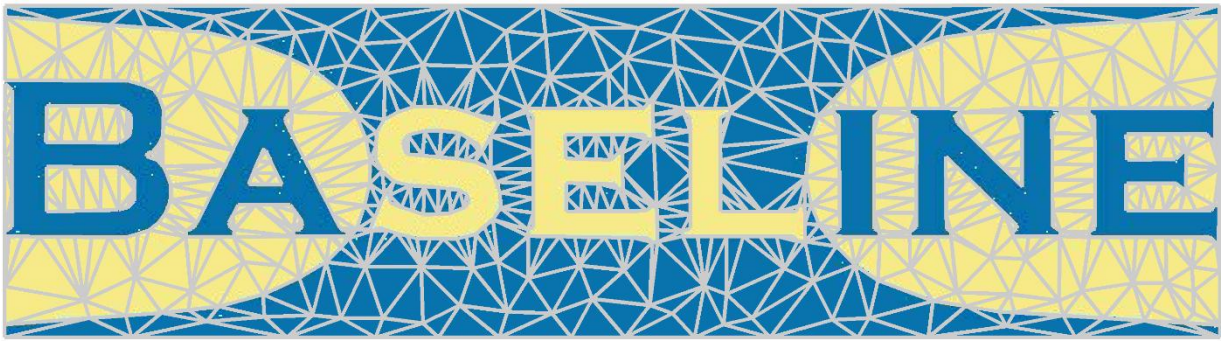
Rijkswaterstaat
Ministry of Infrastructure and Water Management



User Manual Baseline 7

Date July 2026
Status Valid for Baseline 7.2.0 (and higher)

Deltares maintains and supports Baseline according the de Service Level Agreements (SLA) with the Ministry of Infrastructure & Water Management (I&W).

Deltares
Enabling Delta Life

User Manual Baseline 7

Date July 2026
Status Valid for Baseline 7.2.0 (and higher)

Colofon

Published by	Rijkswaterstaat / Deltares
Information	Erik van Rooijen / Willem Ottevanger
Authors	Frans Hoefsloot (WSP), Paul van der Wal (WSP), Reint Jansen (Esri)
Date	July 2026
Status	Valid for Baseline 7.2.0 (and higher)
Versienummer	7.2

Contents

1 Introduction

Baseline 7 is a combination of an ArcGIS add-in and a geographical database. The database is meant for storage, editing, consulting and presenting of data of hydrodynamic systems used in calculations. In addition, Baseline can be used to convert data to suitable files for use in D-HYDRO Suite and SWAN. The ArcGIS add-in and database are suitable for use with ArcGIS Pro 3.0 or higher with an ArcGIS-Advanced license and 3D-analyst extension. For the conversion of an gridcell-averaged elevation to D-HYDRO and SWAN a Spatial Analist extension is required.

The geographical database and extension are developed on behalf of Deltares and Rijkswaterstaat. However Baseline can be used whenever a structured storage of river data and/or a link of flow models with a GIS environment is desired.

Baseline 7 is designed by WSP and updated to ArcGIS Pro by Esri. The predecessor of Baseline 7 is Baseline 6. The functionality of both versions is mostly kept intact. The predecessor of Baselien 6 is Baseline 5. Baseline 5 was developed by Geodan IT and LievenseCSO (before CSO). Both Baseline 5 and Baseline 6 are based on ArcObjects and extensions of ArcGIS.

Although Baseline has been developed as an ArcGIS Pro add-in, the application is primarily aimed at users of hydrodynamic models. Basic knowledge of ArcGIS is required.

The purpose of this document is to guide the user in using Baseline 7. This manual explains the structure and usage of the database and describes the functionality. Subsequently, the operation of the individual functions of Baseline 7 are described.

Since Baseline 6, the datamodel and workflow is completely revised compared to Baseline 5. Data that was stored twice (in Baseline 5) is now stored in one place only. This still applies to Baseline 7.

2 Background Baseline, D-HYDRO Suite and SWAN

2.1 Introduction

Baseline can be used to generate files which can be used for D-HYDRO Suite (Delft3D-Flexible Mesh Suite) and SWAN. This chapter explains the functionalities of Baseline and how it can be used to generate the necessary files for hydrodynamic models. In addition a brief description is given of D-HYDRO Suite.

2.2 Baseline and hydrodynamic models

2.2.1 *Baseline*

General

Baseline 7 is a combination of an add-in of ArcGIS Pro and a geographical database. This database is meant for storing, editing, consulting and presenting of data of hydrodynamic systems used in hydrodynamic simulations. In Baseline spatial data is stored in such a way that an adequate geographical description of an area is obtained. This geographical description is converted to input for a hydrodynamic model.

In addition to this document, a Baseline 7 protocol is created, this document also includes the main changes from Baseline 5 to Baseline 6/7, and from Baseline 6 to Baseline 7. A detailed description of Baseline 7 (the purpose and use) is given in Chapter 4.

Baseline is a GIS tool used for hydrodynamic models. All functionalities in Baseline are controlled via menus. With these menus it is possible to change and maintain the geographical data stored in the database and convert these data to hydrodynamic models.

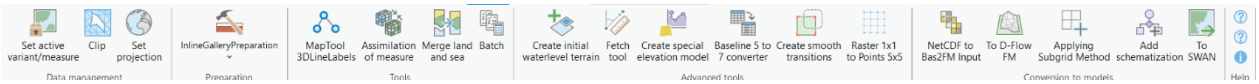


Figure 2.2.1-1 Baseline toolbar

All functions are designed to make spatial data unambiguously available for conversion to hydrodynamic model files. By formalizing procedures and documenting the data model, a high degree of reproducibility of schematizations is obtained.

The function “Data management” offers the possibility to create an empty template for new areas, variants and schematizations. Also an active variant of measure can be loaded.

The menu “Preparation” contains the input module for building measures.
The application enables the user to make a cut of the model (clip) or to erase a part of the model (erase). This can be done with the menu “Tools”. This menu also is used to start mixing measures or go into batch-mode. Finally this menu offers a check on protocol and content for variants and measures and some advanced tools.

The menu “Models” can be used for converting the geographical database to files usable for hydrodynamic models. Hydrodynamic models which are supported are SWAN and D-HYDRO Suite, module D-Flow Flexible Mesh (D-Flow FM).

2.3 Hydrodynamic models

2.3.1 *D-HYDRO Suite*

The D-HYDRO Suite or Delft3D Flexible Mesh (Delft3D FM) Suite is the successor of (structured) SIMONA/WAQUA or Delft3D 4 Suite. Like Delft3D 4, the Delft3D FM Suite can simulate storm surges, hurricanes, tsunamis, detailed flows and water levels, waves, sediment transport and morphology, water quality and ecology, and is capable of handling the interactions between these processes. The suite is designed for use by domain experts and non-experts alike, which may range from consultants and engineers or contractors, to regulators and government officials, all of whom are active in one or more of the stages of the design, implementation and management cycle.

The key component of D-HYDRO Suite is the D-Flow Flexible Mesh (D-Flow FM) engine for hydrodynamical simulations on unstructured grids in 1D-2D-3D.

All Feature classes of the Baseline database can be converted to inputfiles for D-Flow Flexible Mesh.

2.3.2 *SWAN*

SWAN is a numerical model for the simulation of waves in deep, moderate and shallow water. SWAN uses a rectilinear, curvilinear or unstructured (flexible) mesh, like D-Flow Flexible Mesh.

A SWAN model can be used for the following processes:

- Wave reproduction in time and space, shoaling, refraction due to flow and depth, frequency shift due to flow and non-stationary depth;
- Wave formation through wind;
- Non-linear interaction between waves (both quadruplets and triads);
- Blocking of waves by flow.

More information on: www.swan.tudelft.nl

The conversion of the Baseline data is limited to the elevation model, weirs and thin dams (included in the weirs file). These are the elevation and the wave reducing obstacles used in SWAN.

3 Database en Files Baseline

3.1 Introduction

The Baseline database is designed to save and process files in a reproducible manner. This chapter discusses the Baseline database and files. To enable a relatively easy and structured method the Dataprotocol Baseline 7 [1] is developed. In the dataprotocol the entire database and several files are explained more thoroughly.

3.2 Baseline database

The Baseline database is a structured collection of folders with Baseline-files. When starting with Baseline the first step is to build a database with the Baseline structure. This structure is showed below:

<Name can vary> , File name, [Directory-Name]

<area>
- <variant>
  - [Data]
    - [Source] (Location of source files)
    - [Import] (Actually non-functional,)
    - [Export] (Export location for example to. ASCII)
    - [Layers] (Users defined layer-files for deviation of standard layers)
  - [Metainfo]
    - [lists] (batch-lists, measurelists and erase- and appendlists.
    - [Logs] (Location of log files)
    - Metadata.doc
    - Status.json (Keeps track of the version & edit dates)
    - Other .doc-bestanden
  - [Models]
    - [SWAN]
      - [<Schematization_name>]
        - <rooster>.swn
        - (<rooster>.rgf)
        - (<rooster>.grd)
        - rooster.gdb
        - [Geometry]
          - [Output_locations]
    - [DflowFM]
      - [<schematization_name>]
        - [boundary_conditions]
        - [computations]
        - [general]
        - [geometry]
          - [cross_sections]
          - [output_locations]
          - [sources_sinks]
          - [structures]
        - [initial_conditions
        - [metainfo]
        - [rtc]
  - [Baseline.gdb]

An extensive description is available in the Dataprotocol Baseline 7 [2].

4 Baseline Functions

4.1 Introduction

The Baseline add-in is a toolbar in ArcGIS Pro. This toolbar contains all Baseline functions. This chapter discusses all available buttons. Every button of the toolbar is discussed in a separate section. Per menu all submenus and/or functions are discussed. The purpose of each function is described.

As can be seen in figure 1, the following menus are available:

- Data management
- Preparation
- Tools
- Models
- Help

4.2 Data management

This paragraph describes the following functions:

1. Set active variant
2. Clip
3. Set projection

4.2.1 *Set Active variant*

Purpose:
With this function an active variant is set. The active variant is the database in which changes are made with the Baseline functions.

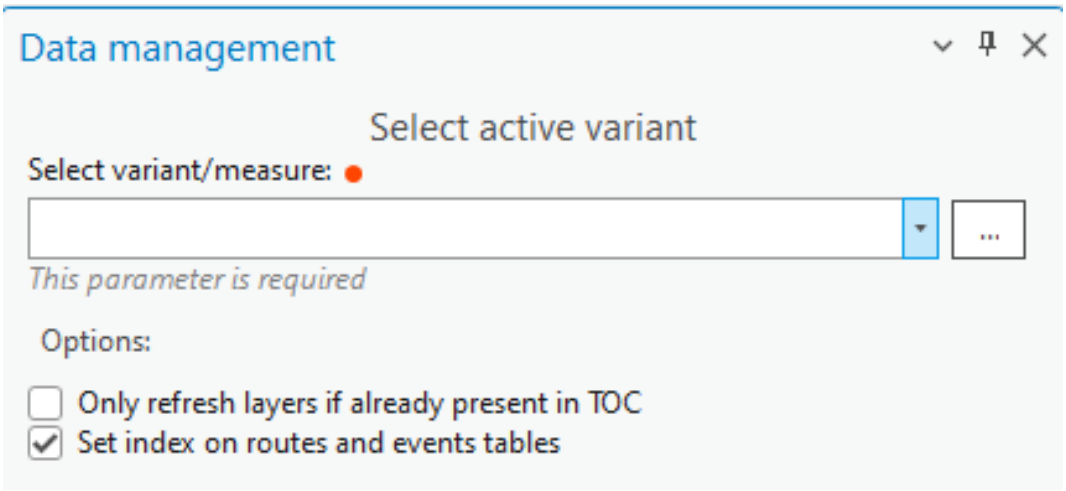


Figure 4.2.1-1 Dockpane

Explanation:
The active variant is selected using a "Windows Explorer structure" where the user navigates to the variant or measure. A variant is always located in an area, which is always located in the directory Baseline: "Baseline/<area>/<variant>. A measure must be located in the directory Baseline: "Baseline/<area> or within a subfolder of this directory. Subfolders up to 3 levels deep are allowed. When a chosen directory is not a variant or measure, an error will appear. After selecting an active variant this variant/measure will be loaded in ArcGIS Pro, using the default legend. There is a legend for a variant and one for a measure. Differences between those two are the presence of an elevation model and erasefiles for instance.

For Baseline data on sea, outside the domain of the Rijksdriehoelsstelsel there are sea variants and measures in WGS1984 projection available.

The first time an active variant is set in Baseline 7 the TemplateDefault folder is copied to Template. In this Template folder there are subfolders for variants and measures containing default gdb templates and layers. Here you can replace these default layer by your own customised layer. The first time a specific variant or measure is set as active variant the layer form the template is copied into the subfolder \data\layers of the variant/measure.

For measures there are 2 layers available; measure.lyrx and measure_complete.lyrx. The first one contains only the symbology for the most used Feature classes, eg. Elevation, landuse and model Feature classes. The second one contains all available Feature classes in a measure. This distinction is made for editing purposes, when a measure is constructed in the edit mode the create menu shows an extensive list of editable Feature classes\objects. This list is reduced in the default measure.lyrx compared with the measure_complete.lyrx. If less used Feature classes must be edited then the measure.lyrx can be renamed measure_short.lyrx and measure_complete.lyrx renamed to measure.lyrx. The create mernu then shows the complete list of editable Feature classes\objects.

Note: Baseline recognizes a measure by the presence of an erase_list.txt and an append_list.txt.

If a user uses the option "Only refresh layers if already present in TOC" a previous loaded variant will be refreshed. When this option is ignored a previous loaded variant will be duplicated in the TOC. When closing Baseline the last active variant set is remembered.

If a user uses the option "Set index on routes and events tables" (default checked) an index is set on all routes and events for the field NUMBER. When this option is checked but an index is present no action will be carried out. As a result of this index the performance is significantly increased. However, when no index is present yet this action will take some extra time.

4.2.2 *Clip*

Purpose:
This function clips all Feature Classes within the selected Baseline database. This function copies these cutouts to a new Baseline database. The Clip dataset can only be a polygon feature class inside a File Geodatabase.

The clip dataset might contain more than one clip polygon. If so, the dataset must contain a field NAME in which the name of the output directory (target Baseline directory) is defined.

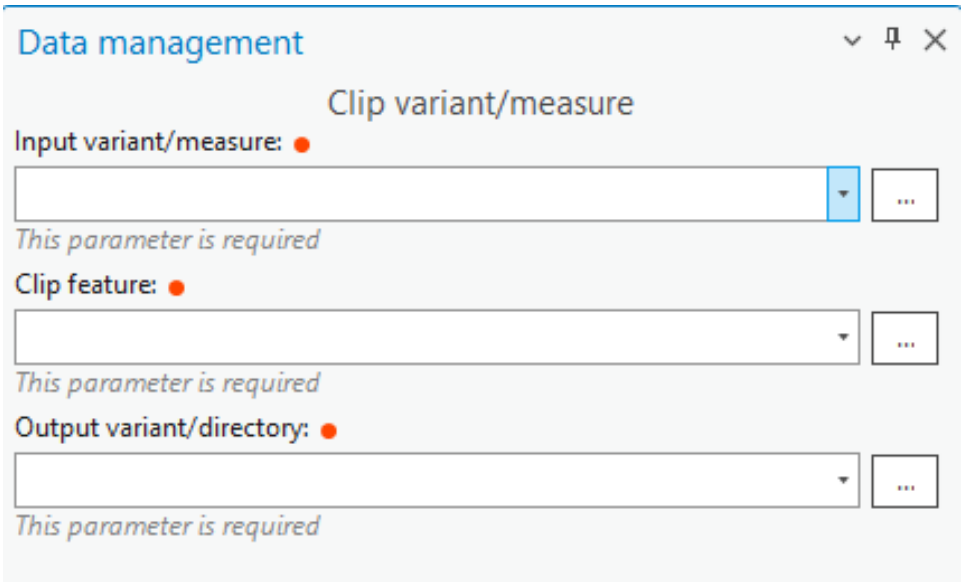


Figure 4.2.2-1 Dockpane

Explanation:
To use this function an input folder should be selected. This should be a Baseline variant or measure. Next a polygon Feature class in a File geodatabase should be selected. This Feature class contains one or more polygons defining the extent of the desired clip(s). At last the path to an empty output directory has to be selected in case of a single clip, this directory is converted to a variant/measure. In case of a clip with multiple clips, the clips are pasted inside this directory, following the name structure given in the NAME attribute of the clip feature class.

Note: This function is not intended to clip sea databases; a sea database is always used to merge with a lands database, an eventual clip is done in the merging process, see also chapter 3. If a sea varaint is clipped using the clip variant/measure function the elevation_mosaic is not updated.

4.2.3 *Set projection*

Purpose:
This function allows the user to choose a projection which is used. This projection is used for the template geodatabases of variants and measures. When a projection is selected by the user, these templates are copied to the user template directory of Baseline. This function needs to be used to work with a projection different than RD_NEW (Rijksdriehoeksstelsel). This function does not need to be used if RD_NEW is used unless another projection is used previously.

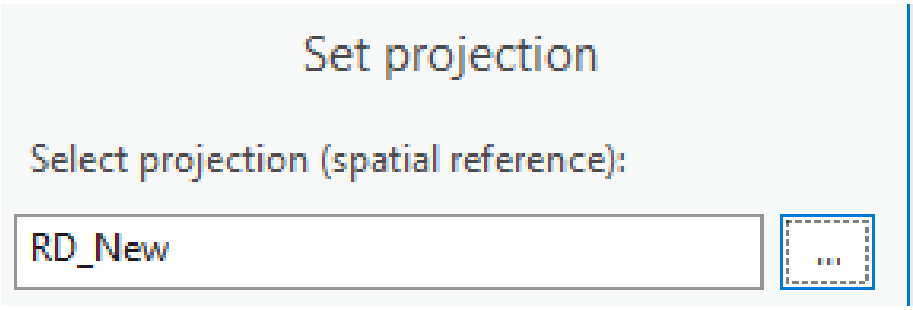


Figure 4.2.3-1 Dockpane

Explanation:
Default template databases have an RD_NEW projection by default. When a user would like to work with another projection it is necessary to convert these default database to the desired projection. These are located in the user template directory \Users\<username>\AppData\Local\Baseline\Template. Hereafter Baseline continues working with the chosen projection until another projection is set. If the user selects a variant with a projection other than the selected projection a message is displayed.

Note: Baseline is only able to save a chosen projection if it is a default ArcGIS projection (provided with a ESPG-code; http://spatialreference.org). Moreover only **projected** coordinate systems can be used. This is a restriction due to the presence of terrains in the Baseline database. Terrains can not be made using non-projected data.

For this reason a second set of templates is introduced to cope with non-projected data. These templates are "variant_sea" and "measure_sea", both have a WGS1984 projection. These templates are used in order to work with Baseline in the North Sea and the Atlantic Ocean outside the spatial domain of the Rijksdriehoeksstelsel. It is not necessary to change the projection using 'set projection' to set sea variants or measures as active variant. More information on sea databases is given in paragraph 4.4.3 and in the dataprotocol.

4.3 Preparation

The Preparation menu contains tools to assist the Baseline user when creating and updating measures, without the need of converting to other GIS-dataformats (shapefiles etc). The use of these tools is not mandatory, although strongly recommended.

Using the preparation tools the user can automate a number of (time-consuming or complex) processes. Also, a larger degree of uniformity can be achieved, which enhances the overall quality of Baseline measures.

For the tools which manipulate elevation data, 3D Analyst is required.

The menu contains various tools focussing on specific editing operations in ArcGIS Pro of mainly the elevation lines in Baseline measures. The Baseline 6/7 dataprotocol does use (3D) routes and events and 3D lines. This is similar to Baseline 5.
However, ArcGIS Pro does not contain straightforward edit tools for (3D) routes and events and 3D lines. In order to cope with this problem the (3D) routes and events or 3D lines must be converted in lines and points before editing. These lines and points can be edited with standard ArcGIS tools or specific Baseline preparation tools, after which they can be converted again to routes and events. During this conversion it is checked whether the points are positioned on the lines.

Note: When a dock pane of a tool is closed after use and then opened again, sometimes the dockpane does not appear. This is an ArcGIS problem. Solution is to open and close anther tool first and then open the desired tool again, then the dockpane will open.

4.3.1 *Add variant/measure*

Purpose:
The user can add a variant or measure directory. According to the name conventions a measure name should start 2 characters representing the waterbody, followed by an underscore, then maximum 20 characters/numbers for the actual name (no spaces and special tokens allowed), another underscore, ending with one characters and one number. For example *wl_dynrivbeheergeldprt_a1* refers to waterbody Waal, dynamisch rivierbeheer Gelderse poort, version a1.

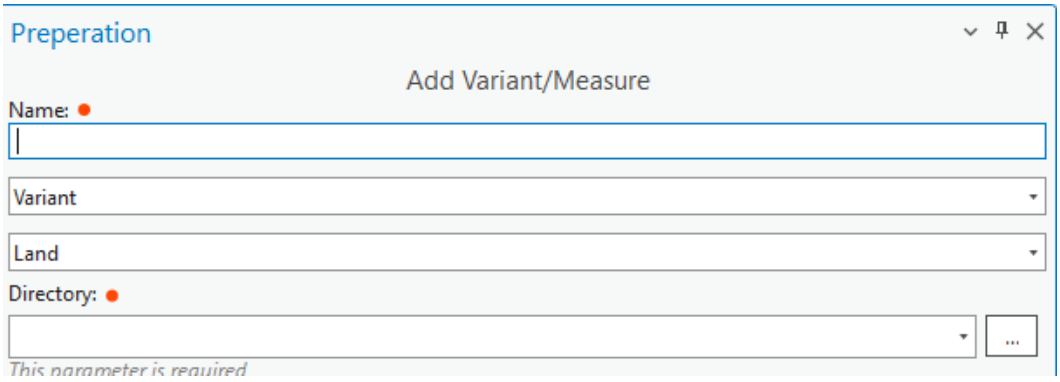


Figure 4.3.1-1 Dockpane

Explanation:
After adding an area, the user adds a variant directory. This will be located directly in Baseline/<area>. The user can choose if it is a "land" or "sea" database. There are some small differences between the two types. These are documented in the dataprotocol.
The user determines the name of the variant. This variant directory has the following structure:

[Data]
- [Source]
- [Import]
- [Export]

[Layers]

[Metainfo]
[Lists]
[Logs]
[Models]
[dflowFM]
[swan]
baseline.gdb

The File Geodatabase baseline.gdb contains the following Feature Datasets:

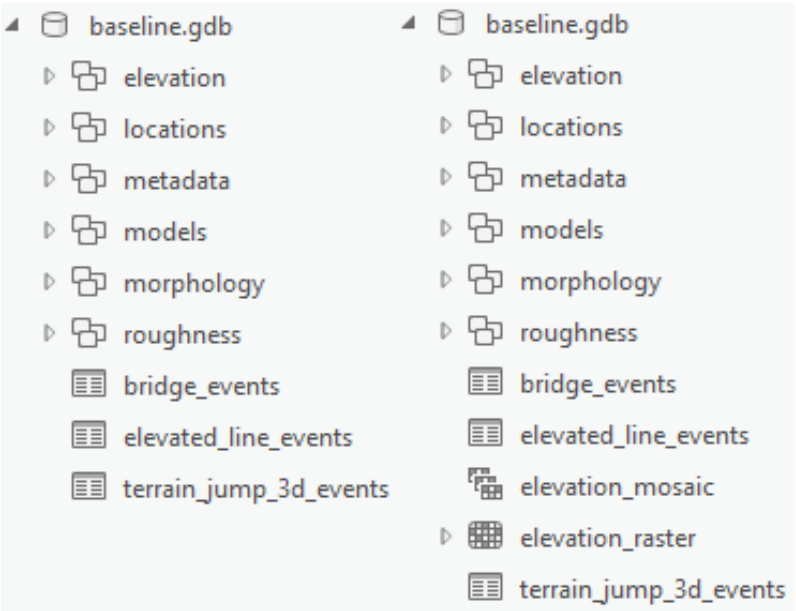


Figure 4.3.1-2 Variant dataset structure

Within these Feature Datasets the Feature Classes will be created.

When instead of a variant a measure is added a similar process is executed. However in that case the Feature Dataset "erase_polygons" is created with the corresponding Feature Classes. Also instead of variant.lyr the layerfile measure.lyr is loaded.

4.3.2 *3D Lines,(3D)Routes & Events to Lines & Points*

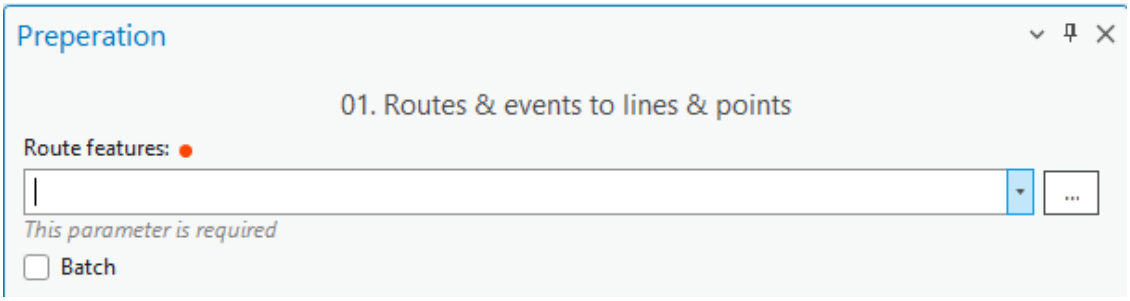


Figure 4.3.2-1 Dockpane

This tool converts (3D) routes and events or 3D lines to lines and points. Both attributes of feature class and table are preserved. The selected (3D) routes/3D lines might be an empty feature class or a feature class containing actual (3D) routes or 3D lines. In the feature dataset *elevation* a line- and point feature class are made (_lines and _points). In case of the conversion of bridges routes and events these lines and points will be placed in the feature dataset *locations.* After this conversion the original (3D) feature class and corresponding event table or 3D line feature class

are deleted. In this way the user is forced, after editing the lines and points, to do the backward conversion to (3D) routes and events or 3D lines to prevent version issues.
If only a single route feature class must be converted this feature class should be entered; when batch is checked a Baseline database folder should be selected and all present route feature classes will be converted to lines and points.
Before the conversion the current routes Feature classes and event tables are copied in \data\source\elevation.gdb including a timestamp as a backup.

4.3.3 *Position Points on Polylines*

The tool 'Position Points on Polylines' adds points on the (selected) lines of the line feature class. If a selection of lines is active only points are added on these lines, if no lines are selected, points are added to all lines. However, the selection only is recognized if the line Feature class is selected by using the dropdown arrow in the dockpane (1) and not when the line Feature class is selected by using the browser window (2), see Figure 4.3.1-1. If points are already situated on a line, the tool adds a vertex to the line if not already present. This is conform Baseline protocol.

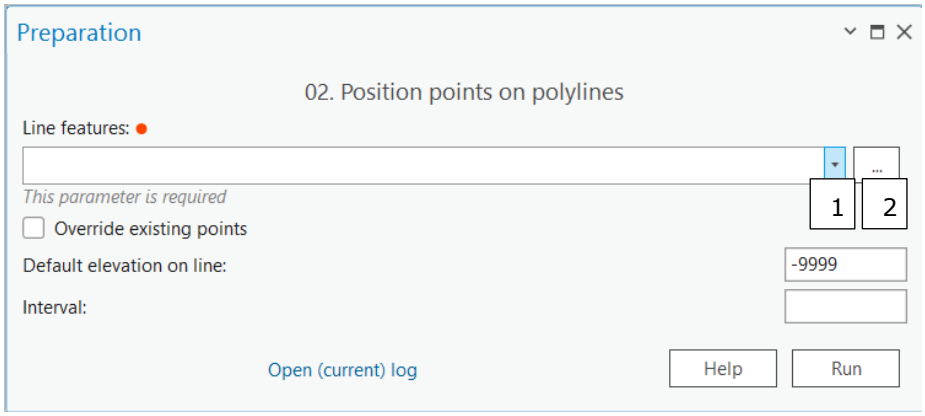


Figure 4.3.3-1 Dockpane

- When 'Overwrite existing points' is NOT checked lines already having points are skipped and these existing points (having the same NUMBER as the line) will be preserved. If the number of vertices of the line is smaller than the number of points on the line, extra vertices are added. If the line contains more vertices than points, extra points will be added, having the default 'standard crest elevation'. For these points, realistic elevation values must be defined with tool 3 (see paragraph 4.3.4).

  If checked, existing points are deleted and a complete rebuild is made.

- 'Default elevation on line' is the default crest elevation which is filled in the field ELEVATION_CREST. Default is -9999. So it can be easily detected if ELEVATION_CREST is already determined.

- The 'interval' determines on what distance extra points are added on the lines in between the existing vertices. No interval results in points only on existing vertices of the lines.

The default spatial tolerance in ArcGIS and Baseline is 0.001 m (1 mm). The default precision of the field MEASURE of events is 4 decimals. The location of an event along a route can vary from 0.000% to 100.0000%. Combining the 1 mm tolerance and the measure precision the maximum route length is 1000 meters. Longer routes might result in the coincidence of 2 events along the route. When longer routes are used it is recommended that the events on the routes have an interval of at least 1 cm in case of a 10-kilometre route and 10 cm in case of a 100-kilometre route.

4.3.4 *Position points on bridges*

The tool 'Position points on bridges' adds points on the (selected) lines of the line feature class of bridges. If in ArcMap a selection of lines is active only points are added on these lines, if no lines are selected, points are added to all lines. If points are already situated on a line, the tool adds a vertex to the line if not already present. This is conform Baseline protocol.

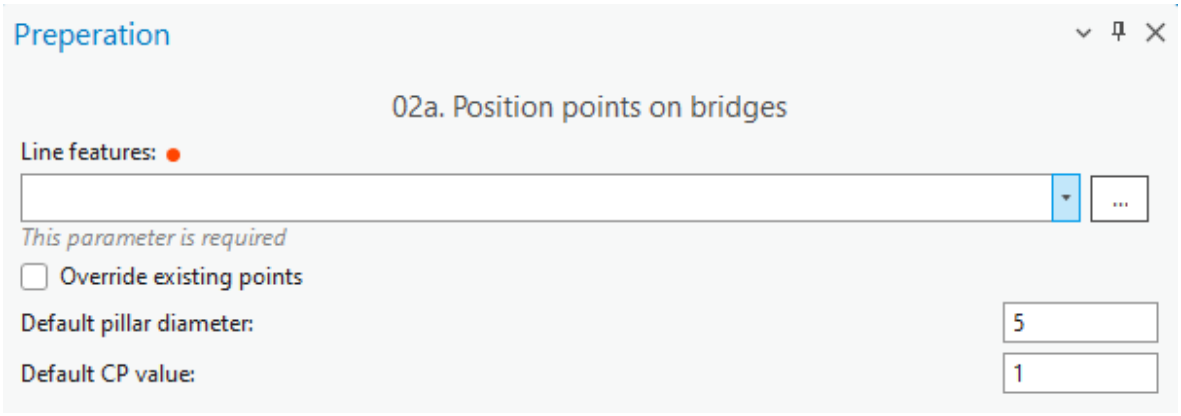


Figure 4.3.4-1 Dockpane

When 'Overwrite existing points' is NOT checked lines already having points are skipped and these existing points (having the same NUMBER as the line) will be preserved. If the number of vertices of the line is smaller than the number of points on the line, extra vertices are added. If the line contains more vertices than points, extra points will be added, having the default values for pilar diameter and CP value.

If checked existing points are deleted and a complete rebuild is made.

- 'Default pillar diameter' is the default diameter which is filled in. Default is 5. The default CP value is 1.

- When extra vertices are needed to decribe the start- and end-location of the bridge_route or when the bridge_route contains bends then these vertices are also converted to bridge_events. When there is no pillar at these locations the diameter should be -999. D-Hydro recognizes these locations as dummy pillars.

4.3.5 *Set crest elevation*

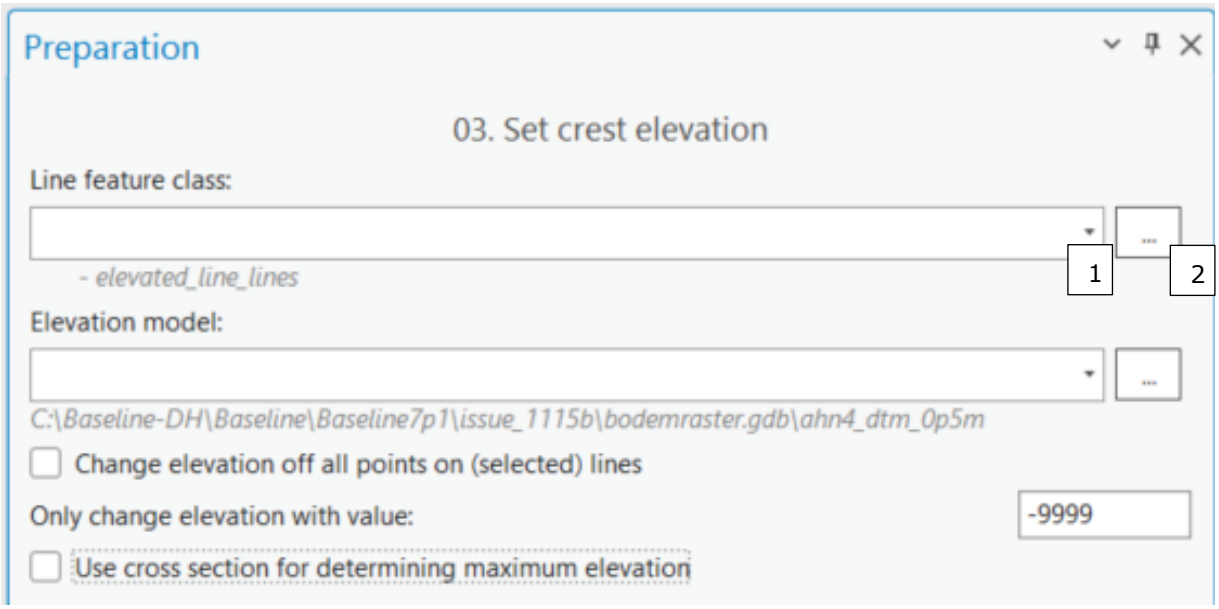


Figure 4.3.5-1 Dockpane

This tool fills in the (crest) elevation in the field ELEVATION_CREST for all points on the selected lines. The elevation is determined using an elevation model. Valid elevation models are TIN, Terrain or Raster (Esri grid, Erdas Imagine img, …). For this tool 3D Analyst is needed.
Lines might contain: a feature class, or a selected set of lines in the corresponding layer. If in ArcGIS Pro a selection of lines is active the crest elevation of points is added on these lines only. If no lines are selected, the crest elevation is added to all points. However, the selection only is recognized if the line Feature class is selected by using the dropdown arrow in the dockpane (1) and not when the line Feature class is selected by using the browser window (2), see Figure 4.3.5-1.

Change elevation of all (selected) points is unchecked by default. When unchecked 'Elevation to change' is activated. When a value is entered, only points with this elevation are updated (e.g. update of -9999 values, while keeping other elevation data).

For elevated_lines the above-mentioned options can be combined with the choice to determine the CREST_ELEVATION based on the maximum value in cross sections ('Use cross section for determining maximum elevation'). When this option is chosen also the CREST_WIDTH is derived from the cross sections. In this case the only valid Elevation model type is a raster.

4.3.6 *Set toe elevation*

This tool fills in the (toe) elevations in the fields ELEVATION_LEFT and ELEVATION_RIGHT for all points on the (selected) lines. The elevations are determined using an elevation model. Valid elevation models are TIN, Terrain or Raster (Esri grid, Erdas Imagine img, …). For this tool 3D Analyst is needed.
Lines might contain: a feature class, or a selected set of lines. If a selection of lines is active the toe elevations of points are added on these lines only. If no lines are selected, the toe elevations are added to all points. However, the selection only is recognized if the line Feature class is selected by using the dropdown arrow in the dockpane (1) and not when the line Feature class is selected by using the browser window (2), see Figure 4.3.5-1.

1 2

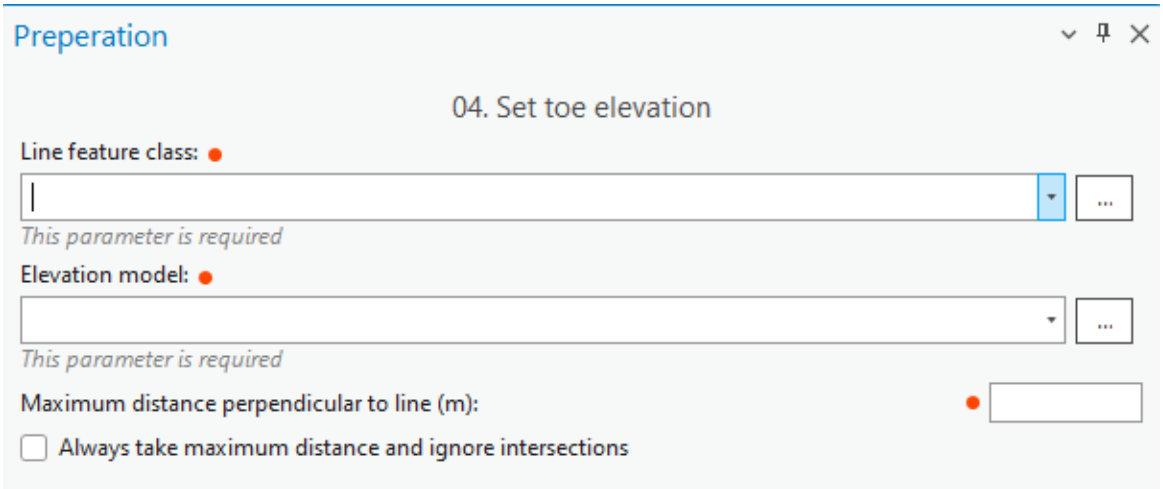


Figure 4.3.65-1 Dockpane

The determination of toe elevation is done in several steps:

- For each point on a line left and right of this line perpendicular lines are constructed using the maximum distance parameter, default 10 meters (see Figure 4.3.5-1). When a line changes direction both neighbouring line segment directions are considered in the construction of the perpendicular line.
- When the input elevation model is a TIN or Terrain, breaklines of this model are selected and intersected with the perpendicular lines. The intersection closest to the original line is interpreted as the toe location and the corresponding elevation is added to the point. If no breakline is found within the maximum distance the toe elevation is determined at the end of the perpendicular line.
- When the input elevation model is a Raster, the toe elevation is determined at the end of the perpendicular line (defined by the maximum distance).
- If the line Feature class is a *terrain_jump_lines*, toe elevations on both sides of the line are compared with the crest elevation, at the side with the lowest sill the toe elevation gets the same height as the crest elevation.

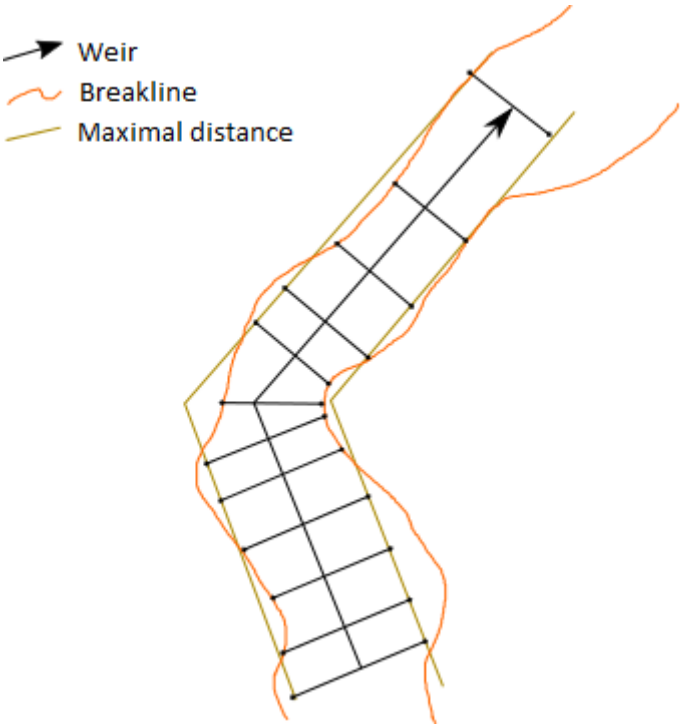


Figure 4.3.6-2 Toe elevation determination

4.3.8 *Add terrain_edge within open water bodies*

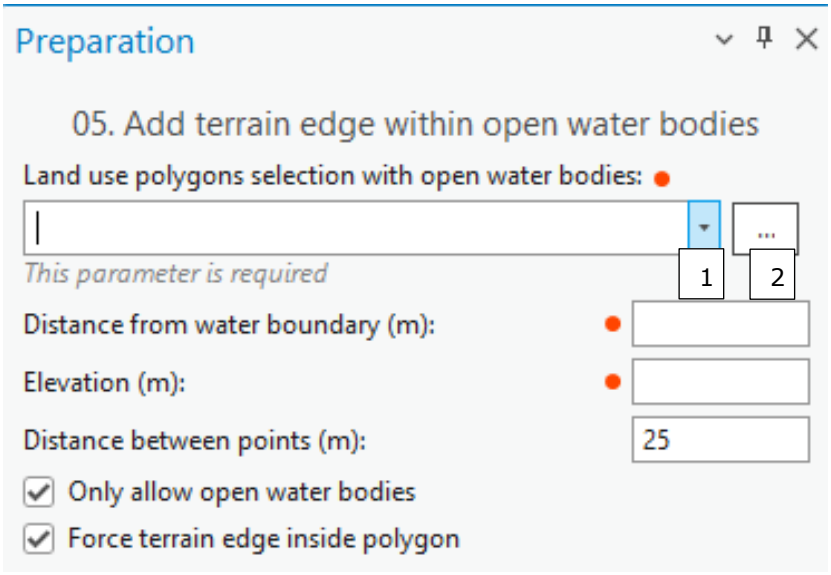


Figure 4.3.8-1 Dockpane

This tool constructs (bottom) terrain_edge_3d_lines in open water, usually this is done in open water where no bathymetry is available.

This tool only works when there is a selection made in land_use_polygons. However, the selection only is recognized if the Feature class is selected by using the dropdown arrow in the dockpane (1) and not when the Feature class is selected by using the browser window (2), see Figure 4.3.8-2.

The user has to fill in 5 additional parameters:

- The distance (m) from the terrain edge to the shoreline,
- The elevation of the terrain edge.
- Only allow open water bodies.
  When this is checked all selected polygons must have ROUGHNESS_CODE 102, 103, 104, 105, 106, 201, 302, 303, 304 or 305. The tool performs a check and will not execute if other polygons are selected. When unchecked: the tool will not validate the ROUGHNESS_CODE. All ROUGHNESS_CODEs are allowed.
- Force terrain edge polygon.
  When checked: the terrain edges are always located inside the selected polygon. When unchecked: the location of the terrain edges depend on the sign of the distance. A negative sign means a terrain edge inside the polygon, a positive sign means a terrain edge outside the polygon.

NB. If terrain edges are already present in the Feature class *terrain edges* the new constructed terrainedges are added to this Feature class.

4.3.9 *Lines and Points to Routes & Events*

After finishing all edits in the elevation or bridges, lines and points must be converted back to routes and events in order to be consistent with the Baseline data protocol.

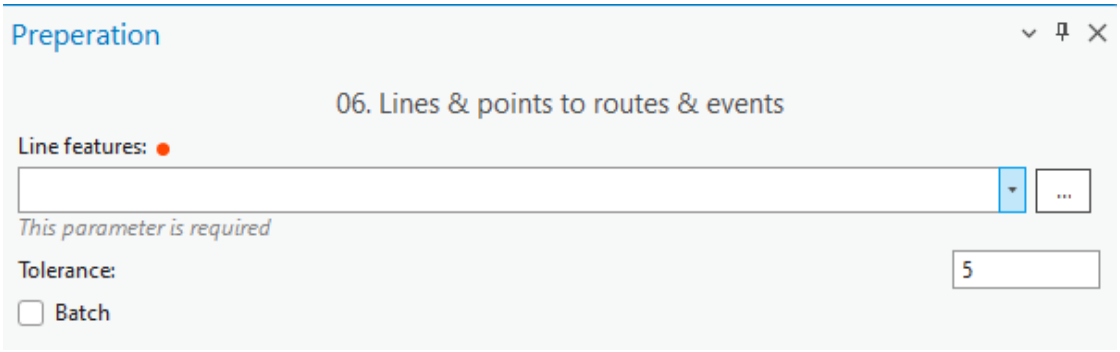


Figure 4.3.9-1 Dockpane

The tolerance is the distance that a point may be located from the line with the same NUMBER to be incorporated as event on the route. Points with the same NUMBER as the line but outside this tolerance distance will not be added to the event table. These points are preserved in the _points Feature class. After a succesfull conversion the line and point feature classes are deleted.
If only a single line feature class must be converted this feature class should be entered; when batch is checked a Baseline database folder should be selected and all present line and point feature classes will be converted to routes and events.
Before the conversion the current line and point Feature classes are copied in \data\source\elevation.gdb including a timestamp as a backup.

4.3.10 *Renumber routes and events*

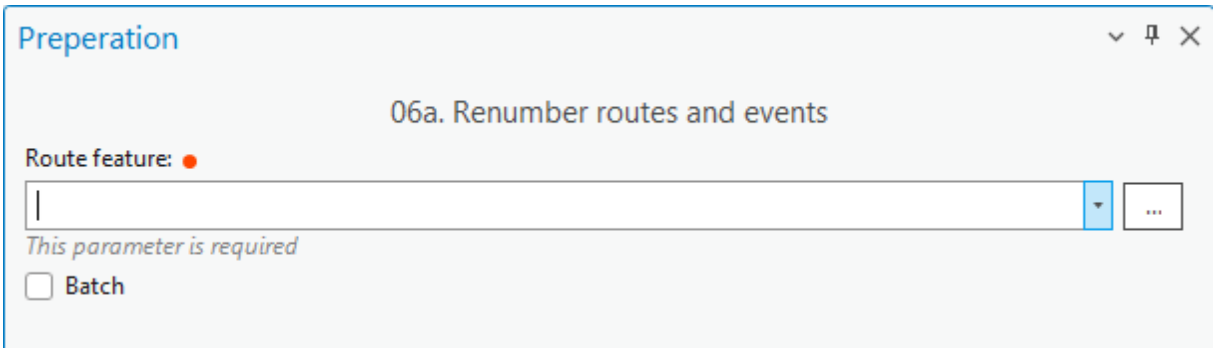


Figure 4.3.10-1 Dockpane

This tool alters the content of the NUMBER field in elevated_line_routes, terrain_jump_3d_routes and terrain_edge_3d_lines. Records (both routes and events) are renumbered from 1 to the number of records in the route Feature class.
If only a single route feature class must be renumbered this feature class should be entered; when batch is checked a folder containing a Baseline database should be selected and all present route feature classes (elevated_line_routes, terrain_jump_3d_routes) and terrain_edge_3d_lines will be renumbered.

Note: this fuction only works with measures, NOT with variants.

4.3.11 *Reposition points in a regular grid*

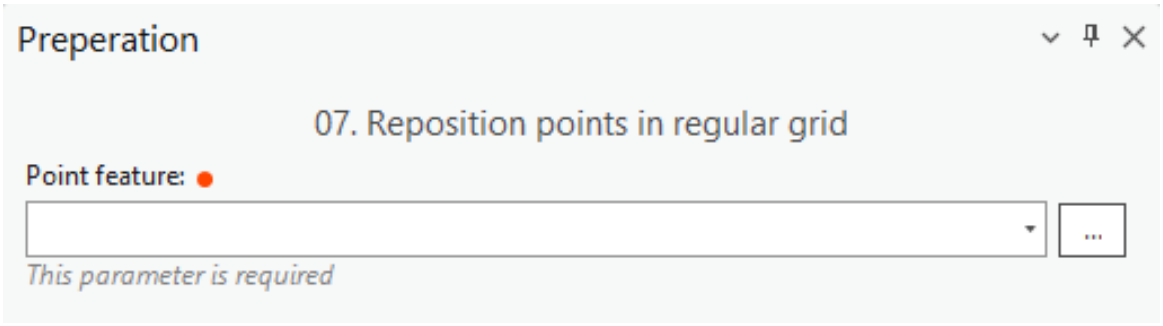


Figure 4.3.11-1 Dockpane

This tool replaces the former functionality of 'Bascon' and must be used to process elevation points derived from an ArcGIS grid or raster. These points have the property to have coordinates on a regular mesh. Using such points in a TIN or Terrain, ArcGIS/Baseline cannot produce a single solution triangulation which results in non reproducable results. This tool shifts all points with a maximal shift of 1 milimeter in a random direction. As a result, Baseline can make a single solution Terrain. NB. This tool should be used just once per point Feature class. Never use this tool on points in a variant database because then the reference data are altered.

The point dataset must be in a geodatabase; shapefiles are not supported.

4.3.12 *Create measure contour*

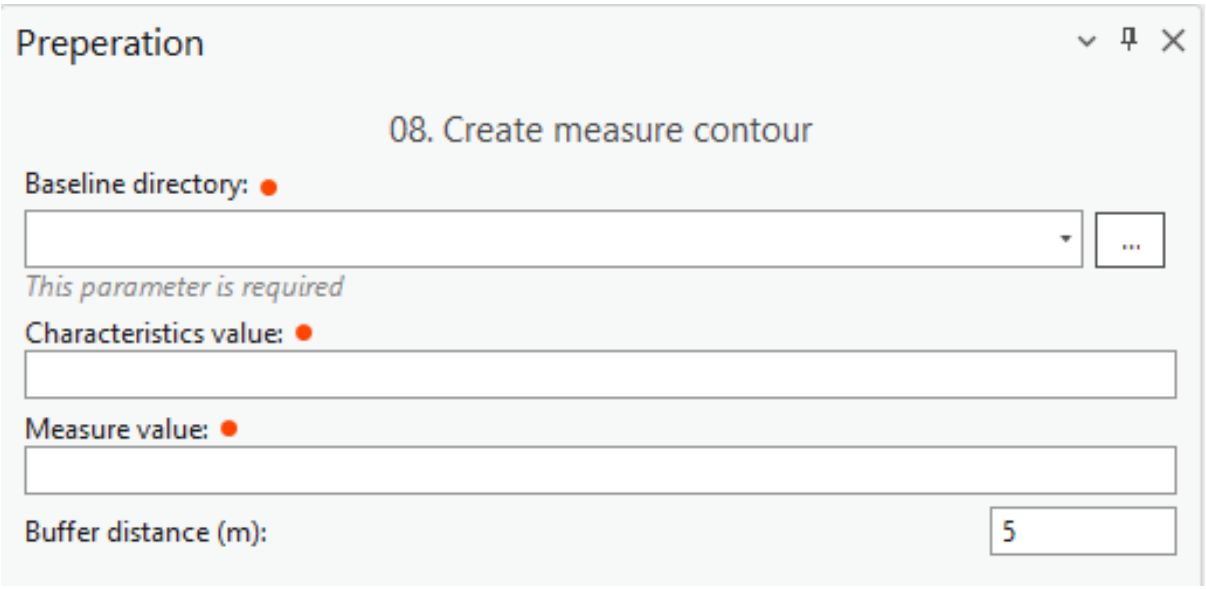


Figure 4.3.12-1 Dockpane

This tool constructs a hull polygon of all objects in the Baseline measure; both erase- and add-polygon-, polyline-, route- and point-Feature Classes are used. All polygons, polylines, routes and points are buffered with a variable distance (default 5 meter) and the resulting (generalised) polygons are added to *measure_contour_polygons.*
Four parameters must be entered:

- The Baseline measure (directory),
- The string to be filled in in the field CHARACTERISTICS (usually the name of the measure),
- The string to be filled in in the field MEASURE (this is the name of the measure),
- Buffer distance (m) to create around all measure objects representing lines and points (default 5m).

4.3.13 *Update all CHARACTERISTICS values in measure*

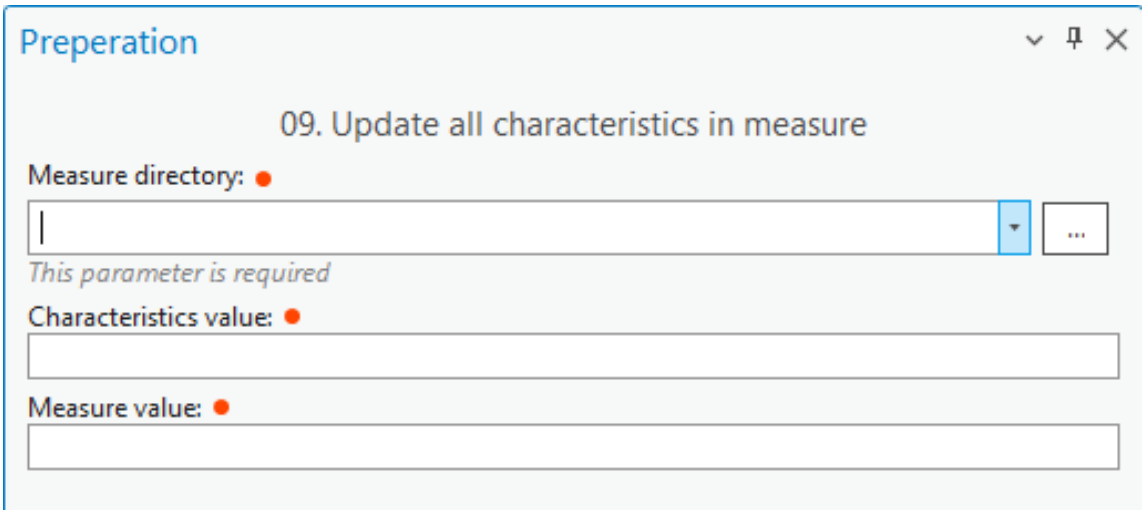


Figure 4.3.13-1 Dockpane

This tool fills the field CHARACTERISTICS in a Baseline measure for all objects.
Three parameters must be entered:

- The Baseline measure (directory),
- The string to be filled in in the field CHARACTERISTICS (usually the name of the measure),

- De string to be filled in in the field MEASURE (usually the name of the measure).

4.3.14 *Clean Baseline Database*

This tool removes all empty Feature classes within a Baseline database.

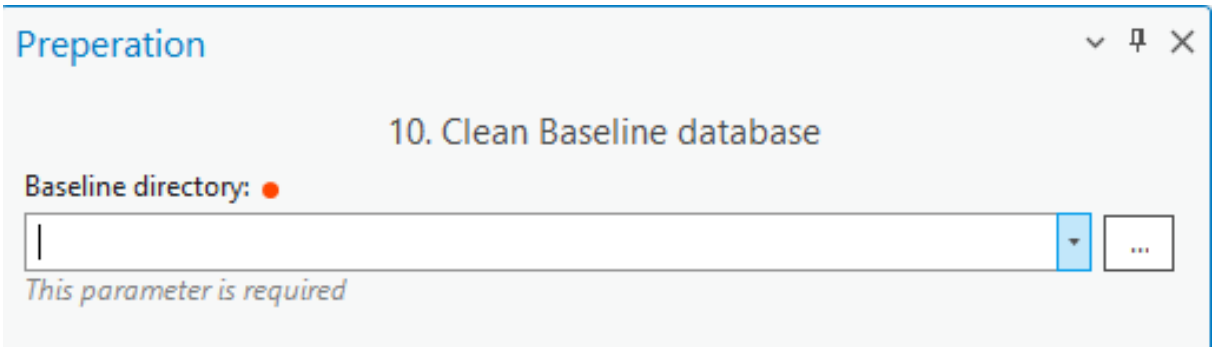


Figure 4.3.14-1 Dockpane

4.3.15 *Create erase- and appendlist*

This tool creates the files erase_list.txt and append_list.txt in the selected Baseline measure database. Only feature classes that are present and contain data are included in the list. Empty feature classes are ignored.

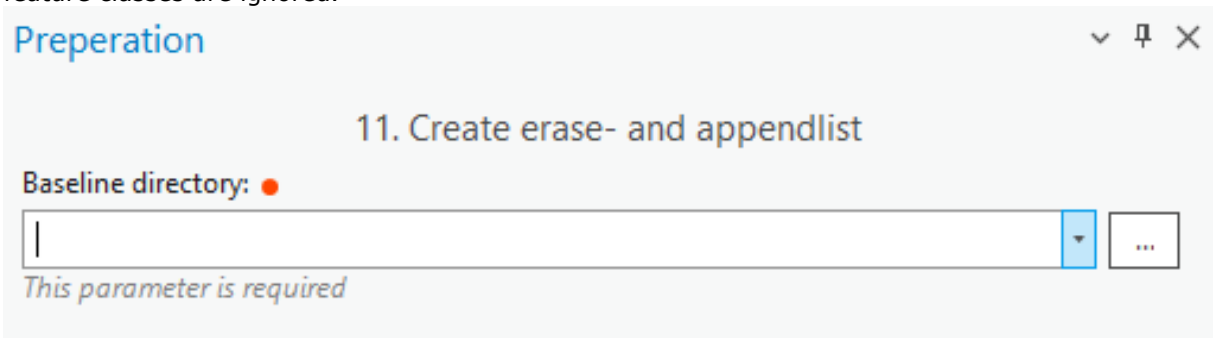


Figure 4.3.15-1 Dockpane

4.3.16 *Create land_use_polygons*

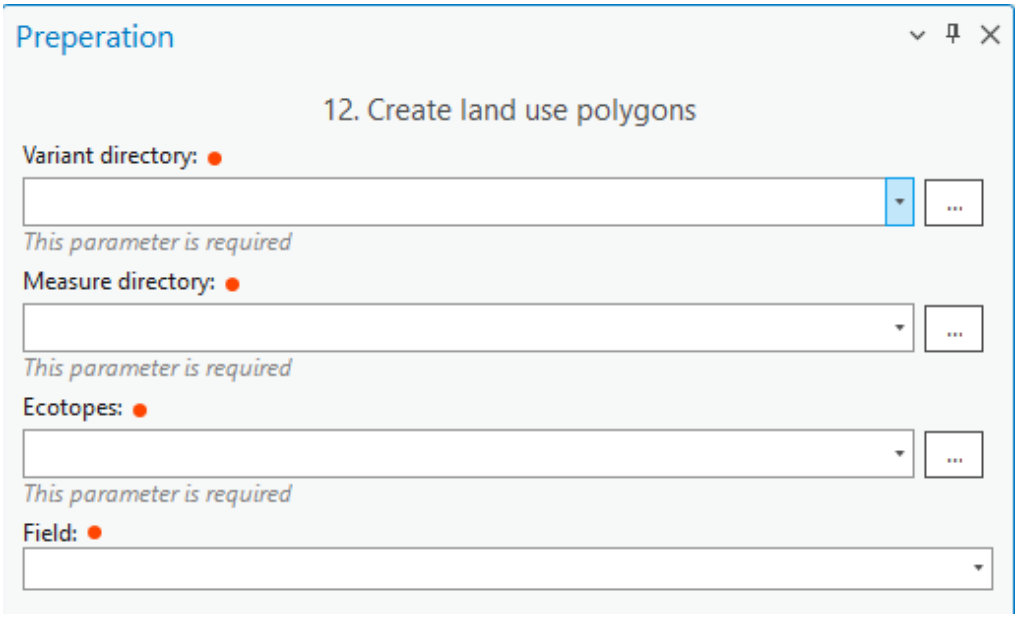


Figure 4.3.16-1 Dockpane

This tool creates *land_use_polygons* in an empty measure based on new ecotopes combined with the corresponding Baseline variant. This is the Baseline 6 duplicate of the Baseline 5 tool "Clustering ecopen". However, since Baseline 6 only contains one *land_use_polygons* file (instead of multiple types as was the case in Baseline 5) some information of the variant is needed to maintain data with a higher precision (like waterbodies). Also, calibrations polygons should be maintained.

Therefore, this tool selects all features from *land_use_polygons* (of the variant) with ROUGHNESS_CODE equal to 1 (buildings) or 102-106, 201, 302-305 (waterbodies). In addition, all features present in *calibration_section_input_polygons* are selected in *land_use_polygons*. All selected features are used to erase the ecotopes. The remaining ecotopes are transferred to the allocated measure with the corresponding ROUGHNESS_CODE.

The translation form ecotope code to ROUGHNESS_CODE is done using *landuse_key.csv* in python toolbox folder the install folder (eg. C:\Program Files (x86)\Deltares\Baseline7\Pythontools\Scripts\BaselineInputModule). It is recommended to check and eventually update this file when new ecotope codes are introduced.

Four parameters must be entered:

- The Baseline variant (directory),
- The new ecotopes (shapefile/ feature class),
- Field in the new ecotopes files containing the ecotope codes,
- The target (empty) Baseline measure (directory).

4.3.17 *Check a variant/measure:*

Purpose:
This function is used to perform two validations of a variant or measure:

1. Validation if a variant/measure complies with the protocol;
2. Validation of the content of the variant/measure.

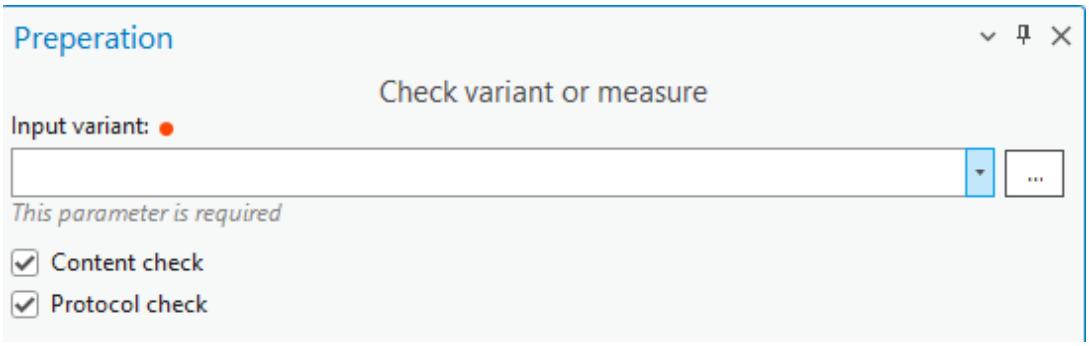


Figure 4.3.17-1 Dockpane

Explanation:
With this function it is possible to perform two checks on variants and measures. Both the protocol and content log file will be cleared before running the respective check.

Check Protocol
The selected variant or measure is compared with the templates of Baseline. The result of this validation is saved in ProtocolCheck.log. When this validation is successful, the tool succeeds and the status.json file is updated to indicate that the protocol check has succeeded (the last line of the log is still "Protocol succeeded" to make it compatible with Baseline 6). Attachement B presents the performed tests. *Note: The domains of the field TYPE are not checked due to a difference in spelling in certain Baseline 6 datasets compared to the templates of Baseline 7.*

Check Content
The selected variant or measure is checked on content. All fields in the Baseline Database are revised where wrong or strange values are reported. The result of this validation is saved in ContentCheck.log in metainfo\logs. Attachement B presents the performed tests.

Note: These validations can take a very long time, especially checking if there are double points with deviating heights in the various Feature Classes.

4.4 Tools

In this paragraph the following functions are described:

1. MapTool 3D Line vertices
2. Assimilation of measure
3. Merge Land and Sea
4. Batch

4.4.1 *Identify 3D line vertices*

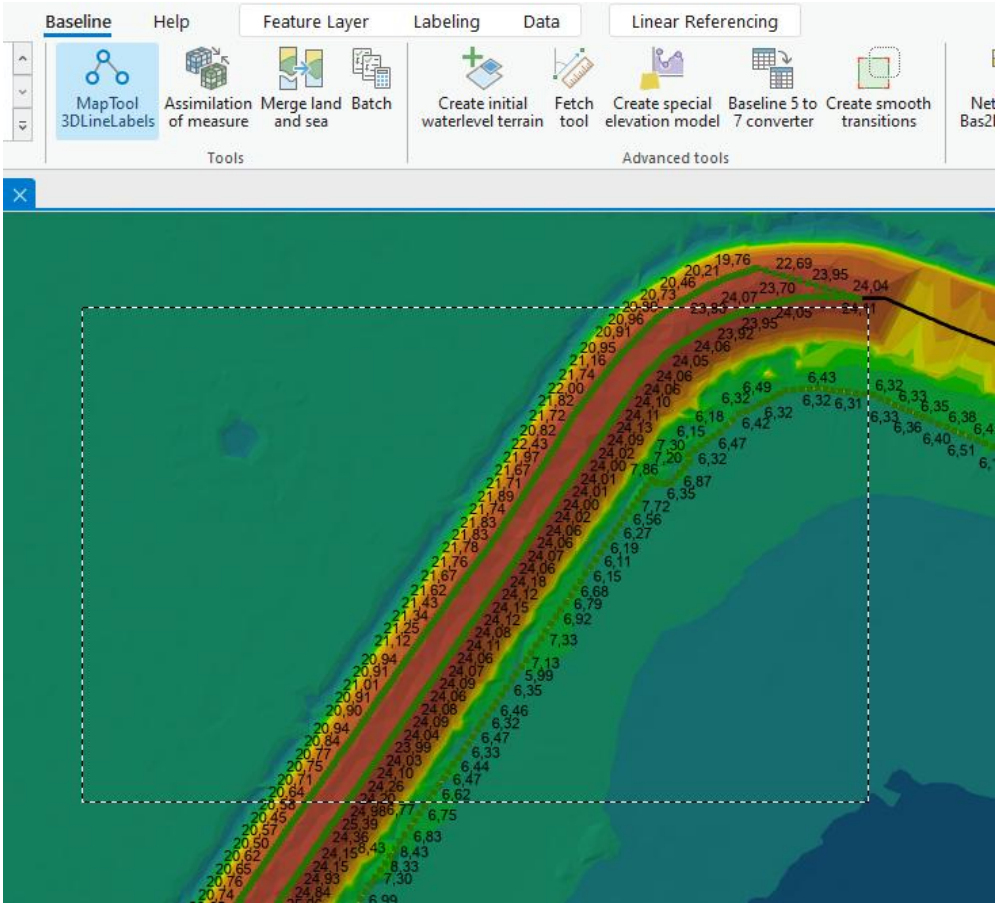


Figure 4.4.1-1 Tool usecase example

Goal:
With this function 3D lines in a Baseline database can be examined on their Z-value.

The function "Identify 3D line vertices" uses the active variant and only works for terrain_edge_3d_lines and terrain_jump_3d_routes. After activating the button a new temporary layer "vertexCoordinates" is added to the TOC and the cursor changes into a cross. Using this cross a box can be drawn, this will reveal the vertices of all visible 3D lines overlapping this box and show the Z-coordinate. More boxes can be drawn subsequently by activating the button.
Do not remove this temporary layer "vertexCoordinates", after removal this function will no longer work.

4.4.2 *Assimilate measures*

Purpose:
When performing "Assimilate measures" Baseline measures will be included in the active variant. These measures are defined in measure_list.txt in the folder metainfo/lists of the active variant.

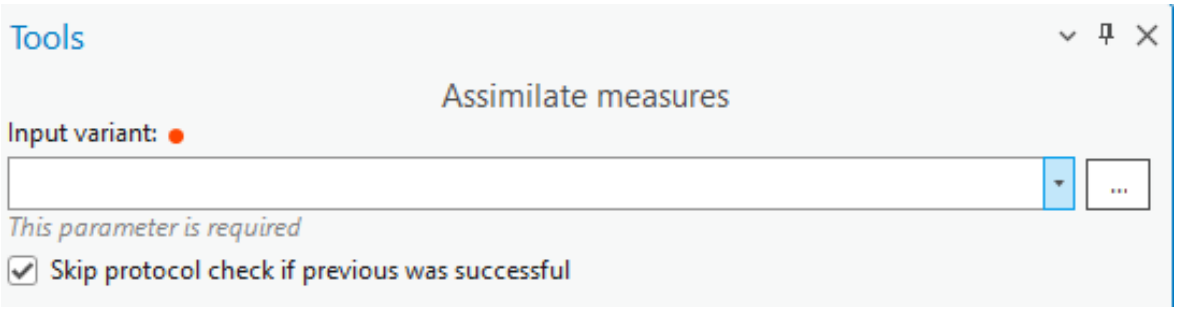


Figure 4.4.2-1 Dockpane

The measure_list.txt defines the location of the Baseline measures which should be included in the active variant. Each measure contains a folder metainfo\lists. This folder contains two files: append_list and erase_list. The erase_list.txt contains a list of polygons, indicating which areas should be erased from the active variant. Such a polygon is always called erase_<Feature Class>, where <Feature Class> is a Feature Class in the Baseline database in accordance with the Baseline protocol. Append_list.txt contains Feature Classes which should be included in the active variant after completing the erase_list.txt.

An exact description of all lists and Feature Classes is included in the Dataprotocol Baseline 7.

Explanation:
User checks on the shown display if the correct active variant is set before clicking on "OK". After that the following validations are performed:

1. Presence of measure_list.txt in the active variant
2. Presence of the measure called in measure_list.txt
3. Presence of append- and eraselist.txt in metainfo/lists
4. Presence of all Feature Classes in measure and variant called in erase- and appendlist.txt.
5. Are the variant and measure listed in accordance with the protocol.

If any of the above points is not valid, the tool will not run properly

A direct reference to a log-file is given. The file is named Assimilate.log and is located in metainfo\Logs of the active variant.

When all points are valid the assimilation of measures can be executed or aborted. When there is an error during the assimilation an error message is shown, which is also included in the log-file.

Action 5 includes checking the active variant and all included measures whether they meet all protocol requirements. This can take some time. To prevent this the user can check the variant and all measures before this function is performed. This is done by a protocol check. If this protocol check succeeded (as a separate function, or before the assimilation), a status.json file is created/edited to record the protocol result (true if succeeded, false if not). By checking checkbox "Skip when previous check was successful" these checks are skipped when a the last recorded protocol result succeeded according to the status.json file (and if it was not edited after the last protocol check). When this is not the case, the protocol check will be executed before the function. The measures are always checked, and the function will not run if this check fails.

4.4.3 *Merge Land and Sea*

Purpose:
This function is used to merge a "land" database and a "sea" database.

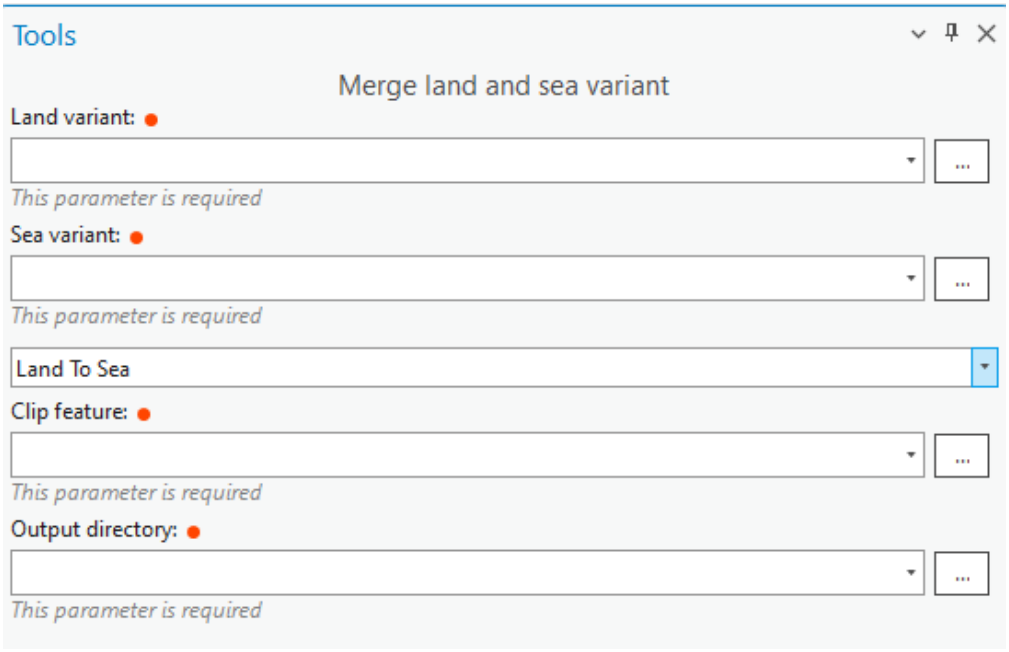


Figure 4.4.3-1 Dockpane

Explanation:
This function is used to merge a "land" database and a "sea" database. For instance the Rijn-Maasmonding model with the Northsea. When this is done it can be treated as a regular model.

- First a "land" and "sea" database must be selected. These databases are available at Deltares. There is a slight overlap (approximately 150 meters) between these databases in order to prevent nodata in the elevation_raster at the connection of both databases.
- Then a choice should be made which type of dataset the results should be. In other words: should the "land" dataset be converted to "sea" or the other way around.
- Moreover a clip shape should be selected defining the area of the target database which is partly overlapping with Baseline-Land and partly with Baseline-Sea.
- The output folder that will contain the merged variant also needs to be selected. The log is also placed at this location (not in the metainfo folder).
- The default transformation is 'Amersfoort_To_WGS_1984_4X'. The geographic_transformations.pdf can be found in the Baseline install folder.

The vertical reference for Baseline-Land is NAP and for Baseline-Sea MSL (Mean sea Level). In Baseline the assumption is done that NAP is equal to MSL, therefore **no vertical transformation** is done when merging Land and Sea databases. Data from Rijkswaterstaat CIV show that in the coastal zone, where Baseline data are transferred from land to sea or vice versa, MSL and NAP differ only slightly (eg. less then 5 cm).

The function generates the following results, however, only the merged output variant will be kept (in case of an error, it is possible that some of the in between results are kept):

1. Land_clip (the clipped land variant within the clip feature)
2. Sea_clip (the clipped sea variant within the clip feature)
3. Projected_<SurfaceType> (the variant projected to the output surface type)
4. Projected_<SurfaceType>_measure (the variant from point 3 transformed to a measure)
5. <OutputPath> (merged output variant, where result 4 is assimilated in result 1 or 2, depending on the output surface type) (the main result)

When a land database is converted to a sea variant the elevation_model_terrain is converted to 5x5m elevation_raster. In the resulting merged sea variant this raster is named elevation_raster_land and this raster is added to the mosaic dataset named elevation_mosaic.

When a sea database is converted to a land variant the elevation_raster is converted to points. In the resulting merged land variant these points are added to the elevation_model_terrain.

4.4.4 *Batch*

Purpose:
The purpose of the batch function is to perform tasks in Baseline without the intervention of a person. This involves all functions available in Baseline. This allows a user to create inputfiles for hydrodynamic models from the beginning Baseline database through a number of tasks without the user needs to run the individual functions. These functions are performed automatically.

Explanation:
When starting the Batch function the following display will appear:

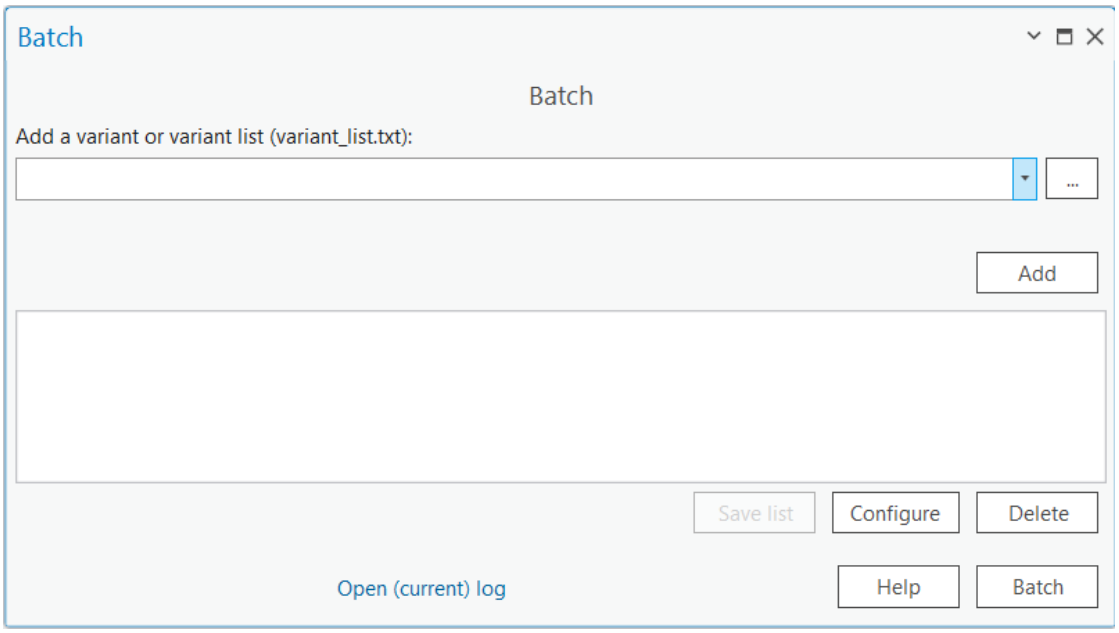


Figure 4.4.4-1 Dockpane of the variant selection

It is possible to add multiple variants with the "Add" button. If an area directory with a variant_list.txt is selected, it will add all the variants in that list, if a variant is selected, it will only add that variant. In any other case, it will do nothing.

When selecting a variant the properties of the batch can be set with "Configure", or removed from the list with the "Delete".

Before it is possible to run the batch a number batch list should be created (as described in the Dataprotocol (lists/batch_list.json)). This list should be created with the "Configure" button. This opens a dockpane where the user can enable or disable functions and edit the parameters if applicable, see figure 4.4.4-2. When the "Save" button is pressed in this panel, the batch_list is created or updated for the selected variant.

The 'Use Grid netCDF to Bas2FM parameters' button fills in the values of the Grid netCDF to Bas2FM tool in to the respective fields of the 'Conversion to D-Flow Flexible Mesh' tool.

When using the Assimilate measure, it is assumed the measure_list.txt already is created. The batch will not create any other lists besides the required batch_list.json.

From the panel "Configure" can be clicked on "Set Parameters" this button will show the parameters for the function and will allow the user to set the parameters for that function. The

Conversion to D-Flow FM parameters contain a button to fill in the values of the Convert Grid to Bas2FM parameters.

If the variant does not contain a batch file, the parameters always need to be set for the 'Convert Grid netCDF to Bas2FM' and 'Conversion to D-Flow Flexible Mesh' functions, otherwise the batch will not start. If it contains a batch file, it will use the values of those parameters by default.

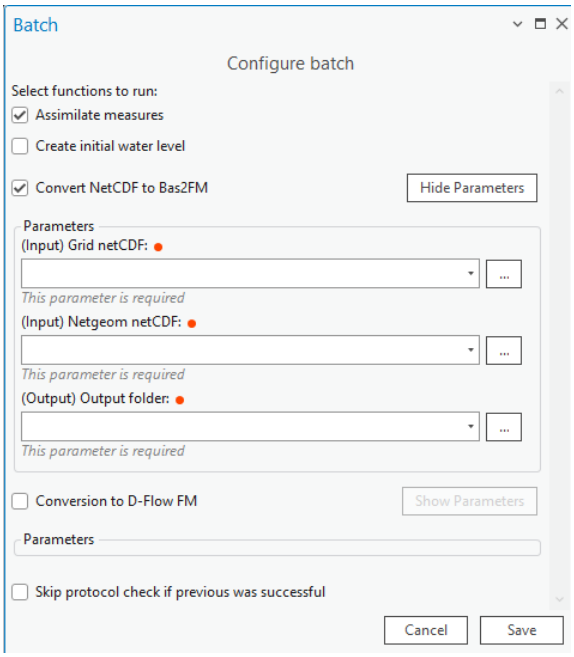


Figure 4.4.4-2 Dockpane of the function selection

The log files can be found in the specific variant, per function (the batch.log shows a summery of the batch process for the specific variant), those logs will contain any errors if they occurred. If any (serious) error takes place while running the batch, the batch operation is stopped for that specific variant. It will (try to) continue with the next variant.

When the user runs the batch, all the variants and batch_files will be validated first, a message will be returned if a validation fails and the batch will not start. If the batch fails due to an error during the batch, the batch is skipped for that variant.

Note: *The batch needs to be configured again if the variant was previously used in Baseline 6, the batch will not copy the values of the old batch_list.txt*

4.5 Advanced Tools

4.5.1 *Create Initial Waterlevel terrain*

Introduction
The *Create Initial Waterlevel terrain* is a tool to create the Initial Waterlevel Conditions for a simulation with D-Flow Flexible Mesh (D-Flow FM). The tool has the same objective as the previously used function 'waqini'. However, this tool does not require a hydrodynamic model calculation. The (initial) water level is completely derived using the land/water boundary as stored in the Baseline database. The function does not work with sea-variants.

General method
The *initial_waterlevel_terrain* is derived from the *elevation_model_terrain* and information from *sections* en *landuse_polygons*.

From *sections* both section 1 and 2 polygons (river and banks) are selected together with the 'wet' *landuse_polygons* (roughness codes[1] 104, 105, 106, 201, 302, 303, 304). These polygons are merged and dissolved and the result is converted to a polyline, the 'water boundary'. The elevation of all vertices from this water boundary is determined using the *elevation_model_terrain*. This elevation represents the 'year average waterlevel' of the river and lakes in the floodplain (or more accurate the water boundary at the time the source data of these Feature classes were collected). The resulting polylineZ is not a flat line so some smoothing/averaging is done:

- The elevation of the water boundary of the river (section 1 and 2) is smoothed using a smoothing routine which use a window average pointlist of 500 vertices. Before this smoothing is applied all vertices of the waterboundary within waterbodies are deleted. In this way no waterlevel is determined within connected waterbodies.
- The next step is to determine the elevation of the waterboundary of waterbodies which are not part of section 1 and 2. Therefore three separate routines are processed.
    - First a selection is made of isolated waterbodies, waterbodies with a single connection to the river or waterbodies with multiple connections to the river.
    - When waterbodies with multiple connections to the river are found these boundaries are used to update the waterboundary of section 1 and 2. With this routines multiple connected waterbodies are embedded into the smoothing routine of de waterlevel in the river.
    - When a waterbody is just connected to the river at one location the waterlevel of the start and end vertices of the waterboundary is retrieved at the connection. The highest value of these waterlevels is used for the waterlevel in the waterbody.
    - When a waterbody is not connected to the river the elevation of all vertices of isolated water bodies in the floodplain is averaged (median) to retrieve a single waterlevel.
- After the waterlevel within the model area is determined the waterlevel at the open water boundaries are determined. At these locations no water boundary is available so at these locations no accurate waterlevel can be derived form the *elevation_model_terrain.* To make the best possible assumption a boundary is determined. For this boundary a subdivision is made for boundaries in a river or at sea. This is done based on the length of the boundary. Assumed is that when a boundary is less than 1000 meter it is a boundary in a river if longer it is a boundary at sea. A boundary at sea is given a value of 0 meter. For a boundary in a river the waterlevel is determined based on the elevation on the start en end of the boundary.

Using this ingredients the *water_level_terrain* is built. The terrain is still not a smooth water level, so after 'Conversion to D-Flow Flexible Mesh' a simulation should be run in order to derive a stable initial water level field. Because the *water_level_terrain* is based on the model geometry rather than on a simulation there is no direct relation between this water level terrain and a specific discharge. Also because of this shortcomings in the model geometry can result in an incorrect water_level_terrain.

[1] Roughness code 102 is not taken into account as requested by Rijkswaterstaat

Considering these knowledge still the *water_level_terrain* is a good starting point to derive a real initial state for a D-Flow FM model.

Comparison with waqini
Unlike waqini there is no real boundary condition associated with this water level terrain model, it serves as a starting point for simulations and approaches a bankfull situation. The accuracy of this waterlevel terrain model depends on the accuracy of the position of the land/water boundary in relation to the terrain elevation model, both stored in the Baseline database. As a result of inconsistencies between these two data sources the water level terrain model has no flat or slightly declining surface; it contains pits and spikes of water. A simulation is needed to level out these inconsistencies but the simulation needs less time to become stable because all water bodies are filled up already.

Environment
The tool is part of the Baseline 7 software package and cannot be used without Baseline 7 being installed. The tool runs on systems with ArcGIS Pro 3.0 and higher.

Input
The tool uses a Baseline variant as input, additionally a suffix should be given for the name of the waterlevel terain model.

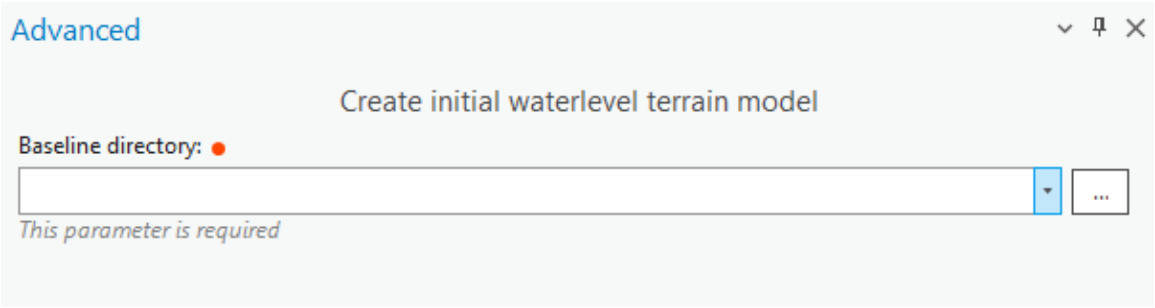


Figure 4.5.1-1 Dockpane

In the Baseline variant the following data should be present:

- Sections_polygons
- Roughness_polygons
- Elevation_model_terrain

Output
The tool generates the following data in de Feature dataset Models:

- Initial_water_level_terrain, the actual waterlevel terrain model
- Initial_water_level_input1, the elevation of the boundaryline of connected waterbodies
- Initial_water_level_input2, the elevation of the boundaryline section1/2 –section 3
- Initial_water_level_input3, the elevation of the open waterboundaries
- Initial_water_level_input4, the outer boundary of the waterlevel terrain model

Points of attention
The operation of the tool is based on the elevation in the Elevation_model_terrain at the boundary of waterbodies; the boundary wet/dry. This boundary is present in Baseline databases describing the river domain of the Netherlands, eg. Rhine and Meuse. In models describing both river and sea domain (like the Rhine-Meuse estuary and the Southwestern Delta) in the sea part of the domain the boundary wet/dry is not present at the western borders of the domain. As a result no

waterlevel can be determined from the Baseline data in this part of the domain. The waterlevel at the sea is therefore set on 0 meter.

Also, when there is a discrepancy in the geometry of *sections* and the *elevation_model_terrain* (eg. a *terrain_edge_3d_line* representing a shoreline in *elevation_model_terrain* is not situated at the boundary between section 1 or 2 and 3) the resulting *initial_water_level_terrain* will not show a smooth surface.

This tool does not work on Baseline sea variants.

4.5.2 *Fetch tool*

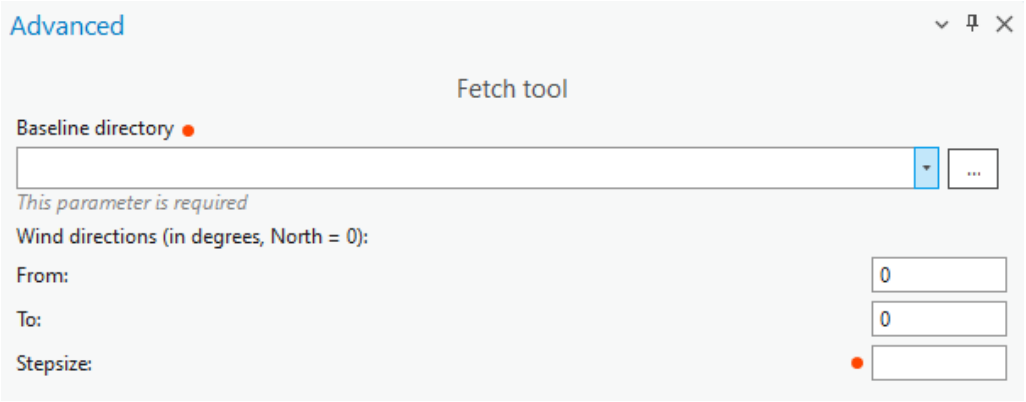


Figure 4.5.2-1 Dockpane

The Fetch tool is a tool that enables the user to calculate the so-called *fetch* at a location.

The fetch is the distance over which the wind can blow to a certain location. This distance is a parameter for the calculation of wave height. The fetch is restricted by dykes and flood free areas.

The fetch is calculated by the following method (Technische Adviescommissie voor de Waterkeringen, 1985).

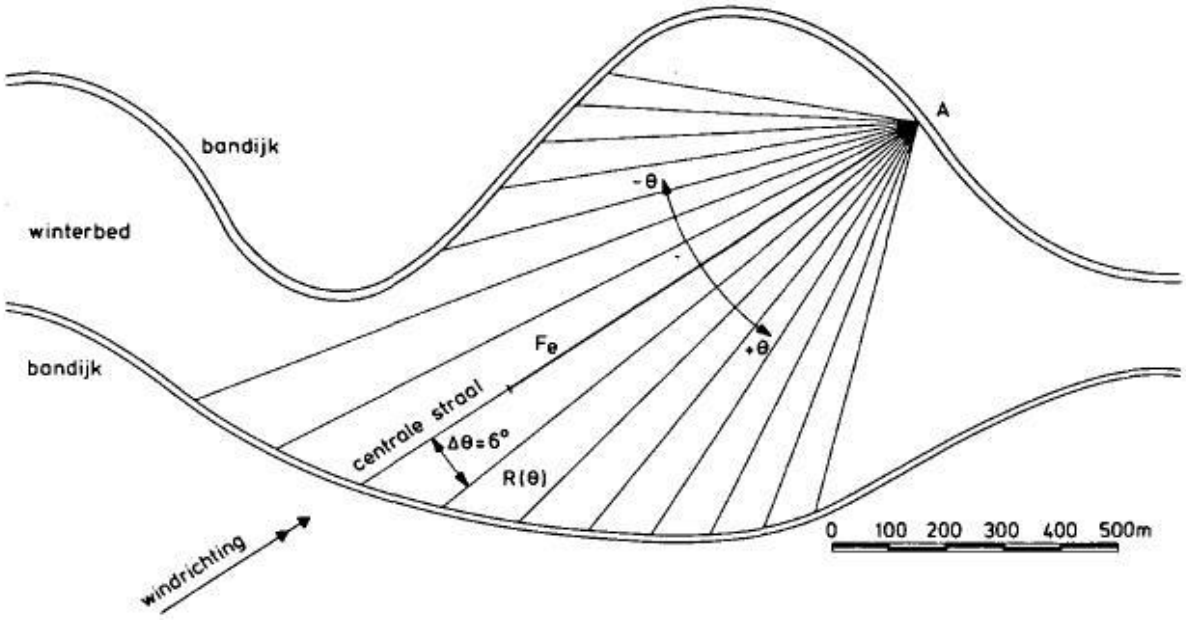


Figure 4.5.2-2 Fetch calculations

For every wind direction an array of fetch transects is created in steps of 6° from the wind and a maximum angle of 42° on both sides. This gives a total of 15 fetch transects: one in the wind direction and seven on either side.
Using the angle and length of the transects, the effective fetch is calculated with the following formula:

$$F_e = \frac{\sum R(\theta)\cos^2(\theta)}{\sum \cos\theta}$$

Where F_e is the effective fetch, R is the length of the transect and θ is the angle to the wind direction.

An example is showed in the following table (taken from (Technische Adviescommissie voor de Waterkeringen, 1985)).

θ [graden]	$\cos(\theta)$	$\cos^2(\theta)$	$R(\theta)$ [m]	$R(\theta)\cos^2(\theta)$
-42	0.743	0.552	520	287
-36	0.809	0.654	570	373
-30	0.866	0.750	640	480
-24	0.914	0.835	720	601
-18	0.951	0.904	830	750
-12	0.978	0.956	1340	1281
-6	0.995	0.990	1240	1228
0	1.000	1.000	1140	1140
6	0.995	0.990	1050	1040
12	0.978	0.956	980	937
18	0.951	0.904	920	832
24	0.914	0.835	880	735
30	0.866	0.750	830	623
36	0.809	0.654	780	510
42	0.743	0.552	730	403
$\sum\cos(\theta)=13.512$		$\sum R(\theta)\cos^2(\theta)=11220$		

De effectieve strijklengte, Fe volgt uit:

$$F_e = \frac{\sum R(\theta)\cos^2(\theta)}{\sum \cos(\theta)}$$

$$F_e = \frac{11220}{13.512} = 830 \text{ m}$$

Figure 4.5.2-3 Fetch table (Technische Adviescommissie voor de Waterkeringen, 1985)

4.5.3 *Create Special Elevation model*

Introduction
The Special elevation model tool is built to provide functionality for Baseline users to build custom elevation models with different elevation data as input, including the ones that are not available in the standard Baseline application (weirs and groynes).

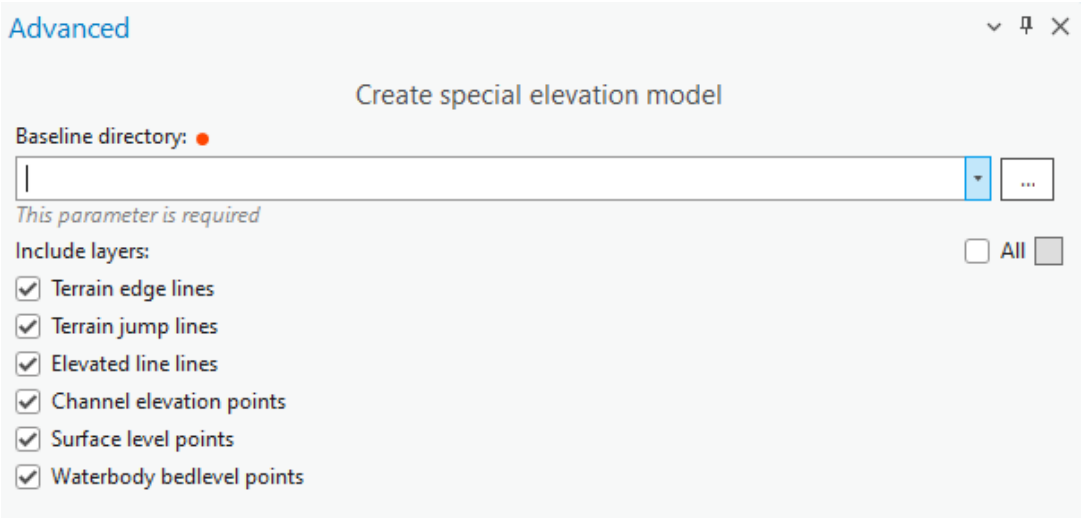


Figure 4.5.3-1 Dockpane

Environment
The tool is part of the Baseline 7 software package and cannot be used without Baseline 7 being installed. Therefore, the same systems requirements as for Baseline 7 have to be met.

Input
The tool uses a Baseline variant as input, the function does not work with sea-variants. The user chooses which elevation data to include in the resulting model.

Output
The tool generates a File geodatabase in the /data/source folder, named special_elevationmodel_<<datum>>_<<time>>.gdb. In this database, a feature dataset hoogtemodel is stored with 3D lines, an area file (dissolved secties) and the elevation model.

It is decided not to include these datasets in the general baseline.gdb database, as these files do not comply with the Baseline dataprotocol.

Note that there can be some, local differences between the Baseline elevation model and the model generated with this tool. Effort has been put into optimizing the method to resemble the standard Baseline method as close as possible. Further optimizations are planned.

For conversion to hydrodynamic models, the original Baseline data and models have to be used. The elevation model resulting from running this tool is only to be used for visual inspection.

This tool does not work on Baseline sea variants.

4.5.4 *Baseline 5 to Baseline 7 converter*

The Converter translates Baseline 5 references, variants and measures to Baseline 7 (the output will also work with Baseline 6). References and variants can be converted on their own while a

measure is converted with respect to underlying measures and the underlying reference. In the last section of this paragraph, more details are given on the converter.

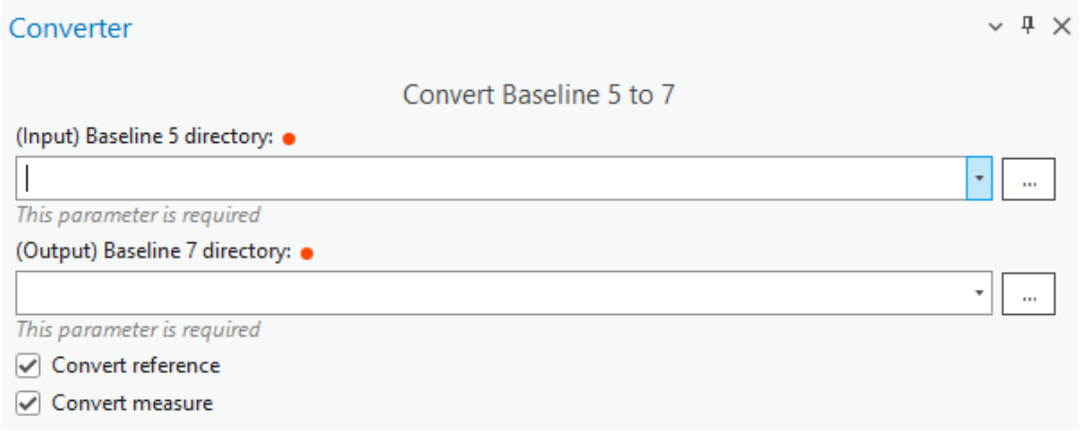


Figure 4.5.4-1 Dockpane

Step 1: Prepare your Baseline 5 data
There are some important things to consider before using the converter

- Make sure all Baseline 5 data is correct and valid. Data which does not comply with the Baseline 5 data protocol can give unexpected results when converting to Baseline 7.
- Make sure there is enough disc space available to store the Baseline 7 data. The Baseline 7 data will use about the same amount of disc space as Baseline 5.
- It is advised to make a backup of all Baseline 5 data, before using the converter.

Step 2: Run the converter

1. Open the 'Baseline 7' tool. The dialogue box of the converter opens

Note: It is strongly recommended to run the converter in ArcGIS Pro while no variants or measures are loaded in other projects (in ArcMap/ArcGIS Pro). Otherwise, problems with file-locking are expected to occur.

2. The first checkbox 'Convert Baseline 5 reference to Baseline 7 (optional)' is used when a Baseline reference or variant must be converted.

3. The second checkbox 'Convert Baseline 5 measures to Baseline 7 (optional)' is used when Baseline has to convert one or more measures.

4. In the field 'Baseline 5 reference directory', browse to an input folder which holds the Baseline 5 reference. The measure-list in this reference will be used to locate the Baseline 5 measures which need to be converted. **NOTE: the convertor only performs well when relative pathnames are used in this measure-list.**

5. In the field 'Baseline 7 directory', browse to a folder in which your Baseline 7 data will be stored. When a reference is converted, the converter will place the Baseline 7 reference in this directory. The converter will create a directory called 'measures_baseline6' in this folder. All converted measures will be stored in the 'measures_baseline6' directory in the same tree-structure as before in Baseline 5.

6. Click OK. The conversion starts.

The duration of the conversion depends strongly on the size of the reference and measures. A reference plus a set of 200 measures can take up to 12 hours to complete. During conversion, a log will be written to the directory indicated at the 'Baseline 6 directory'-field of the converter interface

Step 3: Validate the results

When the converter has finished, two more steps are required. At first, it is recommended to check the results of the converter manually by checking the log-file. Secondly, one might be required to make manual changes to the converted data. This paragraph explains these steps.

Reading the log-file
The log-file is located in the directory provided in the 'Baseline 7 directory'-field of the converter interface. Each time the converter is used, a new log-file is created.

Check the following lines in the log-file:
- When the converter has finished, the last line of the log-file should read 'INFO: Finished in XX hours, XX minutes and XX seconds.
- No lines with the label 'ERROR' should appear in the log. When they do appear, a description of the error is given in the log.
- Pay attention to lines labeled 'WARNING', in some cases they point to changes that have to be made manually. See the next paragraph for more information.

Manual changes to the converted data
In two cases, the Baseline 7 Feature Classes contains more detailed information about features than Baseline 5. In those cases, the converter will transfer the features from Baseline 5 to Baseline 6 and use default values for certain fields. The log-file indicates which fields in which feature classes must be enriched manually after the converter has finished.

The following feature classes/fields need to be enriched manually after conversion:

- In Feature Class 'locations/output_locations_points', the field 'SORTING' must be filled manually. The default value is NULL.
- In Feature Class bridge_points, the fields 'NAME', 'NUMBER' and 'CP' have to be filled manually. The default value for PC is -999, the others have NULL as the default value. These fields are only used in export to D-HYDRO, see the manuals of D-HYDRO for more information about these fields.

Inner workings of the converter
The main reason for the introduction of Baseline 6 is the development of a next generation of models, based on D-HYDRO Suite. This requires new export formats for the data, managed by Baseline. Furthermore, there is the need to store more detailed information in Baseline, for example bridge pillars and - in the future - river morphology of the river bed.

The introduction of Baseline 6 is also used to improve the existing data model using experience and knowledge gained over the past years. The biggest change in the existing data is the removal of derivative files for roughness's, weirs and elevation. Finally, the new data model of Baseline 6 uses English terminology instead of Dutch.

The aim of the converter is to convert Baseline 5 data to Baseline 6/7 automatically, without manual changes. Due to the big difference in the data model, especially those regarding roughness, the converter performs a complex operation to ensure a smooth transition from Baseline 5 to Baseline 6/7. This chapter explains the main difference between Baseline 5 and 6/7 and discusses the inner workings of the converter.

Main differences between Baseline 5 and Baseline 6/7
Besides the transition from Dutch terminology to English, there are 6 important differences between Baseline 5 and Baseline 6/7. These are briefly described below.

1. Removal of weirs as derivative file and the introduction of weirs as basic-file. In Baseline 5 there were 2 basic files containing weirs: *kades* and *kribben*. After assimilation of measures, both files were combined into one derivative file called *overlaten*. In Baseline 6, both types of weirs are stored in the same weirs file, which removes the need of a derivative file.

2. Adding bandijken to both terrain_edges_lines (teenhoogte) and elevated_lines/weirs (kruinhoogte)
3. Removing *lanen* and put those trees as individual trees in roughness_points, together with trees (bomen).
4. Combining ecotopen_ruwheid, plassen and hoogwatervrij_vlakken with roughness code (ruwheidscode) 1 to roughness_poly. This removes the need for a derivative file for roughnesses.
5. Moving hoogwatervrij_vlakken with roughness code (ruwheidscode) 2 to flowblocking_polygons.
6. Moving hoogwatervrij_vlakken with roughness code (ruwheidscode) 3 to bridges_points
7. Removing hoogtemodel as derivative file and introducing elevationmodel_terrain which is automatically created after assimilation of measures.
8. A straightforward combination of similar data types into one file. Each of the following Baseline 6/7 files has a field called 'type' indicating the Baseline 5 source file.
    a. Combining *oeverhoogtes* and *winterbedhoogtes* into landelevation_points.
    b. Combining *meetpunten*, *uitvoerlocaties*, and *rivierkilometer_punten* into output_location_points.

The conversion process
For most files (feature classes) and erase covers the converter simply moves the data from Baseline 5 to Baseline 6/7. Table 1 gives an overview of the conversion on file (feature class) level. In the six cases mentioned above, the converter performs a more complex operation, which all happen automatically and without user intervention.

Table 1: Conversion of Baseline 5 to Baseline 6 on file (Feature Class) level.

Baseline 6/7	**Baseline 5**
FDS Elevation	
terrain_edge_3d_lines	breuklijnen_routes /bandijken_routes (toe)
elevation_model_terrain	hoogtemodel (not transferred but newly build during conversion)
bedlevel_points	zomerbedhoogtes
surfacelevel_points	winterbedhoogtes, oeverhoogtes
waterbody_bedlevel_points	plashoogtes
terrain_jump_3d_routes	hoogteverschillijnen_routes
elevated_line_routes	kades, kribben, bandijken_routes (crest)
model_area_polygon	Dissolve(models/sections_poly)
FDS Locations	
cross_section_lines	rivierkilometer_lijnen
source_sink_points	bronnen_putten
output_location_points	meetpunten, uitvoerlocaties, rivierkilometer_punten
structure_lines	kunstwerken
bridge_routes bridge_points	*New, no data is transferred to this Feature Class* hoogwatervrij_vlakken (ruwheidscode 3)
FDS Roughness	
land_use_lines	heggen
land_use_points	bomen, lanen
land_use_polygons	hoogwatervrij_vlakken (huwheidscode 1), ecotopen_ruwheid, plassen
FDS Metadata	
measure_contour_polygons	omtrek_maatregel

FDS Models	
branch_1d_lines	rivierassen
calibration_section_input_polygons	zomerbed
calibration_section_polygons	*New, no data is transferred to this Feature Class*
cross_section_1d_polygons	*New, no data is transferred to this Feature Class*
flow_blocking_polygons	hoogwatervrij_vlakken (ruwheidscode 2)
flow_blocking_lines	hoogwatervrij_lijnen
initial_water_level_terrain	*New, no data is transferred to this Feature Class*
section_polygons	secties
FDS Morphology	
D50_points	*New, no data is transferred to this Feature Class*
dredging_polygons	*New, no data is transferred to this Feature Class*
suppletion_polygons	*New, no data is transferred to this Feature Class*
bed_characteristics_input_polygons	*New, no data is transferred to this Feature Class*
bed_characteristics_polygons	*New, no data is transferred to this Feature Class*

Conversion to land_use_polygons
The conversion of ecotopen_ruwheid, plassen and hoogwatervrij_vlakken with ruwheidscode 1 to land_use_polygons is a complex operation which dictates the overall working of the converter. In Baseline 5, removing an open water body (plas) requires an erase_plassen. However, in Baseline 6 it requires the definition of a new (ecotopen) roughness. When assimilating such a measure in Baseline 5, the Baseline 5 application would remove the open water body and put back the ecotopen_ruwheid as defined by one of the previous measures of reference in the area of the former open water body. To convert such a measure, the converter needs to mimic this Baseline 5 assimilation process in order to create the new ecotopen_ruwheid for the measure. This process is further explained in Table 2. The described method or conversion dictates that a measure can only be converted in context of the underlying/previous measures and reference.

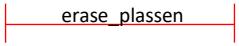
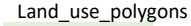
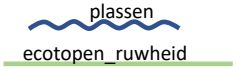
Conversion of erase covers
Erase covers are translated to Baseline 7 as well. Regarding the combination of erase covers for combined files (e.g. the erase cover for terrainjumps, which is a combination of the erase covers for hoogteverschillijnen and bandijken), the converter creates one new Baseline 6 erase cover that only removes the features as intended in Baseline 5. This results in an erase cover with very small polygons around the features that need to be removed, leaving other features untouched.

Known issues
As mentioned before, the results of the converter should be validated in detail, particularly in the case of converted measures.
Special attention should be given to the following situations:

- Polygons from the Baseline 5 Feature class hoogwatervrij_vlakken with ruwheidscode = 3 are translated to the Baseline 6 Feature class bridge_points. In order to complete the data the corresponding Featurclass bridge_lines needs to be constructed according to the dataprotocol. After finishing the bridge_lines both bridge_lines and bridge_points are converted to bridge_routes and events using preparation tool 6.
- Sometimes measures change the landuse close to the main channel. In this case land_use_polygons might overlap section 1. This is not allowed, so this overlap should be removed manually from land_use_polygons.
- When a measure results in a wider main channel (section 1 in Feature class section_polygons) the enlarged part should be added to land_use_polygons with the roughness_code of the corresponding roughness_code of the main channel in the reference. No changes in Feature class calibration_section_polygons should be made.

- When a measure results in a smaller main channel (section 1 in Feature class section_polygons) the reduced part should be added to land_use_polygons with information from the most recent ecotope measure.
- When a measure results in a removal of the primary embankment (Feature class Elevated_line_routes, type Primary Embankment) also an erase_terrain_edge_3d_lines Feature class should be constructed to remove the corresponding terrain edge in the reference situated beneath the primary embankment.
- In some rare occasions information form the Feature class land_use_polygons needs to be deleted. If so a Feature class erase_land_use_polygons should be constructed manually because the converter does not.
- When a Baseline 5 measure contains Feature classes erase_bomen and/or erase_heggen the converter creates new Feature classes erase_land_use_points and/or erase_land_use_lines by selecting all features to be erased within or crossing the erase polygon and buffering the selected features with a small buffer distance. This result in very small polygons which are sometimes difficult to distinghuis in GIS. Check wether the right polygons are made. If not, construct them manually.

Table 2: Schematization of conversion to roughness_poly in Baseline 6

		In the field	**Baseline 5**	**Baseline 6/7**	**Converter**
		This column describes the changes in the field which are schematized in Baseline.	This column describes the way in which Baseline 5 is handling the changes in schematization.	This column describes the way in which Baseline 6 is handling the changes in schematization.	This column describes the actions carried out by the converter to convert measures from Baseline 5 to Baseline 6.
↑ time	Measure 2	The open water body is muted. Vegetation around the open water body remains untouched.	An erase_plassen indicated which open water bodies (plassen) needs to be erased. 	A new vegetation is defined on the location of the former open water body. 	The converter is placing the most recent vegetation from previous measures and reference back in land_use_polygons at the extent of erase_plassen. In this particular case this will be the vegetation of measure 1 for the surrounding of the open water body. For the open water body itself the vegetation of the reference is taken.
	Measure 1	An open water body is being dug and the vegetation around the open water body is changed. 	An open water body is schematiszed in plassen, new vegetation is schematized in ecotopen_ruwheid. 	Open water body and new vegetation is schematized using land_use_polygons. 	The converter merges plassen and ecotopen_ruwheid to land_use_polygons.
	Reference		A vegetation is present in ecotopen_ruwheid.	A land_use_polygons with vegetation is present.	The converter copies all data from ecotopen_ruwheid to land_use_polygons.

4.5.5 *Create Smooth Transitions*

Purpose of the tool:
Both *calibration_section_input_polygons* and *bed_characteristic_input_polygons* contain information on hydraulic roughness in longitudal direction of the main channel of the river. The polygons in these Feature classes result, after conversion to D-Hydro input files, in discrete (abrupt) transitions in the hydraulic roughness of the main channel. Especially morphological models are sensitive to abrupt transitions. In order to prevent this the tool generates smaller stepwise transitions between two adjacent polygons. As a result the D-Hydro input files contain smoother transitions in main channel roughness.

This tool creates *calibration_section_polygons* or *bed_characteristic_polygons* with smooth transitions using the feature class *calibration_section_input_polygons* or *bed_characteristic_input_polygons* and the field *Transition*. If *bed_characteristic_input_polygons* is not present in the Baseline database this Feature class is derived from *land_use_polygons* selecting all polygons with Roughness_code > 1999.

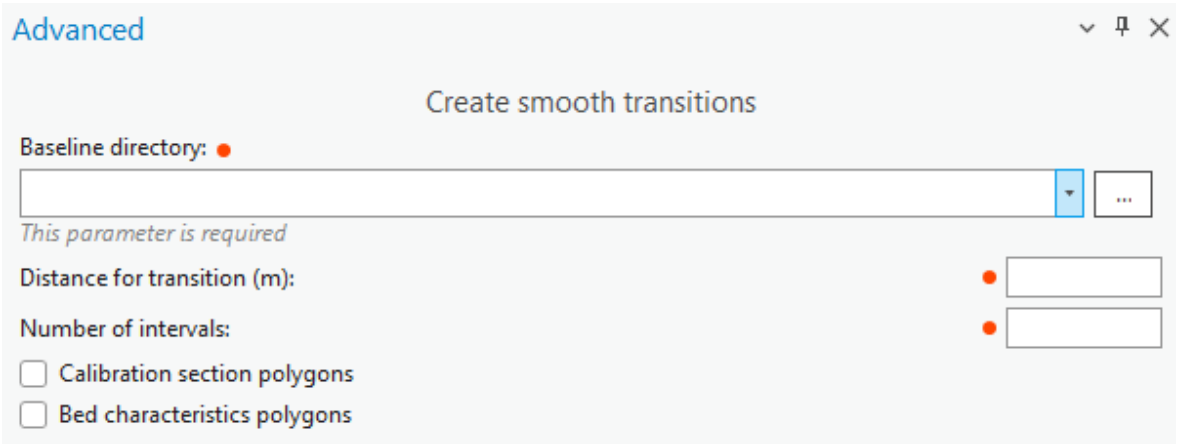


Figure 4.5.5-1 Dockpane

When *Calibration_section_input_polygons* is input the tool will create the Feature class *models\calibration_section_polygons* with two calibration codes and two calibration fractions for each polygon to make the transition. The interval must be an odd number (integer).
Bas2FM converts *calibration_section_polygons* to <model name>*_calibration_sections.cll*.

When *Bed_characteristic_input_polygons* is input the tool will create the Feature class *morphology\bed_characteristic_polygons* with two roughness codes and two roughness fractions for each polygon to make the transition. The interval must be an odd number (integer).

When *Bed_characteristic_input_polygons* is not present *bed_characteristic_input_polygons* is derived from *land_use_polygons* selecting all polygons with Roughness_code > 1999. The resulting Feature class *morphology\bed_characteristic_polygons* with two roughness codes and two roughness fractions for each polygon to make the transition. The interval must be an odd number (integer).

Bas2FM converts *bed_characteristic_polygons* to *<model name>_trachitopes.arl* by updating *land_use_polygons* within the main channel.

Both the *Calibration_section_input_polygons* and *Bed_characteristic_input_polygons* Feature class should not contain fixed layers and boundaries between polygons must lay between maximum two adjacent polygons. At bifurcations the polygon boundaries must be positioned downstream of the bifurcation at a distance of at least half the distance for the transition.

The tool will not give valid results when complex geometries are used.

The picture below gives an impression of the result (left calibration_section_input_polygons, right calibration_section_polygons).

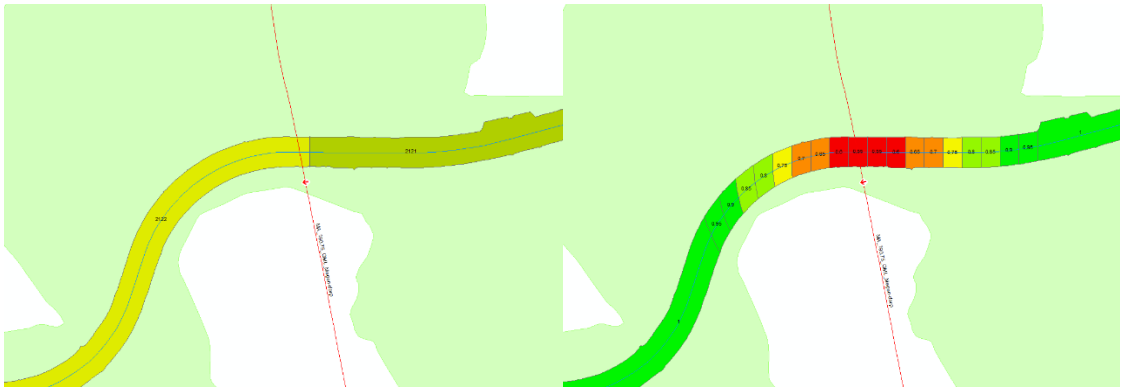


Figure 4.5.5-2 Example of the result

The CALIBRATION_FRACTION1 is plotted from 0.5 (red) to 1 (green). The calibration code will flip in the middle of the transition.
The tool will give a warning when a smooth transition is given where a calibration section input polygon does not span the whole width of a channel but will create this transition anyway.

4.5.6 *Raster 1x1 to Points 5x5*

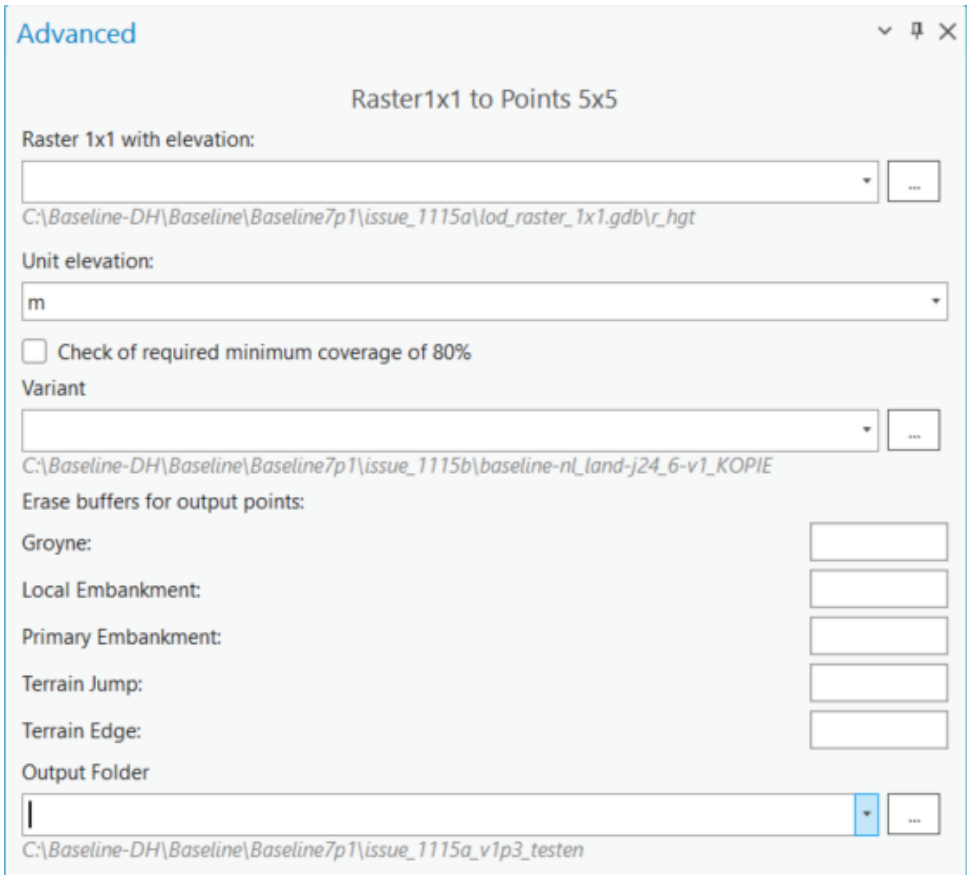


Figure 3: Dockpane

Purpose:
With this tool a bathymetry raster file with a resolution of 1x1m can be converted into a Feature class with points every 5x5m.

Parameters:
Provide a raster with a reolution of 1mx1m.

The unit of the input raster can be either cm, dm or m.

By checking 'Check of required minimum coverage of 80%' you can choose to only let a point be created when atleast 80% of the 1x1 raster cells carry valid data.

To guarantee that no points are generated too close to objects a distance (unit meters) to groynes, local_embankments, primary_embankments, terrain_jumps and terrain_edges can be given. When these 5 fields stay empty defaults are used in correspondence to the Data Protocol (5m distance to weirs, and 2,5 m distance to terrain_jumps and terrain_edges).

In the output folder a filegeodatabase with the output feature_class "Point_[name_input_raster]_5x5" is created, along with e few files in the calculation process. You can check if the outcome of the tool suits your goal.

4.6 Models

This paragraph describes the following functions:

1. Convert Grid netCDF to Bas2FM input
2. Conversion to D-Flow Flexible Mesh
3. Applying subgrid method
4. Conversion to SWAN

4.6.1 *Convert Grid netCDF to Bas2FM input*

Before Bas2FM can be run, as described in next paragraph, the relevant Grid netCDF (<>_net.nc) file should be translated to GIS Feature classes in order to be available for ArcGIS-functions. The resulting Feature classes are stored in a File geodatabase with the same name as the Grid netCDF.
Warning: The following characters are not allowed in file- and folder names (see also https://docs.microsoft.com/en-us/windows/win32/fileio/naming-a-file).
Use any character in the current code page for a name, including Unicode characters and characters in the extended character set (128–255), except for the following:
< (less than)
> (greater than)
: (colon)
" (double quote)
/ (forward slash)
\ (backslash)
| (vertical bar or pipe)
? (question mark)
* (asterisk)
Integer value zero, sometimes referred to as the ASCII NUL character.

Each time a Bas2FM conversion is run using the same Grid netCDF (<>_net.nc) file, this File geodatabase can be used.

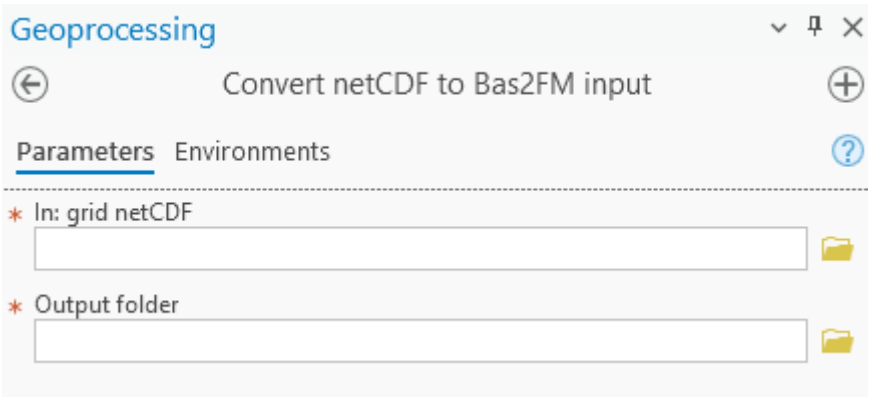


Figure 4.6.1-1 Dockpane

Parameters:

- In: Grid netCDF ➔ the Grid netCDF file (created by D-Grid Editor / RGFGRID). A copy of this file will be placed in the output directory. The bathymetry / elevation will be added. Therefore the output directory and the location of the grid file cannot be the same;
- Output folder; folder where the resulting File geodatabase is stored. This can be the same folder where the Grid netCDF file is located.

Five Feature classes are created: depth_FM, face_depth_FM, faces_FM, flownet_FM and ZonalFC.

ZonalFC is a Feature class containing the representative zones (polygons) surrounding the nodes in Feature class depth_FM and are derived using Thiessen polygons around the nodes. Flownet_FM is a Feature class containing the Perot-areas/polygons, the representative areas for the grid faces in Feature class faces_FM. These polygons are used to create the trachytope files.

4.6.2 *Conversion to D-Flow Flexible Mesh*

Bas2FM is a module for Baseline 7 that enables the user to convert Baseline 7 data to input for D-HYDRO Suite / D-Flow Flexible Mesh (FM).

It uses the Baseline 7 database in conjunction with Grid netCDF to create required D-Flow FM model input files. The tool does not create a fully functional D-Flow FM model model.

Warning: when large grids are used the computational time is extensive due to some elaborate GIS operations. For example the conversion of the RhineMeuse Model (RMM) takes about 36 hours.

Baseline database to D-Flow FM model input files

The main tool 'To D-Flow FM' to be used for conversion from Baseline 7 to D-Flow FM model input data is incorporated in the Baseline 7 toolbar.

In 4.6.2 the main tool is presented. This tool is used for conversion from Baseline 7 to D-Flow FM model input data. The tool consists of multiple parameters of which some are **mandatory**:

Parameters:

- **In: Baseline directory ➔ A Baseline variant tree (e.g. C:\Baseline\maas\j15_5-v1)**
- **In: Model name ➔ extension of all model files. (e.g *maas15*);**
- **In: GRID NETCDF GDB ➔ the GDB (created by 'Convert Grid netCDF to Bas2FM input') storing the GIS Feature classes of the Grid netCDF file relevant for the 'Conversion to D-Flow Flexible Mesh';**
- **In: Grid netCDF ➔ the Grid netCDF file (created by D-HYDRO / RGFGRID). A copy of this file will be placed in the output directory. Therefore output directory and the location of the grid file cannot be the same;**
- In: model boundary (optional) ➔ polygon feature class. If one is entered, only Baseline data intersecting these polygons will be converted (no clipping will occur);
- Use cdf enclosure if present ➔ if the Grid netCDF contains an enclosure container, this can be used as a clip-boundary. If both model bounds and cdf enclosure are selected, the combined polygon will be used. If neither one is selected, all Baseline data will be converted – this function will be discarded.
- Convert checkboxes ➔ depending on presence of required feature classes in the selected Baseline database, the available conversion options are shown;

Output:

- Output dir containing D-Flow FM input files:
  <baseline dir>/models/dflowFM/<<model name>>
    - boundary_conditions
    - computatons
    - general

- geometry
  - sources_sinks
    - <model name>_sources_sinks.pli
    - <model name>_<name1>.pli
    - <model name>_<name2>.pli
    - Etc…
  - structures
    - <model name>_structures.pli
    - <model name>_<name1>.pli
    - <model name>_<name2>.pli
    - Etc…
  - output_locations
    - <model name>_0_all_obs.xyn
    - <model name>_1_kilometer_obs.xyn
    - <model name>_2_output_obs.xyn
    - <model name>_3_measurement_obs.xyn
    - <model name>_4_boi_obs.xyn
    - <model name>_5_100m_obs.xyn
    - <model name>_6_20m_obs.xyn
    - <model name>_7_obs.xyn
    - <model name>_8_obs.xyn
    - <model name>_9_obs.xyn
    - <model name>_10_obs.xyn
    - <model name>_11_obs.xyn
    - <model name>_12_obs.xyn
    - <model name>_13_obs.xyn
    - <model name>_14_obs.xyn
    - <model name>_15_obs.xyn
    - <model name>_16_obs.xyn
    - <model name>_17_obs.xyn
    - <model name>_18_obs.xyn
    - <model name>_19_obs.xyn
    - <model name>_20_obs.xyn
  - cross_sections
    - <model name>_0_all_crs.pli
    - <model name>_1_kilometer_crs.pli
    - <model name>_2_output_crs.pli
    - <model name>_3_measurement_crs.pli
    - <model name>_4_calibration_crs.pli
    - <model name>_5_crs.pli
    - <model name>_6_crs.pli
    - <model name>_7_crs.pli
    - <model name>_8_crs.pli
    - <model name>_9_crs.pli
    - <model name>_10_crs.pli
  - <model name>_fxw.pliz
  - <model name>_thd.pli
  - <model name>_dry.pol
  - <model name>_enc.pol
  - <model name>_net.nc
  - <model name>_trachytopes.arl
  - <model name>_calibration.cll
  - <model name>_bridges.pliz
  - <model name>_initial_water_level.xyz
  - <model name>.gdb
    - *Gdb files*
  - {Other files which are not created with Bas2FM}
- initial_conditions

    - metainfo
    - rtc
- <baseline_dir>\metainfo\logs\Bas2FM.log. The log-file with D-Flow FM-conversion history. New log-information is appended to the file. Subsequent runs are separated by the following line:
  ==========

The used methods to create this output are described in Appendix C.

Partial conversion

Sometimes it is not necessary to convert an entire Baseline database to D-Flow FM. For example if in a variant only elevated_lines are modified compared to the reference then a conversion of only fixed weirs has to be done, all other D-Flow FM input files remain the same. The 'Conversion to D-Flow Flexible Mesh' facilitates partial conversion and this reduces the conversion time considerably.

However, because some Feature classes are combined in the D-Flow FM input files it is necessary to check these combinations when a partial conversion is done. The following combinations should be used:

1. *trachytopes.arl* > *land_use_polygons, land_use_lines, land_use_points* and *bed_characteristics_polygons*[2]
2. *dry areas.pol* > *section_polygons* and *flow_blocking_polygons*

[2] bed_characteristics_polygons only must be used when smooth transitions in the main channel are required

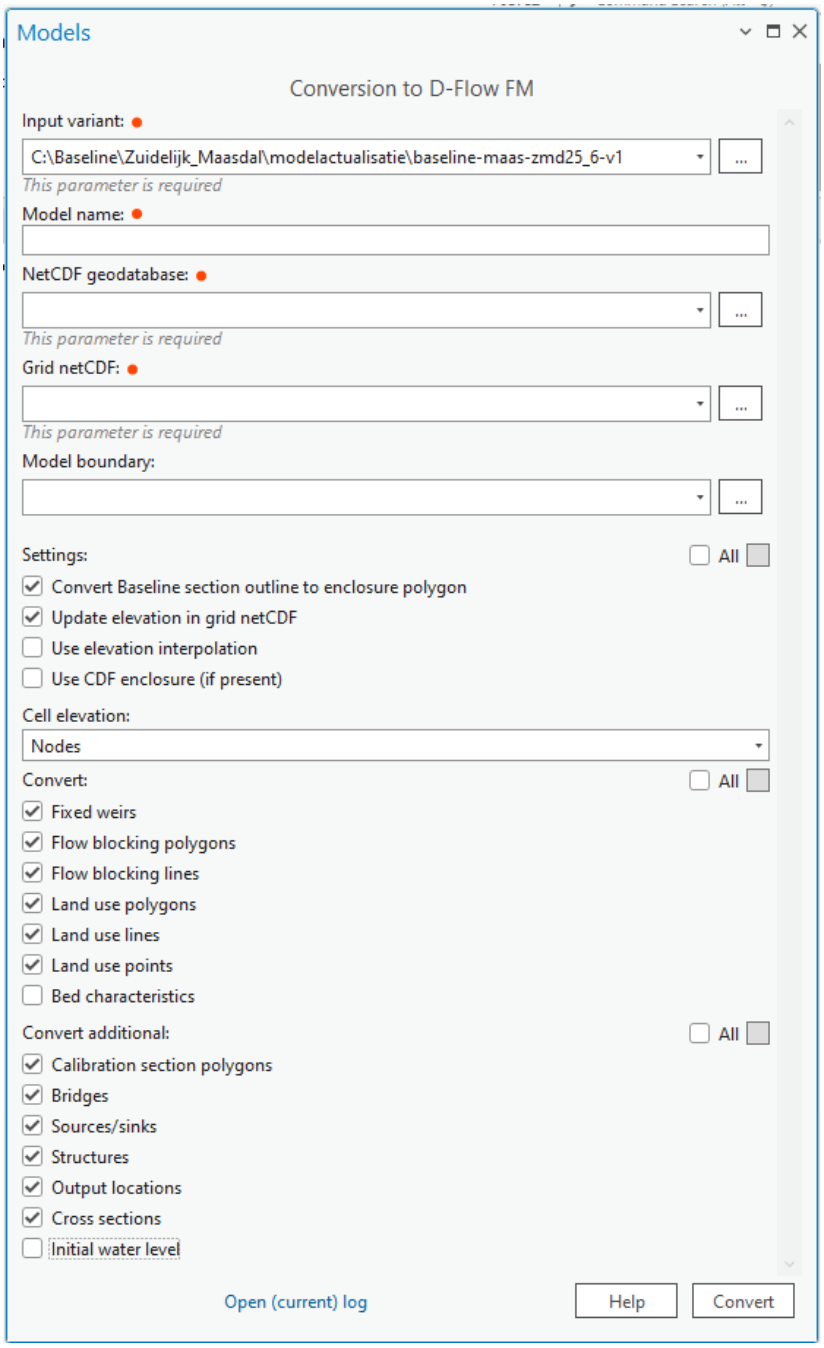


Figure 4.6.2-1 Dockpane

Convert polyline or polygon to landboundaries-file
Tool to create .ldb files which can be used as landboundaries-file in D-HYDRO suite. This tool can be added as ArcToolbox, see installation manual.

4.6.3 *Applying subgrid method*

Purpose:
When the Grid netCDF is derived from the Baseline-database (Baseline-land) by default the elevation is determined at the nodes of the cells in the mesh. When certain very thin low elements are present in the Baseline database (like small side channels) these might not be

represented well in the resulting elevationin the Grid netCDF. This might lead to an underestimation of the effects of these elements in the modelling results.

This tool can be used to calculate an alternative elevation for a Grid netCDF file based on the **average** elevation in the elevation_model_terrain found in a cell in the mesh instead of the point value at the cell node. Then the effect of such measures is better approximated through the schematisation.

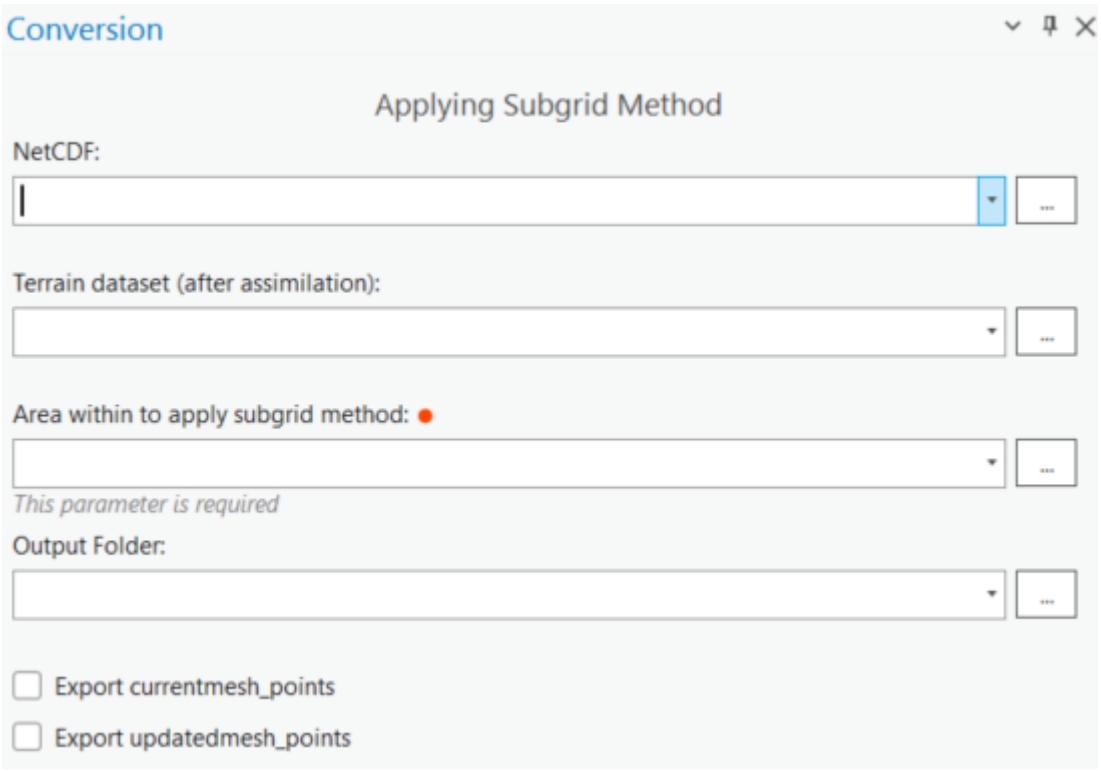


Figure 2: Dockpane

Parameters:
Give the name of the Grid netCDF file (this is the result of the default conversion to D-Flow FM).

Give in the elevation_model_terrain of the corresponding variant.

Provide a polygon feature-class representing the contour(s) of the small element(s) within which the alternative elevation is calculated. The new calculated elevations overrule the default elevation of the input Grid netCDF file and it is combined in the new Grid netCDF file with the name [input]_subgrid.nc

In the Output Folder the additional outputs will be written.

4.6.4 *Add schematization*

Purpose:
This function adds a SWAN directory with the name of the grid. In this directory the input files for hydrodynamic models are created with the function "Conversion to SWAN.

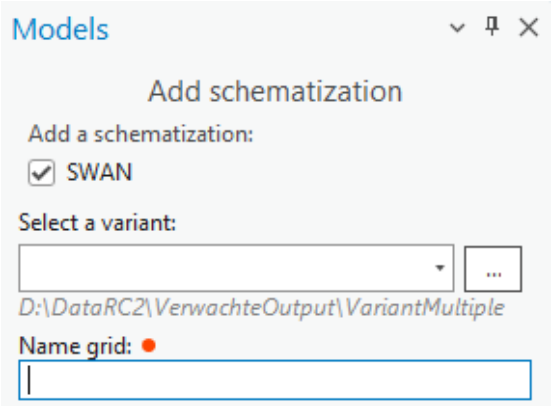


Figure 4.6.4-1 Dockpane

Explanation:
The grid directory is created in the model directory. The name of this directory is determined by the user. In this directory a default structure is created. After creating this grid directory the <>.swn file and eventually <>.grd file should be placed in this directory.

The default structure:
SWAN: /Baseline<area>/<variant>/models/swan/<model name>

4.6.5 *Conversion to SWAN*

Goal:
The function "Conversion to SWAN" converts data from the Baseline database to inputfiles for SWAN.

Explanation:
The function "conversion to SWAN" uses the active variant. This should be set by the user. There are several possibilities for SWAN input, three of these are mandatory:

1. The Model name is the identification which is used as prefix for SWAN input files.
2. At "Select SWAN inputfile" the user selects a <>.swn inputfile. This file should be located in a SWAN schematization directory. This directory should be made with the Baseline application. This directory is used for writing the SWAN inputfiles.

After this some properties of the grid are shown.

Next to these properties the user can define which inputfiles should be generated. The available check-boxes are automatically checked if the necessary files are present in the Baseline database of the active variant. This selection can be undone. Only selected features will be converted.

3. Depth form terrain/raster: Is a representation of the elevation at the elevationpoints. These are the elevationpoints of the SWAN grid. These are written to *<model name>\geometry\<model name>.bot*.
4. Obstacels: Obstacles/fixed weirs consists of lines/routes and points/events. When this checkbox is selected *<model name>.fxw* is created. This file represents weirs as line elements with relative height on the pointlocations. Different types of line elements are incorporated:
    a. Elevated_line_routes (containing primary dikes, summer dikes and groynes)
    b. Terrain_jump_3d_routes (containing terrain_jumps)

   c. Flow_blocking_polygons, these will become non floodable elements (elevation 999m)
   d. Flow_blocking_lines, these will also become non floodable elements (elevation 999m)
5. Output_locs: output_locations, these are written to *<model name>\geometry\output_locations\<model name>_<location type>_obs.pnt*.
6. Elevation interpolation: instead of the default method of determining the elevation (using "LINEAR" interpolation for land-databases and "BILINEAR" interpolation for sea-databases) at the locations of rooster_bp_punten in rooster.gdb gridcell averaging (zonal statistics MEAN) is used within the polygons of rooster_bp_vlakken in rooster.gdb. This might give a better representation of for example the bathymetry of smaller tidal channels. For this option the Spatial Analist extension is required.

Finally a Feature class containing a modelboundary clip polygon can be defined to "clip" the swan input files using a model boundary (output_locations are not clipped).

The result of the conversion to SWAN is written to the folder geometry in the folder *<model name>*. However the inputfiles are written in several directories, see the data protocol.

The converted files are placed on location on the SWAN model name (gridcells and crossings of links) and can now be used as input for SWAN models.

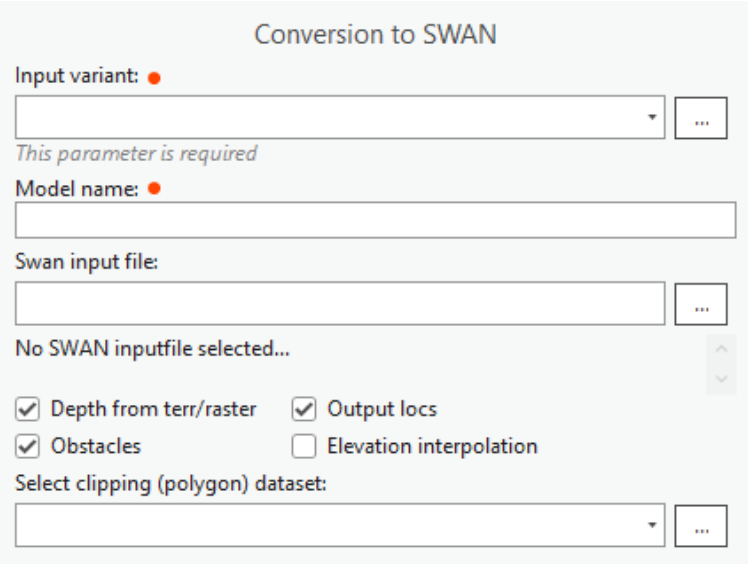


Figure 4.6.5-1 Dockpane

A part of the converted files is also converted back to a Feature Class. The purpose of this is that the user can do a visual check in ArcGIS whether the conversion from Baseline to SWAN is done properly. Alle files converted to a Feature Class are being saved in the database *Invoer.gdb*. This database can be found in the directory input in *models/SWAN/<model name>/geometry*. After conversion these files are automatically added to the Table Of Contents (TOC) using default layers.

4.7 Help

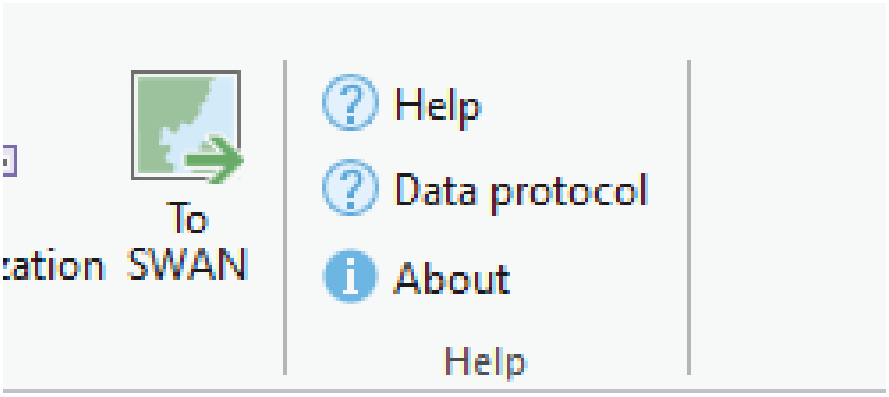


Figure 4.6.5-1 Help information

In this paragraph the following functions are described:

1. Baseline Help
2. Baseline Dataprotocol
3. About Baseline

4.7.1 *Baseline Help*

This function displays the User Manual, this document in a PDF-reader (if it is installed). Otherwise the pdf can be opened via the explorer in an internet browser.

Note: This document is also opened with the Help-buttons of other Baseline functions.

4.7.2 *Baseline Dataprotocol*

This function displays the Baseline Dataprotocol [1] in a PDF-reader (if it is installed). Otherwise the pdf can be opened via the explorer in an internet browser.

4.7.3 *About Baseline*

This function shows information about the release.

Figure 4.7.3-1 Baseline about panel

4.8 General

4.8.1 *Executing a tool*

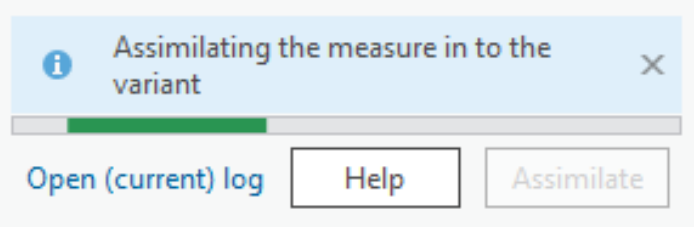


Figure 4.8.1-1 Running a tool (Assimilate measures is used in this example)

When a tool is opened in the toolbar, it opens a dockpane in most cases. This dockpane can be used to set the input parameters and to run the tool, this can be done by clicking the respective run button. When a tool is running, the progress bar is active and progress messages will be shown (blue: information, yellow: warning, green: confirmation, red: error). Additionally, the current log can be opened by pressing the 'Open (current) log' text. If no log currently used, the general log file will be opened by default.

5 References

[1] Dataprotocol Baseline 7. WSP, 2024 (updated by Esri)

[2] Installation Manual Baseline 7. WSP, 2024 (updated by Esri).

Appendices

Appendix A: Baseline database and files

Location in database	**Feature Class/file**
	elevation_mosaic (sea_variant)
	elevation_raster (sea variant/measure)
	elevation_raster_land (merge sea variant)
	bridge_events
	elevated_line_events
	terrain_jump_3d_events
Baseline.gdb/Elevation	terrain_edge_3d_lines
	elevation_model_terrain
	bedlevel_points
	surfacelevel_points
	waterbody_bedlevel_points
	terrain_jumps_3d_routes
	Elevated_line_routes (not included in elevation_model_terrain)
	model_area_polygon
Baseline.gdb/Locations	Bridge_routes/events
	cross_section_lines
	Source_sink_points
	output_location_points
	structure_lines
Baseline.gdb/Roughness	land_use_lines
	land_use_points
	land_use_polygons
Baseline.gdb/Metadata	measure_contour_polygons
Baseline.gdb/Models	branch_1d_lines
	calibration_section_input_polygons
	calibration_section_polygons
	Cross_section_1d_polygons
	flowblocking_lines
	flowblocking_polygons
	Initial_water_level_terrain
	section_polygons
Baseline.gdb/Morphology	D50_points
	dredging_polygons
	suppletion_polygons
	bed_characteristics_input_polygons

Location in database	Feature Class/file
	bed_characteristics_polygons
Models/dflowFM/<model name>/geometry/	<model name>_landboundaries.ldb
	<model name>_net.nc
	<model name>_fxw.pliz
	<model name>_dry.pol
	<model name>_thd.pli
	<model name>_trachitopes.arl
	<model name>_calibration.cll
	<model name>_bridges.pliz
	<model name>_initial_water_level.xyz
Models/dflowFM/<model name>/geometry/ sources_sinks/	<model name>_sources_sinks.pli <model name>_<name1>.pli <model name>_<name2>.pli Etc…
Models/dflowFM/<model name>/geometry/ structures/	<model name>_structures.pli <model name>_<name1>.pli <model name>_<name2>.pli Etc…
Models/dflowFM/<model name>/geometry/ output_locations/	<model name>_0_all_obs.xyn <model name>_1_kilometer_obs.xyn <model name>_2_output_obs.xyn <model name>_3_measurement_obs.xyn <model name>_4_boi_obs.xyn <model name>_5_100m_obs.xyn <model name>_6_20m_obs.xyn <model name>_7_obs.xyn <model name>_8_obs.xyn <model name>_9_obs.xyn <model name>_10_obs.xyn <model name>_11_obs.xyn <model name>_12_obs.xyn <model name>_13_obs.xyn <model name>_14_obs.xyn <model name>_15_obs.xyn <model name>_16_obs.xyn <model name>_17_obs.xyn <model name>_18_obs.xyn <model name>_19_obs.xyn <model name>_20_obs.xyn
Models/dflowFM/<model name>/geometry/ cross_sections/	<model name>_0_all_crs.pli <model name>_1_kilometer_crs.pli <model name>_2_output_crs.pli <model name>_3_measurement_crs.pli <model name>_4_calibration_crs.pli <model name>_5_crs.pli <model name>_6_crs.pli <model name>_7_crs.pli

Location in database	Feature Class/file
	<model name>_8_crs.pli <model name>_9_crs.pli <model name>_10_crs.pli
Models/dflowFM/<model name>/geometry/ <model name>.gdb	Depth_FM
	Flownet_FM
	Flownet_FM_roughpoly
	Flownet_FMroughnessline
	Flownet_FMruwpoint
	Enclosure_FM
	Initial_waterlevel_FM
Models/swan/<model name>	<grid>.swn <grid>.grd <grid>.rgf
Models/swan/<model name>/geometry/	<model name>.bot
	<model name>.fxw
Models/swan/<model name>/geometry/output_locations	<model name>_0_all_obs.xyn <model name>_1_kilometer_obs.xyn <model name>_2_output_obs.xyn <model name>_3_measurement_obs.xyn <model name>_4_boi_obs.xyn <model name>_5_100m_obs.xyn <model name>_6_20m_obs.xyn <model name>_7_obs.xyn <model name>_8_obs.xyn <model name>_9_obs.xyn <model name>_10_obs.xyn <model name>_11_obs.xyn <model name>_12_obs.xyn <model name>_13_obs.xyn <model name>_14_obs.xyn <model name>_15_obs.xyn <model name>_16_obs.xyn <model name>_17_obs.xyn <model name>_18_obs.xyn <model name>_19_obs.xyn <model name>_20_obs.xyn

Appendix B: Overview of all checks in function "Check variant/measure"

Legend with "Baseline validation module Baseline 6":

- Note: OK, no errors.
- Warning: Be carefull but no error
- Error: Validation not succesfull
- When the validation is succesfull this is displayed with a note

	Tool	Scriptversion	Error	Note/Error/Warning	Remark
1	ProtocolCheck.py	26	Directory metainfo not present	Note.	If not present, it will be created
2	ProtocolCheck.py	26	Baseline 6 not installed (check registry)	Error.	If Baseline 6 is not installed the validation will stop
3	ProtocolCheck.py	26	ArcGIS 10.3.1 / ArcGIS 10.4 or ArcGIS 10.5 is not installed	Error.	
4	ProtocolCheck.py	26	template not present in Baseline 6installdir	Error.	Template is neccesary for further validation
5	ProtocolCheck.py	26	Name measure/variant contains space	Error.	spaces are forbidden according to protocol
6	ProtocolCheck.py	26	Name measure/variant contains more than 26 characters	Error.	More than 26 characters are forbidden according to protocol
7	ProtocolCheck.py	26	Baseline.gdb not present in measure/variant	Error.	Baseline.gdb needs to be present
8	ProtocolCheck.py	26	structure of directories not according protocol (including TEMP)	Error.	Mandatory directories are: models, data, metainfo, baseline.gdb and optional temp
9	ProtocolCheck.py	26	structure of directory DATA not according protocol	Error.	Mandatory directories in DATA are: export, import, layers, source
10	ProtocolCheck.py	26	structure of directory Models not according protocol	Error.	Mandatory directories in Models are: FM
11	ProtocolCheck.py	26	structure of directory Metainfo not according protocol	Error.	Mandatory directories in Metainfo are: Lists and Logs
12	ProtocolCheck.py	26	Other directories than listed at 8 in measure-rootdir	Error.	No other directories are allowed

	Tool	Scriptversion	Error	Note/Error/Warning	Remark
13	ProtocolCheck.py	26	Feature Dataset in Baseline.gdb is missing	Error.	All FD's need to be present in a variant/measure
14	ProtocolCheck.py	26	Unnecessary Feature Dataset are present in Baseline.gdb	Error.	FD's which are not present in the templates are not allowed in baseline.gdb
15	ProtocolCheck.py	26	Unnecessary FeatureClas are present in Baseline.gdb	Warning.	FC's which are not present in the templates are not allowed in baseline.gdb. However this is the responsibility of the user
		26			This also means that Elevation does not contain **_lines, **_points or **_locate_errors.
16	ProtocolCheck.py	26	Fields in FC's not according protocol	Error.	Fields in FC's not according protocol
17	ProtocolCheck.py	26	Unnecessary Tabellen in Baseline.gdb	Error.	Only event-tables are allowed in the database
18	ProtocolCheck.py	26	Table field not according protocol	Error.	Check if Fields are according protocol/template
19	ProtocolCheck.py	26	erase_list and append_list is missing or not complete	Error.	Check if erase_list and append_list are present
		26		Error.	Check if item in list is present in Baseline.gdb
		26		Warning.	Check if FC in baseline.gdb is present in lists.
20	ProtocolCheck.py	26	check on singlepart 1 polygon dissolved sections	Error.	If Als er meer dan 1 omhullende is dan crasht baswaq.
21	ContentCheck.py	38	Directory metainfo not present	Note.	If not present, it will be created
22	ContentCheck.py	38	ArcGIS 10.3.1 / ArcGIS 10.4 or ArcGIS 10.5 is not installed	Error.	
23	ContentCheck.py	38	FC measure_contours is missing	Error.	FC measure_contours is needed to check if all FC's are inside this polygon
24	ContentCheck.py	38	List of all Characteristics in an FC or Event Table is made	Note.	List of all Characteristics in an FC or Event Table is listed in logfile
25	ContentCheck.py	38	Characteristics not filled	Warning.	Characteristics should be filled
26	ContentCheck.py	38	Lijst met items in LOCATIE	Note.	In het logbestand wordt per FC weggeschreven welke LOCATIEs zijn opgegeven.
27	ContentCheck.py	38	LOCATIE is verkeerd	Warning.	Locatie mag alleen LINKEROEVER of RECHTEROEVER zijn
28	ContentCheck.py	38	KENMERK niet hwatvrij bij poly's met opp > 2500m2	Warning.	Een oppervlak van 2500m2 of groter moeteigenlijk hwatvrij zijn.

	Tool	Scriptversion	Error	Note/Error/Warning	Remark
29	ContentCheck.py	38	In measure_contours, item MEASURE not equal to measure name	Error.	In measure_contours each polygon should contain the measure name in the MEASURE field.
30	ContentCheck.py	38	Land_use_polygon: connected is connected or not connected	Error.	Connected should be connected or not connected where ROUGHNESS_CODE is 104,105,106 or 201
31	ContentCheck.py	38	Features in a measure are outside measure_contours	Error.	All features are inside measure_contours (buffer of 10cm)
32	ContentCheck.py	38	Overlapping polygons in a polygon-FC	Error.	No overlapping polygons are allowed, exception is measure_contours.
33	ContentCheck.py	38	Elevation lines with a Number which is not present in the corresponding event table.	Error.	Elevation lines with a Number which is not present in the corresponding event table.
34	ContentCheck.py	38	Elevation points with value = -9999	Error.	-9999 is an incorrect value'
35	ContentCheck.py	38	Elevation points with value = 0	Warning.	Check if value is correct
36	ContentCheck.py	38	Elevation points with values < -30 or > +70	Warning.	Check if value is correct
37	ContentCheck.py	38	CRESTHEIGHT < LEFT, RIGHT	Error.	Is not permitted
38	ContentCheck.py	38	Elevated_line: ELEVATION_LEFT and _RIGHT both non-equal to ELEVATION_CREST.	Error.	Check if value is correct
39	ContentCheck.py	38	Multiple elevation points on the same locations with different elevations (rounded to centimeters)	Error.	Different Elevation values on the same locations are not permitted
40	ContentCheck.py	38	bedlevel_points, surfacelevel_points or waterbody_bedlevel_points are placed in a regular grid	Warning.	This is not permitted but hard to check
41	ContentCheck.py	38	A measure containing waterbody_bedlevelpoints which are outside a waterbody in the same measure.	Warning.	Not logical to append waterbody_bedlevel_points outside waterbodies

	Tool	Scriptversion	Error	Note/Error/Warning	Remark
42	ContentCheck.py	38	Land_use_points or land_use_lines with HEIGTH, DIAMETER of DENSITY = -9999	Error.	-9999 is an incorrect value'
43	ContentCheck.py	38	Land_use_points or land_use_lines with HEIGTH, DIAMETER of DENSITY <0	Error.	Value should be positive
44	ContentCheck.py	38	Land_use_points or Land_use_lines : HEIGHT > 15	Warning.	Check if value is correct
45	ContentCheck.py	38	Land_use_lines: DENSITY is incorrect	Error.	DENSITY is open, very open or dense
46	ContentCheck.py	38	Al fc's with ROUGHNESS_CODE: List of alle used codes	Note.	
47	ContentCheck.py	38	Routes with number = 0	Error.	Numbers should be greater than zero for baswaq to run
48	ContentCheck.py	38	Multipart polygons	Error.	No multipart features are allowed
49	ContentCheck.py	38	Check if polyline feature classes have overlapping records	Error.	Polylines should not overlap
50	ContentCheck.py	38	GROUNDLEVEL waterbody = -9999	Error.	-9999 is an incorrect value'

Appendix C: Conversion to D-Flow FM Methods and technical details

Overview
As described in chapter 4.6.2, Conversion to D-Flow FM generates the following input data for D-HYDRO Suite / D-Flow FM. See chapter 4.6.2 for the directory structure.

Window: Baseline to D-Flow FM		
	Baseline – used as input	**Generated D-HYDRO input**
Convert boundaries	Section_polygons	<model name>_enc.pol <model name>_dry.pol[3]
Convert elevation in Grid netCDF	Elevation_model_terrain[4] Elevation_ raster Elevation_raster_land	<model name>_net.nc
Convert weirs	Elevated_line (routes and events) Terrainjump_line (routes and events)	<model name>_fxw.pliz
Convert areas outside flood zone	Flow_blocking_polygons	<model name>_dry.pol <model name>_thd.pli
Convert flow blocking lines	Flow_blocking_lines	<model name>_thd.pli
Convert roughness polygons	Land_use_polygons	<model name>_trachytopes.arl
Convert roughness lines	Land_use_lines[5]	
Convert roughness points	Land_use_points[5]	
Convert bed characteristics	Bed_characteristics_polygons	
Convert calibration sections	Calibration_section_polygons	<model name>_calibration.cll
Convert bridges/pillars	Bridges_routes/events	<model name>_bridges.pliz
Convert cross-sections	Cross_section_lines	<model name>_<nr>_<type>_crs.pli
Convert lateral sources/sinks	Source_sink_points	<model name>_sources_sinks.pli *Plus files with individual sources/sinks*
Convert barriers	Structure_lines	<model name>_structures.pli *Plus files with individual structures*
Convert Waterlevel in water bodies	initial_water_level_terrain	_<model name>initial_water_level.xyz
Convert output locations	output_location_points	<model name>_0_all_obs.xyn *Plus files with individual feature class data.*

Method for elevation on cell nodes

[3] In case of inner rings in sections
[4] Elevation _model_terrain in case of a land-variant, elevation_raster in case of a sea-variant, elevation_raster and elevation_raster_land in case of a merged sea-variant
[5] When a sea varaint is converted no roughness lines or points are taken in to account due to a negligible area

For all grid points the default method in which the elevation is determined is by the standard ArcGIS-function to interpolate in a Terrain ("Add Surface Information"), the method used is "LINEAR" for land-databases and "BILINEAR" for sea-databases. This is compliant with previous Baseline releases. Tests have shown that the resulting elevation model for curvilinear (Waqua)grids is almost similar: only at some limited areas there are differences.

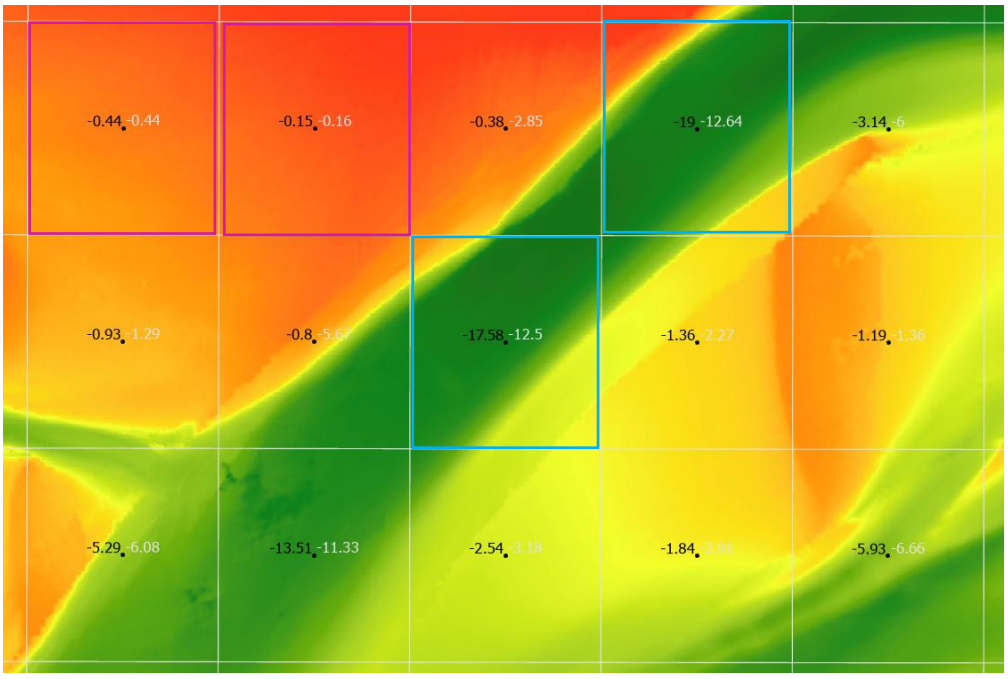
Alternative elevation interpolation

Besides this default method the elevation can be determined using a cell-averaging method. In the Conversion to D-Flow FM menu "use elevation interpolation" should be checked. For this method the Spatial Analist ArcGIS extension is required. The following steps are conducted:

- When a land variant is input the elevation_model_terrain is converted to a 5x5m raster named elevation_raster_land.
- Using the function "zonalstatistics" mean values are calculated within the polygons of ZonalFC (in <modelname>_net.gdb) from:
    - elevation_raster in case of a sea variant
    - elevation_raster and elevation_raster_land in case of a merged sea variant
    - elevation_raster_land in case of a land variant

In the next figure the elevation_raster_land (5x5m) of part of the Waddensea is shown in colors together with the nodes of the DCSM grid, the nodes are the black dots and the elevation at the nodes is plotted in black. The representative areas surrounding the nodes (the zones in the ZonalFC) are the white polygons and the mean elevation of this representative areas is plotted in white.

In the figure can be seen that in flat locations both elevations are almost equal (purple cells) while in locations with a lot of variation (blue cells) there is a significant difference between the 2 methods. This example shows results for a model grid with very large cells (112x70m) compared to a high resolution elevation grid (5x5m). When the model grid has a finer resolution the differences between both methods will be smaller.



From November 2024 onwards, most Dutch hydrodynamic models use Grid netCDF where the elevation is determined using the default method in the 'Conversion to D-Flow Flexible Mesh'. Only hydrodynamic models of North Sea and Continental Shelf use Grid netCDF where the elevation is determined using the interpolation method in the 'Conversion to D-Flow Flexible Mesh', because this results in a better connectivity of tidal channels.

Technical details:

- Input:
  - Elevation_model_terrain/ Elevation_raster/ Elevation_raster_land
- Output – intended for use in D-HYDRO Suite:
  - <model name>_net.nc
  - See D-HYDRO Suite documentation for file-format and definitions
- Conversion to D-Flow FM reads and uses the z_positive = 'up/down' from the <model name>_net.nc-file
- Conversion to D-Flow FM reads the "fill-value" from the <model name>_net.nc-file, for example '-999'
- Conversion to D-Flow FM writes the fill-value where ArcGIS cannot determine the elevation. So, on all grid points an elevation is written!

Method for elevation on cell centers

The determination of elevation is similar to the ones described before. Only the location where elevation is derived from is different; cell centers instead of cell nodes.

This method is only possible for Grid netCDF files with UGRID format=1.0 or higher.

If the cell-center elevation attribute is not present in the Grid netCDF file, Conversion to D-Flow FM will add it to the file prior to defining data on these locations.

Method for elevation near grid enclosure

If one or more of the neighbouring nodes of a cell-center do not have a valid elevation D-Flow FM uses the value "bedleveluni" (if this value is -999 there will be a depression in the model). In an earlier version of 'Conversion to D-Flow Flexible Mesh'only nodes within or on the edge of the enclosure (Feature class section_terrain) received a z-value which resulted in cell-centers having no z-values.
In order to avoid this, 'Conversion to D-Flow Flexible Mesh'extrapolates existing z-values over a distance of **250 meters** outside the enclosure (the z-value is taken form the closest vertex on the edge of the Baseline elevation_model_terrain, so the extrapolation is always horizontal). This value of **250 meters** might be insufficient when Baseline is used in areas where the grid resolution is coarser then 250 meters, eg. in North Sea models.

Method for vegetation (Trachytopes / roughness polygons, lines and points)

The option 'Conversion to D-Flow FM' intersects for all flow/net-links the so-called Perot-areas/polygones with all the Land_use_polygones. This is described in paragraph 4.6.2 Conversion to D-Flow Flexible Mesh.
The Perot-areas/polygons need to be generated in advance, as described in paragraph 4.6.1 Convert Grid netCDF to Bas2FM input. The polygons are derived from the Grid netCDF file (<model name>_net.nc) and stored in Feature class Flownet_FM.

Definitions:

- In D-HYDRO flow/net-links are defined as the lines connecting grid points. Water "flows" from one grid cell to the other, passing the flow/net-link.
- A Perot-area is the rectangle given by 2 lines perpendicular to the flow-link, through the grid points on both sides of the flow-link and 2 lines parallel to the flow-link, through the opposing cell-centres.

Technical details:

- Input:
  - Land_use_polygons
  - Land_use_lines
  - Land_use_points
  - Bed_characteristics_polygons (only when present in the variant)
- Output – intended for use in D-HYDRO Suite:
  - <model name>_trachytopes.arl
  - See D-HYDRO Suite documentation for file-format and definitions
- Mark that Perot-areas can overlap. In those cases the roughness is counted double. Still, the Perot-area can be seen as the area that "influences" the flow over a flow-link. Mark also that in case of curvilinear grids this method is identical to the method used in previous Baseline releases, in casu Baswaq / WAQUA.
- Mark that Perot-areas are limited to the grid. This is a minor improvement to the method used in Baswaq.
- Roughness points:
  - are transposed into surface-areas derived from the diameter of the tree. The sum of all tree-surface-areas is also written to the trachytopes -file.
  - (roughness percentage for a tree) = (∑(diameter of the tree) / (tree-surface-area))
- Roughness lines
  - are one by one projected on flow/net-links. The total projected length is written in the trachytopes -file with the roughness code of the 'heg'.
  - Bas2FM writes NO information as Thin dams. This is equivalent to the former Baswaq, that did not write any information in the schotarea-u/v.
- Bed_characteristics_polygons
  - When this Feature class is checked it is used to overwrite the underlying information from land_use_polygons.
- Mark that total area-percentage per cell as stored in the trachytope-file exceeds 1.0 (100%) when roughness lines or points are located within this cell. This is identical to the Baswaq / WAQUA method.

Method for calibration sections

Calibration sections are converted in the same manner as land_use_polygons.

Method for modelling Enclosure, Areas outside flood zone and Flow blocking lines

Areas outside flood zone (flow_blocking_polygons) are ('flood free') areas/buildings where no water will flow (no flow, no precipitation, no source/sink). These areas are modelled as Dry areas and/or Thin dams and as trachytopes.

Flow blocking lines (flow_blocking_lines) block the flow of water, modelled in D-HYDRO Suite as Thin dams.

Technical details:

- Input:
  - flow_blocking_polygons
  - flow_blocking_lines
  - section_polygons (polygons that will be dissolved during Bas2FM conversion process)
- Output – intended for use in D-HYDRO Suite:
  - <model name>_thd.pli
  - <model name>_dry.pol
  - <model name>_enc.pol
  - See D-HYDRO Suite documentation for file-format and definitions
- For enclosure polygons (sections_polygons)
  - Only the outside polygon(s) are modelled as individual enclosures.
  - Inner rings (e.g. islands that will not be part of the model area) are modelled as Dry area
- For areas outside flood zone (flow_blocking_polygons):
  - Gridcells which are completely covered by a flow_blocking_polygon or building are modelled as Dry area.
  - The remaining parts of these former flow_blocking_polygons and buildings are written as Thin dams and processed by D-HYDRO Suite: the part of a flow_blocking_polygon overlapping a gridcell is projected on the flow/net-link.
  - D-HYDRO Suite treats a flow/net-link like a Thin dam (no flow of water) if it intersects a flow_blocking_polygon or flow_blocking_line.
- For Flow_blocking_lines:
  - Flow_blocking_lines are modelled as Thin dams, In Baseline they are stored as (poly)lines and cannot be written as Dry areas.
  - D-HYDRO Suite treats a flow/net-link like a Thin dam (no flow of water) if it intersects a flow_blocking_polygon or flow_blocking_line.

Modelinformation on lines
Modelinformation on lines must be projected on flow/net-links. This done by D-HYDRO Suite. Bas2FM passes the information "as is" in the format of D-HYDRO Suite (*.pli).

Weirs
Elevated lines and terrain_jumps are modelled in D-HYDRO Suite as Fixed weirs.

Technical details:

- Input:
  - Elevated lines (routes and events)
  - Terrain_jumps (routes and events)
- Output – intended for use in D-HYDRO Suite:
  - <model name>_fxw.pliz
- Mark that D-HYDRO Suite requires the following definitions:

  - Crest level (m, NAP)
  - Sill height (left and right) (m, relative to the Crest level, positive down)
  - Slope-left and right (e.g. 4.0, which means 1 metre down on every 4.0 metres)
  - See D-HYDRO Suite documentation for file-format and definitions

Structures (Lines)
Barriers are modelled in D-HYDRO Suite as Structures.

Technical details:

- Input:
  - Structure_lines
- Output – intended for use in D-HYDRO Suite:
  - <model name>_structures.pli
  - See D-HYDRO Suite documentation for file-format and definitions

Bridges
Bridges are modelled in D-HYDRO Suite as bridge (pillars).

Technical details:

- Input:
  - bridges (routes and events)
- Output – intended for use in D-HYDRO Suite:
  - <model name>_bridges.pliz
- See D-HYDRO Suite documentation for file-format and definitions

Cross_sections (Lines)
Cross_sections are modelled in D-HYDRO Suite as Cross_sections.

Technical details:

- Input:
  - Cross_section_lines
- Output – intended for use in D-HYDRO Suite:
  - <model name>_<nr>_<type>_crs.pli
  - See D-HYDRO Suite documentation for file-format and definitions

Modelinformation on points
Again, like modelinformation on lines, 'Conversion to D-Flow Flexible Mesh'passes the information on points "as is" in the format of D-HYDRO Suite (*.xyn).

Lateral sources/sinks (points)
The location of lateral sources/sinks ('bronnen/putten') is written.

Technical details:

- Input:
  - Source_sinks_points
- Output – intended for use in D-HYDRO Suite:
  - <model name>_sources_sinks.xyn

    - See D-HYDRO Suite documentation for file-format and definitions

Output locations
Output_locations ('Uitvoerlocaties') are modelled as output location points

Technical details:
- Input:
    - Output_location_points
- Output – intended for use in D-HYDRO Suite:
    - <model name>_<nr>_<type>_obs.xyn
    - See D-HYDRO Suite documentation for file-format and definitions

Measuring stations
Measuring stations are modelled as output location points
Technical details:
- Input:
    - Output_location_points
- Output – intended for use in D-HYDRO Suite:
    - <model name>_<nr>_<type>_obs.xyn
    - See D-HYDRO Suite documentation for file-format and definitions

Kilometer locations
Kilometer locations (points) are modelled as output location points.
(Kilometer locations ('Rivierkilometerraaien', lines) will be converted to (D-Flow FM) Cross sections.)

Technical details:
- Input:
    - Output_location_points
    - cross_sections_lines
- Output – intended for use in D-HYDRO Suite:
    - <model name>_<nr>_<type>_obs.xyn
    - <model name>_<nr>_<type>_crs.pli
    - See D-HYDRO Suite / D-Flow FM documentation for file-format and definitions

Using a modelboundary
When a modelboundary is defined in the 'Conversion to D-Flow Flexible Mesh'the geometric data of a Baseline database can be partially converted to D-Flow FM.

Some remarks on the use of this modelboundary:

1. All input point data are clipped using this modelboundary, so only FM-output is written of data within the clip polygon.
2. All input polylines which are within the modelboundary or intersect the modelboundary are written to FM-output.
3. All nodes or cell-centers that are within the modelboundary are written to FM-output.

   a. At locations where the modelboundary is smaller then the Baseline database, elevation at nodes or cell-centers is also written to the <model name>_net.nc-file in a zone up to 250m from the modelboundary.
   b. At locations where the modelboundary is exeeds the Baseline database, elevation at nodes or cell-centers is also written to the <model name>_net.nc-file in a zone up to 250m from the boundary of the Baseline database.
4. All roughness information in the flow faces which are within the modelboundary or intersect the modelboundary is written to FM-output.

Appendix D: Bas2SWAN Methods and technical details

Overview
As described in chapter 4.6.5, Bas2SWAN (baswaq.exe) generates the following input data for the SWAN model. See chapter 4.6.5 for the directory structure.

Window: Baseline to SWAN		
	Baseline – used as input	**Generated SWAN input**
Convert boundaries	Section_polygons	<model name>_dry.pol <model name>_dry.thd[6]
Convert elevation in Grid netCDF	Elevation_model_terrain[7] Elevation_ raster Elevation_raster_land	<model name>.bot
Convert weirs	Elevated_line (routes and events) Terrainjump_line (routes and events)	<model name>.fxw
Convert areas outside flood zone	Flow_blocking_polygons[8]	<model name>.fxw
Convert flow blocking lines	Flow_blocking_lines[9]	<model name>.fxw
Convert output locations	Output_location_points	<model name>_0_all_obs.pnt *Plus files with individual feature class data.*

Method for elevation on cellnodes
For all gridpoints the elevation is determined by the standard ArcGIS- function to interpolate in a Terrain ("Add Surface Information"), the method used is "LINEAR" for land-databases and "BILINEAR" for sea-databases. This is compliant with previous Baseline releases. Tests have shown that the resulting elevation model for curvilinear (Waqua-)grids is almost similar: only at some limited areas there are differences.

Technical details:

- Input:
  - Elevation_model_terrain/ Elevation_raster/ Elevation_raster_land
- Output – intended for use in SWAN models:
  - <model name>.bot
- Bas2SWAN reads and uses the z_positive = 'up/down' from the <model name>_net.nc-file

Modelinformation on lines
Modelinformation on lines must be projected on grid cell boundaries.

[6] These files are generated by baswaq, they are not used in SWAN models
[7] Elevation _model_terrain in case of a land-variant, elevation_raster in case of a sea-variant, elevation_raster and elevation_raster_land in case of a merged sea-variant
[8] Flow_blocking_polygons are converted to polylines and added as fixed weirs with an elevation of 999 m
[9] Flow_blocking_lines are added as fixed weirs with an elevation of 999 m

Weirs/obstacles
Elevated lines, terrain_jumps, flow_blocking_polygons and , flow_blocking_lines are modelled in SWAN as Fixed weirs/obstacles.

Technical details:

- Input:
  - Elevated lines (routes and events)
  - Terrain_jumps (routes and events)
  - Flow_blocking_lines (vertices get elevation 999)
  - Flow_blocking_polygons (are first converted to polylines, then vertices get elevation 999)
- Output – intended for use in SWAN:
  - <model name>.fxw
- Mark that SWAN requires the following definitions:
  - Crest level (m, NAP)
  - Sill height (left and right) (m, relative to the Crest level, positive down)

Output locations
Output_locations ('Uitvoerlocaties') are modelled as output location points

Technical details:

- Input:
  - Output_location_points
- Output – intended for use in SWAN:
  - <model name>_<nr>_<type>_obs.pnt

Using a modelboundary
When a modelboundary is defined in the bas2swan conversion the geometric data of a Baseline database can be partially converted to SWAN.

Some remarks on the use of this modelboundary:

1. All input polylines which are within the modelboundary are written to FM-output.
2. All nodes or cell-centers that are within the modelboundary are written to FM-output.