
        </NaviEditTask>
    </PipetrackerMergeTask>
```

## 3.49 Recalc Bathy

Description:

```

<!-- Batch Job Pressure to Depth -->
<RecalcBathyTask po="" name="Batch Pressure to Depth"
logKeyword="{SBDFile.Filename}" output="" OnError="">
    <NaviEditTask version="1.0">
        <SqlServer>{SqlServer}</SqlServer> <!-- Could also be "{SqlServer}" -->
        <DatabaseName>{DatabaseName}</DatabaseName> <!-- Could also be "DatabaseName" -->
            <BatchProcess>
                <BlockId>{SBDBlockId}</BlockId>
            <!-- Multiple allowed, could also be {SBDBlockId}-->
```

```

<batchjob>
  <edititem target="Bathy[Active]" action="Recalculate Bathy
(ocean)">
    <sensortype>4</sensortype>
    <actionid>34002</actionid>
    <!-- recalc bathy task id -->
    <selection>2</selection>
    <!-- 0 = ACTIVE_SENSOR,
        1 = ALL_SENSORS_OR_CHANNEL_OF_TIS_TYPE,
        2 = THIS_SENSOR_INDEX -->
    <index>2</index>
    <recalcbathyocean>
      <gravity>9.80173664581538</gravity>
      <surfpres>1.01325</surfpres>
      <waterdensity>0</waterdensity>
      <calctype>3</calctype>
      <!-- 0 = SIMPLE (DEPTH CORRECTION),
          1 = SIMPLE (CONST DENSITY),
          2 = SAUNDERS AND FOFONOFF,
          3 = UNESCO (REQ. CTD) -->
      <selectpressure>0</selectpressure>
      <!-- 0 = FixedPressure,
          1 = FromAPressureBlock,
          2 = ABathySeqFromThisBlock -->
      <refpressure>0</refpressure>
      <pressureid>-1</pressureid>
      <surfacePressureSeq>0</surfacePressureSeq>
    </recalcbathyocean>
  </edititem>
</batchjob>
</BatchProcess>
</NaviEditTask>
</RecalcBathyTask>

```

## 3.50 Recalc KP

Description: This is an example recalculation of KP values via the BatchJob functionality.

```

<!-- Recalc KP -->
<RecalcKPTask po=" " name="Recalculate KP" logKeyword=" " output=" "
OnError=" ">

```

```

<NaviEditTask version="1.0">
<Input xml="{SqlServer}"/>
<Input xml="{DatabaseName}"/>
<Input xml="{Geodesy}"/>
  <BatchProcess>
    <BlockId>5</BlockId> <!-- Multiple allowed-->
    <batchjob>
      <edititem action="Recalculate KP" target="Navigation[Active]">
        <sensortype>9</sensortype>
        <actionid>34012</actionid>
        <selection>0</selection>
        <index>0</index>
      </edititem>
    </batchjob>
  </BatchProcess>
</NaviEditTask>
</RecalcKPTask>

```

## 3.51 Reson S7K Import

```

<!-- Reson S7K Import -->
<ResonImportTask po="" name="" output="" logKeyword="{SBDFile.Filename}">
  <ResonS7KInterpreter version="1.0">
    <Destination matchFolder="no" importSVP="no" linkSVP="no">
      <Folder>
        \S7K
      </Folder>
    </Destination>
    <Source>
      <FileList>
        d:\Data\Oceaneering\Freedom_S7K_LAS_SprintNAV\Freedom FreeSpan\20210713-140923.615_Freesp
      </FileList>
    </Source>
    <Filters>
      <!-- Note when the setDeletedFlag is set to yes, the beams below the threshold are marked as deleted/rejected. When the quality threshold flag is set in the HeaderEditor, the beams are not marked as deleted/rejected. Beams marked as deleted/rejected can be manually accepted in NaviModel -->
      <MBEQuality setDeletedFlag="yes" qualityThreshold="0"/>
    </Filters>
    <Options disableBathy="no" useOnlineSV="no" onlineSVStart="-1.1" onlineSVEnd="15.5" scanRedu
  </ResonS7KInterpreter>
</ResonImportTask>

```

## 3.52 Recalculation of GPS Height

Description: BatchJob example for Recalculation of GPS Height.

```

<BatchSmoothTask po="" name="" logKeyword="">
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}"/>
    <Input xml="{DatabaseName}"/>
    <Input xml="{Geodesy}"/>

```

```

<BatchProcess>
    <BlockId>{AllBlockId}</BlockId> <!-- Multiple allowed-->
        <batchjob>
            <edititem action="Recalc Bathy or GPSTide (singlebeam) by GPSHeight"
target="Bathy[Active]">
                <sensortype>4</sensortype>
                <!-- 4: Bathy-->
                <actionid>34001</actionid>
                <!-- 34001: RecalcBathy/GPSTide -->
                <selection>0</selection>
                <index>0</index>
                <recalcbathygpsheight>
                    <bGpsTide>false</bGpsTide>
                    <SpeedCorr>0</SpeedCorr>
                    <SelectGeoid>1</SelectGeoid>
                    <nTimeOffset>0</nTimeOffset>
                    <fGeoidHeight>40.5</fGeoidHeight>
                    <fAdditionalGeoidHeight>0</fAdditionalGeoidHeight>
                    <fSC2>0</fSC2>
                    <fSC1>0</fSC1>
                    <fSC0>0</fSC0>
                    <sFileName>D:\VORF_ETRFtoCD_HI1561.gbin</sFileName>
                    <!-- or {GeoidFilePath} -->
                    <sSpeedCorrFile>[No file selected]</sSpeedCorrFile>
                    <m_bUseDynDraft>false</m_bUseDynDraft>
                    <m_DynDraftSeq>-1</m_DynDraftSeq>
                    <m_fDynDraftScale>0.0</m_fDynDraftScale>
                </recalcbathygpsheight>
            </edititem>
        </batchjob>
    </BatchProcess>
    <NaviEditTask>
        </BatchSmoothTask>
    </NaviEditTask>
</BatchSmoothTask>

```

### 3.53 SBD Import

Description:

```

<!-- SBD import -->
<SbdImportTask po="" name="Import in NE" output="SBDBlockId"
logKeyword="{SBDFile.Filename}">
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}" />
        <Input xml="{DatabaseName}" />
        <Geodesy useSbdGeodesy="no">
            <Source>
                <Ellipsoid epsg="7030" name="WGS 84">
                    <!-- eiva does not use the ellipsoid epss number, but indirectly by
use of the projection epss number-->
                    <SemiMajorAxis>6378137.0</SemiMajorAxis>
                    <InverseFlattening>298.257223563</InverseFlattening>
                </Ellipsoid>
                <Projection epsg="32632" name="UTM (north)">
                    <!-- epss 32632 requires/implies wgs 84 ellipsoid -->
                    <EivaType>5</EivaType>

```

```

<!-- integer PROJPAC.h UTM N = 5 -->
<FalseEasting>500000</FalseEasting>
<FalseNorthing>0</FalseNorthing>
<FirstParallel>0</FirstParallel>
<SecondParallel></SecondParallel>
<Scale>.9996</Scale>
<OrigLatitude>0</OrigLatitude>
<OrigLongitude>9</OrigLongitude>
<Zone>32</Zone>
</Projection>
</Source>
<Destination>
    <Ellipsoid epsg="7030" name="WGS 84">
        <!-- eiva does not use the ellipsoid epsgs number, but indirectly by
use of the projection epsgs number-->
        <SemiMajorAxis>6378137.0</SemiMajorAxis>
        <InverseFlattening>298.257223563</InverseFlattening>
    </Ellipsoid>
    <Projection epsg="32632" name="UTM (north)">
        <!-- epsg 32632 requires/implies wgs 84 ellipsoid -->
        <EivaType>5</EivaType> <!-- integer PROJPAC.h UTM N = 5 -->
        <FalseEasting>500000</FalseEasting>
        <FalseNorthing>0</FalseNorthing>
        <FirstParallel>0</FirstParallel>
        <SecondParallel></SecondParallel>
        <Scale>0.9996</Scale>
        <OrigLatitude>0</OrigLatitude>
        <OrigLongitude>9</OrigLongitude>
        <Zone>32</Zone>
    </Projection>
</Destination>
<Source></Source>
<Shift type="1" method="2" name="No shift">
    <!-- Type: 0: No Shift, 1: Normal (BW), 2: North Sea (NO) --> <!--
Method: 0: No Shift, 1 User Defined, 2-21 as found in -->
    <Translation X="90.36" Y="101.13" Z="123.38" /> <!-- unit is meter-->
    <Rotation X="-0.333" Y="-0.077" Z="-0.894" /> <!-- unit is arc
seconds-->
    <ScaleFactor ppm="-1.994" />
</Shift>
</Geodesy>
<SbdInterpreter version="1.0">
    <Destination matchFolder="yes" importSVP="no" linkSVP="no">
        <Folder>
            \EIVADays\2019
        </Folder>
    </Destination>
<Source>
    <FileList>
        d:\WFM\Kuda\WatchDir\140917N000.SBD
    </FileList>
</Source>
<Filters>
    <!-- Note when the setDeletedFlag is set to yes, the beams below the
threshold are marked as deleted/rejected.

```

When the quality threshold flag is set in the HeaderEditor, the beams are not marked as deleted, but marked as filtered in NaviModel.

Beams marked as deleted/rejected can be manually accepted in NaviModel -->

```

<MBEQuality setDeletedFlag="yes" qualityThreshold="12"/>
</Filters>
<Options disableBathy="no" useOnlineSV="no" onlineSVStart="-1.1"
onlineSEnd="15.5" scanReductionImportEvery="1"
skipScansWithoutMotion="true"/>
</SbdInterpreter>
</NaviEditTask>
</SbdImportTask>

```

## 3.54 SBD Import task

Description: SBD Import with extended properties i.e. rather than relying on the parameters last used in the NaviEdit Interpreter wizard.

It is not possible at this point to use the extended properties in the SBD import, only Geodesy.

```

<!-- SBD import task -->
<SbdImportTask po=" " name=Import in NE output="SBDBlockId"
logKeyword="{SBDFile.Filename}">
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}"/>
        <Input xml="{DatabaseName}"/>
        <Geodesy useSbdGeodesy="yes">
        </Geodesy>
        <SbdInterpreter version="1.0">
            <Destination importSVP="no" importSVP="no" linkSVP="no"
importGPSTide="no" linkGPSTide="no">
                <Folder>{DateName}</Folder>
            </Destination>
            <Source>
                <FileList>{SBDFile}</FileList>
            </Source>
            <Filters>
                <!-- Note when the setDeletedFlag is set to yes, the beams below the threshold are marked as deleted/rejected can be manually accepted in NaviModel -->
                <MBEQuality setDeletedFlag="yes" qualityThreshold="12"/>
            </Filters>
            <Options disableBathy="no" useOnlineSV="no" onlineSVStart="-1.1"
onlineSEnd="15.5" scanReductionImportEvery="1"
skipScansWithoutMotion="true"/>
        </SbdInterpreter>
    </NaviEditTask>
</SbdImportTask>

```

## 3.55 Set Geodesy

Description:

```

<SetGeodesyTask po="" name="" logKeyword="" output="" />
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />
    <SetGeodesy>
      <BlockId>53</BlockId>
      <!-- Multiple allowed, use space as separator -->
    </SetGeodesy>
    <Geodesy useSbdGeodesy="no">
      <Source>
        <Ellipsoid epsg="7019" name="GRS 80">
          <!-- epsg number is not in use -->
          <SemiMajorAxis>6378137.0</SemiMajorAxis>
          <InverseFlattening>298.257222101</InverseFlattening>
        </Ellipsoid>
        <Projection epsg="25833" name="UTM (north)">
          <!-- epsg number is not in use -->
          <!-- EivaType 0: Transverse Mercator,
              1: Mercator,
              2: Polar Stereographic,
              3: Equatorial Stereographic,
              4: Oblique Stereographic,
              5: UTM N,
              6: UTM S,
              7: Gauss Krueger,
              8: UPS N, 9: UPS S,
              10: RD_STEREOGRAPHIC (National grid in Holland),
              11: Lamber's conical (Two parallels),
              12: Lamber's conical (One parallel),
              13: Rectified Skew Orthomorphic,
              14: New Zealand Map Grid,
              15: American Polyconic
            -->
          <EivaType>5</EivaType>
          <FalseEasting>500000</FalseEasting>
          <FalseNorthing>0</FalseNorthing>
          <FirstParallel>0</FirstParallel>
          <SecondParallel></SecondParallel>
          <Scale>0.9996</Scale>
          <OrigLatitude>0</OrigLatitude>
          <OrigLongitude>15</OrigLongitude>
          <Zone>33</Zone>
        </Projection>
      </Source>
      <Destination>
        <Ellipsoid epsg="7019" name="GRS 80">
          <!-- epsg number is not in use -->
          <SemiMajorAxis>6378137.0</SemiMajorAxis>
          <InverseFlattening>298.257222101</InverseFlattening>
        </Ellipsoid>
        <Projection epsg="25833" name="UTM (north)">
      </Destination>
    </Geodesy>
  </NaviEditTask>

```

```

<!-- epsg number is not in use -->
<EivaType>5</EivaType>
<FalseEasting>500000</FalseEasting>
<FalseNorthing>0</FalseNorthing>
<FirstParallel>0</FirstParallel>
<SecondParallel></SecondParallel>
<Scale>0.9996</Scale>
<OrigLatitude>0</OrigLatitude>
<OrigLongitude>15</OrigLongitude>
<Zone>33</Zone>
</Projection>
</Destination>
<Shift method="1" name="User Defined Datum Shift">
<!-- method: use '1'
      name: Only use the names specified in c:\EIVA\NaviEdit\Setup\datumshift.cf like 'None',
            'User Defined Datum Shift', 'WGS84 to ED50 (NGO method)' ... -->
<Translation x="1.1" y="2.2" z="3.3" />
<!-- In Meters -->
<Rotation x="0.44" y="0.55" z="0.66" />
<!-- In Arc Seconds -->
<ScaleFactor>42</ScaleFactor>
<!-- In ppm -->
</Shift>
</Geodesy>
</NaviEditTask>
</SetGeodesyTask>

```

## 3.56 More on the Set geodesy task

Description:

To understand the source and destination entries.

Do I have to define both entries both when setting geodesy and changing geodesy?

In our .all flow we have a set geodesy entry with both source and destination, and in the sbd xml file example.

For the case of having one geodesy set up in naviscan and it was wrong so I want to set a new projection in the import step, how should I set up the xml then?

If you look at the top <Geodesy> element it is reused later on the XML file, just like the <SqlServer> and <DatabaseName> elements.

```

<Setup StateDatabaseLocation="d:\WFM\TaskStateDatabase.db">
  <Log TimingLog="c:\Eiva\Logs\Timing.csv"/>
  <GroupTask name="Workflow">
    <SqlServer>testpc2</SqlServer>
    <DatabaseName>WFM</DatabaseName>
    <Geodesy>
      <Destination>
        <Ellipsoid epsg="7019" name="GRS 1980">

```

When the workflow XML file is parsed by the Workflow Manager all “WFM variables” with “Input xml=” followed by curly brackets (named a Tuborg sign in Danish ☐) are replaced with the elements in the top of the XML file.

```
<AllImportTask po="2" name="Import in NE" Output="AllBlockId"
logKeyword="{AllFile.Filename}">
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />
    <Input xml="{Geodesy}" />
    <EmInterpreter version="1.0" emType="3">
      <!-- emType EM121A = 0, EM1000 = 1, EM3000HUGIN2001 = 2, EMSERIES = 3,
KMALL = 4) -->
      <Destination matchFolder="yes" importSVP="yes" linkSVP="yes">
        <Folder>\LineName\SubFolder</Folder>
```

This implies you only need to specify the Geodesy once in a XML file. All other places you should be able to reuse it by just specifying <Input xml="{Geodesy}" />  
In other words the following:

```
<SetGeodesyTask po="4" name="Change Geodesy" logKeyword="{SBDFFile.Filename}">
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />
    <SetGeodesy>
      <BlockId>
        {SBDBlockId}
      </BlockId>
      <!-- Multiple allowed, use space as separator -->
    </SetGeodesy>
    <Geodesy useSbdGeodesy="no">
      <Source>
        <Ellipsoid epsg="7019" name="GRS 80">
          <!-- epsg number is not in use -->
          <SemiMajorAxis>6378137.0</SemiMajorAxis>
          <InverseFlattening>298.257222101</InverseFlattening>
        </Ellipsoid>
        ...
      </Source>
    </Geodesy>
  </NaviEditTask>
</SetGeodesyTask>
```

Could simply be replaced by

```
<SetGeodesyTask po="4" name="Change Geodesy" logKeyword="{SBDFFile.Filename}">
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />
    <SetGeodesy>
      <BlockId>
        {SBDBlockId}
      </BlockId>
      <!-- Multiple allowed, use space as separator -->
    </SetGeodesy>
    <Input xml="{Geodesy}" />
  </NaviEditTask>
</SetGeodesyTask>
```

In general I would recommend always to specify both source and destination even though the source and destination are equal. (Right now the SBD importer and the Kongsberg all importer are only reading the destination projection)

If you have a wrong projection only in NaviScan, you could specify the right in the SbdImportTask. If you need to do a datum shift, you will need to use a SetGeodesy task. (If you specify a full Geodesy including datum shift in the SbdImportTask, the datum shift will – as far as I remember – just be ignored)

## 3.57 Set Block Quality Threshold

Description:

```
<!-- Set block quality threshold -->
<BlockSettingsTask po="" name="Move block" logKeyword="{AllFile.Filename}">
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />
    <BlockSettings version="1.0">
      <BlockId>158 197</BlockId>
      <QualityThreshold quality="6"/>
    </BlockSettings>
  </NaviEditTask>
</BlockSettingsTask>
```

## 3.58 SetMounting

Example:

```
<BlockSettingsTask po="" name="Move block" logKeyword="{AllFile.Filename}">
  <NaviEditTask version="1.0">
    <SqlServer>localhost</SqlServer>
    <DatabaseName>WFM</DatabaseName>
    <SetMounting>
      <BlockId>2639</BlockId>
      <!-- type:
        0 Absolute
        1 Bathymetry
        2 Motion
        3 Pipetracker
        4 Doppler Log
        5 Gyro
        6 Position
        7 Remote position
        8 Waterline
        9 Gps Antenna
        10 User defined offset
        11 Auxiliary
        12 Relative Position
      -->
```

```

15 Laser
16 Echosounder
17 Echosounder Rx
18 Sidescan
20 Combined position
256 Single data
-->

<!-- Position -->
<Mounting type="6" seq="0">
    <!-- In meters -->
    <Offset x="1.1" y="2.2" z="3.3" />
</Mounting>

<!-- Several Mountings can be specified-->
<!-- Echo sounder-->
<Mounting type="16" seq="0">
    <!-- In meters -->
    <Offset x="3.3" y="2.2" z="1.1" />
    <!-- In degrees -->
    <Angle roll="-21.1" pitch="0.123" heading="0.111"/>
    <Multibeam isRollCompensated="no" isPitchCompensated="no" />
</Mounting>

<!-- Single beam, example with sequence 2 -->
<Mounting type="256" seq="2">
    <!-- In meters -->
    <Offset x="0.11" y="0.22" z="3.33" />
    <!-- In degrees -->
    <Angle roll="0.0" pitch="0.0" heading="0.0"/>
    <Singlebeam useRawSingleData="no" />
</Mounting>

</SetMounting>
</NaviEditTask>
</BlockSettingsTask>

```

## 3.59 Set window

Description:

```

<BlockSettingsTask po="" name="Move block" logKeyword="{AllFile.Filename}">
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}" />
        <Input xml="{DatabaseName}" />
        <BlockSettings version="1.0">
            <BlockId>9707</BlockId>
            <Window> <!-- All units in meters -->
                <Easting max="2147483647" min="-2147483648"/>
                <Northing max="2147483647" min="-2147483648"/>
                <DepthRelRefPoint max="20000" min="-20000"/>
                <XDistRelRefPoint max="500" min="-500"/>
            </Window>
        </BlockSettings>
    </NaviEditTask>
</BlockSettingsTask>

```

```

    </NaviEditTask>
</BlockSettingsTask>
```

## 3.60 Smooth Surface Pressure

Description: This function allows you to smooth surface pressure data via the BatchJob functionality.

```

<BatchSmoothTask po="" name="Smooth tide" logKeyword="" >
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}"/>
    <Input xml="{DatabaseName}"/>
    <BatchProcess>
      <BlockId>5815</BlockId>
      <!-- Multiple allowed, but smoothed as single blocks -->
      <batchjob>
        <edititem target="Surface Pressure[Active]" action="Smooth points">
          <sensortype>22</sensortype>
          <actionid>40100</actionid>
          <selection>0</selection>
          <index>0</index>
          <smooth>
            <wavelength>3600</wavelength> <!-- seconds -->
          </smooth>
        </edititem>
      </batchjob>
    </BatchProcess>
  </NaviEditTask>
</BatchSmoothTask>
```

## 3.61 Smooth Tide

Description: This function allows you to smooth tide data via the BatchJob functionality.

```

<BatchSmoothTask po="" name="Smooth tide" logKeyword="" >
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}"/>
    <Input xml="{DatabaseName}"/>
    <BatchProcess>
      <BlockId>5814</BlockId>
      <!-- Multiple allowed, but smoothed as single blocks -->
      <batchjob>
        <edititem target="Tide[Active]" action="Smooth points">
          <sensortype>15</sensortype>
          <actionid>40100</actionid> <!-- TODO -->
          <selection>0</selection>
          <index>0</index>
          <smooth>
            <wavelength>3600</wavelength> <!-- seconds -->
          </smooth>
        </edititem>
      </batchjob>
</BatchSmoothTask>
```

```

    </BatchProcess>
    </NaviEditTask>
</BatchSmoothTask>
```

1b)

Note the batchjob-element is exactly as a saved batch job, so if you want to change any of the parameters try to make a batch job and save it.

```

<BatchSmoothTask po="2" name="Smooth Tide">
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}" />
        <Input xml="{DatabaseName}" />
        <BatchProcess>
            <BlockId><!-- Multiple allowed, but smoothed as single blocks -->
                {TideBlocks}
            </BlockId>
            <batchjob>
                <edititem action="Smooth points" target="Tide[Active]">
                    <sensortype>15</sensortype><!-- Tide -->
                    <actionid>40100</actionid><!-- Smooth -->
                    <selection>0</selection><!-- Active = 0, All = 1, Index = 2 -->
                    <index>0</index>
                    <smooth>
                        <wavelength>3600</wavelength><!-- seconds -->
                    </smooth>
                </edititem>
            </batchjob>
        </BatchProcess>
    </NaviEditTask>
</BatchSmoothTask>
```

## 3.62 Turbidity Merge

Description: Merge Turbidity Ascii file.

```

<!-- Merge Turbidity -->
<TurbidityMergeTask po="" name="" logKeyword="" output="">
    <NaviEditTask version="1.0">
        <Input xml="{SqlServer}" />
        <Input xml="{DatabaseName}" />
        <Input xml="{Geodesy}" />
        <Merge version="1.0">
            <Destination>
                <BlockIds>{AllBlockId}</BlockIds>
            </Destination>
            <Source addNew="yes" name="Turbidity" inputType="21">
                <!-- inputType 21 = Turbidity -->
                <Turbidity>
                    <Mounting z="41" y="43" x="42"/>
                    <FileList>\{previousFile.Filename}.turbidity</FileList>
                </Turbidity>
            </Source>
        </Merge>
    </NaviEditTask>
```

```
<TurbidityMergeTask>
```

### 3.63 Wait for previous NaviEdit block

Description:

```
<WaitForPreviousNaviEditBlockTask po="" name="Wait for previous file"
Output="PrevID" logKeyword="{AllFile.Filename}" timeoutInSeconds="300">
  <NaviEditTask version="1.0">
    <Input xml="{SqlServer}" />
    <Input xml="{DatabaseName}" />
    <DoesPreviousBlockExist maxSecondsPreviousBlockEnd="10"
maxSecondsPreviousBlockStart="400">
      <BlockId>{AllBlockId}</BlockId>
    </DoesPreviousBlockExist>
  </NaviEditTask>
</WaitForPreviousNaviEditBlockTask>
```

### 3.64 Yaw Stabilization Warning

```
<!-- Yaw Stabilization Warning -->
<BlockSettingsTask po="" name="Move block" logKeyword="{AllFile.Filename}" OnError="">
  <NaviEditTask version="1.0">
    <SqlServer>testpc2</SqlServer>
    <DatabaseName>WFM</DatabaseName>
    <AbortIf yawStabilizationActive="yes">
      <BlockId>19</BlockId>
    </AbortIf>
  </NaviEditTask>
</BlockSettingsTask>
```

## 4 NaviModel Tasks

### 4.1 Automatic Flagging

```
<!-- Automatic Flagging -->
<NMCLITask po="" name="AutomaticFlags" logKeyword="AutomaticFlags" output="" >
  <Command name="AutomaticFlags">
    <input>
      <dbfiles>
        \\eiva.local\data\Datasets\Internal_use_only\_TestData\NaviModel\cli\automaticflags
      </dbfiles>
      <Runline>
        \\eiva.local\data\Datasets\Internal_use_only\_TestData\NaviModel\cli\automaticflags
      </Runline>
      <DigitizedLines>
        \\eiva.local\data\Datasets\Internal_use_only\_TestData\NaviModel\cli\automaticflags
      </DigitizedLines>
      <Pipe>
        <Step>0.002 kp</Step>
        <Diameter>1.1 m</Diameter>
      </Pipe>
      <Markers>
        -8 -0.5 0.5 8
      </Markers>
    </input>
    <Output>
      <FlagFilePath>c:\temp\ddd.eff</FlagFilePath>
    </Output>
  </Command>
</NMCLITask>
```

### 4.2 Camera Eventing

Description:

```
<!-- Camera Eventing -->
<NMCLITask po="" name="Camera Eventing" logKeyword="" output="" >
  <Command name="CameraEventing">
    <Event collection="" />
    <Export path="C:\Data\cameraEventing_NMCLI"/>
    <Server port="443" ip="10.6.0.63"/>
    <Model name="Pipe Inspection Imagery"/>
    <Track path="c:\Data\DATA_20170803095456248\20170803095456248@current.csv" template="Visuals" />
    <Video path="c:\Data\DATA_20170803095456248"/>
  </Command>
</NMCLITask>
```

## 4.3 Cell Stdev

Description: NaviModel task.

```
<!-- Cell Stdev -->
<NMCLITask po="" name="Cell standard deviation" logKeyword="" output="" >
  <Command name="CellStdev">
    <Input StandardDeviationLimit_m="0.1" RequiredPercentageOfCells_pct="90" dbfile="c:\Data\re>
```

## 4.4 Compare Event Collections

Description:

```
<!-- Compare Event Collections Settings -->
<NMCLITask po="" name="Compare Event Collections Settings" logKeyword="" >
  <Command name="CompareEventCollectionsSettings">
    <EventDistanceMetric>XYZ</EventDistanceMetric>
    <EventDistanceThreshold>10</EventDistanceThreshold>
    <EventCollectionInput1Path>c:\Data\first.xml</EventCollectionInput1Path>
    <EventCollectionInput2Path>c:\Data\second.xml</EventCollectionInput2Path>
    <EventCollectionOutputPath>c:\Data\compared.xml</EventCollectionOutputPath>
    <EventCollectionOutputName>compared</EventCollectionOutputName>
    <EventTypeMatchByName>true</EventTypeMatchByName>
    <EventTypeMatchByNameSimilar>true</EventTypeMatchByNameSimilar>
    <EventTypeMatchByEvents>true</EventTypeMatchByEvents>
    <EventTypeMatchByRecursiveEvents>true</EventTypeMatchByRecursiveEvents>
    <EventTypeMatchHierarchyPreference>PreferHierarchy</EventTypeMatchHierarchyPreference>
  </Command>
</NMCLITask>
```

## 4.5 Corridor Coverage

Description:

Syntax: nmcli - command.

There are two ways to provide parameters to corridor coverage now. The older version of the xml still works. The new version now has a **revision** tag in it, so when the corridor coverage command is running, it will recognize which one. If there is no <revision> tag, it will assume that this is the old format of the xml. Also the command generates an event file that you can drop back into NaviModel.

Older version:

```
<NMCLITask po="" name="CorridorCoverage" logKeyword="CorridorCoverage" output="" >
  <Command name="CorridorCoverage">
    <dbfiles>
      \\eiva.local\data\DataSet\Internal_use_only\_TestData\NaviModel\cli\corridorcoverage\c
      \\eiva.local\data\DataSet\Internal_use_only\_TestData\NaviModel\cli\corridorcoverage\c
    </dbfiles>
    <runline>
      <File>\\eiva.local\data\DataSet\Internal_use_only\_TestData\NaviModel\cli\corridorcov
```

```

        <TrimToData>yes</TrimToData> <!-- Cut the line so it as has the same lenght as the dtm
    </runline>
    <CorridorWidth>20 m</CorridorWidth>
    <ErrorIf>
        <gap_longer_than>10 m</gap_longer_than>
        <with_coverage_less_than_pct>80</with_coverage_less_than_pct>
    </ErrorIf>
</Command>
</NMCLITask>
```

Newer version:

```

<NMCLITask po="" name="CorridorCoverage" logKeyword="CorridorCoverage" output="" >
    <Command name="CorridorCoverage">
        <revision>1</revision>
        <dbfiles>
            \\eiva.local\data\DataSet\Internal_use_only\_TestData\NaviModel\cli\corridorcoverage\c
            \\eiva.local\data\DataSet\Internal_use_only\_TestData\NaviModel\cli\corridorcoverage\c
        </dbfiles>
        <runline>\\eiva.local\data\DataSet\Internal_use_only\_TestData\NaviModel\cli\corridorcoverage\c
        <track>\\eiva.local\data\DataSet\Internal_use_only\_TestData\NaviModel\cli\corridorcoverage\c
        <TrimToData>yes</TrimToData> <!-- Cut the line so it as has the same lenght as the dtm -->
        <CorridorWidth>20 m</CorridorWidth>
        <ErrorIf>
            <gap_longer_than>3 m</gap_longer_than>
            <with_coverage_less_than_pct>80</with_coverage_less_than_pct>
        </ErrorIf>
        <output>
            <eventsxmlfile>c:\temp\corridor_gaps.xml</eventsxmlfile>
        </output>
    </Command>
</NMCLITask>
```

## 4.6 Create DTM

```

<!-- Create DTM -->
<NMCLITask po="" name="CreateDTM" logKeyword="CreateDTM" output="" >
    <Command name="CreateDTM">
        <Files>
            \\
            \\eiva.local\data\DataSet\Internal_use_only\_TestData\NaviModel\cli\createddt
            m\file1.xyz
            <!-- \
            \\eiva.local\data\DataSet\Internal_use_only\_TestData\NaviModel\cli\createddt
            m\file2.xyz -->
        </Files>
        <Model>
            <CellSize>1.00 m</CellSize>
            <Path>c:\temp\WFM_Output\CreatedDTM.db</Path>
            <CreateMinimumSurface>yes</CreateMinimumSurface>
            <CreateMaximumSurface>yes</CreateMaximumSurface>
            <CreateMedianSurface>yes</CreateMedianSurface>
            <CreateStdSurface>yes</CreateStdSurface>
            <CreateRejectedCountSurface>yes</CreateRejectedCountSurface>
```

```
<CreateDeletedCountSurface>yes</CreateDeletedCountSurface>
<CreateFileCountSurface>yes</CreateFileCountSurface>
<CreateBackscatterSurface>yes</CreateBackscatterSurface>
<CreateQualitySurface>yes</CreateQualitySurface>
<OverwriteExistingFile>yes</OverwriteExistingFile>
<ReadIntensity>no</ReadIntensity>
<!-- <CellSizeIntensity>0.5 m</CellSizeIntensity> -->
</Model>
<DTMFilters>
  <Filters>
    <!-- Multiple allowed-->
    <DepthFilter>
      <MinThreshold_cm>100</MinThreshold_cm>
      <MaxThreshold_cm>100</MaxThreshold_cm>
    </DepthFilter>
    <RemoveEveryNthScan>
      <NthScan>2</NthScan>
    </RemoveEveryNthScan>
    <KeepOnlyEveryNthScan>
      <NthScan>2</NthScan>
    </KeepOnlyEveryNthScan>
    <RemoveEveryNthPoint>
      <NthPoint>2</NthPoint>
    </RemoveEveryNthPoint>
    <KeepOnlyEveryNthPoint>
      <NthPoint>2</NthPoint>
    </KeepOnlyEveryNthPoint>
    <MedianDepth>
      <MinThreshold_cm>0</MinThreshold_cm>
      <MaxThreshold_cm>100</MaxThreshold_cm>
    </MedianDepth>
    <AverageDepth>
      <MinThreshold_cm>0</MinThreshold_cm>
      <MaxThreshold_cm>100</MaxThreshold_cm>
    </AverageDepth>
    <MinMaxDepth>
      <MinThreshold_cm>-500</MinThreshold_cm>
      <MaxThreshold_cm>0</MaxThreshold_cm>
    </MinMaxDepth>
    <ScaleFilter>
      <EastingScale>1</EastingScale>
      <NorthingScale>1</NorthingScale>
      <DepthScale>1</DepthScale>
    </ScaleFilter>
    <FlipZFilter>
    </FlipZFilter>
  </Filters>
</DTMFilters>

<!--
  <Database>
    <SqlServer>JEK-XPS15</SqlServer>
    <DatabaseName>jesper2</DatabaseName>
    <BlockIds>
      2
    </BlockIds>
```

```

        </Database>
    -->
</Command>
</NMCLITask>
```

## 4.7 Depth Tiff

Description: DepthTiff returns a black&white \*.tif which cannot be opened in the Windows Photo Viewer, but in QGIS.

Please open the \*.tif in QGIS.

```

<!-- Depth Tiff -->
<NMCLITask po="1" name="DepthTiff" logKeyword="DepthTiff" output=" " >
    <Command name="DepthTiff">
        <dbfile>\eiva.local\data\DataSet\Internal_use_only\_TestData\NaviModel\cli\reject_fract
        <tiffile>\eiva.local\data\DataSet\Internal_use_only\_TestData\NaviModel\cli\reject_fract
        <OverwriteExistingFile>true</OverwriteExistingFile>
        <Surface>Average</Surface>
        <ScaleZ>1</ScaleZ>
        <NoDataValue>-9999</NoDataValue>

        <Geodesy>
            <Destination>
                <crs>32632</crs>
            </Destination>
        </Geodesy>
    </Command>
</NMCLITask>
```

## 4.8 DTMintensity2raster

```

<NMCLITask po=" " name=" " logKeyword=" " output=" ">
    <Command name="DtmIntensity2Raster">
        <dbfile>d:\Support\JIRA\SUPPORT-22124\HI1587_TestA.db</dbfile>
        <tiffile overwrite="true" format="Float32" gabfilling="true" compression="LZW" nodatavalue="-65536" >
            <geodesy>
                <Destination crs="3042">
                    <Ellipsoid epsg="7030" name="WGS 84">
                        <!-- SPHEROID["WGS 84",6378137,298.257223563,AUTHORITY["EPSG","7030"]] -->
                        <!-- eiva does not use the ellipsoid epsgs number, but indirectly by use of the projection -->
                        <SemiMajorAxis>6378137.0</SemiMajorAxis>
                        <InverseFlattening>298.257223563</InverseFlattening>
                    </Ellipsoid>
                    <Projection epsg="32632" name="UTM (north)">
                        <!-- 32625 32626 32627 .. 32630 32631 .. 32635 -->
                        <!-- epsg 32632 requires/implies wgs 84 ellipsoid -->
                        <EivaType>5</EivaType>
                        <!-- integer PROJPAC.h UTM N = 5 -->
                        <FalseEasting>500000</FalseEasting>
                        <FalseNorthing>0</FalseNorthing>
                        <FirstParallel>0</FirstParallel>
                        <SecondParallel></SecondParallel>
                    <Scale>0.9996</Scale>
                </Destination>
            </geodesy>
        </tiffile>
    </Command>
</NMCLITask>
```

```

<OrigLatitude>0</OrigLatitude>
<OrigLongitude>9</OrigLongitude>
<!-- -33      -27      -21      ..      -3       3       ..      27 -->
<Zone>32</Zone>
<!-- 25      26      27      ..      30       31       ..      35 -->
</Projection>
</Destination>
<Source crs="3042">
  <Ellipsoid epsg="7030" name="WGS 84">
    <!-- eiva does not use the ellipsoid epsgs number, but indirectly by use of the projection crs -->
    <SemiMajorAxis>6378137.0</SemiMajorAxis>
    <InverseFlattening>298.257223563</InverseFlattening>
  </Ellipsoid>
  <!-- Projection is not in use -->
  <Projection epsg="32632" name="UTM (north)">
    <!-- 32625   32626   32627   ..   32630   32631   ..   32635 -->
    <!-- epsg 32632 requires/implies wgs 84 ellipsoid -->
    <EivaType>5</EivaType>
    <!-- integer PROJPAC.h UTM N = 5 -->
    <FalseEasting>500000</FalseEasting>
    <FalseNorthing>0</FalseNorthing>
    <FirstParallel>0</FirstParallel>
    <SecondParallel></SecondParallel>
    <Scale>0.9996</Scale>
    <OrigLatitude>0</OrigLatitude>
    <OrigLongitude>9</OrigLongitude>
    <!-- -33      -27      -21      ..      -3       3       ..      27 -->
    <Zone>32</Zone>
    <!-- 25      26      27      ..      30       31       ..      35 -->
  </Projection>
</Source>
<Setup></Setup>
<Shift method="0" name="No shift"></Shift>
</geodesy>
</Command>
</DtmIntensity2Raster>

```

## 4.9 DTMsurface2singlebandraster

```

<Command name="DtmSurface2SingleBandRaster">
  <dbfile surface="depth.map" scale="-1.0">d:\Support\JIRA\SUPPORT-22124\HI1587_TestA.db</dbfile>
  <tiff file overwrite="true" format="Float32" gapfilling="false" tiled="true" compression="deflate">
    <Geodesy>
      <Destination crs="3042">
        <Ellipsoid epsg="7030" name="WGS 84">
          <!-- SPHEROID["WGS 84",6378137,298.257223563,AUTHORITY["EPSG","7030"]] -->
          <!-- eiva does not use the ellipsoid epsgs number, but indirectly by use of the projection crs -->
          <SemiMajorAxis>6378137.0</SemiMajorAxis>
          <InverseFlattening>298.257223563</InverseFlattening>
        </Ellipsoid>
        <Projection epsg="32632" name="UTM (north)">
          <!-- 32625   32626   32627   ..   32630   32631   ..   32635 -->
          <!-- epsg 32632 requires/implies wgs 84 ellipsoid -->
          <EivaType>5</EivaType>
        </Projection>
      </Destination>
    </Geodesy>
  </tiff>
</Command>

```

```

<!-- integer PROJPAC.h UTM N = 5 -->
<FalseEasting>500000</FalseEasting>
<FalseNorthing>0</FalseNorthing>
<FirstParallel>0</FirstParallel>
<SecondParallel></SecondParallel>
<Scale>0.9996</Scale>
<OrigLatitude>0</OrigLatitude>
<OrigLongitude>9</OrigLongitude>
<!-- -33      -27      -21      ..      -3      3      ..      27 -->
<Zone>32</Zone>
<!-- 25      26      27      ..      30      31      ..      35 -->
</Projection>
</Destination>
<Source crs="3042">
    <Ellipsoid epsg="7030" name="WGS 84">
        <!-- eiva does not use the ellipsoid epsgs number, but indirectly by use of the projection parameters -->
        <SemiMajorAxis>6378137.0</SemiMajorAxis>
        <InverseFlattening>298.257223563</InverseFlattening>
    </Ellipsoid>
    <!-- Projection is not in use -->
    <Projection epsg="32632" name="UTM (north)">
        <!-- 32625   32626   32627   ..   32630   32631   ..   32635 -->
        <!-- epsg 32632 requires/implies wgs 84 ellipsoid -->
        <EivaType>5</EivaType>
        <!-- integer PROJPAC.h UTM N = 5 -->
        <FalseEasting>500000</FalseEasting>
        <FalseNorthing>0</FalseNorthing>
        <FirstParallel>0</FirstParallel>
        <SecondParallel></SecondParallel>
        <Scale>0.9996</Scale>
        <OrigLatitude>0</OrigLatitude>
        <OrigLongitude>9</OrigLongitude>
        <!-- -33      -27      -21      ..      -3      3      ..      27 -->
        <Zone>32</Zone>
        <!-- 25      26      27      ..      30      31      ..      35 -->
    </Projection>
</Source>
<Setup></Setup>
<Shift method="0" name="No shift"></Shift>
</Geodesy>
</Command>

```

## 4.10 EC-3D Combined

```

<!-- EC- -->D Combined -->
<NMCLITask po="" name="EC3D" logKeyword="EC3D" output="" >
    <Command name="EC3D">
        <!-- UI anad corresponding Method names
            Distance to average value = DistanceToAverage
            Distance to median value = DistanceToMedian
            Distance to multiple points (median Z) = KMeanZ
            Distance to multiple points (median) = KMean
            Distance to nearest point = NearestNeighbour

```

```

        Distance to surface = SurfaceCleaning supports: min, max, median,
avg, std
        Fraction of points in cylinder = PointsInCylinder
        Number of points in sphere = NeighbourInSphere
        VerticalCluster = VerticalCluster
-->

<Input dbfile="\eiva.local\data\DataSet\Internal_use_only\_TestData\NaviModel\cli\ec3dcombined\pipe.db" />
    <Combined>
        <Absolute>
            <CleaningMethod name="DistanceToMedian" radius_m="1"
threshold_min="-0.13" threshold_max="0.14"/> <!-- In NM UI the diameter is
shown -->
            <CleaningMethod name="DistanceToAverage" radius_m="0.5"
threshold_min="-0.23" threshold_max="0.09"/>
            <CleaningMethod name="KMeanZ" radius_m="15"
threshold_max="0.59"/>
        </Absolute>
        <Relative voting ="3">
            <CleaningMethod name="NeighbourInSphere" radius_m="0.5"
threshold_min="39"/> <!-- In NM UI shown as pts -->
            <CleaningMethod name="PointsInCylinder" radius_m="0.5"
threshold_min="0.67"/>
            <CleaningMethod name="NearestNeighbour" threshold_max="0.11"/>
            <CleaningMethod name="KMean" radius_m="15" threshold_max="0.28"/>
        <!-- radius_m is here used for pts -->
        </Relative>
    </Combined>
</Command>
</NMCLITask>

```

	<b>EC-3D method and parameter in NM UI</b>	<b>Method names: Single EC-3D from WFM help file.</b>	<b>Combined EC-3D - Filter and parameter in XML</b>
1	Distance to average value	DistanceToAverage	DistanceToAverage <CleaningMethod name="DistanceToAverage" radius_m="1" threshold_min="-0.16" threshold_max="0.24"/>
2	Distance to median value	DistanceToMedian - Distance to median surface	DistanceToMedian <CleaningMethod name="DistanceToMedian" radius_m="1" threshold_min="-0.06" threshold_max="3.0"/>
3	Distance to multiple points (median Z)	KMeanZ - Median Z distance to multiple points	KMeanZ

			<CleaningMethod name="KMeanZ" radius_m="20" threshold_max="9.21"/>
4	Distance to multiple points (median)	KMean - Median distance to multiple points	KMean <CleaningMethod name="KMean" radius_m="20" threshold_max="9.21"/>  Note, radius_m is used for <b>Value</b> number of points.
5	Distance to nearest point	NeighbourInSphere - Number of points within a sphere  SetProperty TresholdMin="0" SetProperty TresholdMax="0.02"  method="NearestNeighbour" threshold_min="{TresholdMin}" threshold_max="{TresholdMax}" "/>	NearestNeighbour <CleaningMethod name="NearestNeighbour" threshold_max="0.14"/>
6	Distance to surface		<CleaningMethod name="SurfaceCleaning" surface="max" threshold_min="0.0" threshold_max="0.01"/>  Note, surface supports: min, max, median, avg, std.
7	Fraction of points in cylinder	PointsInCylinder - Number of points within a cylinder	PointsInCylinder <CleaningMethod name="PointsInCylinder" radius_m="0.75" threshold_min="0.5"/>
8	Number of points in sphere	Radius,min,max	<CleaningMethod name="NeighbourInSphere" radius_m="0.5" threshold_min="2"/>
9	Vertical cluster	VerticalCluster	<CleaningMethod name="VerticalCluster"

		radius_m="0.5" threshold_min="2"/>
--	--	---------------------------------------

Method	Value
Number of points in sphere	1.00m

Please note, that the value shown is the diameter

## 4.11 EC-3D cleaning

Description:

Syntax: nmcli - command.

In Navimodel it is possible to run the EC-3D with 3 parameters: Radius\_m, threshold\_min, threshold\_max.

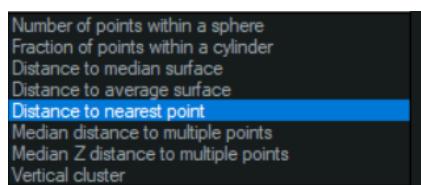
In cli it will select the whole dtm and clean with the method and values selected.

The threshold values can be multiple thing. For nearest neighbor it is in metres maybe 0.3m and for NeighboursInSphere it is the number of neighbours maybe 20.

So the results shown in the display should be similar to the results from cli. In the xml file the threshold\_max is set to 0.004.



The values are shown in NM:



```

<NMCLITask po="1" name="EC3D" logKeyword="EC3D" output=" " >
    <Command name="EC3D">
        <!-- UI and corresponding Method names
            Distance to average value = DistanceToAverage
            Distance to median value = DistanceToMedian
            Distance to multiple points (median Z) = KMeanZ
            Distance to multiple points (median) = KMean
            Distance to nearest point = NeighbourInSphere
            Distance to surface = SurfaceCleaning supports the surfaces: min, max, median, avg, std
            Fraction of points in cylinder = PointsInCylinder
            Number of points in sphere = NeighbourInSphere
            VerticalCluster = VerticalCluster
        -->
        <Input>
            <dbfile>\eiva.local\data\DataSet\Internal_use_only\_TestData\NaviModel\cli\ec3dcleaning
            <method>DistanceToAverage</method>
            <radius_m>1</radius_m>
            <threshold_min>0.0</threshold_min>
            <threshold_max>0.02</threshold_max>
            <DeletePatchFile>yes</DeletePatchFile>
        </Input>
    </Command>
</NMCLITask>

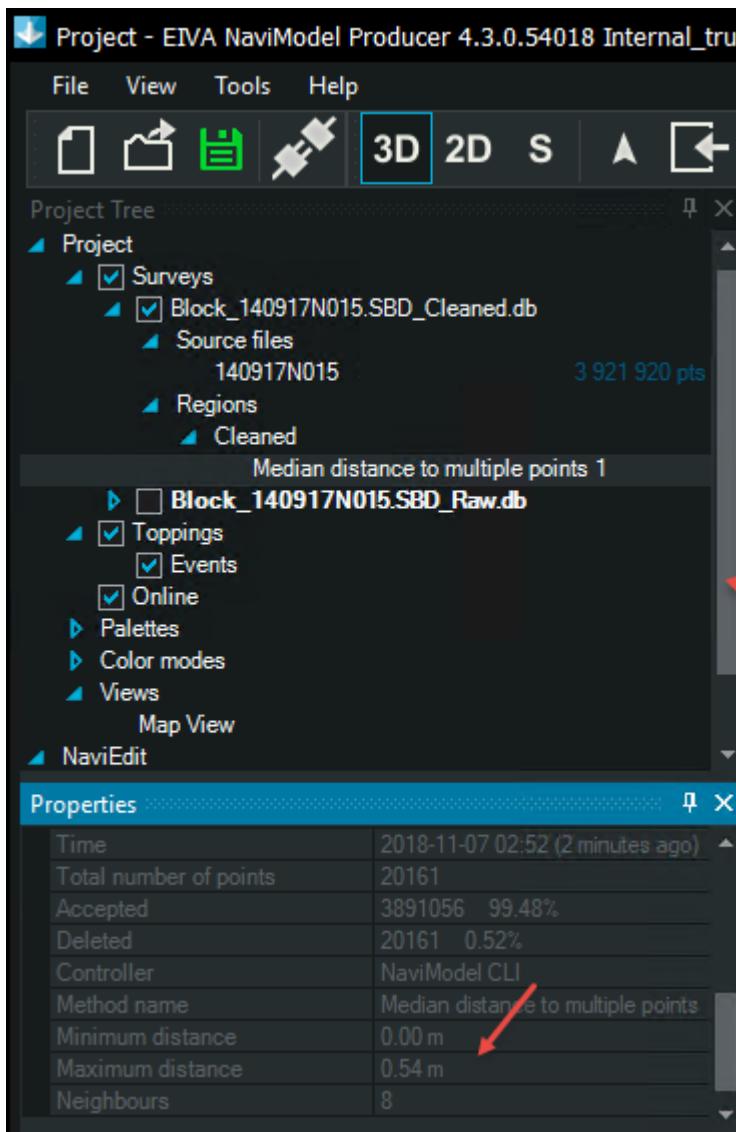
```

The EC-3D thresholds can be defined either at the top of the xml (in the group task) or in the NMCLITask.

```

<GroupTask po="1" name="Define variables">
    < SetPropertyTask po="2" name="Base directory" input="C:\temp\Kuda_Demo" output="BaseDirectory" level="2" >
    < SetPropertyTask po="2" name="Define DTM Cell Size (m)" input="0.5" output="DTMcellsize" level="2" >
    < SetPropertyTask po="2" name="EC3D Radius in Metres" input="0.5" output="RadiusM" level="2" >
        <!-- This line is not used when using cleaning option Distance to nearest point -->
    < SetPropertyTask po="2" name="EC3D Threshold Minimum in Metres" input="0" output="ThresholdMin" level="2" >
    < SetPropertyTask po="2" name="EC3D Threshold Maximum in Metres" input="0.6" output="ThresholdMax" level="2" >
    < SetTitleTask po="3" name="Set Title" title="Kuda Workflow Monitoring: {BaseDirectory}\Watch" >
</GroupTask>

```



## 4.12 Export Grid

Description:

```
<!-- Export grid -->
<NMCLITask po="" name="ExportGrid" logKeyword="ExportGrid" output="" >
  <Command name="ExportGrid">
    <dbfile>\eiva.local\data\DataSet\Internal_use_only\_TestData\NaviModel\cli\reject_fraction</dbfile>
    <Surface>Average</Surface> <!-- depth.map, count.map, min.map, std.map -->
    <CellSize>0.5</CellSize>
    <outfile>c:\temp\1\2\3\qqq.xyz</outfile>
```

```

<OverwriteExistingFile>yes</OverwriteExistingFile>
<separator>,</separator>
<scalexy>1</scalexy>
<scalez>1</scalez>
<numdecimalsxy>2</numdecimalsxy>
<numdecimalsz>2</numdecimalsz>
</Command>
</NMCLITask>

```

## 4.13 Exponential Smoothing

Description:

```

<!-- Exponential Smoothing -->
<NMCLITask po="" name="Exponential smoothing" logKeyword="" output="" >
    <Command name="exponentialsMOOTHING">
        <lines>
            \\eiva.local\\Data\\DataSets\\Internal_use_only\\_TestData\\NaviModel\\cli\\Exponential Smoothing
            \\eiva.local\\Data\\DataSets\\Internal_use_only\\_TestData\\NaviModel\\cli\\Exponential Smoothing
        </lines>
        <outputFolder>c:\\Data\\Output\\</outputFolder>
        <outputName>Top of Pipe</outputName>
        <runline>c:\\Runline 1.rlx</runline>
        <smoothing>500</smoothing>
    </Command>
</NMCLITask>

```

## 4.14 Export points

Description: Exports accepted points

```

<!-- Export points -->
<NMCLITask po="" name="ExportPoints" logKeyword="ExportPoints" output="" >
    <Command name="ExportPoints">
        <dbfile>\\eiva.local\\data\\DataSets\\Internal_use_only\\_TestData\\NaviModel\\cli\\ExportPoints.db</dbfile>
        <outfile>c:\\temp\\1\\2\\3\\qqq.xyz</outfile>
        <outfolder>c:\\temp\\1\\2\\allfiles</outfolder>
        <OverwriteExistingFile>true</OverwriteExistingFile>
        <separator>,</separator>
        <scalexy>1</scalexy>
        <scalez>1</scalez>
        <numdecimalsxy>3</numdecimalsxy>
        <numdecimalsz>4</numdecimalsz>
    </Command>
</NMCLITask>

```

## 4.15 Extract Event Frames

Description:

```
<!-- Extract Event Frames -->
<NMCLITask po="" name="Extract Event Frames" logKeyword="" output="" >
  <Command name="ExtractEventFrames">
    <Base path="c:\Visualworks\Projects\2019\16R00001\DATA_20160518161620000"/>
    <Event path="events.csv" template="VisualSoft 2"/>
    <Export path="C:\temp\DATA_20160518161620000\Export"/>
    <Track path="20160518161620000@current.csv" template="VisualSoft 2"/>
    <Upload upload="false"/>
    <Video path=". "/>
  </ExtractEventFrames>
</NMCLITask>
```

## 4.16 ExportEventCollection

Convert an ascii event file with an import template created from NaviModel (under ASCII file import) to a xml file containing information of the events, this file can be used with the command NewNaviModelProject with the tag <EventCollection>

- eventsAsciifile: Path to ascii file containing event information.
- eventsTemplatefile: Path to import template file (located in C:\ProgramData\EIVA\NaviModel Producer\ImportTemplates).
- outfile : Path to the result xml event file

```
<!--Export Event Collection -->
<NMCLITask po="" name="ExportEvents" logKeyword="ExportEvents" >
  <Command name="exporteventcollection">
    <eventsAsciifile>Events.SENAI.txt</eventsAsciifile>
    <eventsTemplatefile>Event Collection.SENAI.txt</eventsTemplatefile>
    <outfile>c:\temp\cli_out\events.SENAI.xml</outfile>
  </Command>
</NMCLITask>
```

## 4.17 ExportTrack

Convert an ascii track file with an import template created from NaviModel (under ASCII file import) to a track file .etr

- trackAsciifile : Path to ascii file containing track information.
- trackTemplatefile : Path to import template file (located in C:\ProgramData\EIVA\NaviModel Producer\ImportTemplates).
- outfile : Path to the result track file

```
<!-- Export Track -->
<NMCLITask po="" name="ExportTrack" logKeyword="ExportTrack" >
  <Command name="ExportTrack">
```

```

<trackAsciifile>TrackAscii.txt</trackAsciifile>
<trackTemplatefile>Track.AsciiTemplate.txt</trackTemplatefile>
<outfile>c:\temp\cli_out\track.etr</outfile>
</Command>
</NMCLITask>

```

## 4.18 Find Gaps

Description: This is the NaviModel task, done from the imported dtm with the option **Identify DTM gap events**.

```

<!-- Find gaps -->
<NMCLITask po="" name="FindGaps" logKeyword="FindGaps" output="" >
  <Command name="FindGaps">
    <Input>
      <dbfile>\eiva.local\data\DataSet\Internal_use_only\_TestData\NaviModel\cli\landscape
3237.db</dbfile>
        <PointsRequiredForGoodCell>1</PointsRequiredForGoodCell>
        <CellsRequiredForGap>1</CellsRequiredForGap>
        <GapEventFile>c:\temp\gaps.xml</GapEventFile>
      <Csv>
        <File>c:\temp\gaps.csv</File>
        <Separator>" "</Separator>
        <UseNMSummaryFormat>no</UseNMSummaryFormat>
      </Csv>
    </Input>
  </Command>
</NMCLITask>

```

## 4.19 Find Rocks

```

<!-- Find Rocks -->
<NMCLITask po="" name="FindRocks" logKeyword="FindRocks" output="" >
  <Command name="FindRocks">
    <dbfile>\eiva.local\data\DataSet\Internal_use_only\_TestData\NaviModel\cli\FindRocks\Roo
    <ExportFolder>c:\temp\cli_out\FindRocks</ExportFolder>
    <Slope>12 deg</Slope>
    <boundary>
      <linesfolder>\eiva.local\data\DataSet\Internal_use_only\_TestData\NaviModel\cli\FindR
      <MaxDistance>4 m</MaxDistance>
    </boundary>
    <Filter>
      <MinHeight>0.2 m</MinHeight>
      <MaxHeight>2 m</MaxHeight>
      <MinDiameter>0.5 m</MinDiameter>
      <MaxDiameter>2 m</MaxDiameter>
      <MinVolume>0.2 m3</MinVolume>
      <MaxVolume>2 m3</MaxVolume>
    </Filter>
  </Command>
</NMCLITask>

```

## 4.20 Level Check

Description:

```
<!-- Level Check -->
<NMCLITask po="" name="" logKeyword="" Output="" >
    <Command name="LevelCheck">
        <Input dbfile="c:\Data\surface.db"/>
    </Command>
</NMCLITask>
```

## 4.21 NMCLI

Description:

CLI is the abbreviation for command-line-interface, eg nmcli.exe is a special version of NaviModel that can be executed without using the UI.

Example for an NMCLI task in NaviModel 4.6:

```
<Setup StateDatabaseLocation="c:\WFM\EivaLive\TaskStateDatabase.db">
    <GroupTask name="Workflow">
        <NMCLITask po="1" name="FindGaps" logKeyword="FindGaps" output="" >
            <Command name="findgaps">
                <Input>
                    <dbfile>C:\cli\landscape3237.db</dbfile>
                    <PointsRequiredForGoodCell>1</PointsRequiredForGoodCell>
                    <CellsRequiredForGap>1</CellsRequiredForGap>
                    <GapEventFile>c:\temp\gaps.xml</GapEventFile>
                    <Csv>
                        <File>c:\temp\gaps.csv</File>
                        <Separator>"</Separator>
                        <UseNMSummaryFormat>false</UseNMSummaryFormat>
                    </Csv>
                </Input>
            </Command>
        </NMCLITask>
    </GroupTask>
</Setup>
```

Other commands that can be used are

createdtm, scanscore, ec3d, writeback, rejectfraction, cellstdev, levelcheck, findgaps, track2shape, runline2shape, depthtiff, shadedrelief, exportgrid, extracteventframes, cameraeventing, compareeventcollection, corridorcoverage, automaticflags, newnavimodelproject, exponentialsmoothing.

## 4.22 Merge Intensity

```
<!-- Merge intensity -->
<NMCLITask po="" name="" logKeyword="" output="" >
  <Command name="MergeIntensity">
    <dbfile>C:\Users\jek\Desktop\HI1587_TestA - Copy.db</dbfile>
  </Command>
</NMCLITask>
```

## 4.23 New NaviModel Project

Creates a new NaviModel Project file, and includes the following in the project:

- <Files>
  - DTM (Digital Terrain Model) file (.db)
  - Pipe Flag file (.eff)
  - Runline (.rlx)
  - Digitized Lines (.dig)
  - Track (.etr)
- <ImagingSeries> : Allow to add one or multiple ImagingSeries objects inside a new navimodel project.
  - <ImagingSeries> has
    - a <Name> (Typically name of the camera)
    - and one or multiple <Path> (Path to the folder(s) where images are located)
  - Multiple <ImagingSeries> tags can be added under <input> section.
- <Events> : Allow to add one or multiple EventCollection objects inside a new navimodel project.
  - <Events> has one tag <Folder> to the folder containing the .xml file created by command “exporteventcollection”
  - Only one tag <Events> allowed in command NewNaviModelProject
- <Events> will override all <EventCollection> tags if they’re both used in command NewNaviModelProject
- An extra option <ExistingProjectDir> is available under <input>. It represents the path where the existing project (.nmp folder) is located.
- If this path is correct, then a new project will be created from the existing project.
  - The new project will have a new name defined by <ProjectName>.
  - The content of the toppings folder, together with the settings inside windows.xml and ToolbarLayout.xml from the existing project will be copied to the new project.
  - All new data defined in the <input><files> will be appended to the new project.

```
<!-- New NaviModel Project -->
<NMCLITask po="1" name="New NaviModel Project" logKeyword="New NaviModel project">
  <Command name="NewNaviModelProject">
    <input>
```

```

        <files><!-- Types of files that can be added DB, RLN, RLX,
DIG, EFF, ETR -->
        \newnavimodelproject\Runline 2.rlx
        \newnavimodelproject\0000_20221223_085017_Bintang.db
        \newnavimodelproject\KP 0.000 to KP 0.075.eff
        \newnavimodelproject\Line 1.dig
        \newnavimodelproject\Line 2.dig
        \newnavimodelproject\Track.etr
    </files>
    <ImagingSeries>
        <Name>Centre</Name>
        <Path>\Centre_Stills\Centre_0000</Path>
        <Path>\Centre_Stills\Centre_0001</Path>
        <Path>\Centre_Stills\Centre_0002</Path>
        <Path>\Centre_Stills\Centre_0003</Path>
    </ImagingSeries>
    <ImagingSeries>
        <Name>PortBoom</Name>
        <Path>\PortBoom_Stills\PortBoom_0000</Path>
        <Path>\PortBoom_Stills\PortBoom_0001</Path>
    </ImagingSeries>
    <Events>
        <Folder>\eventcollection</Folder>
    </Events>
    <ProjectName>somename</ProjectName>
    <ExistingProjectDir>ExistingProjectName</ExistingProjectDir>
</input>
<Output>
    <TargetDir>c:\temp\cli_out\NewNaviModelProject</TargetDir>
</Output>
<Geodesy>
    <Destination> <!-- Geodesy can be defined for the new project -->
    <crs>31984</crs>
    </Destination>
</Geodesy>
</Command>
</NMCLITask>

```

## 4.24 Reject Fraction

Description:

```

<!-- Reject fraction -->
<NMCLITask po="" name="" logKeyword="" output="" >
    <Command name="RejectFraction">
        <Input MaxAllowedPercentageDeleted_pct="5" dbfile="c:\Data\fraction.db"/>
    </Command>
</NMCLITask>

```

## 4.25 Runline to Shape

Description:

```
<!-- Runline to Shape -->
<NMCLITask po="" name="" logKeyword="" Output="" >
  <Command name="Runline2ShapeTask">
    <Shape OverwriteExistingFile="yes" Shapefile="c:\Data\kk.shp"/>
    <Database>
      <Input xml="{SqlServer}"/>
      <Input xml="{DatabaseName}"/>
      <Folder>\50\008</Folder>
    </Database>
    <Geodesy>
      ...
    </Geodesy>
  </Command>
</NMCLITask>
```

## 4.26 ShadedRelief

Description: ShadedRelief returns a black&white \*.tif which can be opened in the Windows Photo Viewer.

```
<!-- Exports a Shaded Relief -->
<NMCLITask po="" name="Shaded Relief" logKeyword="Shaded Relief" output="" >
  <Command name="ShadedRelief">
    <Input ImageFile="c:\shadedrelief.tif" OverwriteExistingFile="true" Surface="Average" ScaleZ="1000" >
      <dbfile>{BaseDirectory}\Test.db</dbfile>
      <Color>grey</Color>
    <Input/>
    <!-- <Geodesy>.. <Geodesy> -->
  </Command>
</NMCLITask>
```

## 4.27 ScanScore Scalgo

Description: Runs a Scalgo cleaning task.

```
< SetPropertyTask po="1" name="Define S-CAN Score threshold (cm)" input="4" output="ScoreThreshold" >
<!-- ScanScore Scalgo -->
<NMCLITask po="2" name="S-CAN Score" logKeyword="{AllFile.Filename}" >
  <Command name="ScanScore">
    <Input dbfile="\NM\DB\Cleaned\Block_{AllFile.Filename}_Cleaned.db" threshold="{ScoreThreshold}" TempFolder="C:/_tmp_WFM/scalgotemp" DeletePatchFile="yes"/>
  </Command>
</NMCLITask>
```

## 4.28 Track2Shape

Description:

```
<!-- Track Shape-->
<NMCLITask po="" name="Export Track (.shp) for GIS" logKeyword="ExportTrack (SHP)">
  <Shape path="{BaseDirectory}\Data\Track_.shp" OverwriteExistingFile="yes"/>
  <Command name="Track2Shape">
    <Database>
      <Input xml="{SqlServer}" />
      <Input xml="{DatabaseName}" />
      <BlockIds>
        {SBDBlockID}
      </BlockIds>
    </Database>
    <Input xml="{Geodesy}" />
  </Command>
</NMCLITask>
```

## 4.29 WriteBack

Description: NaviModel task. Write cleaned data back to NaviEdit.

```
<!-- WriteBack -->
<NMCLITask po="" name="WriteBack to NaviEdit" logKeyword="{SBDFile.Filename}" output=" ">
  <Command name="WriteBack">
    <Input path="{BaseDirectory}\Output\Block_{SBDFile.Filename}_Cleaned.db"/>
  </Command>
</NMCLITask>
```

## 5 FAQ

Enter topic text here.

### 5.1 FAQ

**Q.**

How do we note success or failure from an external application? Is there a special message via stdout?

We have progress messages working fine via stdout but cannot flag success/failure.

**A.**

```
<ExternalTask po="4" name="Attach Database" logKeyword="Attaching database."  
Location="C:\EIVA\NaviEdit\bin\JobPlanner.exe" Parameter="-attachdb NaviEditDB  
{Directory}Database\NaviEditDB.mdf {Directory}Database\NaviEditDB_Log.Idf"/>
```

**Q.**

Error message **Block '0' does not exist**

**A.**

Relates to an unlinked tide block.

**Q.**

I get a warning 'SBD Import: 200703162248.SBD - Could not read XML attribute '/NaviEditTask/Geodesy/@useSbdGeodesy' on the element Geodesy. How can I solve it?

**A.**

Find the Geodesy element in the xml and add the attribute.

Eg <Geodesy useSbdGeodesy="no">

