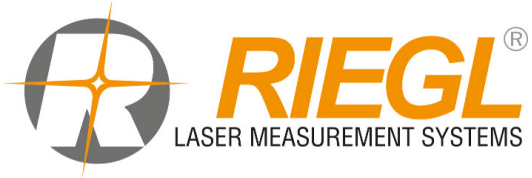
SDCImport



SDCImport

Printed: 02.05.2019.

Table of Contents

Part I

Copyright notice

1 Copyright notice

Part II

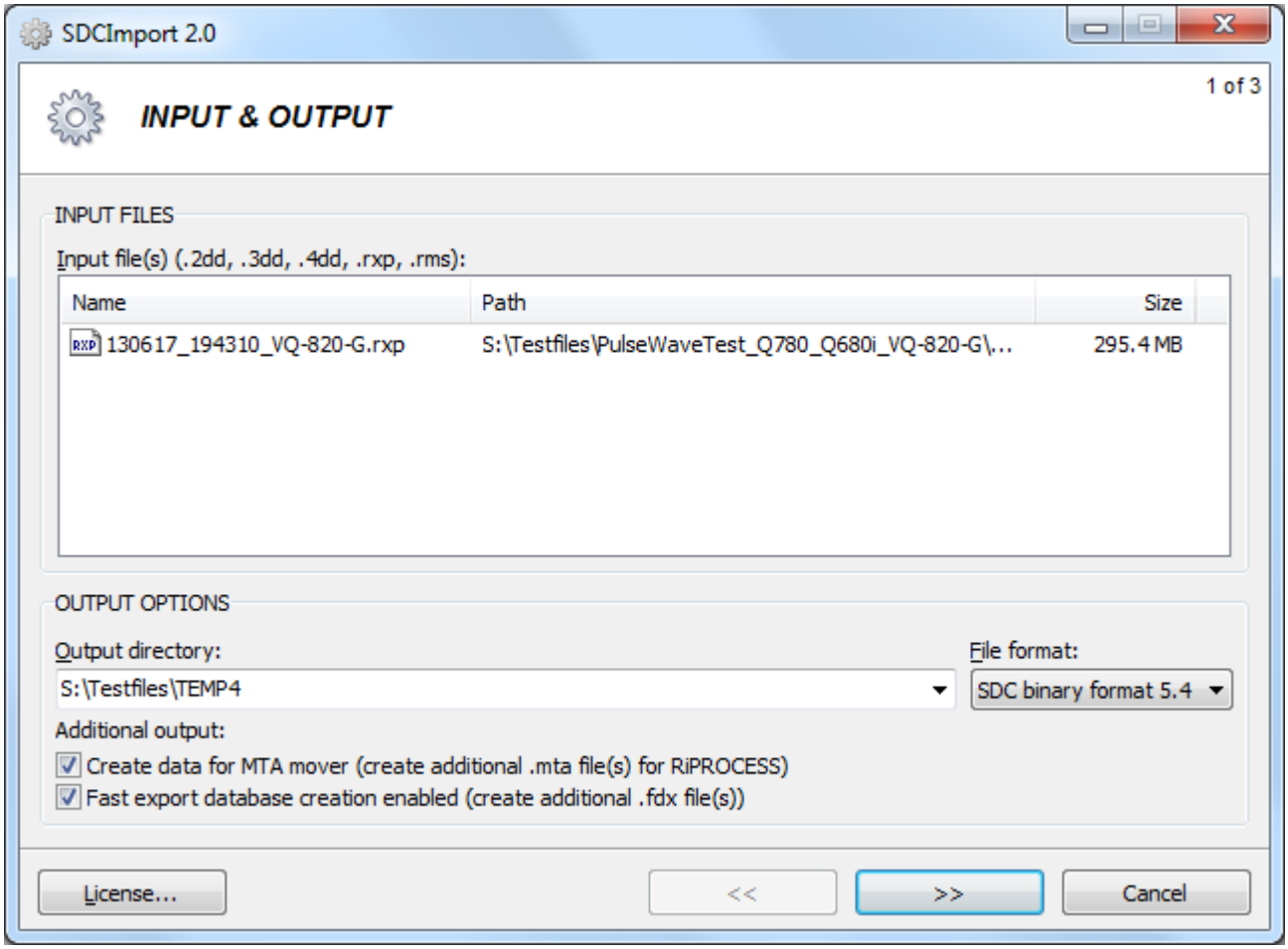
SDCImport

2 SDCImport

2.1 Introduction

SDCImport is a software tool for conversion of scanner data files (*.2dd, *.3dd, *.4dd, *.rxp and *.rms) files into the extracted data file format (*.sdc).
SDC files can be imported in RiPROCESS for further processing (e.g. converting into world coordinated *.sdw files using RiWORLD).

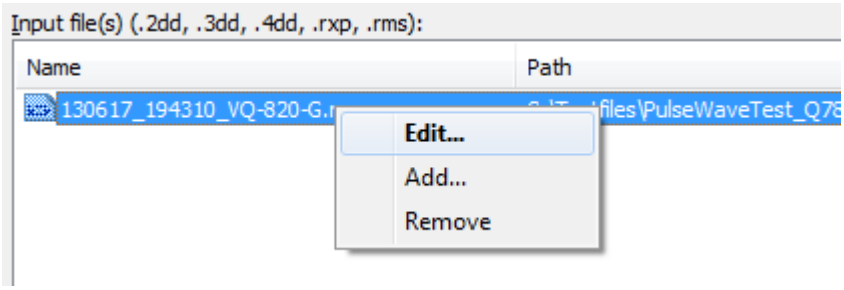
2.2 Input & Output



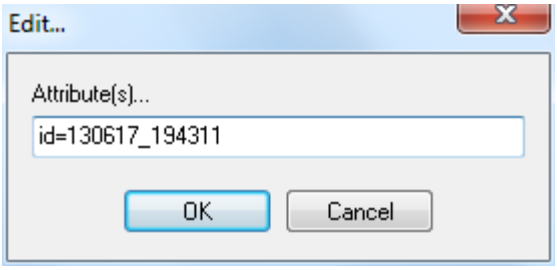
INPUT FILES

Select input data files using the context menu of the file list (right-click the file list), by hitting the Insert key or by drag & drop from Windows Explorer.

Note: .rms files (Riegl Multiplexed Streams) are containers for single or multiple .rxp file(s). If an .rms file is select for import, the name of the contained .rxp file must be specified and the resulting .sdc file is written to the selected Output directory (see below). Specifying the name of the desired stream is done by right-clicking the select .rms file and selecting "Edit...":



Now you can specify options for the selected file:



To extract a single .rxp stream, specify "id=<name_of_stream>" (see example above).

To extract a single segment from an .rxp file, specify "segment=1".

Options are are appended using "&" (ampersand): "id=130617_193411&segment=1".

OUTPUT OPTIONS

Output directory: Specify output directory (using option "Select..." or simply drag & drop the desired output directory from Windows Explorer) to save the converted .sdc files. When selecting "<same folder as input file>" the output files are stored in the same directory the input files are located in.

File format: You can select the desired output format. Please refert to appendix File formats 29 for a detailed description of supported file formats.

Create data for MTA Mover: When enabled, SDCImport creates a .mta/.rx5/.sodx file that can be used for manually moving targets into other zones (see MTA Mover tool in RiPROCESS).

Fast export database creation enabled: Since RiPROCESS version 1.8.0, fast data export option for LAS and PulseWaves (PLS) format is available. In order to speed up export temporary export databases (.fdx files) are created during processing. Activate this option to create these export data files.

Click ">>" to continue.

2.3 Extraction parameters

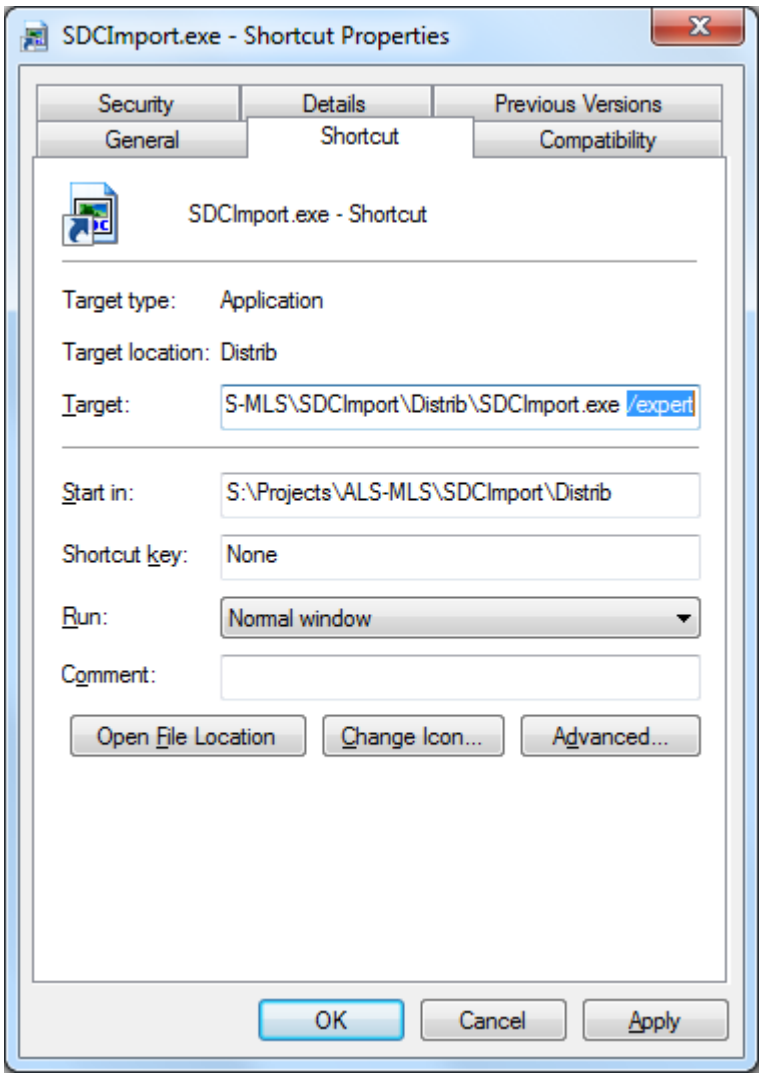
This chapter provides a detailed description of the extraction parameters for SDCImport.

SDCImport provides a special interactive mode named 'Expert mode' for the experienced user. In this operational mode advanced settings and configuration options are available, see Advanced parameters for experts 16.

In order to activate this expert mode, specifiy /expert on the command line when starting SDCImport:

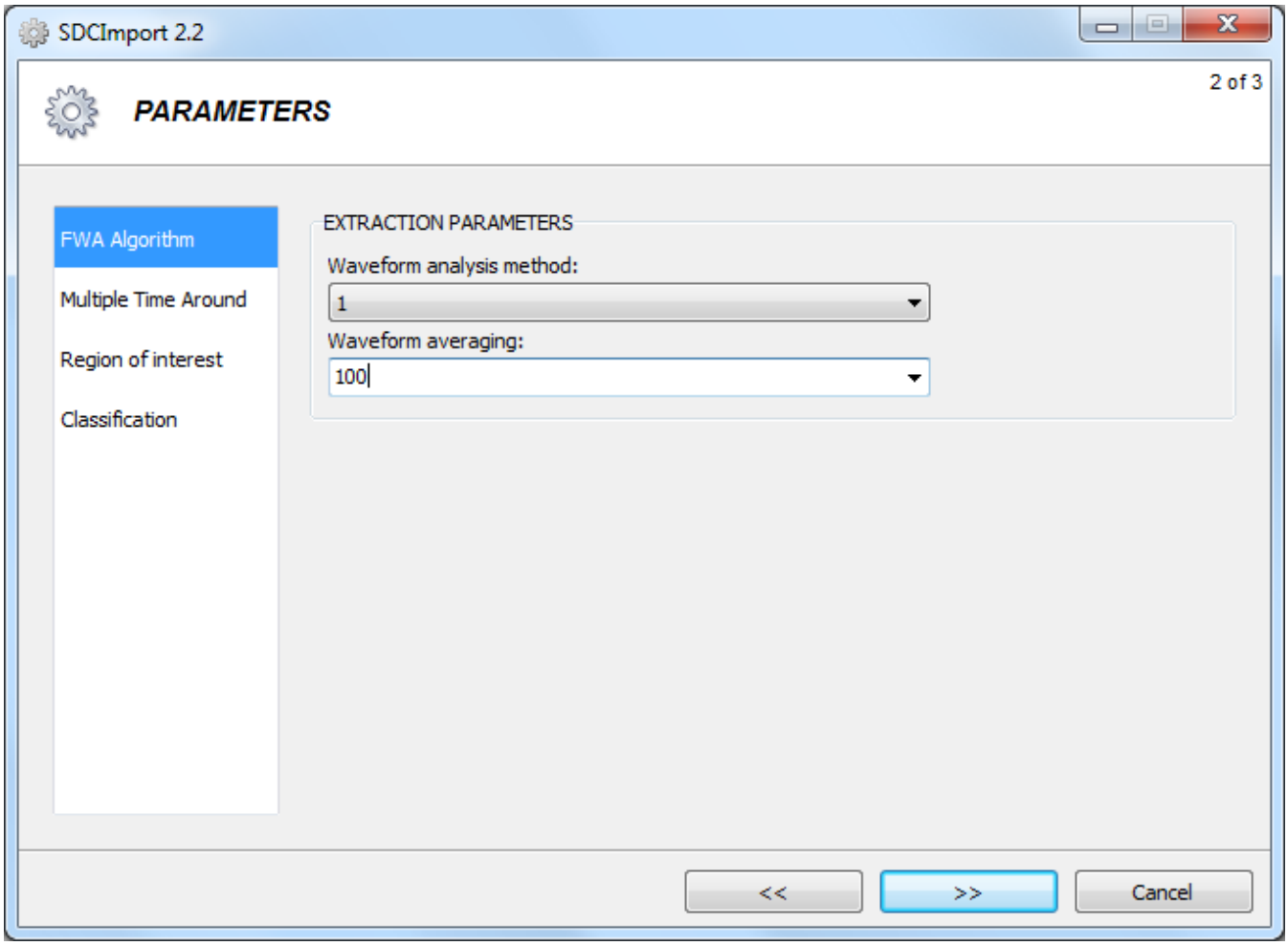
```
SDCImport.exe /expert
```

Of course, you can specify the /expert argument in the "Target:" field of the shortcut properties:



Hint: Don't forget the space between SDCImport.exe and /expert

2.3.1 FWA Algorithm



Waveform analysis method:

RIEGL provides different waveform analysis methods optimized for certain target situations:

- Disabled: Do not perform full wave analysis
- 0: Sequential analysis
- 1: Recursive analysis
- 2 ... For future expansion, currently do not use.

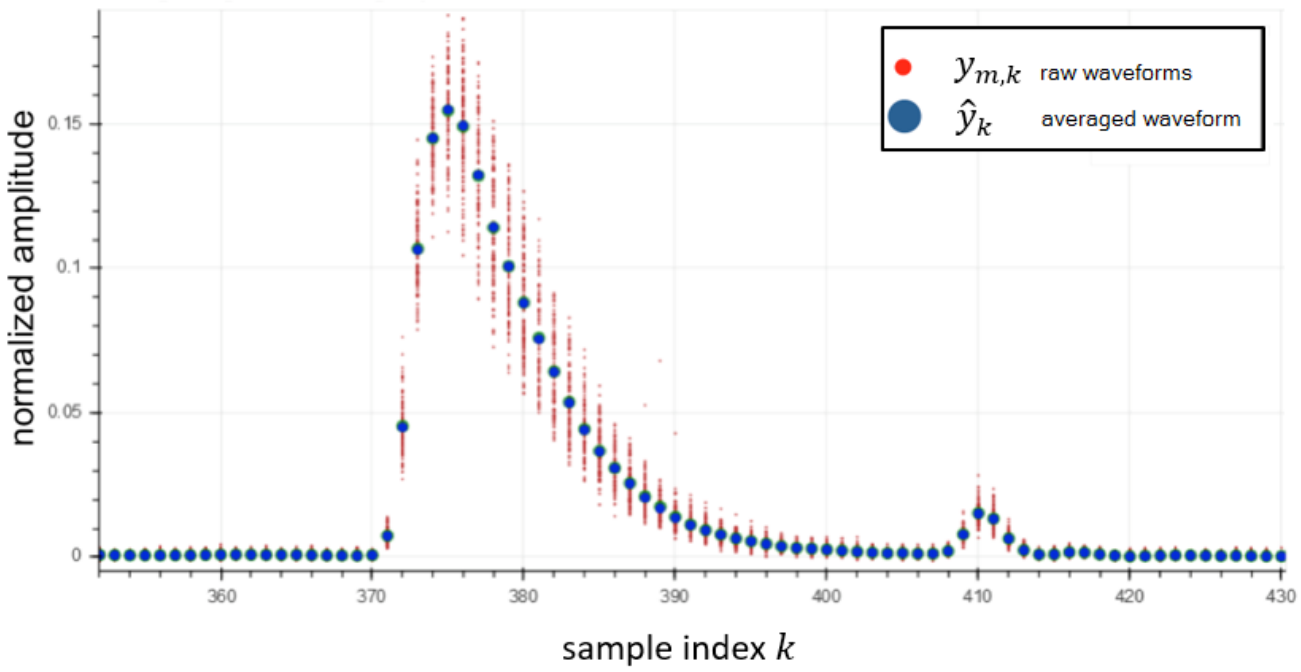
Waveform averaging:

When measuring profiles of waterbodies return signals from the ground typically are of very low amplitude and suffer from a small signal to noise ratio (SNR) causing - depending on the detector design - a high probability of false target detections or a low detection probability. Detection and target estimation are improved when averaging the received signals prior to detection. The key behind the method is that the incoherent noise variance σ^2 is reduced by the factor M, the number of measured and averaged waveforms:

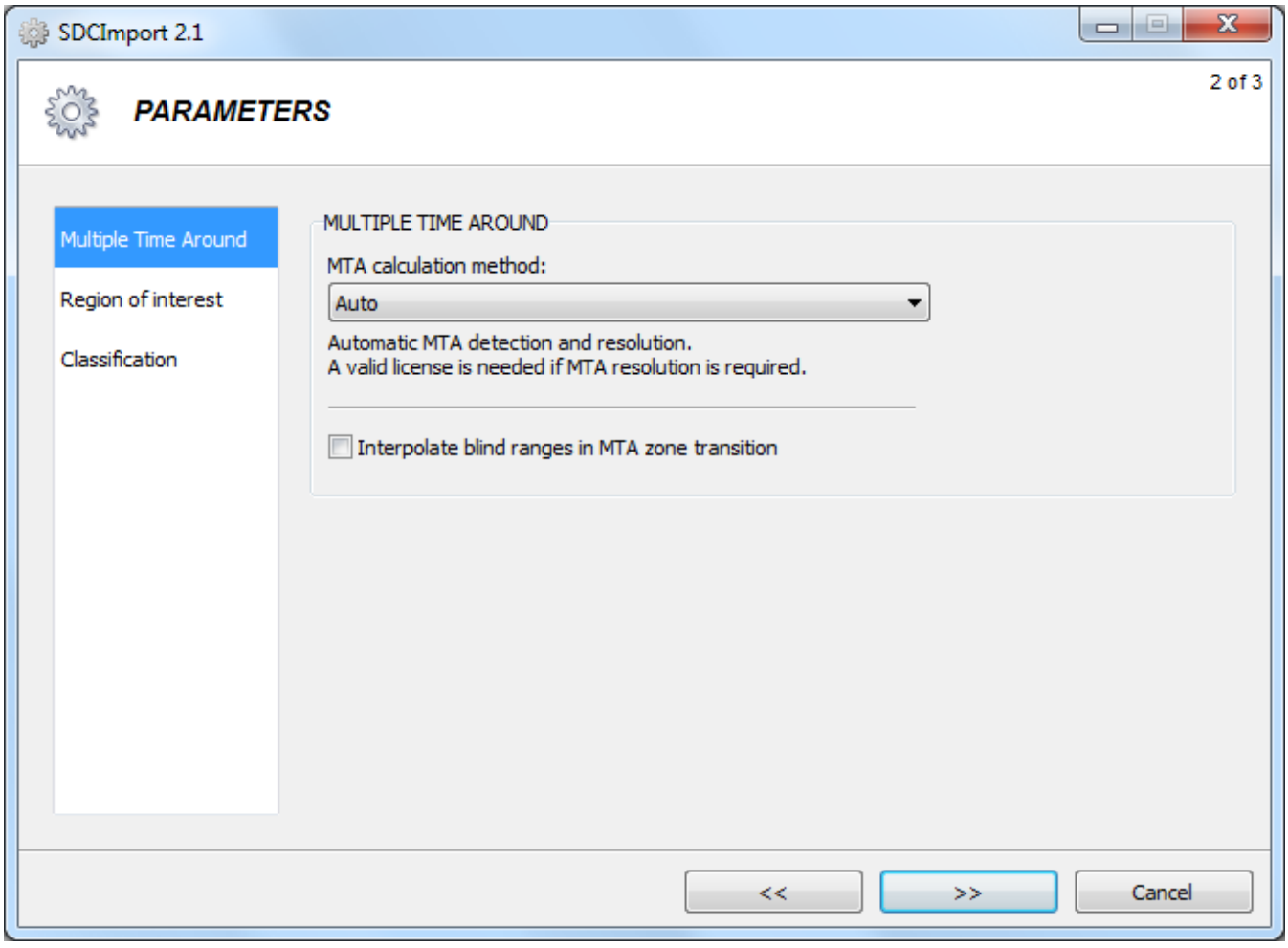
$$\sigma^2 AVG = \sigma^2 / M.$$

A prerequisite is that the target situation does not change significantly during the acquisition of the constituent waveforms.

Before applying waveform averaging outliers must be rejected to avoid that spurious spontaneous signals, caused by the detector, distort the averaged signal. The outlier rejection is implemented as a quantile filter.



2.3.2 Multiple Time Around



MTA calculation method:

- (none): MTA calculation is disabled. Data is delivered depending on scanner configuration.
- Auto: Automatic MTA detection and resolution. Uses RiMTA code-correlation or vicinity method for automatic

zone resolution if possible. Please note that a valid license is needed if MTA resolution is required.

- Fixed in zone: Process data using specified MTA zone.
- Semi-Autodetect: Process data using semi-automatic MTA zone detection. You must specify a range gate defining the minimum and maximum range.
- Start at zone: Process data using automatic MTA zone detection. In this case the MTA zone of the scan start must be specified.

For a detailed description of Multi-Time-Around processing please see Multi-Time-Around (MTA) 19.

Interpolate blind ranges in MTA zone transition: Enable this option to receive calculated targets in MTA zone transition.

Depending on the type of laser scanner short blind ranges may exist. These are caused by the fact that the range finder cannot emit a laser pulse and receive an echo at the same time. Therefore, blind ranges appear in the transitions of consecutive MTA zones which leave small gaps in the point cloud. These gaps have a typical pattern which again is dependent on the type of PRR modulation.

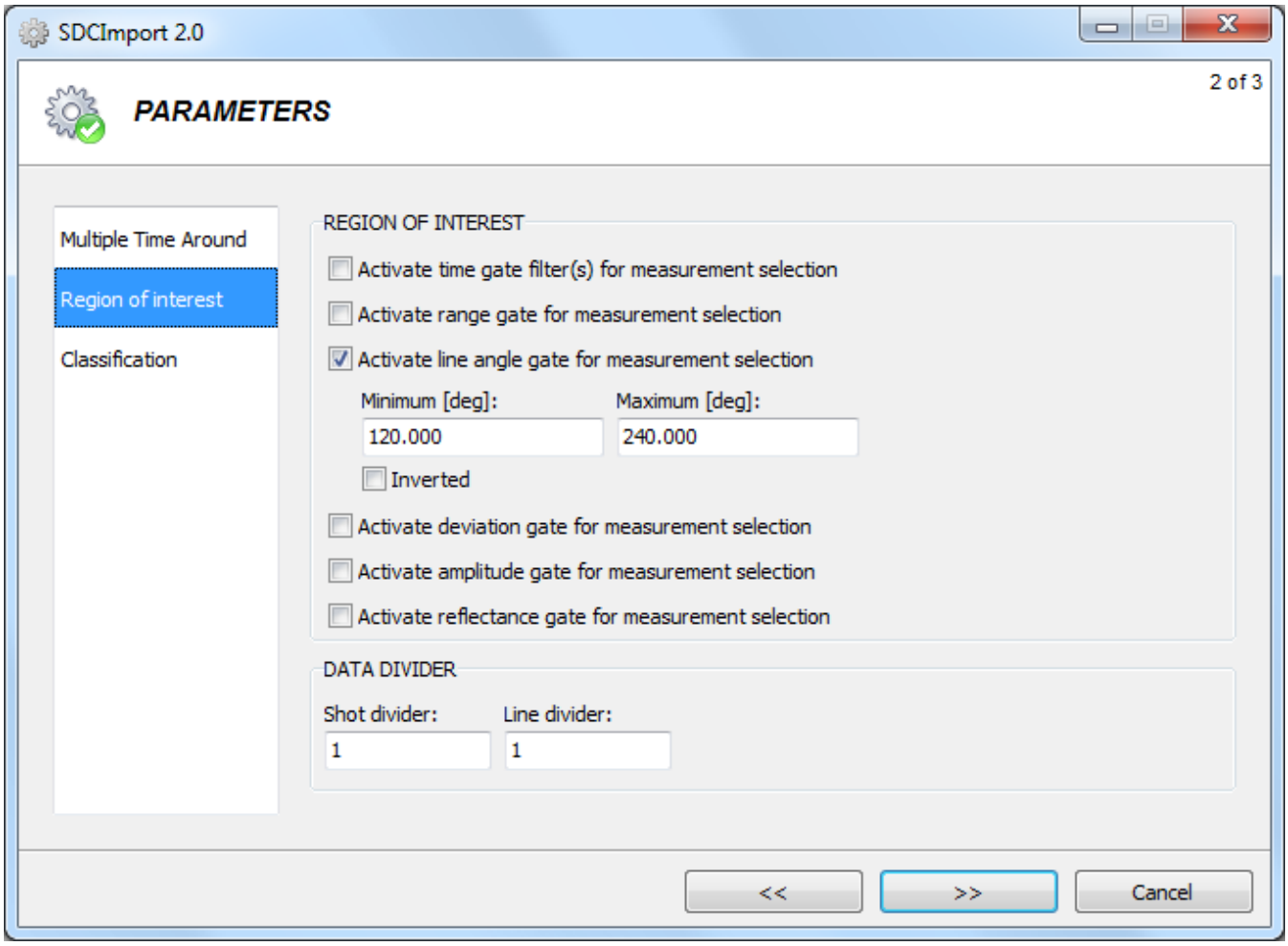
For post processing purposes SDCImport is able to interpolate coordinates within these blind ranges. The interpolation is done scan line by scan line, and considers valid (last) targets on both sides of each gap.

Interpolated coordinates in LAS-files are of class "Ground" (classification value 2, if an appropriate classification method is enabled) and flagged as “synthetic” in the classification bit field, according to the LAS standard.
In SDC- and SDW-files interpolated coordinates are of class "Ground" (classification value 2, if an appropriate classification method is enabled), according to the LAS standard 1.1, and flagged as "synthetic" in the channel descriptor field according to *RIEGL*'s SDC file standard.

The original raw scan data files (i.e. rxp or sdf-files) remain unchanged. Interpolated points are assigned a valid time stamp, which is essential for geo-referencing purposes, but do not carry any full waveform data, as well as no pulse width or pulse deviation values. Amplitude and reflectance values (if available) are calculated/averaged from surrounding points.

The intention of providing interpolated points is to alleviate tasks in point cloud post processing (e.g., terrain modeling).

2.3.3 Region of interest



Time gate filter

A region of interest is specified by a UTC timestamp to start at and a UTC timestamp to stop. You can filter the data to be processed and thus reduce the time needed for processing and the amount of result data. Only measurements between the start and stop timestamps are processed. UTC timestamps are specified in seconds since Sunday midnight (24:00 o'clock or Monday 0:00 o'clock). Please keep in mind that scan data time stamps also can be present in UTC day seconds (i.e. between 0 and 86400) and you must specify the regions using the same unit as the data is presented in the sample data file (*.sdf).
Note: The regions must be sorted ascending (start time stamps) and the regions must not overlap each other (start time stamp of region 2 must be greater than stop time stamp of region 1).
If no regions are specified all measurements are processed.
Additionally, activate the Inverted option if you want to process all measurements outside the specified time gate(s).

Hint: Add (Ctrl+Ins), delete (Ctrl+Del) or clear region(s) using the context menu (right-click).

Range gate filter

Activate the range gate filter for measurement selection and enter a minimum and/or a maximum range value [m] the targets have to fit in to be extracted and processed.
Additionally, activate the Inverted option if you want to process all measurements outside the specified range gate.

Angle gate filter

Activate the angle gate filter for measurement selection and enter a minimum and/or a maximum angle value [deg] the sum of theta angle and rho angle (if available) of the measurement has to fit in to be extracted and processed. In other words: If the sum theta+rho is out of angle gate filter the whole measurement (and all its

targets) is not processed.
Additionally, activate the Inverted option if you want to process all measurements outside the specified angle gate.

Deviation gate

Activate the deviation gate filter for measurement selection and enter a minimum and/or a maximum deviation value the targets have to fit in to be extracted and processed.
Additionally, activate the Inverted option if you want to process all measurements outside the specified deviation gate.

Amplitude gate filter

Activate the amplitude gate filter for measurement selection and enter a minimum and/or a maximum amplitude value [dB] the measurement has to fit in to be extracted and processed. Note that for .2dd/.3dd/.4dd files amplitudes are between 0 and 65535 (not dB).
Additionally, activate the Inverted option if you want to process all measurements outside the specified amplitude gate.

Reflectance gate filter

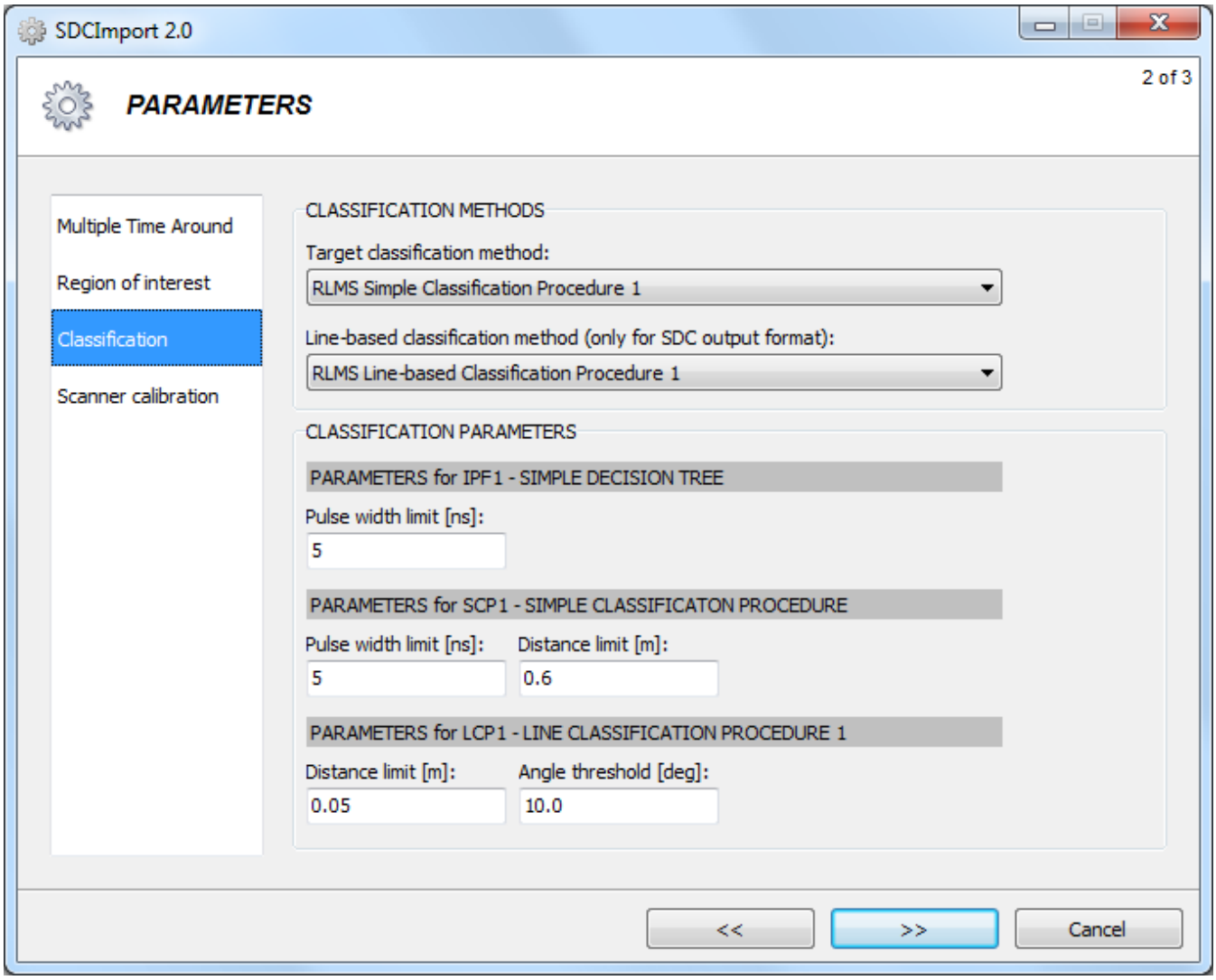
Activate the reflectance gate filter for measurement selection and enter a minimum and/or a maximum reflectance value [dB] the measurement has to fit in to be extracted and processed. Note that for .2dd/.3dd/.4dd files this filter is not available (no reflectance values delivered).
Additionally, activate the Inverted option if you want to process all measurements outside the specified reflectance gate.

DATA DIVIDER

Shot divider: Specify a shot divider (MSM) as stride for filtering measurement data. A shot divider of 1 (default) writes every measurement to the output SDC file. A shot divider of 2 writes every second measurement (and all its echoes) to the output file.

Line divider: Specify a line divider (MSM) as stride for skipping lines. A line divider of 1 (default) writes every line to the output while a shot divider of 2 writes every second scan line to the output file.

2.3.4 Classification

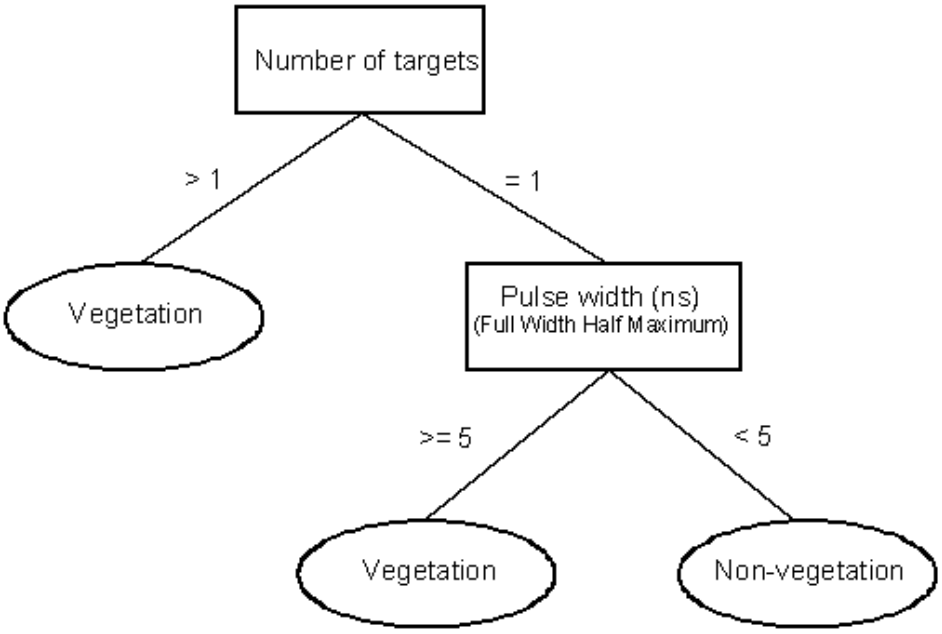


Please note: Classification is available only when MTA calculation is enabled!

Target classification method

- **IPF1 - Simple decision tree (89% probability of accurate target classification)**

  This simple classification algorithm distinguishes between vegetation points and non-vegetation points based on the number of targets (echo signals) and the pulse width:

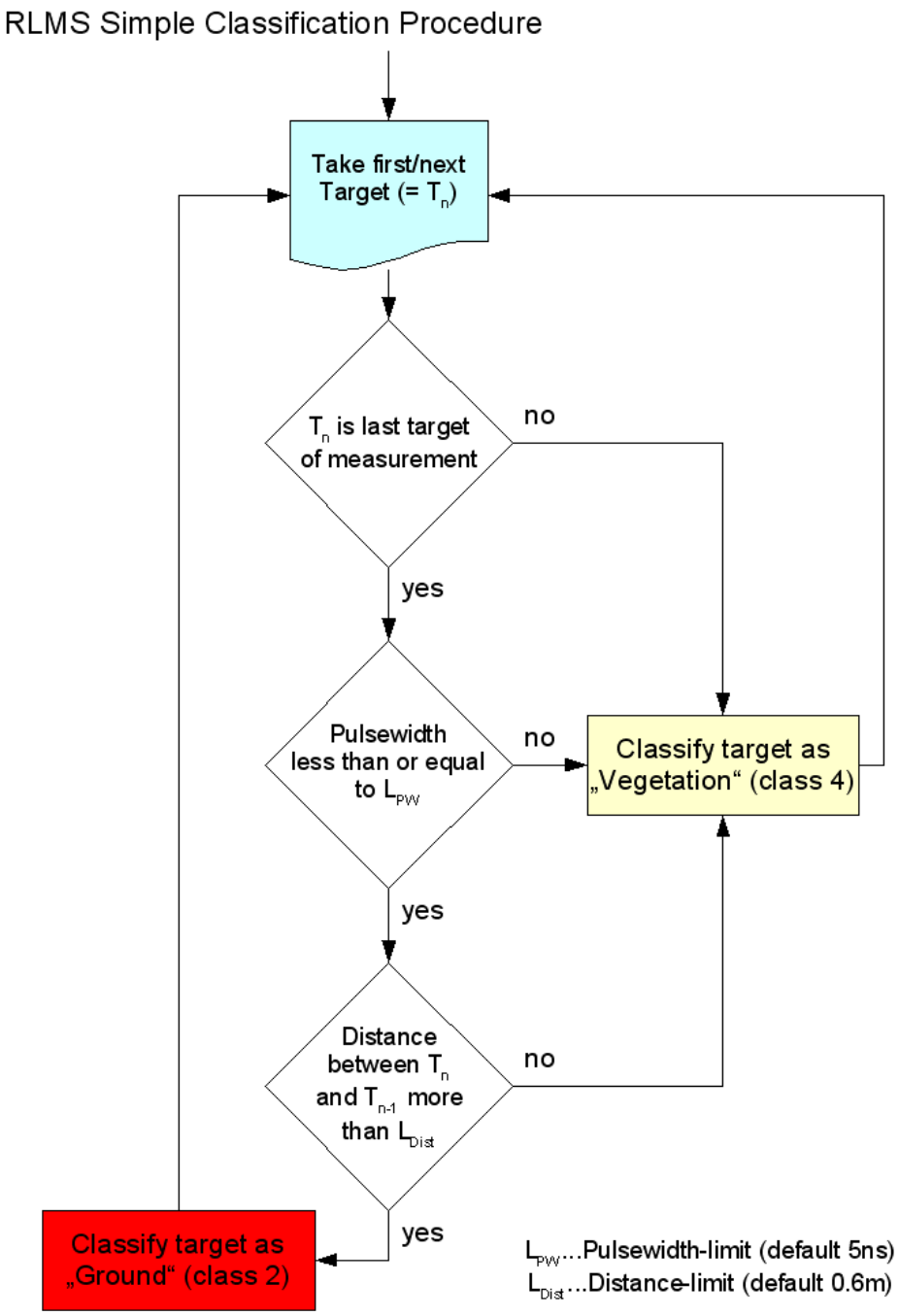


PARAMETERS for IPF1 - SIMPLE DECISION TREE
(available in expert mode)

- **Pulse width limit [ns]:** A target is classified as vegetation point if the number of targets is one and the pulse width is larger than the **Pulse width limit**. The Default value: 5ns.

- **SCP1 - RLMS Simple Classification procedure 1**

  This simple classification procedure distinguishes between vegetation points and ground points based on the number of targets (echo signals), pulse width and distance to previous echo signals (see scheme below).



PARAMETERS for SCP1 - SIMPLE CLASSIFICATION PROCEDURE
(available in expert mode)

- **Pulse width limit [ns]:** A target is classified as vegetation point if the pulse width of the last target is larger than the **Pulse width limit**.
  Default value: 5 ns

- **Distance limit [m]**: The second condition to be fulfilled is the distance of the last target T[n] to the preceding one T[n-1]: The distance of the last target to the preceding target must be greater than **Distance limit** in order to classify the target as ground point.
  Default value: 0.6

- **BDF Standard**

  For automatic classification of BDF-1 data select "**BDF Standard**". The points of a laser shot are assigned to either of the following classes:

- **water surface:** measurement points most likely resulting from echo signals from the water surface
- **ground:** measurement points most likely resulting from echo signals from the ground of a waterbody (seafloor, riverbed)
- **unclassified:** these targets can neither be assigned to water surface nor to ground; in most cases they can safely be ignored
- **never classified:** these are the targets processed by the online waveform processing of the BDF-1 sensor

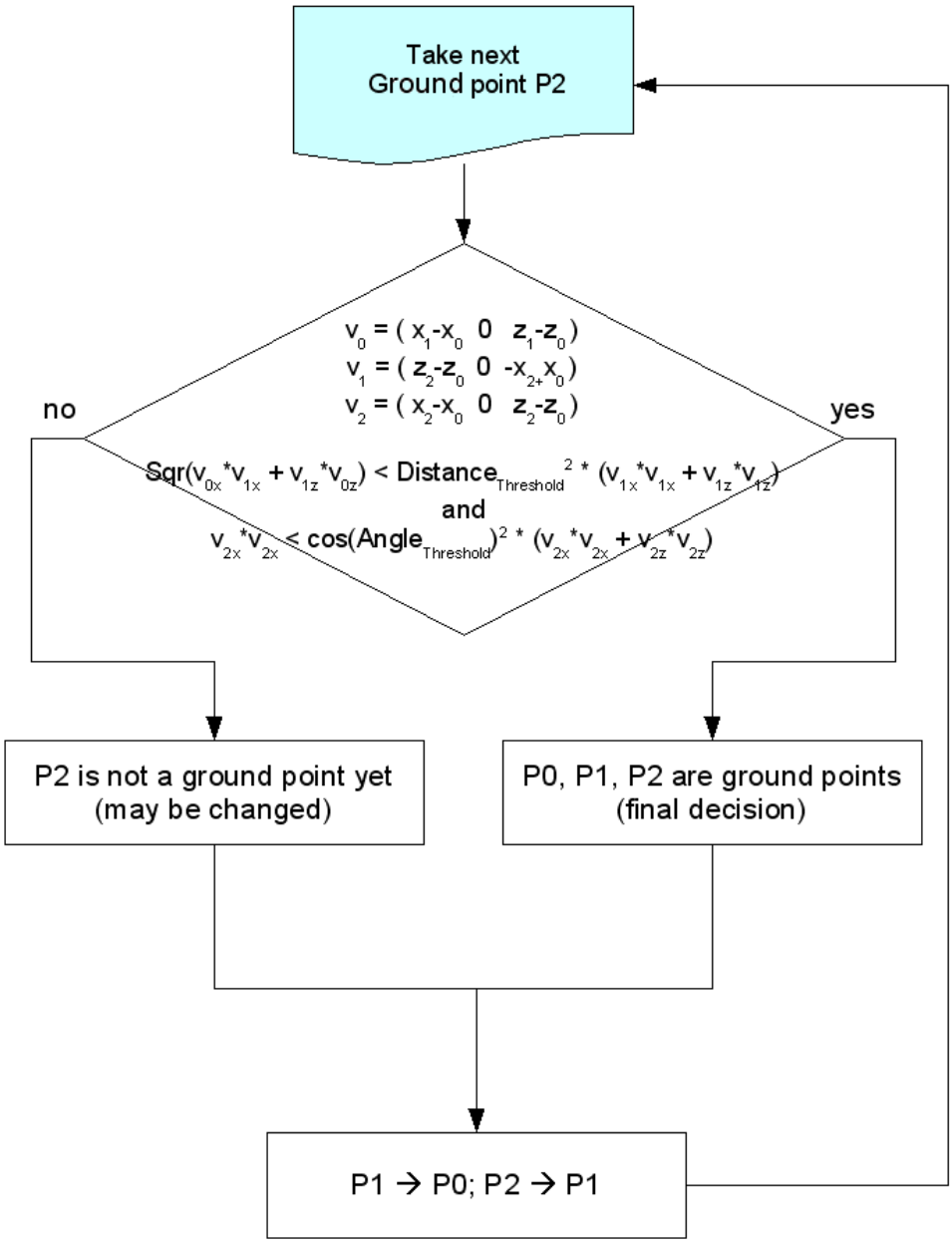
Note: Settings "**BDF Classification 0 to 3**" are for fine tuning the classification. They are performing differently for turbulent waters (e.g. rivers) and quiet waters (e.g. lakes). There is also a certain dependency on the chosen waveform processing algorithm method. Therefore it is advisable to find the combination of settings best suited for the project by trying them out on a small but representative segment of the data.

Line-based classification method

- **RLMS Line-Based Classification Procedure 1**

  Performs a simple re-classification of ground points (classified by IPF1 or SCP1) based on neighborhood relations within the same scan line. The last three ground points P2, P1 and P0 are taken into account.

RLMS Line Classification Procedure 1



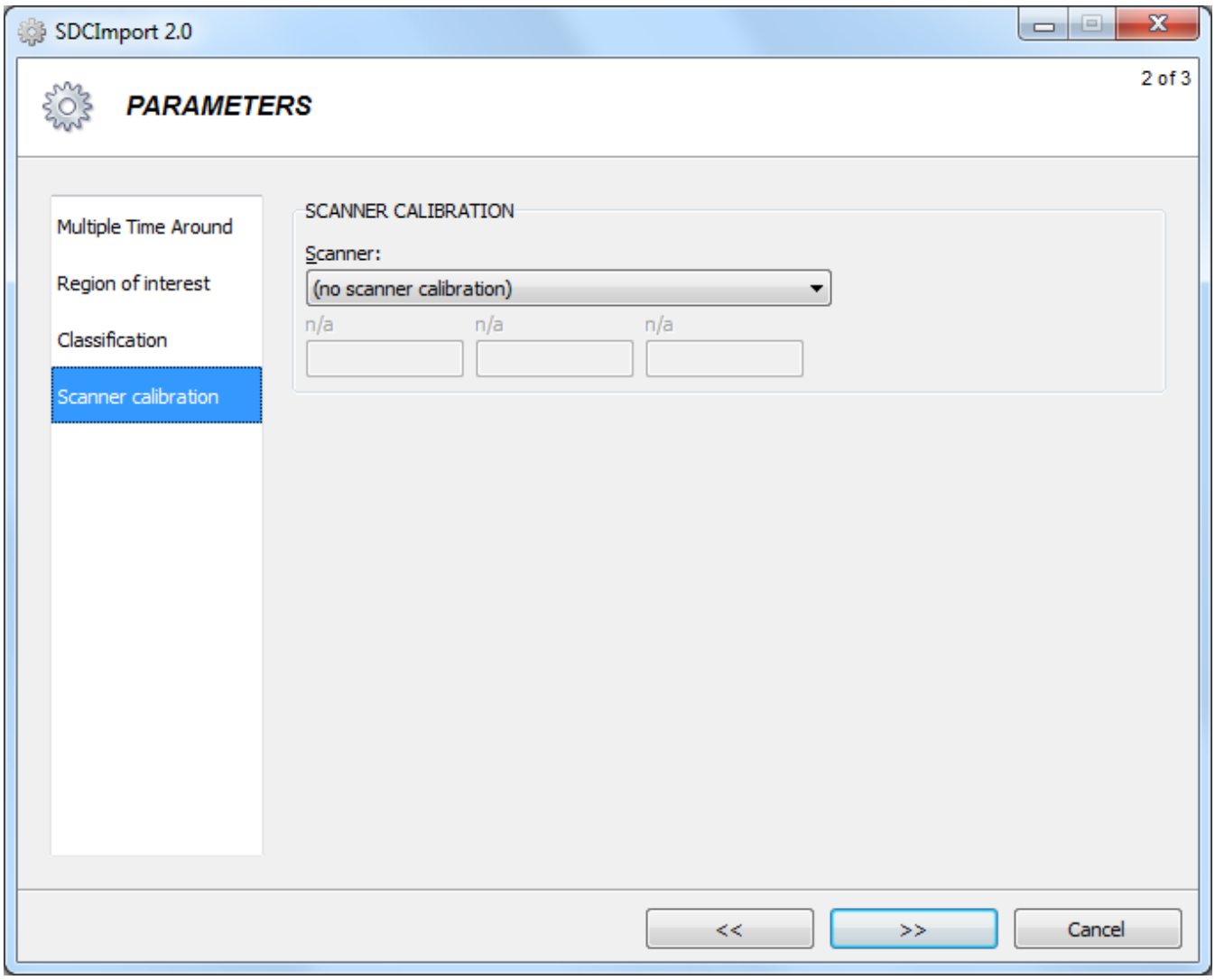
PARAMETERS for LCP1 - LINE CLASSIFICATION PROCEDURE 1
(available in expert mode)

- **Distance limit [m]**: Specifies the maximum distance of the ground points from each other. Default value: 0.03
- **Angle threshold [deg]**: Specifies the maximum angle three ground points are allowed to vary. Default value: 10.0

2.4 Advanced parameters for experts

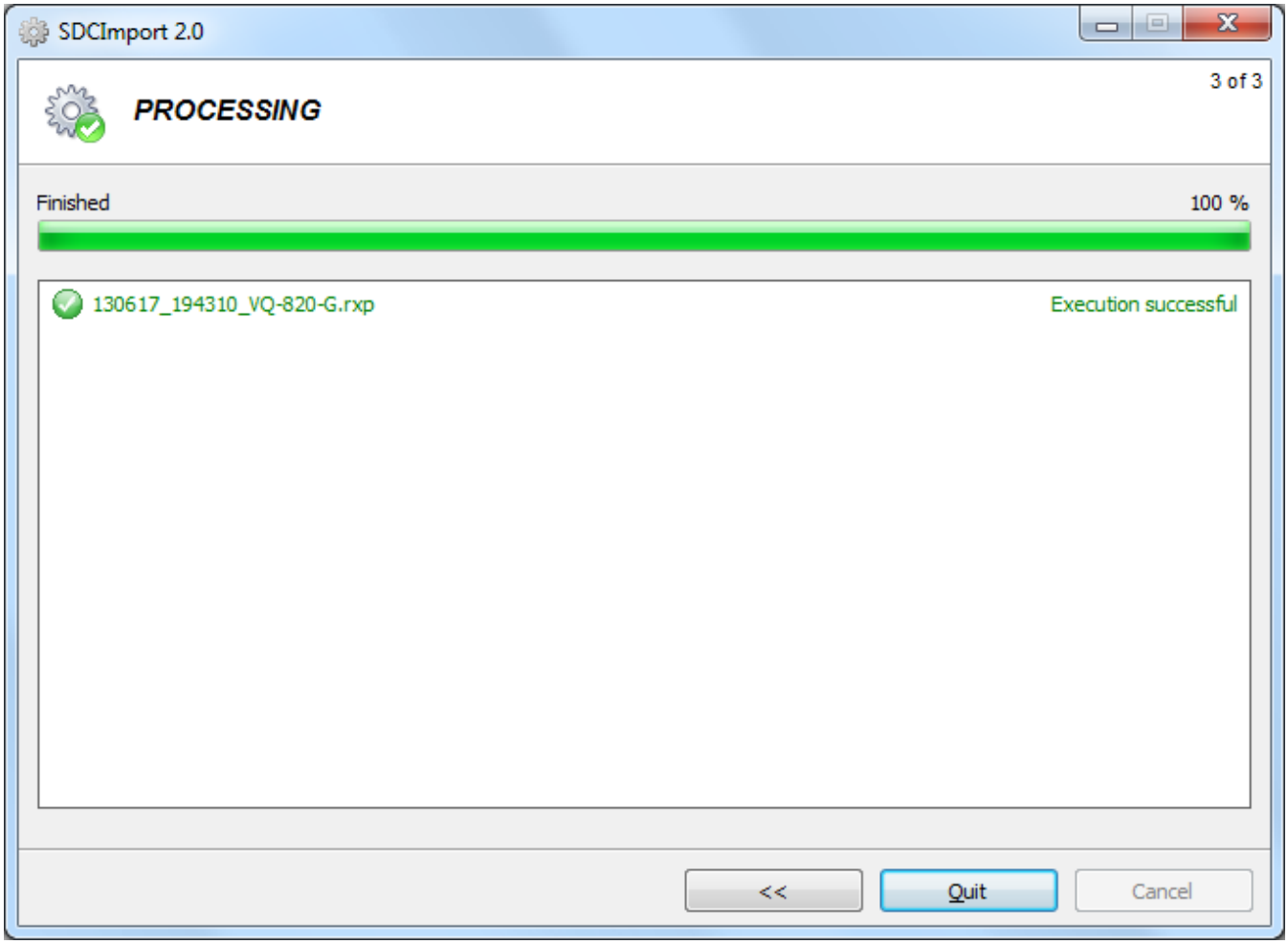
SCANNER CALIBRATION

The following settings are used by RLMS for analysis and available in expert mode only. Do not modify these settings without instructions provided by RLMS!



Depending on the scanner instrument type different calibration values can be specified.
Do not use this option unless instructed by *RIEGL* LMS.

2.5 Processing



This page shows the current state of the conversion process.

Please check the messages to make sure processing was successful.

2.6 Command line parameters

You may specify input files to be opened on the command line:

```
SDCImport.exe c:\scandata\010107_123456.rxp
```

You can specify multiple input files to be processed:

```
SDCImport c:\scandata\010107_100000.rxp c:\scandata\010107_102000.rxp
```

Hint: You may also define SDCImport as default application for opening .sdf files.

ADDITIONAL COMMAND LINE OPTIONS

```
/IgnoreConfiguration
```

ignore default configuration file (.ini), i.e. work with factory default parameters.

```
/Configuration=alternative.ini
```

use specified configuration file (.ini).

```
/Expert
```

enter expert mode. This option enables advanced parameter settings for fine tuning data processing.
This parameter is only relevant for interactive operation. When in remote controlled mode (used by RiPROCESS), expert mode is automatically used.

Note: If a command line argument contains spaces you have to enclose the argument in quotation marks ("..."):

```
SDCImport.exe "c:\my files\010107_123456.rxp"
```

IMPORTANT INFORMATION

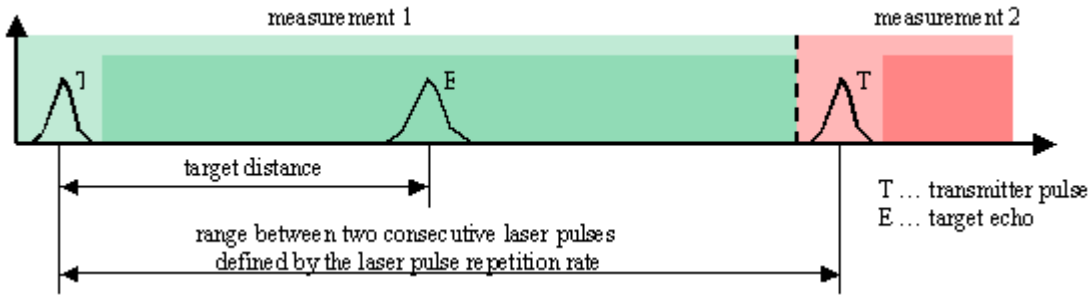
Since version 2.x the SDCImport installation contains a command line tool named sdcrun.exe for advanced batch processing.

On the command line, type
```
sdcrun /?
```
for detailed information on available parameters.

Note: The /expert command line argument is not available because sdcrun.exe always runs in expert mode!

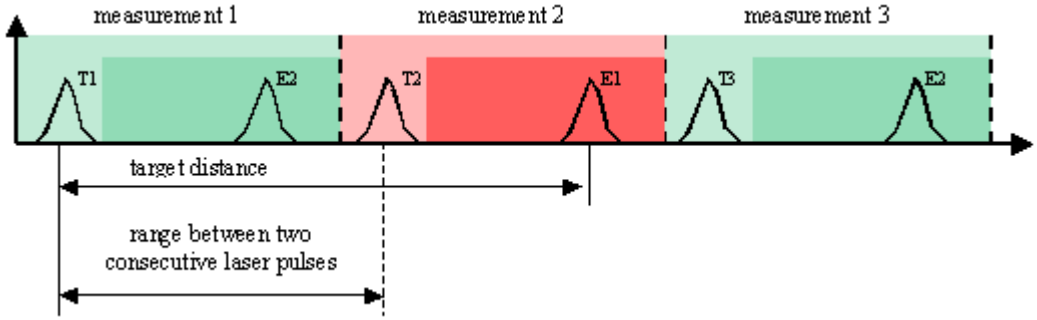
2.7 Multi-Time-Around (MTA)

Multi-Time-Around processing allows the uilization of target echo signals, which have been detected out of of the unambiguous range between two successive laser pulses.



The sketch above shows a situation where target distances are determined measuring the distance between the transmitter pulse and the target echoes of the same measurement.

Multi-time-around processing offers the possibility to determine the target distance of echo pulses, located in one of the following measurements.

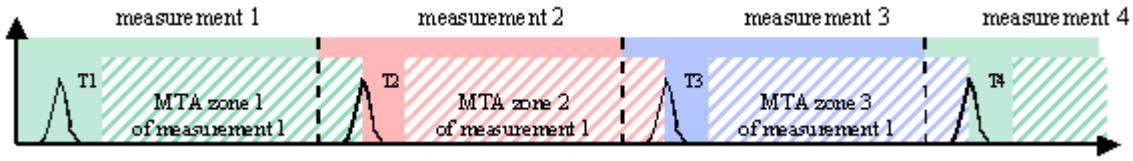


In the sketch above you see a situation with multi-time-around processing. The target distances are determined between the transmitter pulse of one measurement and the target echoes of the next measurement.

If the target echoes of the transmitter pulse of a measurement are returned in the **same** measurement, the echoes are in the **MTA zone 1**.

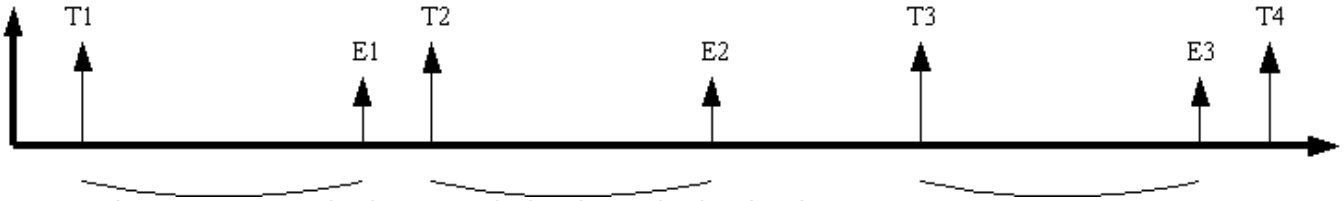
If the target echoes of the transmitter pulse of a measurement are returned in the **next** measurement, the echoes are in the **MTA zone 2**.

If the target echoes of the transmitter pulse of a measurement are returned in the **next but one** measurement, the echoes are in the **MTA zone 3**.



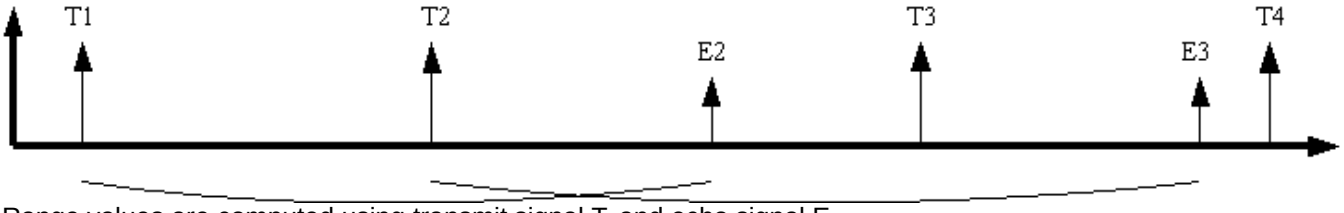
The MTA zone where the target echoes are expected must be configured for post processing with SDCImport. Possible values for MTA zone parameter are 1 to 8.

Example with MTA zone = 1:



Range values are computed using transmit signal T_i and echo signal E_i.

Example with MTA zone = 2:



Range values are computed using transmit signal T_i and echo signal E_{i+1}.

There are some restrictions for correct operation of the multi-time-around processing:

- All appearing target echoes should be in the configured MTA zone.
- Target echoes should not occur before the transmitter pulse of following measurements.
- Strong target echoes in the high power channel should not overlay with the transmitter pulses of next measurements.

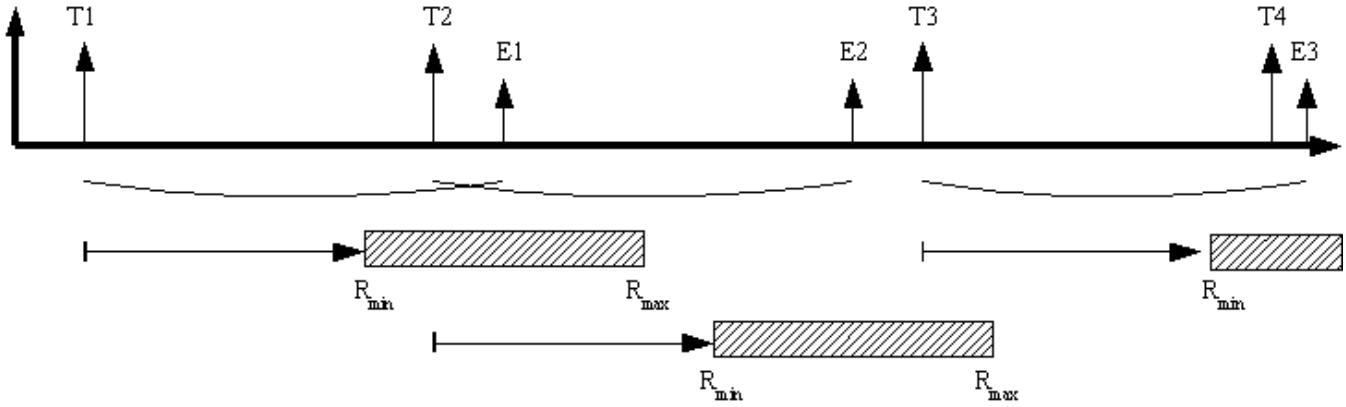
To have the ability to also handle acquired data where the restrictions above are not fulfilled, the pulse repetition rate of the laser is alternated with each second measurement, so that the range between to two consecutive laser pulses varies.

Semi-automatic detection of MTA zone

SDCImport offers a semi-automatic MTA zone detection mechanism for processing. By specifying the minimum and maximum range for a scan data file SDCImport tries to detect the proper MTA zone for every target echo. This mechanism is especially useful if the range values of the targets switch between two MTA zones.

To activate this mode, specify **MTA calculation method** = "Semi-Auto (range based)" and define a **Minimum range [m]** and **Maximum range [m]**.

Example with MTA zone = 0, R_{min} (Range gate minimum) and R_{max} (Range gate maximum):



Range values are computed using T_i and echo signal E_j where the range value must be between R_{min} and R_{max}.

There is a restriction for usage of this mechanism: The range gate (difference of Range Gate Maximum and Range Gate Minimum) must not exceed the range gate of the MTA zone depending of the setting of the laser pulse repetition rate (PRR) for the scan.

Example: Data acquisition with Laser pulse repetition rate = 100000 Hz means that Range Gate Maximum minus Range Gate Minimum must not be greater than 1470 m else the MTA zone cannot be detected properly.

Automatic MTA calculation (start zone needs to be specified)

By specifying the MTA zone for the scan start (i.e. the first measurements of the scan data file) the MTA zone is automatically changed during processing if required so.

To activate this mode, specify **MTA calculation method** = "Start at zone:" and **Zone** = 1, 2, 3, ... Note: Zone specifies the MTA zone to start with.

Full automatic detection of MTA zone (RiMTA code-correlation or RiMTA5 vicinity)

Full automatic MTA zone detection mechanism RiMTA (code-correlation).

To activate this mode, specify **MTA calculation method** = "Auto". If supported by the instrument, "RiMTA (code-correlation)" or "RiMTA5 (vicinity) method is selected.
Please note that a valid license is required to use RiMTA (code-correlation).

2.8 Requirements

Operating system:

Windows 7 Professional, 64 bit operating system.

Note: Please ensure that you have up-to-date device drivers installed (especially for the graphic card).

Memory requirements:

4 GB RAM minimum.

Disk space requirements:

Approx. 125 MB of free disk space for the program.

Hardware requirements for GPU processing:

NVIDIA® CUDA Compute Capability Version 3.2 or higher required (NVIDIA® GPU Geforce 8xx or better).

Note: GPU is used by RiMTA (vicinity) introduced in SDCImport 2.1

Part III

Appendix

3 Appendix

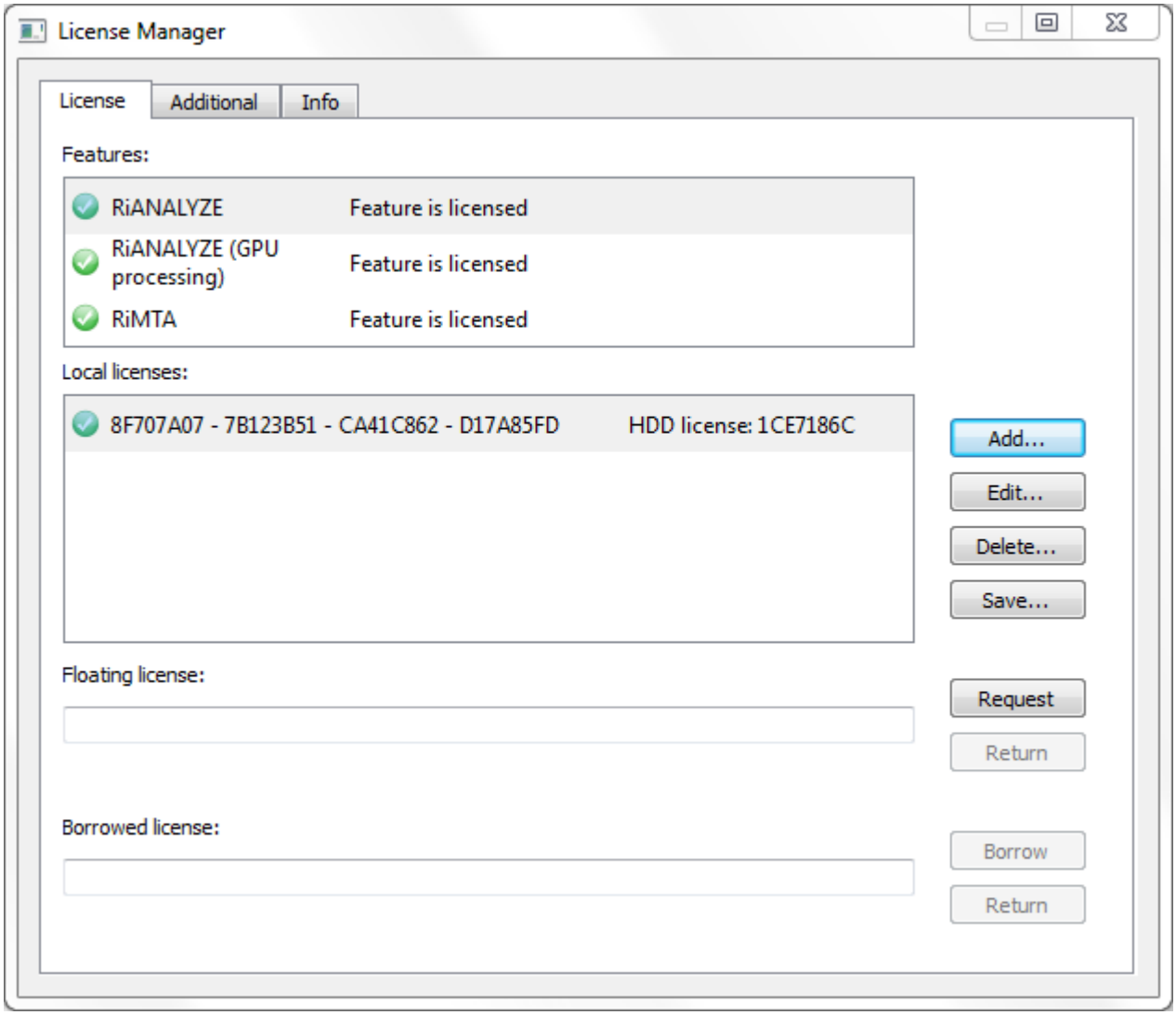
3.1 Licensing

3.1.1 License management

Please refer to the software product's manual/documentation on where to find the product's license manager window. Typically you will find it in the "Tool" menu.

- **Managing licenses**

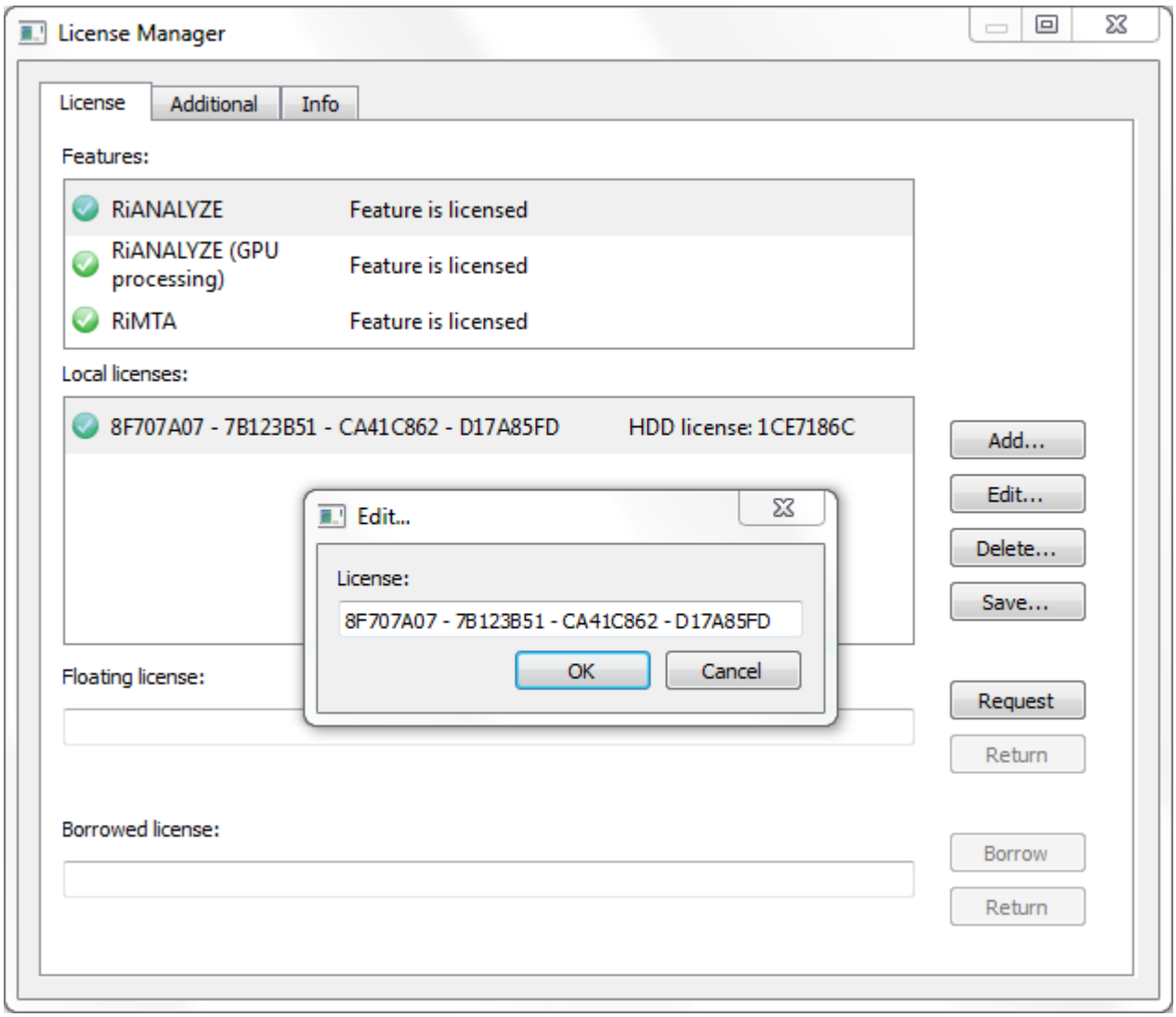
For an application, licenses for different features may be required. In the example below you can see three available features requiring a license: "RiANALYZE", "RiANALYZE (GPU processing)" and "RiMTA". Click on a feature to see all available local licenses:



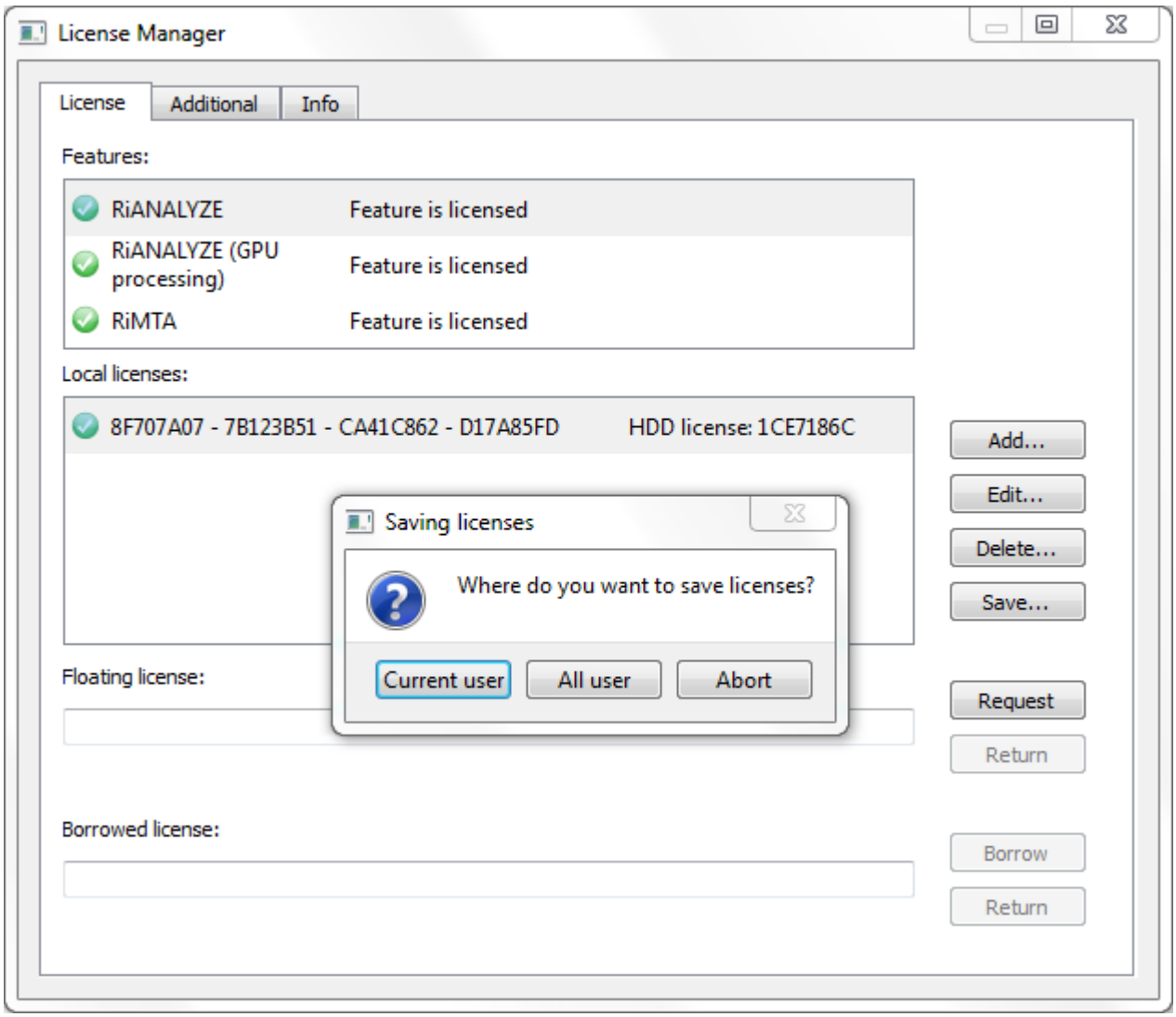
In the example above you can see that RiANALYZE is licensed for harddisk id 1CE7186C. Only licenses stored in the local registry are used yet.

- **Local licenses**

You can add, edit and delete local licenses by using the buttons "Add…", "Edit…" or "Delete":



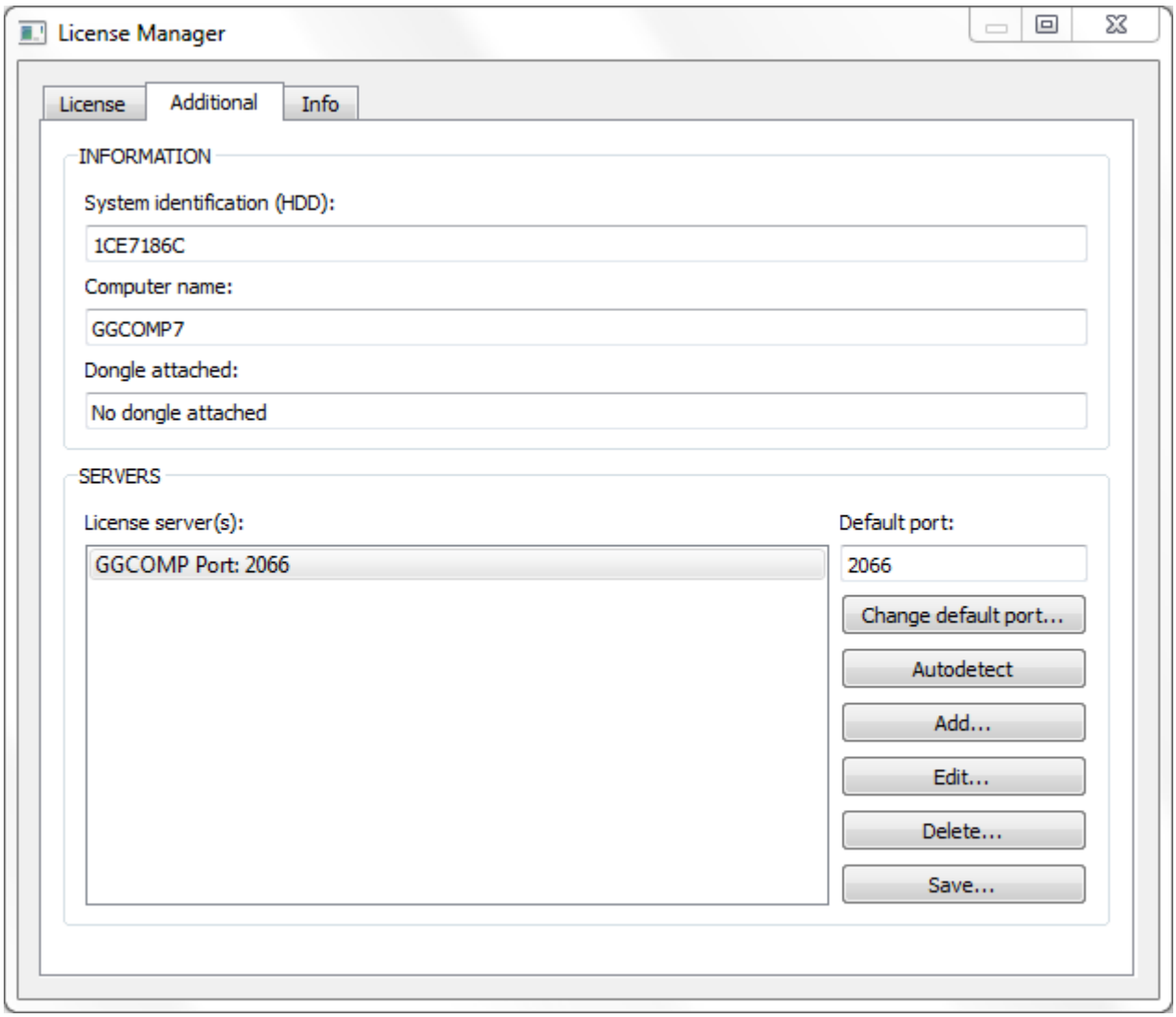
After modifying the license keys, you must not forget to save your changes to the registry. Click "Save..." and select "Current user" or "All users" to save the license keys for current user only or all users on this machine respectively:



If there is no valid local license available for a certain feature, a floating license (see below) may be requested automatically as soon as the feature is activated.

- **Floating licenses**

The *RIEGL* LicenseServer distributes the stored licenses to other *RIEGL* software products requiring a license key. After the LicenseServer's startup it accepts requests for the license key from the client applications. When a *RIEGL* software product requiring a license key is started, it checks the local registry for license keys. If no valid license could be found on the local machine it contacts the LicenseServer for a key. The application needs to know the name (or IP address) of the LicenseServer machine and the port number where the LicenseServer is listening.



Use the "Add..." button add a server to the list. You can change the “Default port” and click “Autodetect” instead. The auto detection feature should show all available LicenseServers in the current network.

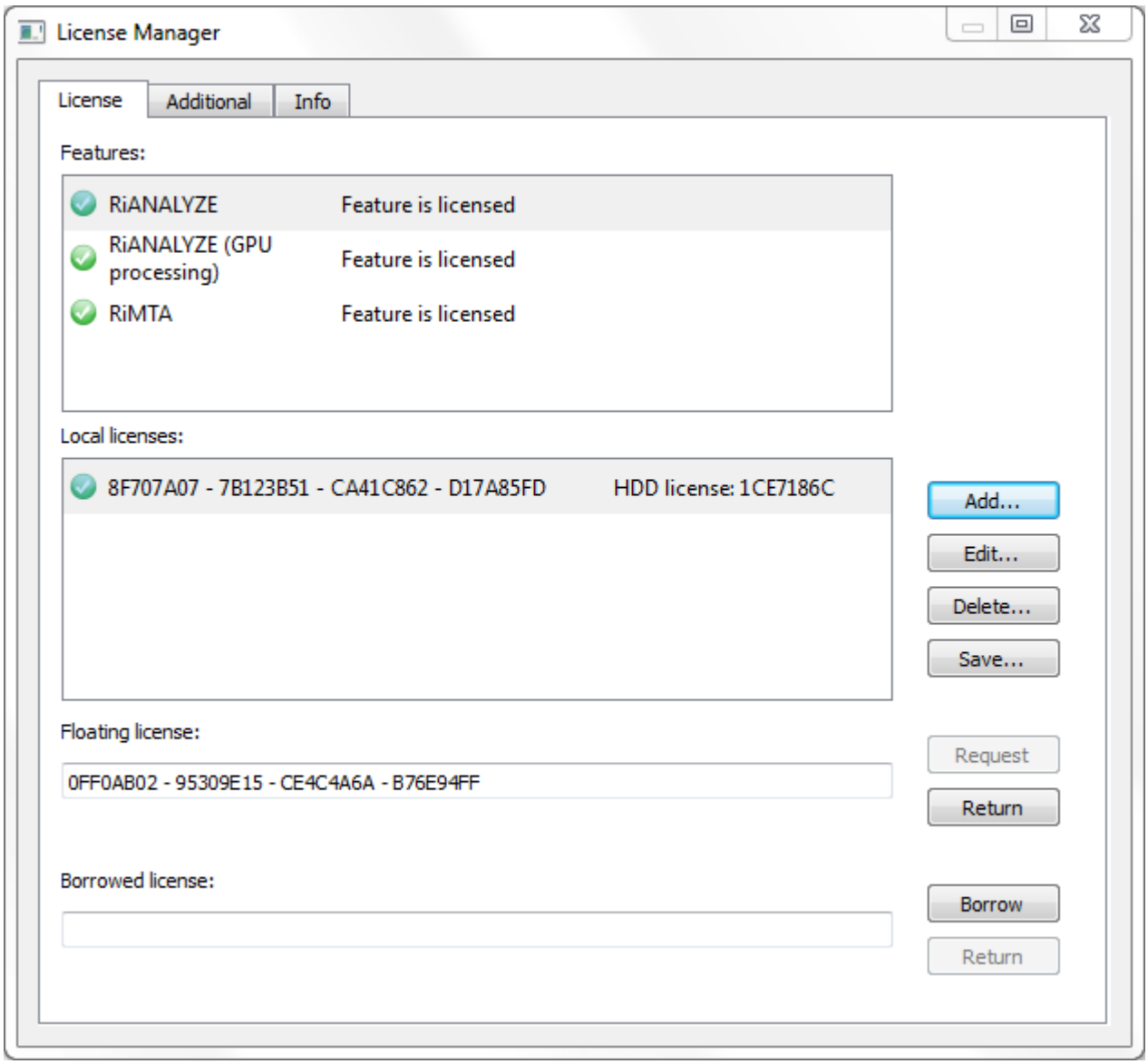
Click “Save...” to save the servers to the registry so they are available for all other modules/products requiring a floating license.

If the application reports that no LicenseServer was found, you need to check the settings of the application and configure the LicenseServer to be contacted. When restarting the application the connection to the LicenseServer should be successful.

Floating licenses are acquired automatically when a feature requires a license and there is no local license available.

Now we will see how to manually request a floating license:

Click “Request" to contact a LicenseServer from the server list to receive a floating license. If the operation is successful, you received a valid license key:



The next section explains how to borrow a floating license for offline-usage (without a network connection, e.g. laptop).

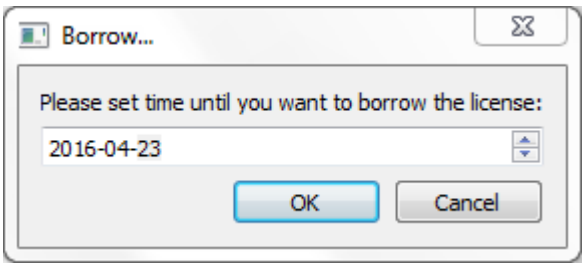
- **Borrowing licenses from a LicenseServer**

In case you want to use a software product requiring a floating license without a network connection (e.g. on a laptop), you can 'borrow' a license from a LicenseServer. To use this feature you need a dongle for the license you want to borrow. Please contact *RIEGL* Laser Measurement Systems for 1 or more dongles. You can use these dongles for all *RIEGL* software products.

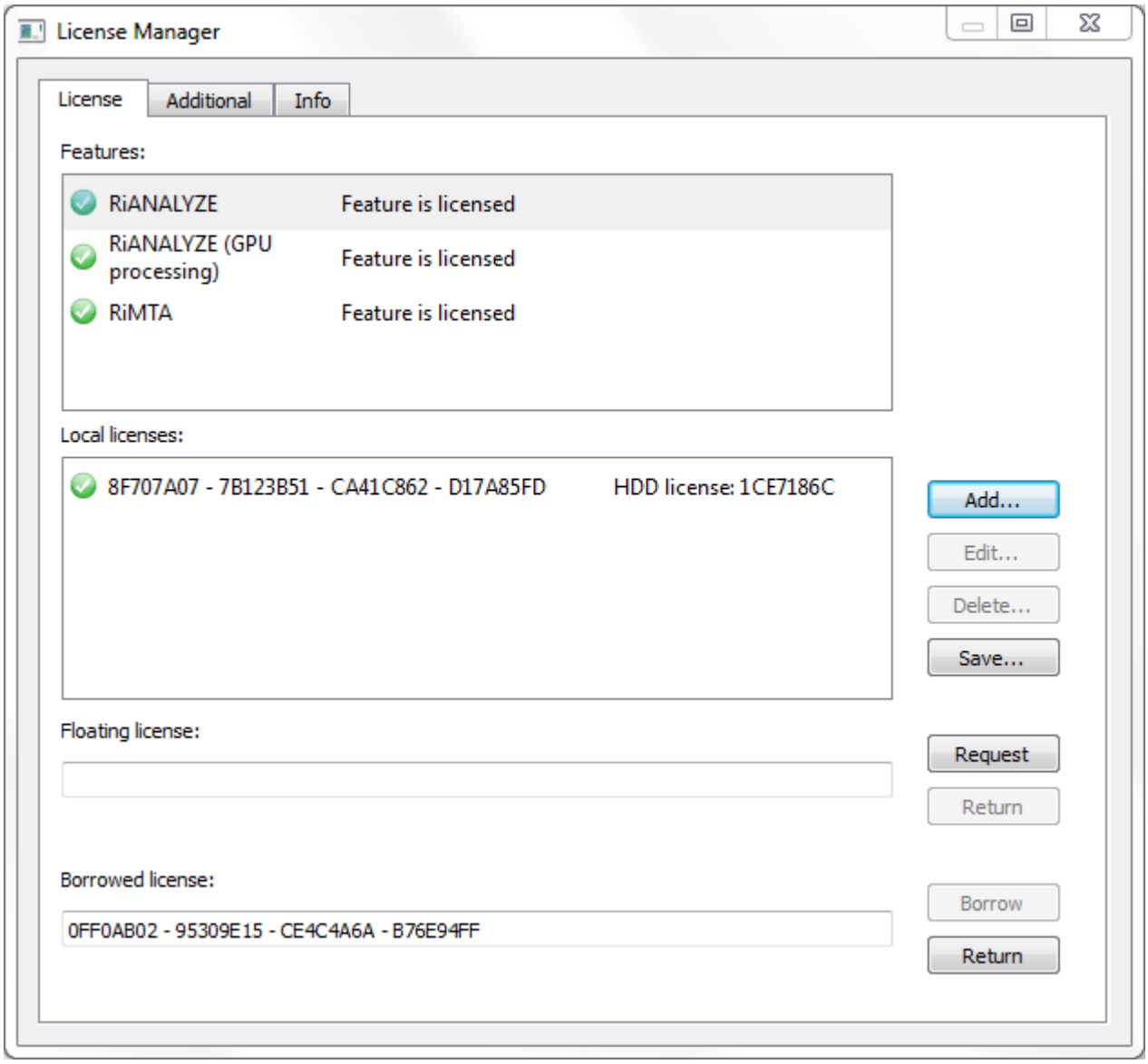
The required license key is requested from a LicenseServer while you are connected to the network. Then you can borrow this license for a specified time (max. 4 weeks). A *RIEGL* dongle must be attached to your laptop. The license is then moved to this dongle.

When offline, you need this dongle (now containing the license) attached to your laptop. After the specified time, the license expires and the LicenseServer allows you to use the license again. Of course you can return the license from the dongle to the LicenseServer at any time.

Click "Borrow" Now you are asked to enter the date (+time) until when you want to use the license offline (max. 4 weeks):



Make sure a dongle is connected and click “OK”.



We can now see that the license has been transferred to the attached dongle.

Note: You can store up to four licenses on a single dongle.

As long as the dongle is attached to your laptop and the license(s) on the dongle have not expired you can use the application without network access. The dongle simply acts as a container for licenses. Expired licenses on a dongle are automatically overwritten when borrowing new licenses.

- **Returning a license to a LicenseServer**

If you want to return the license, simply click “Return”. If the LicenseServer was contacted and accepted the returned license, it will disappear from section “Borrowed license” field.

Note: A borrowed license can only be returned to the LicenseServer it was borrowed from! If the

LicenseServer was moved please ask your network administrator where to find the LicenseServer now.

3.2 File Formats

This chapter describes file formats used by the software.

3.2.1 sdc/sdw

The extracted data files (*.sdc, *.sdw) contain one record for every target detected in the sample data. The binary record data format is different for the different file types and is described below in more detail. Apart from additional valuable information on the detected targets such as amplitude of the reflected echo signal, temporal width of the echo signal, and target number within one laser measurement (first, second, .. last target), the records contain geometry data in different coordinate systems and different data formats.

SDC files additionally contain range and scan angle data for each target.

SDW files contain X, Y, Z values in a world coordinate system, i.e. WGS 84 (double precision values).

These files are binary coded. They consist of a header followed by the data records (= targets). The header must be decoded as follows:

<size_of_header(4)> <major(2)> <minor(2)> <header_information(size_of_header-8)>

Read the first 4 bytes indicating the size of header. Now read size_of_header - 4 bytes to extract the complete header. Major and minor (2 bytes or 1 word each) indicate the version number of the file format (see below). The rest of the header is ASCII text describing the measurement record structure (this is for your information only):

SDC record formats:

```
TRecord_5_4 = packed record   // SDC 5.4
  // 5.0:
  TIME        : DOUBLE;  // seconds of the day or of the week, depending on GPS string
  RANGE       : SINGLE;  // measured range value [m]
  THETA       : SINGLE;  // measured theta value [deg]
  X, Y, Z     : SINGLE;  // x, y, z value of point in scanner's own coordinate system [m]
  AMPLITUDE   : WORD;    // linearized amplitude
  WIDTH       : WORD;    // width of target return (full width half maximum) [0.1 ns] (*)
  TARGETTYPE  : BYTE;    // 0 .. COG, 1 .. PAR, 2 .. GPF, 3 to 5 .. GPE
  TARGET      : BYTE;    // index of target (1 .. NUMTARGET)
  NUMTARGET   : BYTE;    // total number of targets
  RGINDEX     : WORD;    // range gate index of measurement (DEPRECATED)
  CHANNELDESC : BYTE;    // channel descriptor:
                         // Bit7: 1=Waveform available
                         // Bit6: 1=high power channel
                         // Bit5: 1=New line started
                         // Bit4: 1=Facet number available
                         // Bit3: 1=Synthetic
                         // Bit0+1: Facet number [0..3]
  // 5.2:
  CLASSID     : BYTE;    // class identifier (ground, vegetation, ...)
  // 5.3:
  RHO         : SINGLE;  // tile mount angle [deg]
  // 5.4:
  REFLECTANCE : SMALLINT;// reflectance [0.01 dB]
end;
```

(*) Note: If the SDC file has been created from an .rxp (V-line) file, WIDTH contains the deviation of the target echo!
If created from an .sdf file (Q560, Q680 family), WIDTH contains the width of the target pulse as described above.

The deviation has no physical unit but describes the "quality" of the echo pulse. Usually this values is below 30.

SDW record formats:

```
TRecord_2_3 = packed record   // SDW 2.2
  // 2.0:
  TIME        : DOUBLE;  // seconds of the day or of the week, depending on GPS string
  X, Y, Z     : DOUBLE;  // x, y, z value of point in WGS 84 coordinate system [m]
  AMPLITUDE   : WORD;    // linearized amplitude
  WIDTH       : WORD;    // width of target return (full width half maximum) [0.1 ns] (*)
```

```
  TARGETTYPE  : BYTE;    // 0 .. COG, 1 .. PAR, 2 .. GPF, 3 to 5 .. GPE
  TARGET      : BYTE;    // index of target (1 .. NUMTARGET)
  NUMTARGET   : BYTE;    // total number of targets
  // 2.1:
  RGINDEX     : WORD;    // range gate index of measurement (DEPRECATED)
  CHANNELDESC : BYTE;    // channel descriptor:
                         // Bit7: 1=Waveform available
                         // Bit6: 1=high power channel
                         // Bit5: 1=New line started
                         // Bit4: 1=Facet number available
                         // Bit3: 1=Synthetic
                         // Bit0+1: Facet number [0..3]
  // 2.2:
  CLASSID     : BYTE;    // class identifier (ground, vegetation, ...)
  // 2.3: planned, not yet implemented:
  REFLECTANCE : SMALLINT;// reflectance [0.01 dB]
end;
```

(*) Note: If the SDC file has been created from an .rxp (V-line) file, WIDTH contains the deviation of the target echo!
If created from an .sdf file (Q560, Q680 family), WIDTH contains the width of the target pulse as described above.

The possible values of field CLASSID are specified below (based on LAS specification 1.1):

```
CLASS_CREATED           =   0;   // Created, never classified
CLASS_UNCLASSIFIED      =   1;   // Unclassified
CLASS_GROUND            =   2;   // Ground
CLASS_LOW_VEGETATION    =   3;   // Low vegetation
CLASS_MEDIUM_VEGETATION =   4;   // Medium vegetation
CLASS_HIGH_VEGETATION   =   5;   // High vegetation
CLASS_BUILDING          =   6;   // Building
CLASS_LOWPOINT_NOISE    =   7;   // Low point (noise)
CLASS_MODEL_KEYPOINT    =   8;   // Model key-point (mass point)
CLASS_WATER             =   9;   // Water
CLASS_OVERLAP           =  12;   // Overlap
```

These definitions are the same used in the LAS format for LIDAR data. If no classification is used, the CLASSID field is set to CLASS_CREATED (created, never classified).

3.2.2 rxp

RXP Files (*.rxp) contain raw measurement data provided by *RIEGL* V-Line scanners and form the basis for further processing. The description of the RXP format is not presented here but the files can be read in by using the software library RiVLIB. However, details on the RXP structure are not necessary in case the data is processed by RiANALYZE and/or SDCImport.

3.2.3 2dd/3dd/4dd

2dd, 3dd and 4dd files contain raw measurement data provided by *RIEGL* Z- and Q-Series scanners and form the basis for further processing. The description of the file format is not presented here but the files can be read in by using the software library RiSCANLIB. However, details on the file structure are not necessary in case the data is processed by RiANALYZE and/or SDCImport.

3.3 Revision History

3.3.1 SDCImport

SDCImport 2.3.18, 2019-04-30

- RiMTA5 version 5.6.4 (release) upgraded
  Time synchronization issues fixed
  Processing statistics calculation fixed
- Processing Information output from RiMTA5 disabled
- Fixed problem if MTA calculation is disabled

SDCImport 2.3.17, 2019-04-17

- RiMTA5 version 5.6.3 (release) upgraded
  Default values for notch filter updated

SDCImport 2.3.16, 2019-04-11

- RiMTA5 version 5.6.2 (release) upgraded
  Using internal timestamps for MTA resolution

SDCImport 2.3.15, 2019-04-03

- RiMTA5 version 5.6.1 (release) upgraded
  Window echo detection improved
  Time synchronization issues fixed

SDCImport 2.3.14, 2019-04-02

- RiMTA5 version 5.6.0 (release) upgraded

SDCImport 2.3.13, 2018-12-19

- Support for VQ-880-GII, VQ-480II and VQ-580II
- RiMTA5 5.5.12 upgraded:
  Echo interpolation fixed
  Time sync problem fixed

SDCImport 2.3.12, 2018-11-14

- RiMTA5 version 5.5.11 updated
  (norm sec output fixed)
- Week second offset calculation fixed
- RiMTA5 message output added

SDCImport 2.3.11, 2018-11-06

- FastExport database creation (RiMTA5 processing) fixed

SDCImport 2.3.10, 2018-10-18

- RiMTA5 version 5.5.10 (release) updated

SDCImport 2.3.9, 2018-10-15

- FastExport database with day/week seconds from .rxp input
- RiMTA5 version 5.5.9 (release) updated

SDCImport 2.3.8, 2018-09-21

- RiMTA5 version 5.5.7 updated

SDCImport 2.3.7, 2018-06-20

- RiVLib library dev2.5.6 updated
  VQ-780i Window Distortion fixed
- RiMTA5 version 5.5.3 updated
  Week seconds fixed
- FDX cleanup improved (if no waveforms contained in .rxp input)

SDCImport 2.3.6, 2018-06-12

- RiMTA5 version 5.5.1 updated

SDCImport 2.3.5, 2018-06-12

- RiVLib library dev2.5.6 updated
- RiMTA5 version 5.5.0 updated
- MTAReader/RX5Reader buffer initialization fixed
- Window distortion for VQ-780i fixed
- Week seconds time stamps for .sdc if defined by .rxp input

SDCImport 2.3.4, 2018-04-19

- RiMTA5 library updated
  Runtime dependencies (vc100 runtime) fixed

SDCImport 2.3.3, 2018-04-03

- Buffer management problem fixed
- Blocking Fast Data Export fixed
- Window distortion calibration now available for VQ-780i
- RiVLib dev2.5.4 updated (fix processing cut .rxp files)
- RiMTA5 waveform available flag fixed
- RiMTA5 window echo interpolation added
- Command line argument /Date:DataInput added
- Parameter extension files (.pxf) implemented

SDCImport 2.3.2, 2018-02-05

- Logging of RiMTA5 processing parameters added
- RiMTA5 5.3.10 added
- RiVLib binaries upgraded to dev2.5.4

SDCImport 2.3.1, 2018-01-26

- Configuration now stored in <user>/AppData/Riegl_LMS/SDCImport/SDCImport.ini

- Floating point conversion fixed
- Waveform extraction and FastExport for VQ-1560i available
- RiVLib binaries upgraded to 2.5.4
- FastExport database file (.fdx, version 1.1) creation fixed
- RiMTA5 5.3.8 added
- FastExport database file (.fdx) time sync with RiMTA5 fixed (#2970)
- Threading problem with HDF5 library fixed (#2984)
- Logging of RiMTA5 parameters read from configuration file (#2947)

SDCImport 2.3, 2017-12-20

- 64-bit compilation available!
- Note: Default installation directory is now
  C:\Program Files\Riegl_LMS\...
  Please check you RiSERVER configuration!
- RiMTA5 5.3.7 added
  VQ-1560i facet number fixed

SDCImport 2.2.6, 2017-12-15

- PPM swing parameter setting fixed
- RiMTA5 5.3.6 (release) added
- RiMTA5 FastParse support for VQ-580 implemented
- RiMTA5 processing statistics output implemented
- Time scaling problem (VZ-2000i) fixed
- Number of time jump warnings added to processing statistics

SDCImport 2.2.5, 2017-12-04

- RiVLib library dev2.5.3 updated (pulse_position_modulation_1)
- RiMTA5 5.3.4 updated (facet and segment numbers)
- Internal calibration/window distortion for RiMTA5 implemented
- Waveform extraction for VQ-780i enabled
- RiMTA LCP1 Line Classification Method fixed for VQ-250, VQ-450 and VUX
- RiMTA last target/notch echo detection improved
- Waveform extraction fixed (name and path for .wfm file)
- Disable Sync-to-PPS (and auto-detect) if <source>.rxp.log with corresponding entry found
- Re-enable Sync-to-PPS if scan is PPS synchronized (detected)
- User interface for internal calibration of VQ-780i

SDCImport 2.2.4, 2017-09-08

- RiMTA5 5.2.1 library updated
- RiMTA library updated
  (Time/range/angle/amplitude/reflectance gates fixed)

SDCImport 2.2.3, 2017-09-04

- RiMTA5 5.2 library added (RiMTA5 updated to version 5.2)
- Classify synthetic points only if classification is enabled (#2699)

SDCImport 2.2.2, 2017-08-31

- RiMTA5 5.1.2 library added (RiMTA5 updated)
- RiMTA5 detection fixed
- Disable waveform extraction (.wfm) if no waveforms available in source
- Problem overriding internal calibration for VQ-1560i fixed
- Laser beam divergence and Beam diameter at exit aperture now logged
- RiMTA license is checked before creating an .mta file
- Licensing issues handling fixed so RiPROCESS can reset tasks back to ASAP
- RiMTA5 configuration (modswing) invalid -> RiMTA5 (vicinity) disabled
- New point classes for bathymetric classification implemented (#2632)
- Calculate averaged amplitude and reflectance values for synthetic targets

SDCImport 2.2.1, 2017-07-14

- Set amplitude and reflectance to maximum values for synthetic targets

SDCImport 2.2, 2017-05-19

- VQ-1560i direct data processing implemented
- MTA processing speedup
- User interface settings for averaging/analyzing waveforms of BDF-1 implemented
- RiMTA5 5.0.8 updated
- Water column filter (dyntrig elevation) implemented
- Now range and angle gate directly handled by RiMTA
- Processing speedup for BDF-1 (disabling compression for intermediate steps)
- SDC version 1.0 8821 (x86_windows_vc120) added
- MTA version 5.0 8821 (x86_windows_vc120) added
- rxp2wfm.exe (x86_windows_vc110) 2.4.0.1481620731
- rxp2mta.exe (x86_windows_vc120) 3.1.1493115883 using mtaifc (x86_windows_vc120)
  API version: 5.1 build version: 1485530377 with 64-Bit IO
- rms2mta.exe (x86_windows_vc120) 2.1.1493115883 with mtaifc (x86_windows_vc120) 5.1.1485530377
- rxp2rx5-rxp2rx5-4.3.2 (x86_64-Windows-GNU-4.9.2)

- rivaverage-rivanalyze-1.0.0 (x86_64-Windows-GNU-4.9.2)
- rivanalyze-rivanalyze-1.0.0 (x86_64-Windows-GNU-4.9.2)
- rivbdf-rivanalyze-1.0.0 (x86_64-Windows-GNU-4.9.2)
- rivhydro-rivanalyze-1.0.0 (x86_64-Windows-GNU-4.9.2)
- mtaifc-mt-s.dll 5.1 (1485530377 x86_windows_vc120)
- scanifc-mt-s.dll 7.1.6 (1493114912 x86_windows_vc120)

SDCImport 2.1, 2017-02-16

- MTA calculation "RiMTA (vicinity)" introduced
- MTA calculation method "Auto" introduced:
  Selects appropriate MTA resolution method
  - RiMTA (vicinity) or
  - RiMTA (code correlation) or (if no MTA resolution required)
  - Disabled/Fixed in zone 1
- Delete existing .fdx file if FDX output is disabled
- Angle gate filtering fixed for miniVUX
- Reading .rxp file properties fixed/improved
- Waveform extraction for VQ-1560i disabled
- Notchfilter for VQ-1560i (dev notch) improved
- Processing tool chain for BDF-1 implemented
- SDC version 1.0 8458 (x86_windows_vc120)
- MTA version 5.0 8458 (x86_windows_vc120)
- rxp2wfm.exe (x86_windows_vc120) 2.4.0.1481620687
- rxp2mta.exe (x86_windows_vc120) 3.1.1485530377 using mtaifc (x86_windows_vc120) API version: 5.1 build version: 1485530377 with 64-Bit IO
- rms2mta.exe (x86_windows_vc120) 2.1.1485530377 with mtaifc (x86_windows_vc120) 5.1.1485530377
- rxp2rx5.exe-rxp2rx5-4.1.1 (x86_64-Windows-GNU-4.9.2)
- rivaverage.exe-rivanalyze-0.3.1 (x86_64-Windows-GNU-4.9.2)
- rivanalyze.exe-rivanalyze-0.3.1 (x86_64-Windows-GNU-4.9.2)
- rivbdf.exe-rivanalyze-0.3.1 (x86_64-Windows-GNU-4.9.2)
- mtaifc-mt-s.dll 5.1 (1485530377 x86_windows_vc120)
- scanifc-mt-s.dll 7.1.0 (1485529303 x86_windows_vc120)
- RiMTA5 version 5.0.2

SDCImport 2.0.2, 2016-11-21

- Angle gate filtering fixed (#2407)
- Shot/line divider fixed (#2401)
- SDC version 1.0 8209 (x86_windows_vc100)
- MTA version 5.0 8209 (x86_windows_vc100)

- SDCRun 1.0 8209 added
- rxp2wfm.exe (x86_windows_vc110) rc2.4.0.1472468961
- rxp2mta.exe (x86_windows_vc110) 3.1.1478870185 using mtaifc (x86_windows_vc110) API version: 5.1 build version: 1472801110 with 64-Bit IO
- rms2mta.exe (x86_windows_vc110) 2.1.1478870185 with mtaifc (x86_windows_vc110) 5.1.1472801110
- RiMTA dev2.10 (RiVLib rc2.4.0)
- scanifc-mt-s.dll 7.1.0 (1478869391 x86_windows_vc110)
- mtaifc-mt-s.dll 5.1 (1472801110 x86_windows_vc110)

SDCImport 2.0.1, 2016-07-29

- Shot buffer for MTA resolution enlarged for VUX (set to 100000)
- DLL library loading sequence fixed
- Handling of .rms files fixed (rms2mta command line)
- SDC version 1.0 7817 (x86_windows_vc100)
- MTA version 4.1 7817 (x86_windows_vc100)
- rxp2wfm.exe (x86_windows_vc110) 2.2.1.1272
- rxp2mta.exe (x86_windows_vc110) 3.1.1466603313 using mtaifc (x86_windows_vc110) API version: 5.0 build version: 1462271095 with 64-Bit IO
- rms2mta.exe (x86_windows_vc110) 2.1.1466603313 with mtaifc (x86_windows_vc110) 5.0.1462271095
- RiMTA 2.9 (RiVLib rc2.4.0)
- scanifc-mt-s.dll 7.1.0 (1466602656 x86_windows_vc110)
- mtaifc-mt-s.dll 5.0 (1462271095 x86_windows_vc110)

SDCImport 2.0, 2016-07-06

- Completely redesigned
- MTA calculation method "Auto" is now "RiMTA - code-correlation"
- New MTA resolution mode "RiMTA - histogram-correlation" added
- Database creation (.fdx file) for Fast Export implemented
- rxp2wfm.exe (x86_windows_vc110) 2.2.1.1272
- rxp2mta.exe (x86_windows_vc110) 3.1.1466603313 using mtaifc (x86_windows_vc110) API version: 5.0 build version: 1462271095 with 64-Bit IO
- rms2mta.exe (x86_windows_vc110) 2.1.1466603313 with mtaifc (x86_windows_vc110) 5.0.1462271095
- SDCRun 1.0 7708 added
- SDC version 1.0 7708 (x86_windows_vc100)
- MTA version 4.1 7708 (x86_windows_vc100)
  - Barker 7 calculation speedup
  - Blind range interpolation for MTA zone transition added
- RiMTA 2.9 (RiVLib rc2.4.0)
- scanifc-mt-s.dll 7.0.0 (1458742215 x86_windows_vc110)

- mtaifc-mt-s.dll 5.0 (1462271095 x86_windows_vc110)

SDCImport 1.6.4, 2015-12-22

- Reflectance gate filter unit conversion fixed
- Extract header information from .rms files

SDCImport 1.6.3, 2015-10-27

- Near range target calculation fixed (RiMTA)
- Amplitude gate filter unit conversion fixed
- Angle gate filter fixed for VQ-380
- mta.dll 2.3 6564 x86_windows_dcc32-15.0 added

SDCImport 1.6.2, 2015-09-24

- mta.dll 2.3 4602 x86_windows_dcc32-15.0 added
- rxp2mta.exe (x86_windows_vc110) 3.1.1222 using mtaifc (x86_windows_vc110) API version: 5.0 build version: 1222 with 64-Bit IO added
- rms2mta.exe (x86_windows_vc110) 2.1.1222 with mtaifc (x86_windows_vc110) 5.0.1222 added
- rxp2wfm.exe (x86_windows_vc110) 2.1.0.1222 added
- mtaifc-mt.dll 5.0 (1222 x86_windows_vc110) added
- scanifc-mt.dll 7.0.1066 (1222 x86_windows_vc110) added

SDCImport 1.6.1, 2015-03-20

- mta.dll 2.3 (4602 x86_windows_dcc32-15.0) added

SDCImport 1.6, 2015-03-13

- mta.dll 3.2 (5715 x86_windows_vc100) added
- Setup overwrites existing files now
- rxp2mta.exe (x86_windows_vc110) 3.1.1177 using mtaifc (x86_windows_vc110) API version: 5.0 build version: 1181 with 64-Bit IO
- rms2mta.exe (x86_windows_vc110) 2.1.1177 with mtaifc (x86_windows_vc110) 5.0.1181
- rxp2wfm.exe (x86_windows_vc110) 2.0.1.1181 added
- mtaifc-mt.dll 5.0 (1181 x86_windows_vc110) added
- scanifc-mt.dll 7.0.1066 (1177 x86_windows_vc110) added

SDCImport 1.5.14, 2015-01-22

- rxp2mta.exe (x86_windows_vc110) 3.1.1164 using mtaifc (x86_windows_vc110) API version: 5.0 build version: 1164 with 64-Bit IO
- rms2mta.exe (x86_windows_vc110) 2.1.1164 with mtaifc (x86_windows_vc110) 5.0.1164

- rxp2wfm.exe (x86_windows_vc110) 1.44.1164 added
- mtaifc-mt.dll 5.0 (1164 x86_windows_vc110) added
- scanifc-mt.dll 7.0.1066 (1164 x86_windows_vc110) added

SDCImport 1.5.13, 2015-01-16

- rxp2mta.exe (x86_windows_vc110) 3.1.1145 using mtaifc (x86_windows_vc110) API version: 5.0 build version: 1145 with 64-Bit IO added
- rms2mta.exe (x86_windows_vc110) 2.1.1145 with mtaifc (x86_windows_vc110) 5.0.1145 added
- rxp2wfm.exe (x86_windows_vc110) 1.43.1145 added
- mta.dll 2.3 (4602 x86_windows_dcc32-15.0) added (attenuation correction implemented)
- mtaifc-mt.dll 5.0 (1145 x86_windows_vc110) added
- scanifc-mt.dll 7.0.1066 (1145 x86_windows_vc110) added

SDCImport 1.5.12, 2014-12-19

- rxp2mta.exe (x86_windows_vc110) 3.1.1141 using mtaifc (x86_windows_vc110) API version: 5.0 build version: 1141 with 64-Bit IO
- rms2mta.exe (x86_windows_vc110) 2.1.1141 with mtaifc (x86_windows_vc110) 5.0.1141
- rxp2wfm.exe (x86_windows_vc110) 1.42.1141 added
- mtaifc-mt.dll 5.0 (1141 x86_windows_vc110) added
- scanifc-mt.dll 7.0.1066 (1141 x86_windows_vc110) added

SDCImport 1.5.11, 2014-12-17

- rxp2mta.exe (x86_windows_vc110) 3.1.1136:1137 using mtaifc (x86_windows_vc110) API version: 5.0 build version: 1136:1137 with 64-Bit IO added
- rms2mta.exe (x86_windows_vc110) 2.1.1136:1137 with mtaifc (x86_windows_vc110) 5.0.1136:1137 added
- rxp2wfm.exe (x86_windows_vc110) 1.41.1136:1137 added
- mtaifc-mt.dll 5.0 (1136:1137 x86_windows_vc110) added
- scanifc-mt.dll 7.0.1066 (1136:1137 x86_windows_vc110) added

SDCImport 1.5.10, 2014-12-03

- Support for segment extraction (VQ-880-G) implemented
- Failure on waveform extraction produces a warning but processing continues
- mta.dll 2.3 (4602 x86_windows_dcc32-15.0) added
- rxp2wfm.exe (x86_windows_vc110) 1.40.1134 added
- rxp2mta.exe (x86_windows_vc110) 3.1.1134 using mtaifc (x86_windows_vc110) API version: 5.0 build version: 1134 with 64-Bit IO added
- rms2mta.exe (x86_windows_vc110) 2.1.1134 with mtaifc (x86_windows_vc110) 5.0.1134 added
- mtafic-mt(-s).dll 5.0 1134 x86_windows_vc110 added

- scanifc-mt(-s).dll 7.0.1066 (1134 x86_windows_vc110) added

SDCImport 1.5.9, 2014-05-20

- mta.dll 2.3 (4530 x86_windows_dcc32-15.0) added (support for mtaifc version 3.0)
- rxp2wfm.exe (x86_windows_vc100) 1.35.1011 added
- rxp2mta.exe (x86_windows_vc100) 2.0.1011 using mtaifc (x86_windows_vc100) API version: 3.0 build version: 1011 with 64-Bit IO
- rms2mta.exe (x86_windows_vc100) 2.0.1011 with mtaifc (x86_windows_vc100) 3.0.1011 added
- mtaifc-mt.dll 3.0 (1011 x86_windows_vc110) added
- scanifc-mt.dll 6.4.875 (1011 x86_windows_vc100) added

SDCImport 1.5.8, 2014-03-25

- scanifc-mt.dll 6.4.875 (986 x86_windows_vc100) added
- rxp2mta.exe (x86_windows_vc100) 1.3.986 using mtaifc (x86_windows_vc100) API version: 2.0 build version: 969 with 64-Bit IO
- rms2mta.exe (x86_windows_vc100) 1.3.986 with mtaifc (x86_windows_vc100) 2.0.969 added fixing jumps in time stamps

SDCImport 1.5.7, 2014-02-19

- MTA processing: .mta files from .rms stream created in wrong directory [fixed]

SDCImport 1.5.6, 2014-02-14

- Maintainance release for RiPROCESS 1.6
- scanifc-mt.dll 6.4.875 (969 x86_windows_vc100) added
- mtaifc-mt.dll 2.0 (969 x86_windows_vc100) added
- rxp2mta.exe (x86_windows_vc100) 1.3.969 using mtaifc (x86_windows_vc100) API version: 2.0 build version: 969 with 64-Bit IO
- rms2mta.exe (x86_windows_vc100) 1.3.969 with mtaifc (x86_windows_vc100) 2.0.969 added
- rxp2wfm.exe (x86_windows_vc100) 1.32.969 added
- mta.dll 2.2 (4080 x86_windows_dcc32-15.0) added

SDCImport 1.5.5, 2013-11-12

- rxp2wfm.exe (x86_windows_vc100) rc1.31.935 added

SDCImport 1.5.4, 2013-11-08

- scanifc-mt.dll 6.4.875 (933 x86_windows_vc100) added
- mtaifc-mt.dll 2.0 926:927 x86_windows_vc100
- rxp2mta.exe (x86_windows_vc100) 1.3.934 using mtaifc (x86_windows_vc100) API version: 2.0 build version: 926:927 with 64-Bit IO added

- rms2mta.exe (x86_windows_vc100) 1.3.934 with mtaifc (x86_windows_vc100) 2.0.926:927 added

SDCImport 1.5.3, 2013-10-31

- scanifc-mt.dll 6.4.875 (926 x86_windows_vc100) added
- mtaifc-mt.dll 2.0 (926 x86_windows_vc100) added
- rxp2wfm.exe (x86_windows_vc100) rc1.31.926:927 added
- rxp2mta.exe (x86_windows_vc100) 1.3.926 using mtaifc (x86_windows_vc100) API version: 2.0 build version: 926 with 64-Bit IO added
- rms2mta.exe (x86_windows_vc100) 1.3.926 with mtaifc (x86_windows_vc100) 2.0.926 added

SDCImport 1.5.2, 2013-10-04

- Reflectance gate implemented
- mta.dll 2.2 (3363 x86_windows_dcc32-15.0) added

SDCImport 1.5.1, 2013-04-12

- Automatic creation of index files .mta.idx for .mta files implemented
- mta.dll 2.1 (3034 x86_windows_dcc32-15.0) added
- Automatic detection of line angle for AngleGate fixed (VQ-820)
- Calculation of theta angle now from SOCS coordinates (VQ-820)
- Required by RiPROCESS 1.5.8
- scanifc-mt.dll 6.4.752 (754 x86_windows_vc100) added
- mtaifc-mt.dll 2.0 754 x86_windows_vc100 added
- rxp2mta (x86_windows_vc100) 1.3.754 with mtaifc (x86_windows_vc100) 2.0.754 added
- rms2mta (x86_windows_vc100) 1.3.754 with mtaifc (x86_windows_vc100) 2.0.754 added

SDCImport 1.5, 2013-03-14

- rxp2mta (x86_windows_vc100) 1.3.749 with mtaifc (x86_windows_vc100) 2.0.749
- rms2mta (x86_windows_vc100) 1.3.749 with mtaifc (x86_windows_vc100) 2.0.749
- mtaifc-mt.dll 2.0 (749 x86_windows_vc100) added
- scanifc-mt.dll 6.4.740 (749 x86_windows_vc100) added
- mta.dll 2.1 (2978 x86_windows_dcc32-15.0) added
- Automatic detection of line angle for AngleGate implemented (VQ-250, VQ-450)
- rms extraction (reverse job order) handling fixed

SDCImport 1.4.4, 2012-09-12

- mta.dll 3.0 2588 x86_windows_dcc32-15.0 added

SDCImport 1.4.3, 2012-07-13

- MTA calculation of single data stream from RMS files implemented (rms2mta)
- mta.dll 2.0 2421 x86_windows_dcc32-15.0
- rxp2mta (x86_windows_vc100) 1.2.689 with mtaifc (x86_windows_vc100) 1.0.689
- rms2mta (x86_windows_vc100) 1.0.689 with mtaifc (x86_windows_vc100) 1.0.689
- mtaifc-mt.dll 1.0 689 (x86_windows_vc100) added
- scanifc-mt.dll 6.4.667 (689 x86_windows_vc100) added

SDCImport 1.4.2, 2012-05-02

- TimeGate with multiple time slices implemented
- Classification results from RiMTA now written to output
- mta.dll 2.0 2274 x86_windows_dcc32-15.0 added
- scanifc-mt.dll 6.4.621 (664 x86_windows_vc100) added
- mtaifc-mt.dll 1.0 663 x86_windows_vc100 added
- rxp2mta (x86_windows_vc100) 1.2.664 with mtaifc (x86_windows_vc100) 1.0.663

SDCImport 1.4.1, 2012-02-22

- RangeGate/TimeGate/AngleGate inversion issue fixed

SDCImport 1.4, 2012-01-31

- Multiple Time Around (MTA) implemented
- scanifc-mt.dll 631:632 x86_windows_vc100 added
- mta.dll 2.0 2141 x86_windows_dcc32-15.0
- mtaifc-mt.dll 1.0 631:632 x86_windows_vc100 added
- rxp2mta.exe (2012-01-31) added
  (reflectance calculation and near range correction)
- Time gate (filter) implemented
- Angle gate (filter) implemented

SDCImport 1.3.11, 2011-08-17

- Facet number import fixed
- Erroneous rxp file (with invalid gps sync) may cause out of memory error [fixed]

SDCImport 1.3.10, 2011-05-02

- scanifc-mt.dll 6.3.563 (569 x86_windows_vc100) added
- Expert mode implemented

[****]

SDCImport 1.3.9, 2011-03-18

- Line and shot dividers (MSM) implemented

SDCImport 1.3.8, 2011-03-03

- scanifc-mt.dll 6.2.547 (555 x86_windows_vc100) added

SDCImport 1.3.7, 2011-03-03

- Facet number (from .rxp and .rms) now written into .sdc record
- scanifc-mt.dll 6.2.547 (553 x86_windows_vc100) added (decompression of truncated data streams fixed)
- Processing finished successfully on data stream problems (measurements processed so far can still be used)
- now create wave form file (.wfm) from .rxp containing wave forms

SDCImport 1.3.6, 2010-06-25

- Pause/resume implemented
- scanifc-mt.dll 6.0.435 (444) added

SDCImport 1.3.5, 2010-04-29

- scanifc-mt.dll 5.1.397 (420) added (supports .rms files, compression)
- Support for .rms files (multiplexed streams) implemented
- Time jump warnings (forward/backward jumps) implemented

SDCImport 1.3.3, 2010-02-03

- scanifc-mt.dll 5.1.394 (394) added
- Range and theta calculation fixed (rxp file import)

SDCImport 1.3.2, 2010-01-25

- SDCImport documentation updated
- scanifc-mt.dll 5.0.337 (380) added
- Handling of error message from scanifc-mt.dll fixed

SDCImport 1.3.1, 2009-12-16

- Problems with file handling when used by RiPROCESS [fixed]

SDCImport 1.3, 2009-12-07

- Range Gate Filter implemented
- scanifc-mt.dll 5.0.294 added

SDCImport 1.2, 2009-10-05

- Invalid floating point error if no reflectance information found in input file [fixed]

SDCImport 1.1.12, 2009-09-21

- SDC 5.4 output format implemented (reflectance added)

SDCImport 1.1.11, 2009-09-14

- scanifc-mt.dll 4.0.226: Excised files not overrolled properly [fixed]

SDCImport 1.1.10, 2009-07-06

- scanifc-mt.dll 4.0.226: Detection of corrupt files (max. buffer 64KB)
- scanifc-mt.dll 4.0.226: SysTime overroll [fixed]

SDCImport 1.1.9, 2009-06-17

- Output truncated on first empty frame [fixed]

SDCImport 1.1.8, 2009-06-12

- Invalid floating point error was produced when input file was not found [fixed]

SDCImport 1.1.7, 2009-05-29

- Multiple file selection in "Add file" dialog enabled
- Now first and last timestamp of input file logged
- scanifc-mt.dll 3.1.201: Midnight time rollover fixed

SDCImport 1.1.4, 2009-02-09

- scanifc-mt.dll: Not GPS-synchronized data was omitted [fixed]
- scanifc-mt.dll: Time stamp of first data record was invalid [fixed]

SDCImport 1.1.3, 2009-02-05

- Version 1.1.2 created empty output files from .rxp input files [fixed]

SDCImport 1.1.2, 2009-02-04

- scanifc library version now logged
- Auto-Expanding message list
- Message entry now display correct percentage

SDCImport 1.1.1, 2009-01-26

- scanifc-mt.dll now used

3.4 Copyright remarks

This software makes use of data or software from third parties in binary form which is covered by licenses listed in this chapter.

3.4.1 Armadillo C++ Linear Algebra Library

```
Licensed under the Apache License 2.0.
Copyright 2008-2016 Conrad Sanderson (http://conradsanderson.id.au)
Copyright 2008-2016 National ICT Australia (NICTA)

This product includes software developed by Conrad Sanderson
(http://conradsanderson.id.au)
This product includes software developed at National ICT Australia (NICTA)
```

3.4.2 HDF5 Software Library and Utilities

```
Copyright (c) 2006, The HDF Group.

NCSA HDF5 (Hierarchical Data Format 5) Software Library and Utilities
Copyright (c) 1998-2006, The Board of Trustees of the University of Illinois.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted for any purpose  (including commercial purposes) provided that the
following conditions are met:

1. Redistributions  of source  code must retain the above copyright notice, this
   list of conditions, and the following disclaimer.

2. Redistributions in  binary form  must  reproduce  the above copyright notice,
   this  list of  conditions, and the  following disclaimer in the documentation
   and/or materials provided with the distribution.

3. In addition,  redistributions of modified  forms of the source or binary code
   must  carry prominent  notices stating that the original code was changed and
   the date of the change.

4. All  publications or advertising materials mentioning features or use of this
   software are asked, but not required, to acknowledge that it was developed by
   The  HDF Group and by the National  Center for Supercomputing Applications at
   the University of Illinois at Urbana-Champaign and credit the contributors.

5. Neither the  name of The HDF  Group, the name of the University, nor the name
   of any  Contributor  may be used to  endorse or promote products derived from
   this software  without  specific prior  written permission from The HDF Group,
   the University, or the Contributor, respectively.

DISCLAIMER:
THIS  SOFTWARE IS PROVIDED BY THE HDF GROUP AND THE CONTRIBUTORS "AS IS" WITH NO
WARRANTY  OF ANY KIND,  EITHER  EXPRESSED OR IMPLIED.  In no event shall The HDF
Group or the Contributors  be liable  for any  damages  suffered  by  the  users
arising  out of the use of  this software, even if advised of the possibility of
such damage.
```

3.4.3 MINPACK

```
MINPACK, developed by
the University of Chicago, as Operator of Argonne National Laboratory.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

1. Redistributions of  source code must retain  the above copyright notice, this
   list of conditions and the following disclaimer.

2. Redistributions in  binary form  must  reproduce the  above copyright  notice,
   this list  of conditions  and the  following  disclaimer in  the documentation
   and/or other materials provided with the distribution.

3. The end-user documentation  included with  the  redistribution,  if any,  must
   include the following acknowledgment:
```

3.5 Download

This program can be downloaded from:

http://www.riegl.com/members-area/software-downloads/

If you have any further questions please don't hesitate to contact us via email: support@riegl.com

3.6 Contact

RIEGL Laser Measurement Systems GmbH

3580 Horn, Riedenburgstrasse 48, AUSTRIA

Tel.:	+43-2982-4211
Fax.:	+43-2982-4210
email:	office@riegl.com
support:	support@riegl.com
website:	www.riegl.com

Index