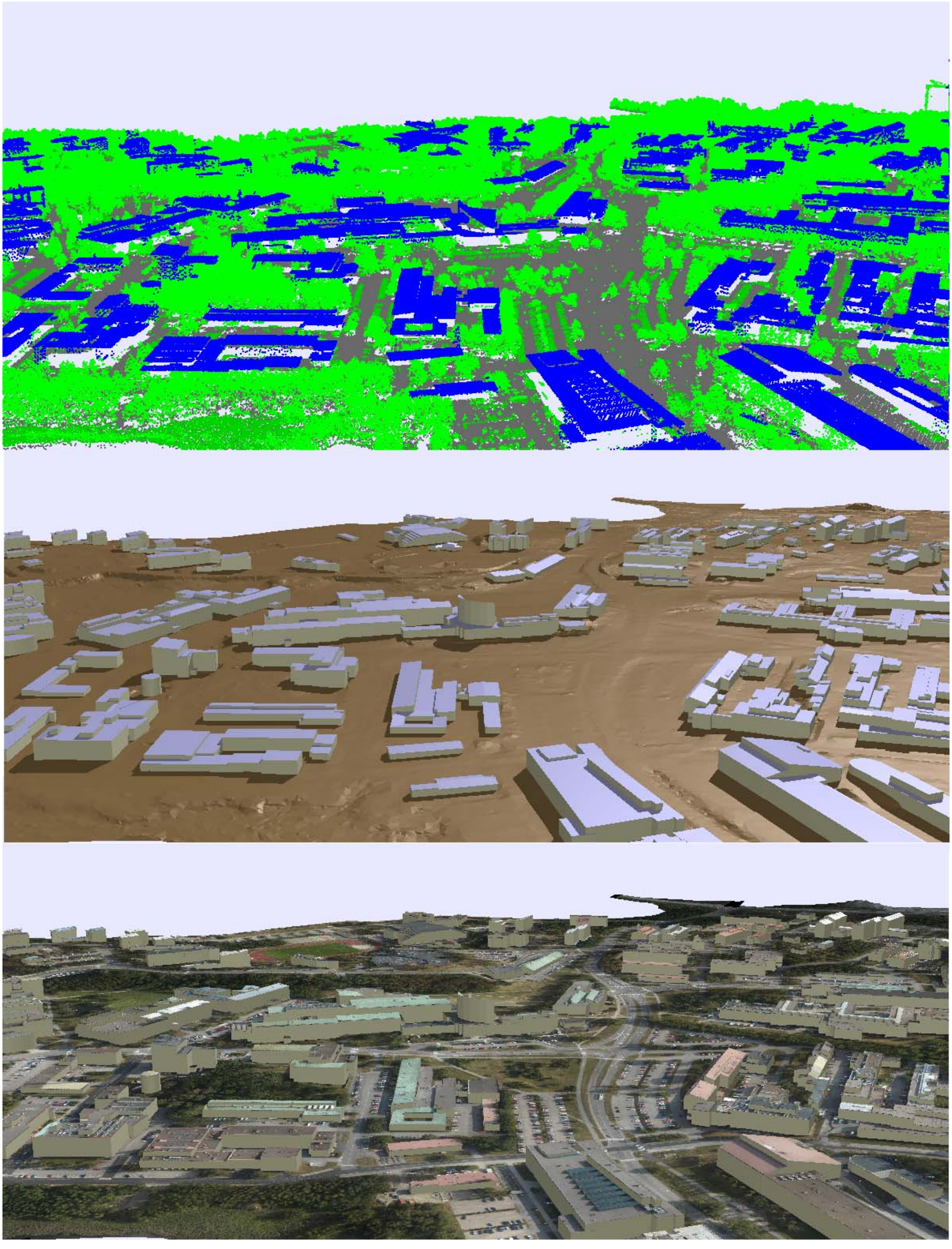
TerraScan User's Guide

--- uncompleted updated version ---



11.09.2014

Trademarks

MicroStation®, MDL® and MicroStation stylized "M" are registered trademarks of Bentley Systems, Incorporated. Bentley Map PowerView, MicroStation PowerDraft, and MicroStation GeoOutlook are trademarks of Bentley Systems, Incorporated.

TerraBore, TerraGas, TerraLink, TerraMatch, TerraModeler, TerraPark, TerraPhoto, TerraPipe, TerraScan, TerraSlave, TerraStereo, TerraStreet, and TerraSurvey are trademarks of Terrasolid Limited.

Windows is a trademark of Microsoft Corporation.

Acrobat Reader is a trademark of Adobe Systems Incorporated.

OSTN02 is a trademark of Ordnance Survey, the national mapping agency of Great Britain.

Intergraph Raster File Formats - Copyright - 1994 Intergraph Corporation. Used with permission.

Copyright

Ordnance Survey®

OSTN02™ enabled

Table of Contents

Getting Started

Tool Reference

Batch Processing Reference

Programming Interface

Additional Information

Getting Started

1 Documentation

About the documentation

This document serves as a user's guide for two versions of TerraScan. The entry-level version, TerraScan Lite, is functionally a subset of the full version, TerraScan. Tools available in TerraScan and TerraScan Lite work identically in the two versions. Tools that are not available in TerraScan Lite are marked as "*Not Lite*" in the documentation.

This User's Guide is divided into several parts:

- **Getting Started** - general information about TerraScan and instructions on how to install and run the application.
- **Tool Reference** - detailed descriptions of the tools and menu commands in TerraScan main toolbox and main window.
- **Batch Processing Reference** - detailed descriptions of classification routines and macro processing.
- **Programming Interface** - information about the addition of user-defined tools, a list of public functions of TerraScan as well as a description of file formats.
- **Additional Information** - information about the installation configuration.

Accessing the documentation

The documentation is accessible as an Acrobat Reader PDF document which serves the role of online help. Accessing the electronic format of the documentation has the following advantages:

- You can conduct automated searches for keywords in topic names or body text.
- You can click hypertext to "jump" to related topics.

Document conventions

The following conventions and symbols appear in this guide:

- **Data click** - click on the data mouse button, usually the left button on a right-hand mouse.
- **Reset click** - click on the reset mouse button, usually the right button on a right-hand mouse.
- **< >** - angle brackets used for keybord keys, for example, <Return>.
- **Key in** - type a command in the key-in line of MicroStation and then press <Return>.
- **OR** - alternate procedures or steps in a procedure.
- **C:/TERRA** - paths to directories of files on a hard disk are written with capital letters.
- Icons used to introduce special information:

Icon:	Appears next to:
✍	Notes and Hints
➢	Procedures

- When no distinction between MicroStation versions is necessary, this document refers to the CAD environment simply as "MicroStation".

MicroStation documentation

This document is written under the assumption that the reader knows how to use basic MicroStation features. You should refer to the printed documentation or online help of MicroStation whenever you need information about using the CAD environment.

2 Introduction to TerraScan

Introduction

TerraScan is a dedicated software solution for processing laser scanning point clouds. It can easily handle millions of points as all routines are tweaked for optimum performance.

Its versatile tools prove useful for a number of application fields, such as transmission lines, flood plains, proposed highways, stock piles, forest areas, city models, road and railroad surveying, and much more.

The application reads points from binary files or text files. It provides tools to:

- view the points three-dimensionally
- define your own point classes such as ground, vegetation, buildings or wires
- organize huge point clouds in projects
- manage trajectory information
- classify points using automatic filter routines
- classify points interactively
- digitize features by snapping to laser points
- detect and vectorize object features, such as buildings, powerline wires and towers, overhead wires, road breaklines, rails
- analyze object conditions, such as road surfaces, lines-of-sight, clearance areas, danger objects for roads, rails and wires, change detection
- create colored point clouds
- export colored raster images
- output classified points into text or binary files
- and much more

TerraScan is fully integrated with MicroStation. This CAD environment provides a huge number of useful tools and capabilities in the areas of view manipulation, visualization, vector placement, labeling and plotting. A basic understanding of MicroStation usage is required in order to be productive with TerraScan. The more familiar you are with MicroStation, the more benefit you can get from its huge feature set.

TerraScan Lite

TerraScan Lite is a light version of TerraScan and provides a subset of the functionality of the full version. It can be used to view point clouds, setup projects, work on loaded points and classify points manually. It provides all the tools for manual 3D building model editing.

TerraScan Lite does not include automatic classification routines, building vectorization and road analysis tools.

Terra application family

Terrasolid developes a full family of civil engineering applications. Almost all Terra applications are tightly integrated with MicroStation presenting an easy-to-use graphical interface to the user.

TerraBore is a solution for reading in, editing, storing and displaying bore hole data. You can triangulate soil layers with the help of TerraModeler.

TerraMatch fixes mismatches between laser points from different data strips automatically. It can be used for the calibration of a laser scanner system and for fixing project data.

TerraModeler creates terrain surface models by triangulation. You can create models of ground, soil layers, or design surfaces. Models can be created based on survey data, graphical elements, laser data, or XYZ text files.

TerraPhoto rectifies digital photographs taken during laser scanning survey flights and produces rectified ortho images.

TerraPipe is used for designing underground pipes. It gives you powerful tools for designing networks of drainage, sewer, potable water, or irrigation pipes.

TerraScan processes laser scanning data. It reads in laser points from text or binary files and lets you view the point cloud three dimensionally, classify the data, and create vector data based on the points.

TerraSlave is a stand-alone application that processes TerraScan macros. It enables distributed processing and scheduling tasks to gain optimal time and working performance.

TerraStereo is a stand-alone application for viewing very large point clouds in mono and stereo mode. It utilizes advanced point rendering techniques and the graphics card memory in order to display huge amounts of points.

TerraStreet is an application for street design. It includes all the terrain modeling capabilities of TerraModeler. The street design process starts with the creation of horizontal and vertical geometries for street alignments.

TerraSurvey reads in traditional survey data and creates a three dimensional survey drawing. The application recognizes a number of survey data formats automatically.

3 Installation

Hardware and software requirements

TerraScan is built on top of MicroStation. You must have a computer system capable of running this CAD software.

To run TerraScan, you must have the following:

- Pentium or higher processor
- Windows 8, 7, Vista, XP, or 2000 (64-bit version recommended)
- mouse
- 1024*768 resolution display or better
- 512 MB RAM or better (at least 2048 MB recommended for production work)
- MicroStation V8, MicroStation V8i (Select Series 2 or higher) or Map PowerView. Check Terrasolid's web pages for a more detailed overview of compatible MicroStation versions.

Installation of TerraScan requires about 2 MB of free hard disk space.

Installation media

TerraScan may be delivered on a CD/USB-Stick or as a zip file.

A **zip package** only contains the actual software - it does not include the PDF User's Guide.

A **Terra Installation CD/USB-Stick** includes the software and the online documentation. When you install from the CD/USB-Stick, the software and the documentation are copied to your hard disk. The CD/USB-Stick may include versions for multiple environments. You should locate the directory which corresponds to your operating system and MicroStation version.

Directory on CD	For operating system	For MicroStation
\setup\eng	Windows	V8 or V8i

Installation from zip file

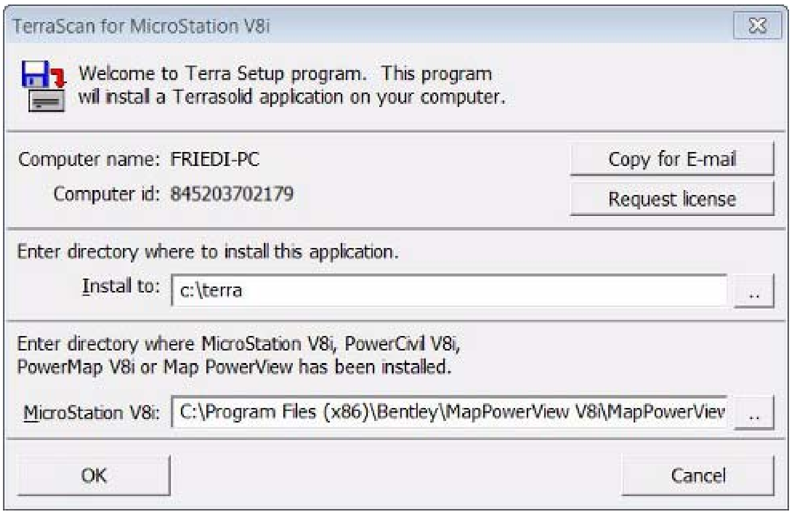
➢ **To install TerraScan from a zip file:**

1. Unpack the zip archive with any zip file manager.
2. Start **SETUP.EXE** which is part of the zip archive.

   This may open a dialog confirming the execution of SETUP.EXE and/or prompting for the administrator password.

   The installation program needs to know where MicroStation has been installed. It automatically searches all local hard disks to find the MicroStation directory.

   The installation dialog opens:



3. Enter the directory where to install TerraScan.

   The default path is C:\TERRA. You may change this to another location. The specified directory is created automatically if it does not exist.

4. Check the MicroStation directory. Replace the path if the correct location was not found automatically.
5. Click OK to start the installation.

   When the installation is finished, a message is displayed.

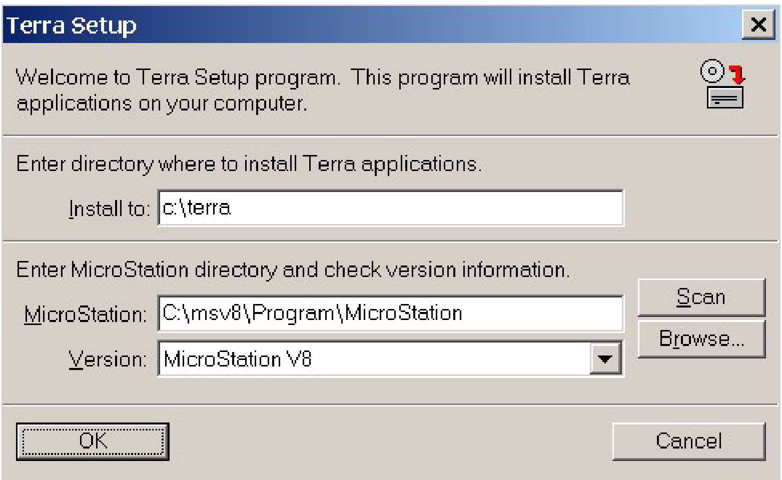
✍ See chapters **Installation Directories** on page 483 and **Configuration Variables** on page 484 for more information.

Installation from CD/USB-Stick

➢ **To install TerraScan from CD:**

1. Insert the **Terra Installation CD/USB-Stick**.
2. Locate the correct directory which corresponds to your computer configuration.
3. Start **SETUP.EXE** from that directory.

   The installation program tries to determine where MicroStation has been installed and opens the **Terra Setup** dialog:



4. Enter the directory where to install the application(s).

   The default path is C:\TERRA. You can change this to another location. The specified directory is created automatically, if it does not exist.

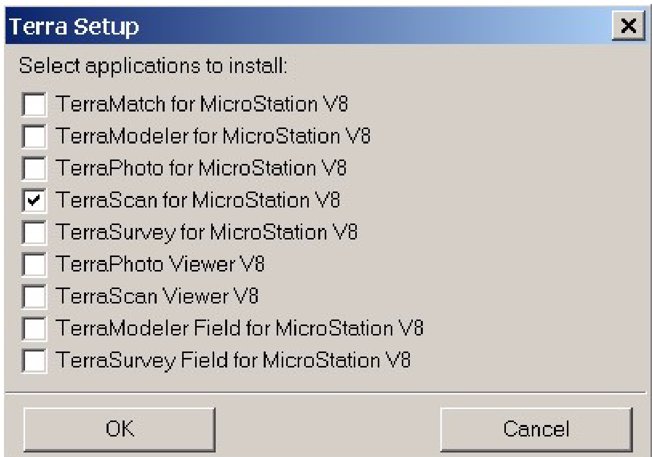
5. Check the **MicroStation** directory. Replace the path if the correct location was not found automatically.

   Alternatively, you can use the **Scan** button to automatically search the hard disk for the MicroStation installation or you can use the **Browse** button to locate the MicroStation executable yourself.

6. Check the MicroStation version information in the **Version** field. Select the correct version if it was not detected automatically.

7. Click OK to continue.

   This opens another **Terra Setup** dialog:



8. Select the **TerraScan for MicroStation** item in the dialog.

   You may select other applications as well for which you have installation files.

9. Click OK to start the installation.

   A message is displayed when the installation is finished.

✍ See chapters **Installation Directories** on page 483 and **Configuration Variables** on page 484 for more information.

4 Starting TerraScan

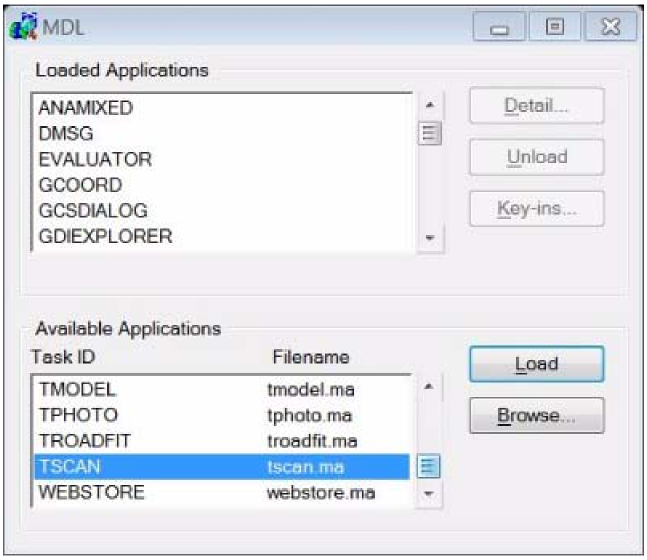
Starting TerraScan

TerraScan is an MDL application that runs within MicroStation.

➢ **To start TerraScan:**

1. Select **MDL Applications** command from the **Utilities** menu in MicroStation.

   The **MDL** dialog opens:

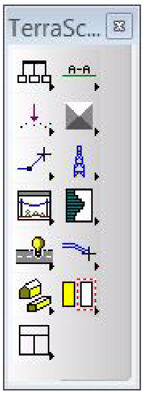


2. In the **Available Applications** list, select TSCAN.
3. Click the **Load** button.

   OR

1. Key in MDL LOAD TSCAN.

When the application is loaded, it adds an **Applications** menu to the MicroStation menu bar and opens the **TerraScan Main window** and **Main tool box**:



✍ The **Available Applications** list shows all MDL applications that MicroStation is able to locate. MicroStation searches for MDL applications in the directories listed in the MS_MDLAPPS configuration variable. If MicroStation can not find TSCAN.MA, you should check the value assigned to this configuration variable. Make sure the directory path of the TSCAN.MA file is included in the variable. To view configuration variables, select **Configuration** command from the **Workspace** pulldown menu in MicroStation. See also Sections **Installation Directories** on page 483 and **Configuration Variables** on page 484 for more information.

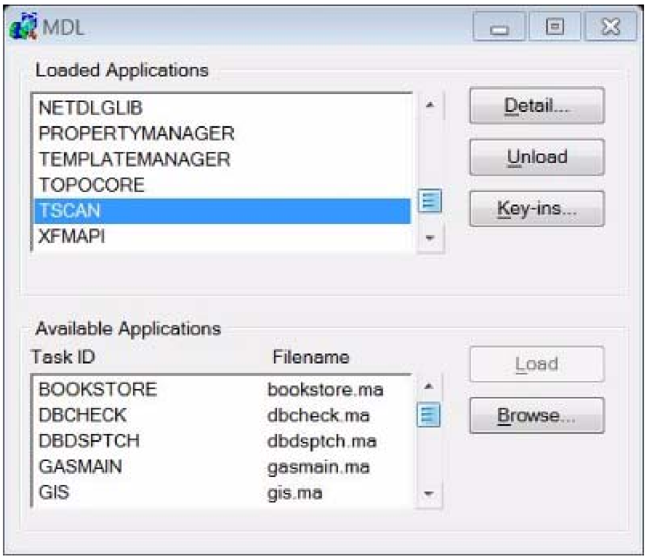
Unloading TerraScan

TerraScan is unloaded automatically when you exit MicroStation. Sometimes you may want to unload the application while continuing to work with MicroStation. This frees up the memory reserved by TerraScan.

➢ **To unload TerraScan:**

1. Select **MDL Applications** command from the **Utilities** pulldown menu in MicroStation.

   The **MDL** dialog opens:



2. In the **Loaded Applications** list, select TSCAN.
3. Click on the **Unload** button.

   OR

1. Key in MDL UNLOAD TSCAN.

   This unloads the application and frees the memory allocated for it.

Tool Reference

5 Basic Tools

TerraScan Settings

Settings control the way how tools and commands of TerraScan work. They are organized in logical categories. The **Settings** dialog is opened by the *Settings* tool.

Settings folder / category:	Settings category
Building vectorization / Editing tools	Alignment reports
Building vectorization / Levels	Block naming formulas
Building vectorization / Model	Classify Fence tool
Component fitting / Colors	Collection shapes
Component fitting / Levels	Default coordinate setup
Component fitting / Operation	Default flightline qualities
Component fitting / Profile	Elevation labels
Component fitting / Weights and styles	Loaded points
Coordinate transformations / Built-in projection systems	Operation
Coordinate transformations / Transformations	Point display
Coordinate transformations / US State Planes	Rail section templates
Coordinate transformations / User projection systems	Road section parameters
File formats / Default storage format	Scanner systems
File formats / EarthData binary format	Scanner waveform profiles
File formats / File name extensions	Section templates
File formats / LAS formats	Target objects
File formats / Leica formats	Tree types
File formats / Optech formats	Undo buffer
File formats / User point formats	
File formats / User trajectory formats	
Powerlines / Active line	
Powerlines / Profile layouts	
Powerlines / Tower functions	
Powerlines / Tower statuses	
Powerlines / Tower types	

Building vectorization / Editing tools

Editing tools category in **Buildings vectorization** folder defines the default tool of the **Building Edges tool box**.

Setting:	Effect:
Start ‘Modify Edge’ as default tool	If on, the *Modify Edge* tool is activated by default when other building model modification tools are reset.

Building vectorization / Levels

Levels category in **Building vectorization** folder sets levels which are used for drawing building models into the design file. These settings are only used for automatic building model vectorization and the *Check Building Models* tool. Tools for the single building construction use other settings defined in the *Construct Planar Building* tool itself. See chapter **3D Building Models** on page 189 for a detailed description of building vectorization options and tools in TerraScan.

Setting:	Effect:
Models to check	Levels for **Roof** and **Wall** polygons if a model is marked for checking. This is the status after automatic detection.
Active model	Levels for **Roof** and **Wall** polygons if a model is active. This is the case if the *Check Building Models* tool is started and a model is selected in the list of vector models.
Approved models	Levels for **Roof** and **Wall** polygons if a model has been approved. This is the status after the model has been checked and approved.

Building vectorization / Model

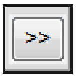
Model category in **Building vectorization** folder defines design settings for drawing automatically detected building models into the design file. See Chapter **3D Building Models** on page 189 for a detailed description of building vectorization options and tools in TerraScan.

Setting:	Effect:
Average elevations	Tolerance of elevation value variation in the ground around the building.
Walls start	Distance between the ground level and the start of walls below the ground.
Roof thickness	Distance between upper and lower level of the roofs. If the value is 0.0, the roof planes are represented by single polygons.
Roof	Color of roof polygons.
Roof sides	Color of roof side polygons. This is ignored if **Roof thickness** is set to 0.0.
Walls	Color of wall polygons.

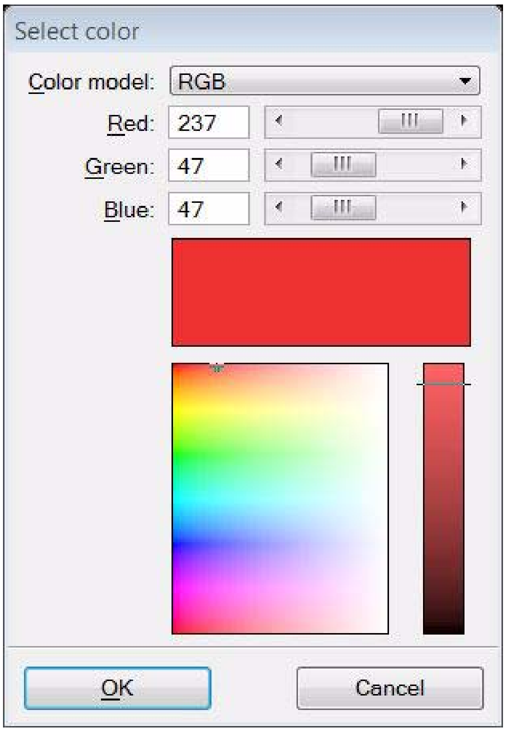
Component fitting / Colors

Colors category in **Component fitting** folder defines colors for drawing the different elements of geometry component fitting. All color values are given in RGB color space. See Chapter **Geometry Component Fitting** on page 390 for more information about the topic.

Setting:	Effect:
Arc	Color of fitted arc elements.
Clothoid	Color of fitted clothoid elements.
Line	Color of fitted line elements.
Hilite	Color of fitted highlighted elements.
Temporary	Color of fitted temporary elements.
Horizontal	Color of horizontal alignment elements.
Vertical	Color of vertical alignment elements.
Label	Color of label elements.
3D shape	Color of the final 3D shape.



The arrow button next to the color values opens a **Select color** dialog. The dialog lets you define a color for the corresponding elements by either typing color values, moving sliders or selecting a color in a color field. The dialog contains two color models, RGB (Red Green Blue) and HSV (Hue Saturation Value).



Component fitting / Levels

Levels category in **Component fitting** folder sets levels used for drawing geometry component fitting elements into the design file. See Chapter **Geometry Component Fitting** on page 390 for more information about the topic.

Setting:	Effect:
Surveyed alignment	Level for drawing the surveyed horizontal alignment elements.
Horizontal components	Level for drawing the fitted horizontal component elements.
Point weights	Level for drawing the point weight values for horizontal elements.
Profile frame	Level for drawing the frame of the profile that displays the geometry fitting elements.
Surveyed alignment	Level for drawing the surveyed vertical alignment elements.
Vertical components	Level for drawing the fitted vertical component elements.
Point weights	Level for drawing the point weight values for vertical elements.
3D shape	Level for drawing the final 3D shape.
Apply to file	Button to enforce the level assignment to already existing component fitting elements.

Component fitting / Operation

Operation category in **Component fitting** folder defines whether results of the geometry component fitting process are saved automatically or not. See Chapter **Geometry Component Fitting** on page 390 for more information about the topic.

Setting:	Effect:
Save geometry automatically	If on, results of the geometry fitting process are saved automatically.

Component fitting / Profile

Profile category in **Component fitting** folder defines the layout of the profile that is used for representing the geometry component fitting elements. See Chapter **Geometry Component Fitting** on page 390 for more information about the topic.

Setting:	Effect:
Horizontal scale	Scale factor for horizontal elements.
Vertical scale	Scale factor for vertical elements.
Elevation grid	Grid size for elevation values.
Relative margin	Size of a margin around the profile.
Station values	Number of decimals for station value labels.
Component values	Number of decimals for geometry component value labels.
Text size	Size of text elements. Given in millimeters plotted on paper.
Font	Font type of text elements. Uses a list of fonts available in MicroStation.

Component fitting / Weights and styles

Weights and styles category in **Component fitting** folder sets the line weights and styles for drawing geometry component fitting elements. It uses the line weights and styles available in MicroStation. See Chapter **Geometry Component Fitting** on page 390 for more information about the topic.

Setting:	Effect:
Line weight	Line weight of component fitting elements.
Hilite weight	Line weight of highlighted elements.
Hilite style	Line style of highlighted elements.
Preview weight	Line weight of preview elements.
Preview style	Line style of preview elements.

Coordinate transformations / Built-in projection systems

Builtin projection systems category in **Coordinate transformations** folder defines what projection systems are available for transformations. This effects lists in dialogs for transforming from WGS84 longitude and latitude coordinates to planar coordinate systems. Currently supported target systems are listed in the following table:

Setting:	Effect:
Belgium LB72/ BEREF2003	If on, transformation to LB72/BEREF2003 can be applied.
Deutsche Bahn GK	If on, transformation to Deutsche Bahn GK1 - GK5 can be applied.
Finnish KKJ	If on, transformation to KKJ using the selected **Equation** can be applied.
Finnish ETRS-TM35FIN and ETRS-GK	If on, transformation to ETRS-TM35FIN and ETRS-Gauss-Krueger zones 19 - 31 can be applied.
Northern Ireland	If on, transformation to Northern Ireland system can be applied.
Republic of Ireland	If on, transformation to Ireland Transverse Mercator system can be applied.
Japan	If on, transformation to Japanese zones 1 - 19 can be applied.
Netherlands RD/NAP 2008	If on, transformation to RD/NAP system can be applied.
South Africa	If on, transformation to South Africa LO system can be applied.
Swedish RT90	If on, transformation to Swedish RT90 system can be applied.
Swedish SWEREF99	If on, transformation to SWEREF99 system can be applied.
UK National Grid	If on, transformation to UK National Grid can be applied.
UTM WGS North	If on, transformation for given UTM **Zones** on the northern hemisphere can be applied.
UTM WGS South	If on, transformation for given UTM **Zones** on the southern hemisphere can be applied.

Coordinate transformations / Transformations

Transformations category in **Coordinate transformations** folder contains a list of coordinate transformations which can be used to transform the position of laser data, trajectories, and other data.

You can **Add**, **Edit**, and **Delete** transformation by using the corresponding buttons in the **Settings** dialog. The **Copy** button copies the selected transformation to the clipboard. With the **Paste** button you can paste a transformation from the clipboard. The **Derive** button can be used for **Deriving a transformation** from a set of control point pairs.

Seven types of coordinate transformations are supported:

- **Linear transformation**
- **Equation transformation**
- **Known points transformation**
- **Xy multiply transformation**
- **3D translate & rotate transformation**
- **3D Affine transformation**
- **Projection change transformation**

➢ **To define a new transformation:**

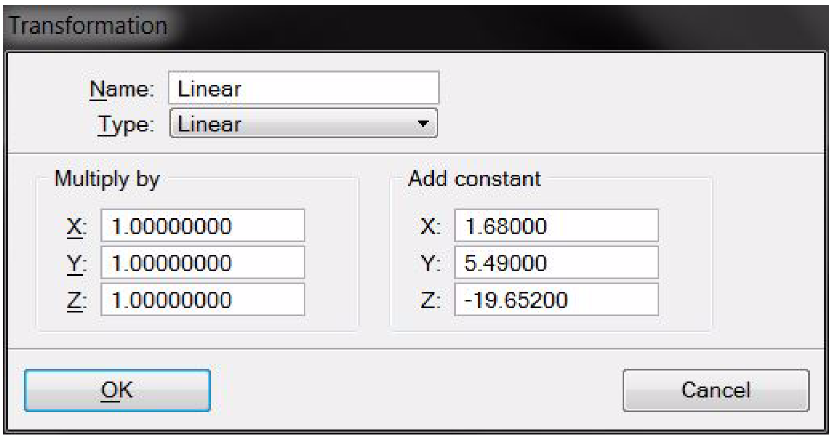
1. Open the **Transformations** category in the **Coordinate transformations** folder.
2. Click **Add** in the **Settings** dialog.

   This opens the **Transformation** dialog.

3. Type a **Name** for the transformation and select a transformation **Type**. Define the other settings depending on the transformation type.
4. Close the **Settings** dialog in order to save the modified settings for TerraScan.

Linear transformation

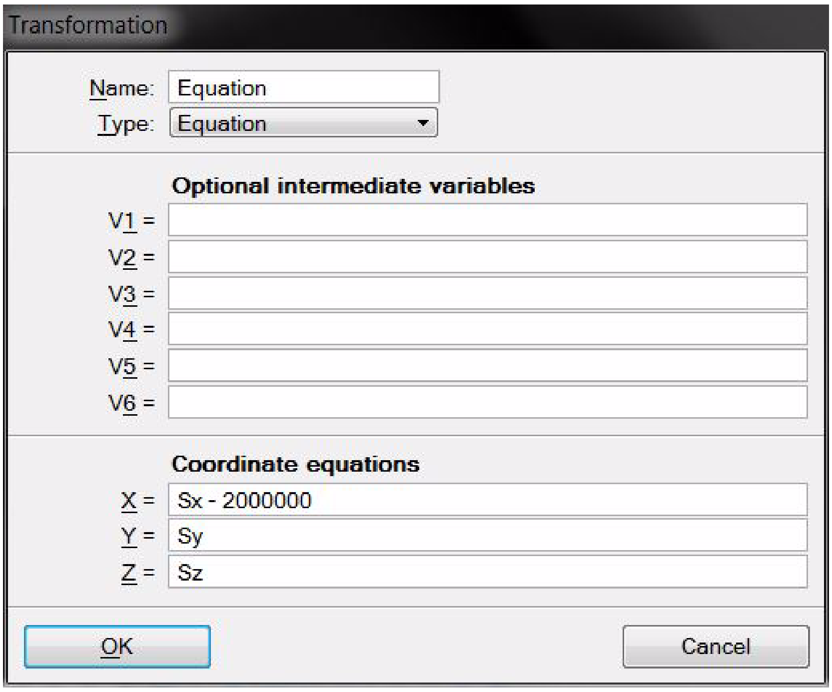
Linear transformation scales and/or translates coordinate values. You can assign a coefficient and a constant offset for each coordinate axis. The target coordinates are computed by multiplying the original coordinates with the given coefficient and by adding a given constant value.



Setting:	Effect:
Multiply by - X	Coefficient for multiplying the easting coordinate.
Multiply by - Y	Coefficient for multiplying the northing coordinate.
Multiply by - Z	Coefficient for multiplying the elevation coordinate.
Add constant - X	Value to add to the easting coordinate.
Add constant - Y	Value to add to the northing coordinate.
Add constant - Z	Value to add to the elevation coordinate.

Equation transformation

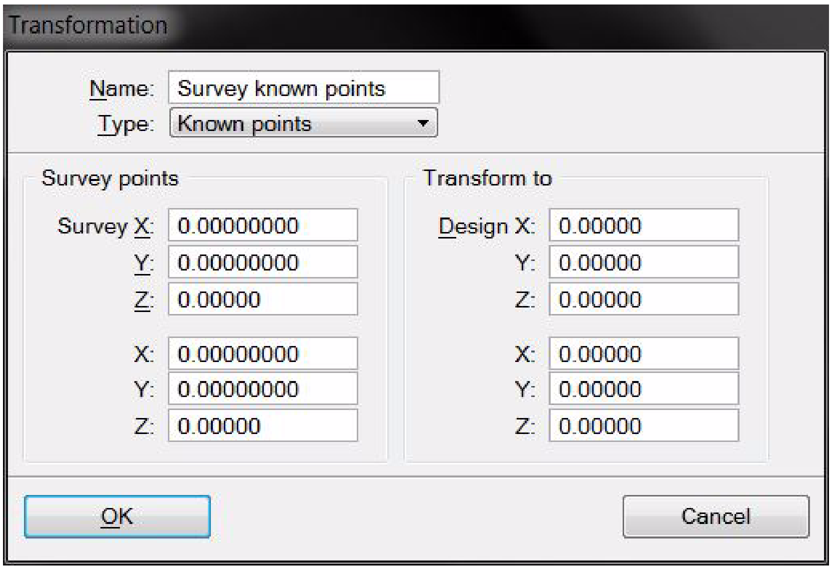
Equation transformation lets you define mathematical equations for computing new easting, northing, and elevation values from the source easting, northing, and elevation coordinates. You can also enter equations for up to six intermediate variables which are computed in order V1, V2, ..., V6 before evaluating new coordinates X, Y and Z.



Setting:	Effect:
V1, V2, ..., V6	Optional equations for calculating intermediate variables V1, V2, V3, V4, V5, and V6.
X, Y, Z	Equations for calculating the easting, northing, and elevation coordinates. The mathematical equation may contain: • Sx - survey file X coordinate. • Sy - survey file Y coordinate. • Sz - survey file Z coordinate. • Intermediate variables V1, V2, V3, V4, V5, and V6. • Mathematical functions such as sin(a), cos(a), tan(a), exp(a), log(a), log10(a), pow(a,b), sqrt(a), ceil(a), fabs(a) and floor(a) where a and b are floating point values.

Known points transformation

Known points transformation lets you specify the coordinates of two known points in the original coordinate system (survey coordinates) and their respective coordinates in the target system (design file coordinates).



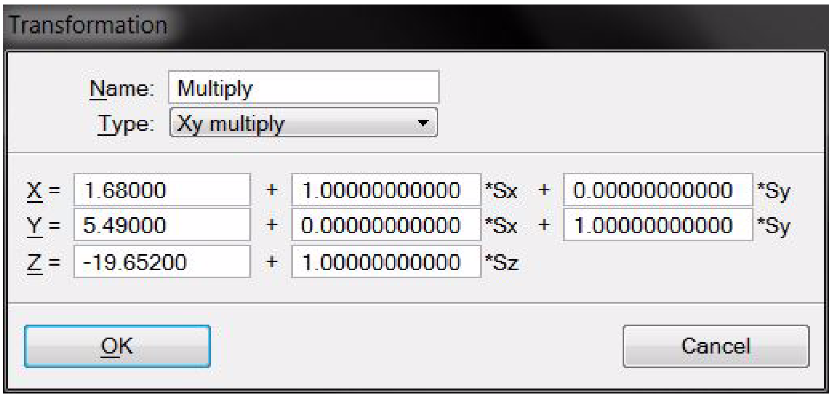
Setting:	Effect:
Survey X, Y, Z	First known point in the original coordinate system.
X, Y, Z	Second known point in the original coordinate system.
Design X, Y, Z	First known point in the target coordinate system.
X, Y, Z	Second known point in the target coordinate system.

Xy multiply transformation

Xy multiply applies a transformation using equations:

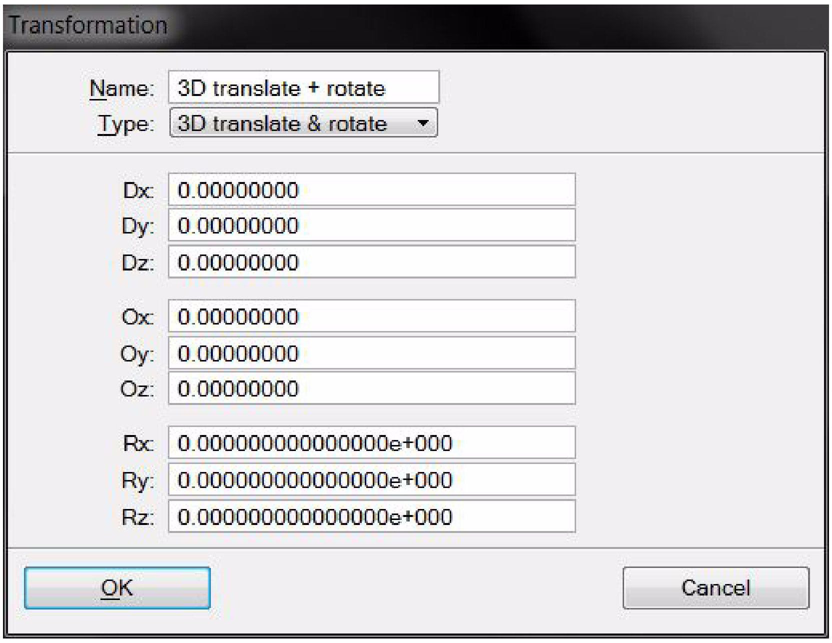
```
NewX = dx + a * Sx + b * Sy
NewY = dy + c * Sx + d * Sy
NewZ = dz + e * Sz
```

where dx, dy, dz, a, b, c, d, and e are constant parameters of the transformation and Sx, Sy, Sz are the original (survey) coordinates. This is often used as *2D Helmert* type of transformation.



3D translate & rotate transformation

3D translate & rotate applies a three dimensional translation and rotation to coordinates.



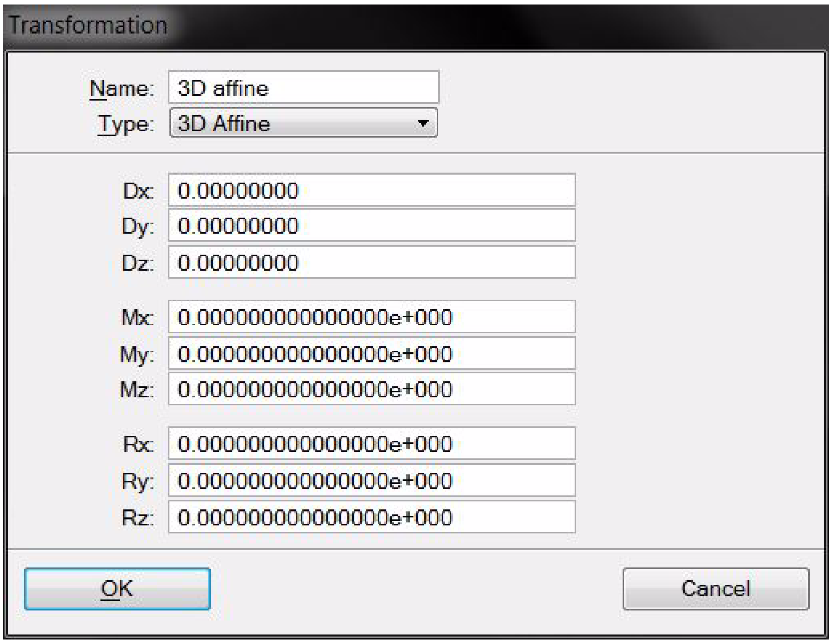
Setting:	Effect:
Dx, Dy, Dz	Values to add to X, Y, Z coordinates.
Ox, Oy, Oz	X, Y, Z coordinates of the rotation center point.
Rx, Ry, Rz	Rotation angle in radians around X, Y, Z axes.

3D Affine transformation

3D Affine applies separate translation, rotation and scaling for each coordinate axis. The transformation is defined by equations:

```
NewX = dx + (1.0 + mx) * X + rz * Y - ry * Z
NewY = dy + (1.0 + my) * Y - rz * X + rx * Z
NewZ = dz + (1.0 + mz) * Z + ry * X - rx * Y
```

where dx, dy, dz, mx, my, mz, rz, ry, and rz are constant parameters of the transformation and X, Y, Z are the original coordinates.



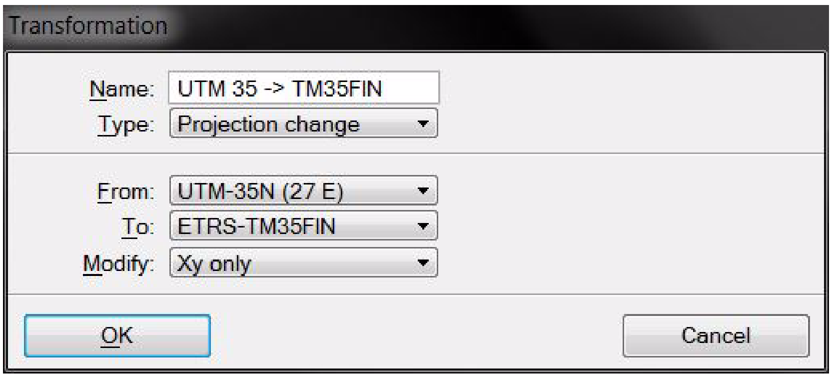
Setting:	**Effect:**
Dx, Dy, Dz	Values to add to X, Y, Z coordinates (translation).
Mx, My, Mz	Factors to scale the data along the X, Y, Z axes.
Rx, Ry, Rz	Rotation angle in radians around X, Y, Z axes.

Projection change transformation

Projection change transforms coordinates from one projection system to another. The software transforms the X, Y, Z coordinates from the source projection system back into WGS84 geocentric X, Y, Z and then computes the transformation into the target projection system.

All projections systems that are active in **Coordinate transformations / Built-in projection systems**, **Coordinate transformations / US State Planes**, or defined in are available for a projection change transformation.

If you already applied a geoid correction, you should run a reverse geoid correction to the data set before using a projection change transformation. This is essential in cases where the source and the target systems use different ellipsoids or datums. A geoid correction or a reverse geoid correction is only applied automatically if the UK National Grid system is used in the transformation.



Setting:	**Effect:**
From	Source projection system.
To	Target projection system.

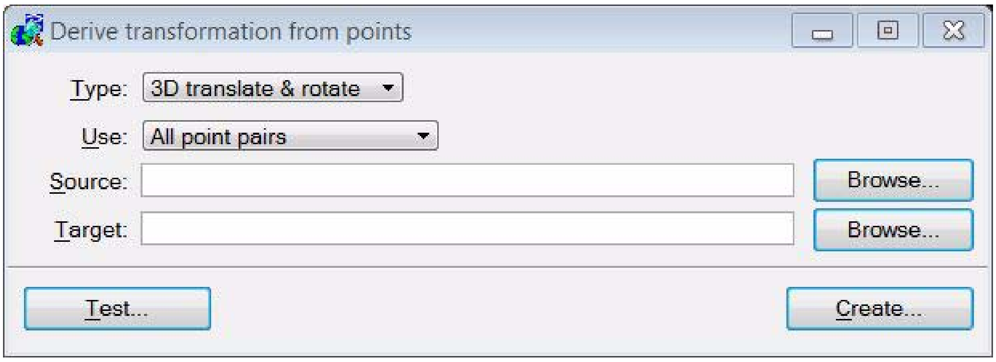
Setting:	Effect:
Modify	Coordinate values to modify: • **Xyz -** modifies all coordinates. • **Xy only -** no changes to elevation values.

✍ You can copy transformations from one Terra application to another. Select the transformation in the **Settings** dialog and click on the **Copy** button to copy the definition to the clipboard. Click on the **Paste** button in the other Terra application to paste the definition.

Deriving a transformation

You can also derive transformation parameter values from point pairs. This requires that identical control points (point pairs) are available in source and target coordinate values. The points must be stored in text files. The number of required control point pairs depends on the transformation type.

To derive a transformation, click on the **Derive** button in the **Settings** dialog. This opens the **Derive transformation from points** dialog:



Setting:	Effect:
Type	Type of the derived transformation: • **2D transformation -** parameter values for a 2D Helmert transformation are derived. • **3D translate & rotate -** parameter values for a 3D translation and rotation transformation are derived. • **7 parameter affine -** parameter values for a 3D affine transformation (7 parameters) are derived. • **9 parameter affine -** parameter values for a 3D affine transformation (9 parameters) are derived.
Use	Points used for deriving the transformation: • **All point pairs -** uses all control point pairs. • **Inside source fence only -** points inside a fence in the source coordinate system are used. • **Inside target fence only -** points inside a fence in the target coordinate system are used.
Source	Text file that contains the point pair coordinates in the source system.
Target	Text file that contains the point pair coordinates in the target system.

The transformation derivation can be tested by using the **Test** button. This computes the parameter values and displays the result in a report window. To create the transformation, click on the **Create** button. This opens the **Transformation** dialog that displays the derived parameter values. Type a **Name** for the transformation and click OK in order to add the transformation to the list in the **Settings** dialog.

Coordinate transformations / US State Planes

US State Planes category in **Coordinate transformations** folder contains a list of US State Plane projection systems using NAD83 datum. Check the toggle box of those state plane systems you want to use.

You can view the parameters of a system by using the **View** button. In case you need to change the parameters of a built-in US State Plane definition, you can use the **Copy** button to copy/paste the system into **Coordinate transformations / User projection systems**.

Coordinate transformations / User projection systems

User projection systems category in **Coordinate transformations** folder contains a list of user defined projection systems. You can define your projection system based on **Transverse Mercator / Gauss-Krueger**, **Lambert conic conformal** or **Hotine oblique mercator** projection.

A projection system definition can be divided into three distinct parts:

- **Ellipsoid** - defined by **Semi-major axis** and **Inverse flattening**.
- **Datum -** defined by seven parameter Bursa/Wolfe transformation.
- **Projection -** defined by the projection type, true origin, false origin, scale factor at the central meridian, and distance unit.

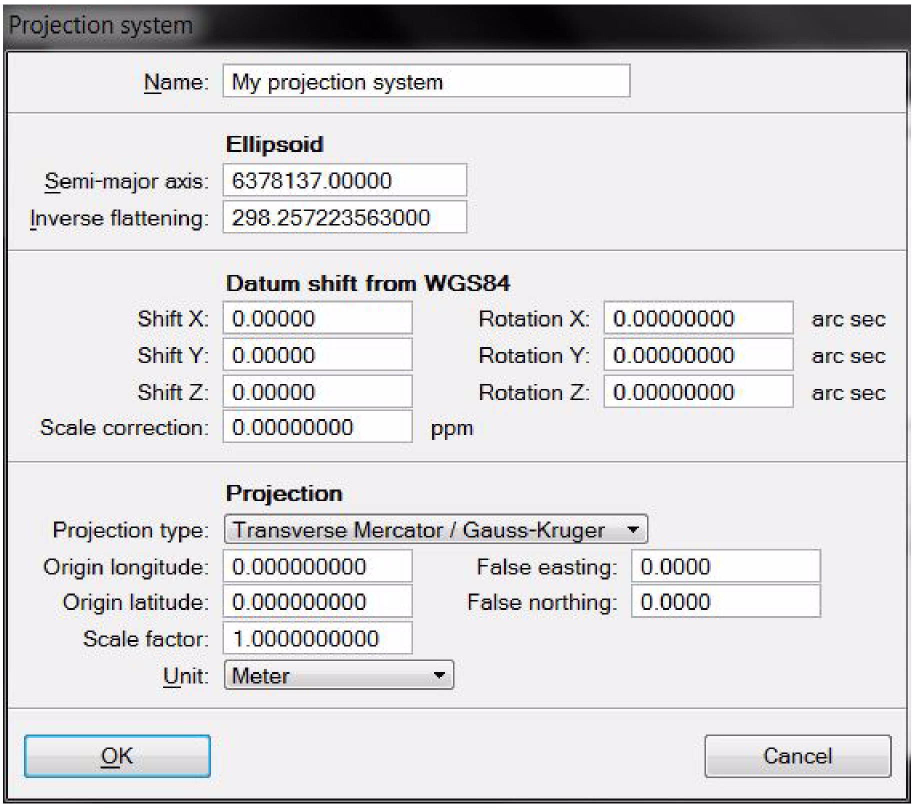
The list of user projection system displays a toggle box for each row. The toggle box indicates whether a projection system is active or not. Only active projection systems can be selected when applying a transformation. To activate or deactivate a projection system, place a data click inside its toggle box in the list.

You can **Add**, **Edit**, and **Delete** user projection systems by using the corresponding buttons in the **Settings** dialog. The **Copy** button copies the selected projections system definition to the clipboard. With the Paste button you can paste a projection system definition from the clipboard.

➢ **To define a new projection system:**

1. Open the **User projection systems** category in the **Coordinate transformations** folder.
2. Click **Add** in the **Settings** dialog.

   This opens the **Projection system** dialog:



3. Define settings and click OK.
4. Activate the projection system.
5. Close the **Settings** dialog in order to save the modified settings for TerraScan.

Setting:	Effect:
Name	Descriptive name for the projection system.
Semi-major axis	Semi-major axis of the target ellipsoid.

Setting:	**Effect:**
Inverse flattening	Inverse flattening of the target ellipsoid.
Shift X	Datum X shift from WGS84 to the target system in meter.
Shift Y	Datum Y shift from WGS84 to the target system in meter.
Shift Z	Datum Z shift from WGS84 to the target system in meter.
Rotation X	Datum rotation around the X axis in arc seconds.
Rotation Y	Datum rotation around the Y axis in arc seconds.
Rotation Z	Datum rotation around the Z axis in arc seconds.
Scale correction	Datum scale correction as parts per million. The actual scale factor is computed as *1.0 + (0.000001 * ScaleFactor)*.
Projection type	Type of the projection system: **Transverse Mercator/Gauss-Kruger**, **Lambert conic conformal**, or **Hotine oblique mercator**.
Origin longitude	Longitude of the true origin in decimal degrees.
Origin latitude	Latitude of the true origin in decimal degrees.
False easting	Map coordinate easting of the true origin.
False northing	Map coordinate northing of the true origin.
Scale factor	Scale factor on the central meridian.
Unit	Distance unit: **Meter**, **International foot**, **US Survey Foot**, or **International yard**.

✍ You can copy user projection systems from one Terra application to another. Select the system in the **Settings** dialog and click on the **Copy** button to copy the definition to the clipboard. Click on the **Paste** button in the other Terra application to paste the definition. You can also paste the definition in a text editor in order to save it into a text file.

File formats / Default storage format

Default storage format category in **File formats** folder defines what binary format is the default storage format for laser data and what GPS time format is the default format for storing time stamps. The formats are used by default for new projects in the **Project information** dialog opened by **New project** command.

Setting:	Effect:
Format	Default format for laser data: **EarthData EEBN**, **EarthData EBN**, **Fast binary**, **LAS 1.0**, **LAS 1.1**, **LAS 1.2**, **Scan binary 16 bit lines**, or **Scan binary 8 bit lines**.
Time type	Default format for storingtime stamps of laser points: **GPS seconds-of-week**, **GPS standard time**

File formats / EarthData binary format

EarthData binary format category in **File formats** folder defines how class codes from EarthData binary files are converted into TerraScan classes. For each class code a corresponding TerraScan class should be selected.

To change a class code conversion, select the line in the list of classes. This activates the **Scan class** list which contains all classes of the active class definitions in TerraScan. Select a new TerraScan class from the pulldown list.

File formats / File name extensions

File name extensions category in **File formats** folder defines default file extensions for various file formats. These extensions are used as default values when you output points from TerraScan.

Setting:	Effect:
East North Z	Extension for plain xyz text files. Default is **xyz**.
Code East North Z	Extension for text files containing point class and coordinates. Default is **txt**.
TerraScan binary	Extension for 8-bit/16-bit binary files in TerraScan format. Default is **bin**.
EarthData binary	Extension for binary files in EarthData format. Default is **ebn**.
Fast binary	Extension for files in TerraScan fast binary format. Default is **fbi**.
LAS binary	Extension for binary files in LAS format. Default is **las**.

File formats / LAS formats

LAS formats category in **File formats** folder defines the bit depth of color values in LAS files. The setting can be used to read LAS files with incorrectly stored color values.

Setting:	Effect:
Bit depth	Bit depth of color values in the LAS file: • **Low 8 bits -** color values are stored as 8 bit values. • **Low 12 bits -** color values are stored as 12 bit values. • **Correct 16 bits -** color values are stored as 16 bit values. This is the correct value according to the LAS standard format definition.

Set the value to **Low 8 bits** or **Low 12 bits** if your read or import LAS files with incorrect color values. Set the value to **Correct 16 bits** in order to store color values correctly.

File formats / Leica formats

Leica formats category in **File formats** folder defines rules how to interpret intensity values coming from specific Leica file formats.

Setting:	Effect:
Read	Reading intensity of Leica LDI files: **Raw intensity** or **Normalized intensity**.

File formats / Optech formats

Optech formats category in **File formats** folder defines rules how to interpret data coming from specific Optech file formats.

Setting:	Effect:
Scale intensity	Factor for scaling intensity values.
Use as last echo	Defines which value is used as last echo: **First xyz**, **Second xyz**, or **Lower xyz**.
Ignore first echoes	If on, TerraScan filters out first echoes from **Optech xyzxyzii** type files based on the elevation difference of the first and last echo.
Less than	First echos less than the given elevation difference above the corresponding last echo are ignored.

File formats / User point formats

User point formats category in **File formats** folder contains a list of user-defined point formats. You can define your own formats which can be used for the input or output of point data. The software can read any text files where each row contains the information of one point and the point attributes are organized into columns (fields). The file format definition determines what fields are included for each point and what is the order of the fields.

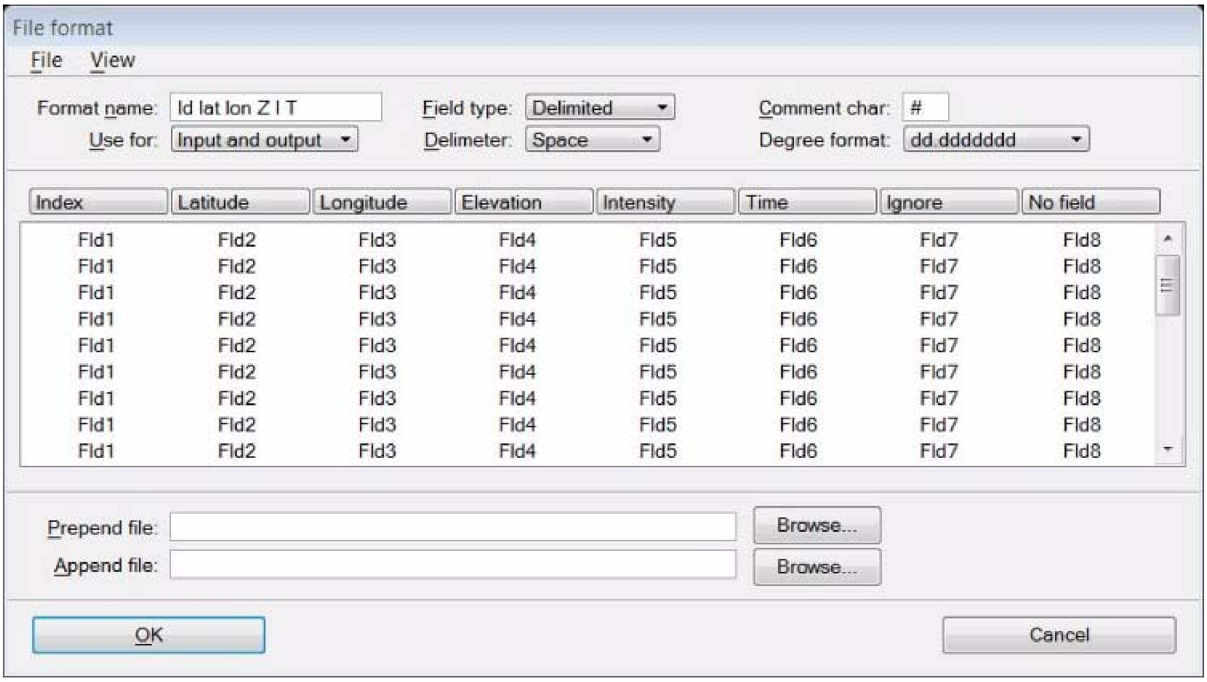
The text file formats may contain delimited fields or fixed length fields. The delimiter can be comma, space, tabulator, or semicolon. A fixed length fields is defined by constant column widths and positions in each row.

You can **Add**, **Edit**, and **Delete** point formats by using the corresponding buttons in the **Settings** dialog. The **Copy** button creates an identical copy of a selected format definition. The **Move up** and **Move down** buttons change the order of formats in the list.

➢ **To define a new point format:**

1. Open the **User point formats** category in the **File formats** folder.
2. Click **Add** in the **Settings** dialog.

   The **File format** dialog opens:



3. (Optional) Select **Load example** command from the **File** pulldown menu of the **File format** dialog.

   This reads the first lines of the text file and shows its content in the field list. The software also tries to detect the **Field type** and the **Delimiter**.

4. If required, change the number of fields that are available in the dialog by using the commands from **View** pulldown menu.
5. Type a **Format name** and define the other settings.
6. Select the correct attribute for a field. The list of attributes is displayed when you click on the **No field** button.
7. Click OK to the **File format** dialog.
8. Close the **Settings** dialog in order to save the modified settings for TerraScan.

Setting:	**Effect:**
Format name	Descriptive name of the new format.

Setting:	**Effect:**
Use for	Defines the usage of the format: • **Input only** - files of this format can be loaded into TerraScan using the **Read points** command or the *Load Airborne Points* tool. • **Output only** - files of this format can be saved into new text files using the **Save points As** command. • **Input and output** - files of this format can be loaded into TerraScan and saved into new text files.
Field type	Defines fields are separated in the text file: **Delimited** or **Fixed length**.
Delimiter	Delimeter character used in text files: **Space**, **Tabulator**, **Comma**, or **Semicolon**. This is only active if **Field type** is set to **Delimited**.
Comment char	Character that introduces comment lines in the text file. Lines beginning with this character are ignored when points are read from a text file.
Degree format	Defines the format of longitude and latitude values in degrees, minutes, and seconds. This is only active if **Longitude** or **Latitude** are selected as attributes.
No field	Selection of what point attribute is stored in the field: • **No field** - no field defined in the text file. • **Ignore** - the column in the text file is ignored. • **Easting**, **Northing**, **Elevation** - xyz coordinates. • **Longitude**, **Latitude** - position in degrees, minutes, and seconds. • **Class** - class number. • **Code** - class code. • **Echo type** - echo type as text string. • **Index** - unique number for each point. • **Intensity** - intensity value as integer. • **Line** - line number. • **Time** - time stamp. • **Collection** - number of collection shape. Only used with **Output collections** command for projects. • **Surface dz** - difference between a point and a TerraModeler surface. • **Red**, **Green**, **Blue** - RGB color values. • **Echo number** - echo number as number. • **Number of echos** - total number of echos at the position of a point. • **Mirror angle** - scan angle in degrees. Values must range between -128 to +127. • **Scanner** - scanner number.
Prepend file	Location of a text file from which the content is added at the beginning of an output file. This is not active if **Use for** is set to **Input only**.
Append file	Location of a text file from which the content is added at the end of an output file. This is not active if **Use for** is set to **Input only**.

✍ There are some text file format already implemented in TerraScan. See **Supported file formats** on page 478 for a list of implemented file formats. User point formats are stored in a configuration file OUTFMT.INF in the TerraScan installation folder. You can copy this file to other computers in order to make point formats available on them.

File formats / User trajectory formats

User trajectory formats category in **File formats** folder contains a list of user-defined trajectory formats. You can define your own file formats which can be used when reading in trajectory information from text files.

For the definition of trajectory formats, the same steps and settings apply as for point formats described above in **File formats / User point formats**. The differences in usage and attributes are listed in the table below.

Setting:	Effect:
Use for	Defines the usage of the format: • **Input only** - files of this format can be loaded into TerraScan using the **Import files** command from the **Manage Trajectories** dialog. • **Output only** - files of this format can be saved into new text files using the **Output positions** command from the **Manage Trajectories** dialog. • **Input and output** - files of this format can be loaded into TerraScan and saved into text files.
No field	Selection of what trajectory position attribute is stored in the field: • **No field** - no field defined in the text file. • **Ignore** - the column in the text file is ignored. • **Time** - time stamp. • **Easting**, **Northing**, **Elevation** - xyz coordinates. • **Longitude**, **Latitude** - position in degrees, minutes, and seconds. • **Heading**, **Roll**, **Pitch** - orientation angles. • **X|Y|Z accuracy** - accuracy estimates for xyz position values. • **Heading|Roll|Pitch accuracy** - accuracy estimates for orientation angle values.

✍ Trajectory formats are stored in a configuration file TRAJFMT.INF in the TerraScan installation folder. You can copy this file to other computers in order to make trajectory formats available on them.

Powerlines / Active line

Active line category in **Powerlines** folder defines settings for the display of an active powerline line string. The settings effect the display of a line string element after it has been selected by the *Activate Powerline* tool.

Setting:	Effect:
Hilite	Parts of the line string that are highlighted: **No hilite**, **Vertices**, or **Line segments**.
Color	Color of a highlighted line string. Uses the active color table of MicroStation.
Weight	Line weight of a highlighted line string. Uses MicroStation line weights.
Style	Line style of a highlighted line string. Uses MicroStation line styles.

Powerlines / Profile layouts

Profile layouts category in **Powerlines** folder contains a list of user defined profile layouts. Each layout definition contains a list of data rows that appear below the profile.

You can **Add**, **Edit**, and **Delete** profile layouts by using the corresponding buttons in the **Settings** dialog. The **Copy** button creates an identical copy of a selected layout definition. You can **Add**, **Edit**, and **Delete** bottom rows to a profile layout by using the corresponding buttons in the **Profile layout** dialog.

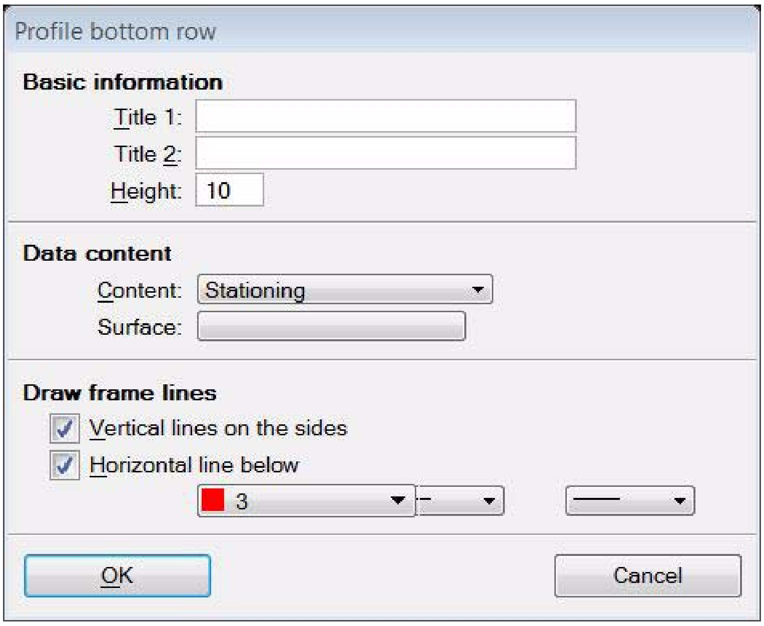
➢ **To define a new profile layout:**

1. Open the **Profile layouts** category in the **Powerlines** folder.
2. Click **Add** in the **Settings** dialog.

   The **Profile layout** dialog opens.

3. Type a **Name** for the profile layout.
4. Click **Add** in the **Profiles layout** dialog in order to add a new data row that is displayed below a profile.

   The **Profile bottom row** dialog opens:



5. Define basic information settings.
6. Select an auto-text option for the **Content** list as well as additional settings depending on the content selection. Choose **Other** as **Content** if nothing of the list entries fit to your data.
7. Select settings for frame lines.
8. Click OK in the **Bottom row** dialog.
9. Add more data rows if necessary.
10. Click OK to the **Profile layout** dialog.
11. Close the **Settings** dialog in order to save the modified settings for TerraScan.

Setting:	Effect:
Title 1	Text used as first line of a title in the bottom row.
Title 2	Text used as second line of a title in the bottom row.
Height	Height of the bottom row. Given in millimeters on paper.

Setting:	Effect:
Content	Defines the type of information displayed in the bottom row: • **Surface elevations -** elevations of surfaces of the given **Surface** type. • **Stationing -** stations along the alignment element of the profile. • **Crossing object stationing -** station along the alignment element where another object crosses the powerline. • **Tower span** • **Tower turn angle** • **Other -** space reservation for any other content that can be added manually.
Vertical lines on the sides	If on, vertical lines are drawn on the left and right side of the bottom row.
Horizontal line below	If on, a horizontal line is drawn below the bottom row using the given symbology.

Powerlines / Tower functions

Tower functions category in **Powerlines** folder contains a list of different functions for powerline towers. Typical function examples are suspension towers, tension towers, and dead-end towers.

You can **Add**, **Edit**, and **Delete** tower functions by using the corresponding buttons in the **Settings** dialog.

A tower function is defined by an **Abbreviation** and a **Description** which can be typed in the fields of the **Tower function** dialog.

The tower function is applied when the tower model is placed using the *Place Tower* tool. The information can be included in a report created by the *Export Powerline* tool.

✍ Tower functions are stored in a configuration file TOWER_FUNCTIONS.INF in the TerraScan installation folder. You can copy this file to other computers in order to make tower functions available on them.

Powerlines / Tower statuses

Tower statuses category in **Powerlines** folder contains a list of different statuses for powerline towers. Status examples may be existing, planned, broken, etc.

You can **Add**, **Edit**, and **Delete** tower statuses by using the corresponding buttons in the **Settings** dialog.

A tower status is defined by an **Abbreviation** and a **Description** which can be typed in the fields of the **Tower status** dialog.

The tower status is applied when the tower model is placed using the *Place Tower* tool. The information can be included in a report created by the *Export Powerline* tool.

✍ Tower statuses are stored in a configuration file TOWER_STATUSES.INF in the TerraScan installation folder. You can copy this file to other computers in order to make tower statuses available on them.

Powerlines / Tower types

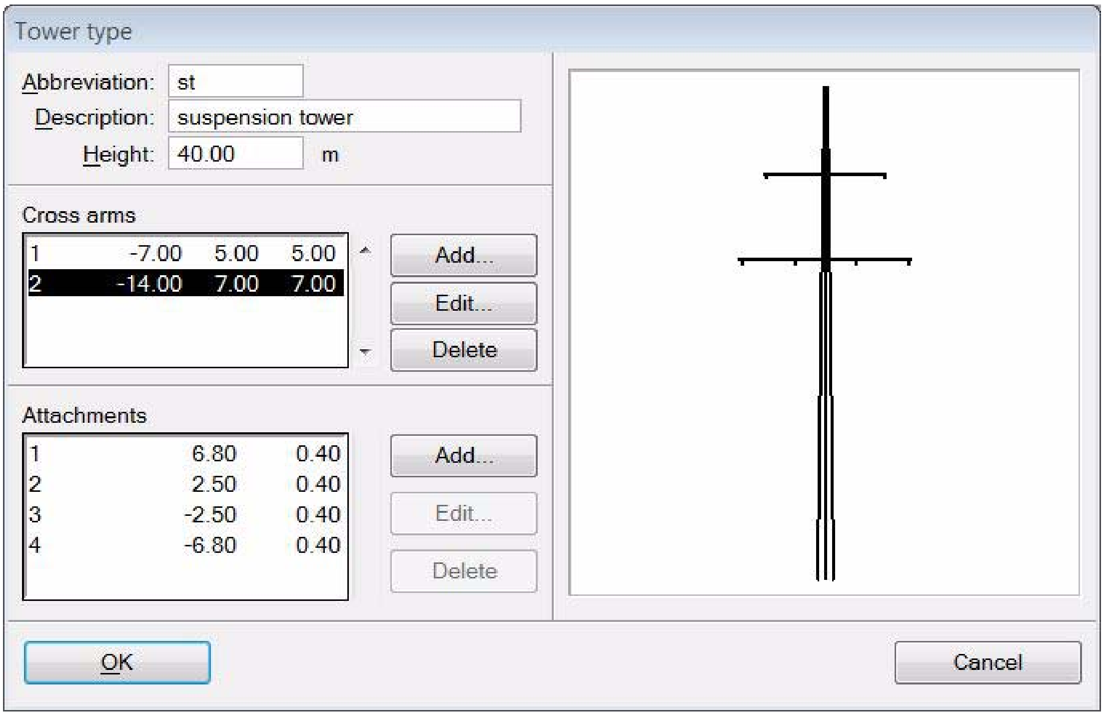
Tower types category in **Powerlines** folder contains a list of user-defined tower types. The tower type determines the general design of a tower, including the number, position and length of cross arms and attachments.

You can **Add**, **Edit**, and **Delete** tower types by using the corresponding buttons in the **Settings** dialog. You can **Add**, **Edit**, and **Delete** cross arms and attachments for a tower type by using the corresponding buttons in the **Tower type** dialog.

➢ **To add a new tower type:**

1. Open the **Tower types** category in the **Powerlines** folder.
2. Click **Add** in the **Settings** dialog.

   The **Tower type** dialog opens:

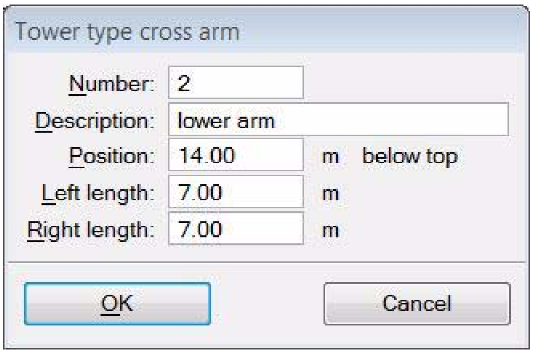


3. Define **Abbreviation**, **Description** and **Height** values.

Setting:	Effect:
Abbreviation	Abbreviation of the tower type.
Description	Description of the tower type.
Height	Height of a tower.
Cross arms	List of cross arms for this tower type. Use buttons next to the list to **Add**, **Edit**, and **Delete** cross arms.
Attachments	List of attachments per cross arm. Select a cross arm and use buttons next to the list to **Add**, **Edit**, and **Delete** cross arms.

4. Click **Add** in the **Tower type** dialog in order to add a cross arm.

This opens the **Tower type cross arm** dialog:

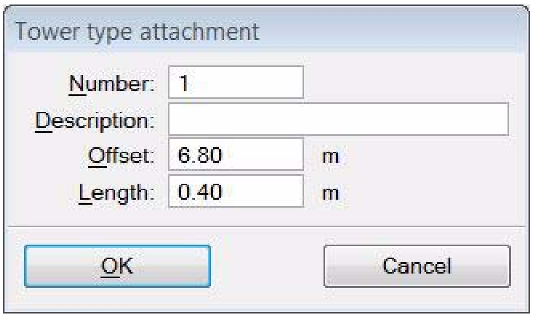


5. Define settings and click OK.

Setting:	**Effect:**
Number	Number of the cross arm.
Description	Description of the cross arm.
Position	Position of the cross arm relative to the top of the tower.
Left lenght	Length of the cross arm to the left side of the tower.
Right length	Length of the cross arm to the right side of the tower.

6. Repeat steps 4 and 5 for all cross arms that belong to this tower type.
7. Select a cross arm and click **Add** in the **Tower type** dialog in order to add an attachment to the selected cross arm.

   This opens the **Tower type attachment** dialog:



8. Define settings and click OK.

Setting:	**Effect:**
Number	Number of the attachment.
Description	Description of the attachment.
Offset	Position of the attachment along the cross arm relativ to the tower center. A positive offset creates an attachment right of the tower center, a negative offset left of the tower center.
Length	Length of the attachment.

9. Repeat steps 7 and 8 for all attachments per cross arm and all cross arms of the tower type.
10. Click OK to the **Tower type** dialog.
11. Close the **Settings** dialog in order to save the modified settings for TerraScan.

The values defined for the tower height as well as position and length of cross arms and attachments do not have to be exact if towers are placed with the *Place Tower* tool using no template. For placing towers using templates the accuracy of the values determine how well a template fits to the real towers design.

It is recommended to enter a text in the **Description** fields of **Tower type** and **Tower type cross arm** dialogs, because the editing tools for powerline processing refer to this field. The other descriptive information is mainly used in reports. See chapter **Powerlines** on page 118 for more information about powerline processing.

✍ Tower types are stored in a configuration file TOWER_TYPES.INF in the TerraScan installation folder. You can copy this file to other computers in order to make tower types available on them.

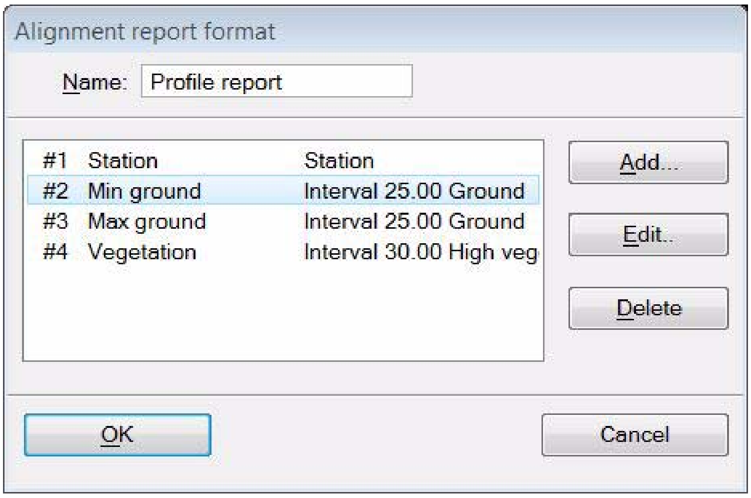
Alignment reports

Alignment reports category contains a list of alignment report formats. The formats are used by the **Output alignment report** command. The report format defines what information is included in the output report along an alignment element. It consists of a descriptive name and a list of columns.

➢ **To define a new alignment report format:**

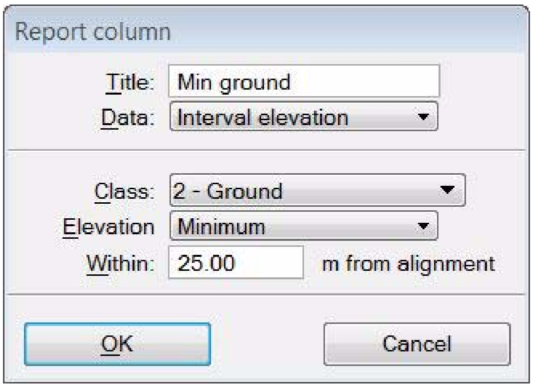
1. Open the **Alignments report** category.
2. Click **Add** in the **Settings** dialog.

   The **Alignment report format** dialog opens:



3. Type a **Name** for the report format.
4. Click **Add** in the **Alignment report format** dialog in order to add a new column definition to the report.

   The **Report column** dialog opens:



5. Define settings and click OK.
6. Add more columns if necessary.
7. Click OK in the **Alignment report format** dialog.
8. Close the **Settings** dialog in order to save the modified settings for TerraScan.

Setting:	Effect:
Title	Title of the report column.

Setting:	**Effect:**
Data	Content of the report column: • **Alignment station -** station value along an alignment element. • **Alignment easting -** easting coordinate at stations along the alignment element. • **Alignment northing -** northing coordinate at stations along the alignment element. • **Alignment elevation -** elevation coordinate at stations along the alignment element. • **Interval elevation -** minimum or maximum **Elevation** of laser points in a given **Class**. A rectangular search area is defined by the offset left and right of the alignment station given in the **Within** field and the interval step size along the alignment element given at report output time. • **Point elevation -** closest, minimum, average, or maximum **Elevation** value from laser points in a given **Class** inside a circular area. The center of the circular area is at the given **Offset** from the alignment station, the size is determined by the **Radius** value. • **Surface elevation -** elevation value of a surface model in TerraModeler. This requires at least on surface model in TerraModeler. The elevation value is computed from the selected **Surface** type at the x and y location of the alignment station plus the given **Offset** value. • **Column difference -** computes the difference between two other **Columns** of the report. • **Alert -** writes an asterisk character (*) in the report if the difference between two columns is bigger or smaller than a given limit.
Offset	Distance from the alignment element. The **Data** value is determined from the location defined by the alignment station and the offset. A negative offset is left, a positive offset right of the alignment element.

✍ Alignment report formats are stored in a configuration file ALREPFMT.INF in the TerraScan installation folder. You can copy this file to other computers in order to make alignment report formats available on them.

Block naming formulas

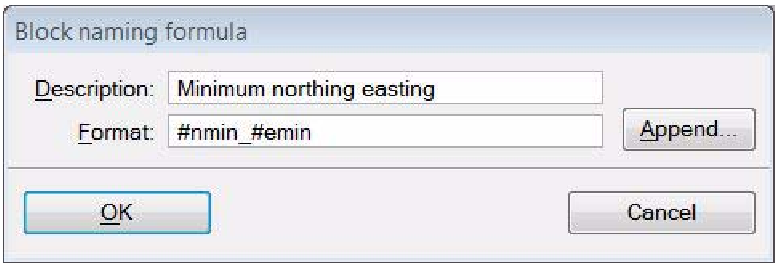
Block naming formulas category shows a list of naming conventions that can be used to create block names for TerraScan projects. The naming formulas are available in **Add by boundaries** dialog which can be opened from the **Block pulldown menu** of the TerraScan **Project** window.

You can **Add**, **Edit**, and **Delete** block naming formulas by using the corresponding buttons in the **Settings** dialog.

➢ **To create a new block naming formula:**

1. Open the **Block naming formula** category.
2. Click **Add** in the **Settings** dialog.

   The **Block naming formula** dialog opens:



3. Define settings and click OK.
4. Close the **Settings** dialog in order to save the modified settings for TerraScan.

Setting:	Effect:
Description	Description of the formula.
Format	Format of the formula.
Append	Opens the **Append field** dialog. The dialog contains a list of variables which can be added to a formula: • **minimum | maximum easting | northing** - minimum or maximum values of easting or northing corner coordinates of a block boundary. • **block size** - block height or width value, whatever is larger. • **block number** - number according to the selection order of block boundaries.

✍ Block naming formulas are stored in a configuration file BLOCKNAMING.INF in the TerraScan installation folder. You can copy this file to other computers in order to make block naming formulas available on them.

Classify Fence tool

Classify Fence tool category determines the symbology of a fence displayed by the *Classify Fence* tool.

Setting:	Effect:
Color	Color of the fence. Uses the active color table of MicroStation.
Weight	Line weight of the fence. Uses MicroStation line weights.
Style	Line style of the fence. Uses MicroStation line styles.

Collection shapes

Collection shapes category shows a list of collection shape types. Collection shapes can be used to group laser points. Typical collection shape type examples are building, road, or tree. The actual grouping is done by placing collection shapes with the *Place Collection Shape* tool. The collection shape type determines what kind of an object the polygon encloses as well as the level and the symbology of the polygon.

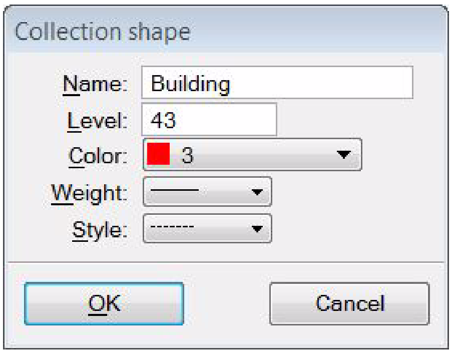
Collection shapes can be further used to output laser points into separate files according to the collection or group they belong to. See **Output collections** command from the **Tools pulldown menu** of the TerraScan **Project** window.

You can **Add**, **Edit**, and **Delete** collection shape types by using the corresponding buttons in the **Settings** dialog.

➢ **To create a new collection shape type:**

1. Open the **Collection shapes** category.
2. Click **Add** in the **Settings** dialog.

   The **Collection shape** dialog opens:



3. Define settings and click OK.
4. Close the **Settings** dialog in order to save the modified settings for TerraScan.

Setting:	**Effect:**
Name	Name of the collection shape type.
Level	Level on which collections shapes are placed.
Color	Color of collections shapes. Uses the active color table of MicroStation.
Weight	Line weight of collection shapes. Uses MicroStation line weights.
Style	Line style of collection shapes. Uses MicroStation line styles.

✍ Collection shapes are stored in a configuration file COLLECTION_SHAPES.INF in the TerraScan installation folder. You can copy this file to other computers in order to make collection shape definitions available on them.

Default coordinate setup

Default coordinate setup category defines the default values for the coordinate setup of Terra applications. The default values can be changed by using the *Define Coordinate Setup* tool.

Setting:	Effect:
Resolution	Default integer steps per master unit in a design file used by Terra applications. This effects the number of decimals stored for laser points in TerraScan.
Easting	Default easting coordinate of the origin in the design file.
Northing	Default northing coordinate of the origin in the design file.
Elevation	Default elevation coordinate of the origin in the design file.

Default flightline qualities

Default flightline qualities category defines quality tags for flightlines. These quality settings are used by **Cut overlap** command in case trajectories have not been imported.

Setting:	Effect:
Bad	Flightline number range with **Bad** quality tag.
Poor	Flightline number range with **Poor** quality tag.
Normal	Flightline number range with **Normal** quality tag.
Good	Flightline number range with **Good** quality tag.
Excellent	Flightline number range with **Excellent** quality tag.

Elevation labels

Elevation labels category defines the format of elevation values drawn as text elements. The settings are used if points are drawn into the design file using the **Write to design file** command and if the points are drawn as **Elevation labels** which is set in the **Define Classes** dialog.

Setting:	Effect:
Accuracy	Number of decimals to display for elevation values.
Display plus	If on, positive elevation values start with a plus sign.
Display minus	If on, negative elevation values start with a minus sign.

Loaded points

Loaded points category defines the symbology for highlighting points in TerraScan. It is used, for example, if the location of several selected points is shown by using the **Show location** button in the **TerraScan Main** window.

Setting:	Effect:
Hilite	Color and line weight for the rectangles drawn around selected points. Uses the active color table of MicroStation and MicroStation line weights.

Operation

Operation category defines actions performed when TerraScan is loaded.

Setting:	Effect:
Create Applications menu	If on, an **Applications** pulldown menu is added to the MicroStation menu. It contains items for opening TerraScan toolsets and its **Main window**.
Open Main window	If on, the TerraScan **Main window** is opened.
Open Main tool box	If on, the TerraScan **Main tool box** is opened.
Close AccuDraw	If on, the MicroStation *AccuDraw* is closed.
Maximum	Maximum amount of threads used for TerraScan processing. Normally, you should set this to the number of processor cores on your computer. The default setting is 2.

Point display

Point display category determines how points in TerraScan are drawn on the screen. It also defines whether coloring schemes for elevation and intensity coloring are fitted automatically to loaded points or not.

Setting:	Effect:
Draw as	Method of drawing points on the screen: • **Points -** points are drawn as true 3D elements on the screen. The display order within the point cloud and compared with vector elements in the design file depends on the true point and element coordinates. • **Raster -** points are drawn as raster pixels on the screen. The point cloud is always drawn in the background of vector elements in the design file.
Weight	Default size of points on the screen. Uses MicroStation line weights.
Speed	Default speed for point display: • **Fast - sparse points -** amount of displayed points depends on the viewing distance. The more you zoom out in a MicroStation view the less points are drawn. • **Normal - all points -** all points are drawn independently of the viewing distance. • **Slow -** all points are drawn slowly. Applies only if points are colored by flightline number.
Use depth	If on, a point is drawn if it is closest to the viewer compared with all other points falling in the same screen pixel. *Not MicroStation V8i*
Fit automatically	If on, color schemes for displaying points by elevation and intensity are fitted automatically to the corresponding values of points when they are loaded into TerraScan. If off, the software keeps the elevation and intensity values of the previous data set for the color schemes.

Rail section templates

Rail section templates category shows a list of rail section definitions that can be used for automatic rail vectorization using the *Find Rails* tool.

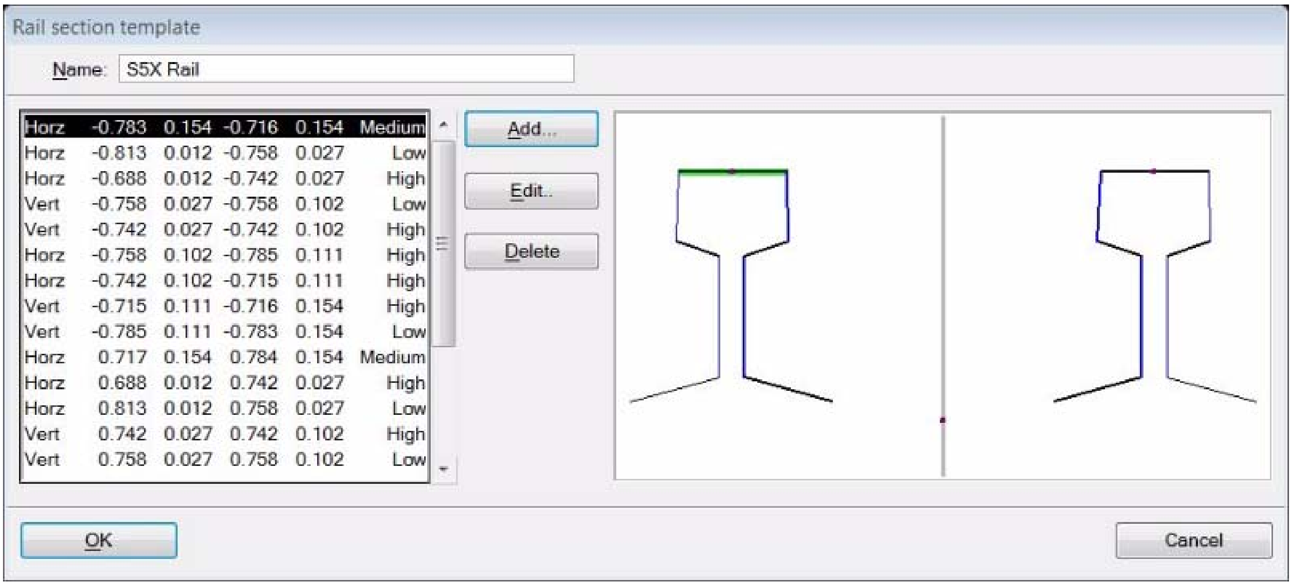
A drawing of the rail section can be used to define the geometry of a rail section template. Both rails of the rail section must be drawn into a design file using the correct measures. It is an advantage to draw them in a way that the center point of the rail section is at the design file origin (coordinates 0,0). As an alternative, you can also define a rail section by typing the start and end point coordinates of section lines in an input dialog.

You can **Add**, **Edit**, and **Delete** rail section templates by using the corresponding buttons in the **Settings** dialog. You can **Add**, **Edit**, and **Delete** parts of a rail section templates by using the corresponding buttons in the **Rail section template** dialog.

➢ **To add a new rail section:**

1. (Optional) Draw the rail section into a design file and select it.
2. Open the **Rail section templates** category.
3. Click **Add** in the **Settings** dialog.

   The **Rail section template** dialog opens:



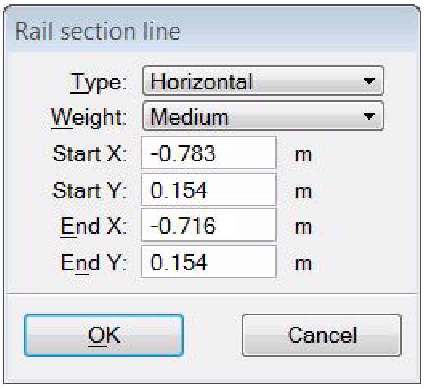
   If a section drawing has been selected, the section definition is shown in the dialog.

4. Type a **Name** for the rail section.
5. Click **Add** in the **Rail section template** dialog in order to add a new element to the section.

   OR

5. Select a line in the list of section elements and click **Edit** in the **Rail section template** dialog in order to edit an existing element of the section.

   The **Rail section line** dialog opens:



6. Define settings and click OK.
7. Add/Edit more elements of the section if necessary.
8. Click OK in the **Rail section template** dialog.
9. Close the **Settings** dialog in order to save the modified settings for TerraScan.

Setting:	Effect:
Type	Type of the rail section line: • **Horizontal** - horizontal line as part of the rail section. Used to find the Z location of a rail in automatic rail detection. • **Vertical** - vertical line as part of the rail section. Used to find the XY location of a rail in automatic rail detection. • **Void** - line that indicates a location without laser data close to rails. • **Alignment** - location of linear vector elements that are drawn in automatic rail vectorization.
Weight	Weight of a rail section line: **Low**, **Medium**, or **High**. A line with higher weight takes priority over lines with lower weights in automatic rail detection. This is only active if **Type** is not set to **Alignment**.
Start X | Y End X | Y	Start and end point coordinates of a rail section line. Given in the rail section’s coordinate system. The origin of the system (0,0) should be in the center of the rail section. This is only active if **Type** is not set to **Alignment**.
Position X | Y	Location of an alignment element. Given in rail section’s coordinates. This is only active if **Type** is set to **Alignment**.

✍ Rail section templates are stored in a configuration file RAIL_SECTIONS.INF in the TerraScan installation folder. You can copy this file to other computers in order to make rail section templates available on them.

Road section parameters

Road section parameters category defines level, color, text size, and unit settings for drawing road section parameter elements into the design file. The settings effect the display of elements that have been detected with the macro action **Compute section parameters** and drawn into the design file using the **Read / Section parameters** command.

Setting:	**Effect:**
Level	Design file level on which the different road section parameters are drawn.
Color	Color of the different road section parameters. Users the active color table of MicroStation.
Text size	Size of text elements that are drawn for the different road section parameters.
Unit	Unit for expressing the different road section parameter values. Slopes can be expressed in **Degree** or **Percentage**, other parameters can be expressed in **Master units** of the design file or in **Millimeters** on paper.

Scanner systems

Scanner systems category shows a list of scanner system configurations. A scanner system can include several scanners where each scanner has its own lever arm definition. The system definition also defines the misalignment between the IMU and the scanner system.

In addition, each scanner in a system can contain a link to a waveform profile. See **Scanner waveform profiles** for more information.

The scanner system number must be unique. It is used to establish a link between a scanner system and trajectory files.

You can **Add**, **Edit**, and **Delete** scanner systems by using the corresponding buttons in the **Settings** dialog. You can **Add**, **Edit**, and **Delete** scanners by using the corresponding buttons in the **Scanner system** dialog.

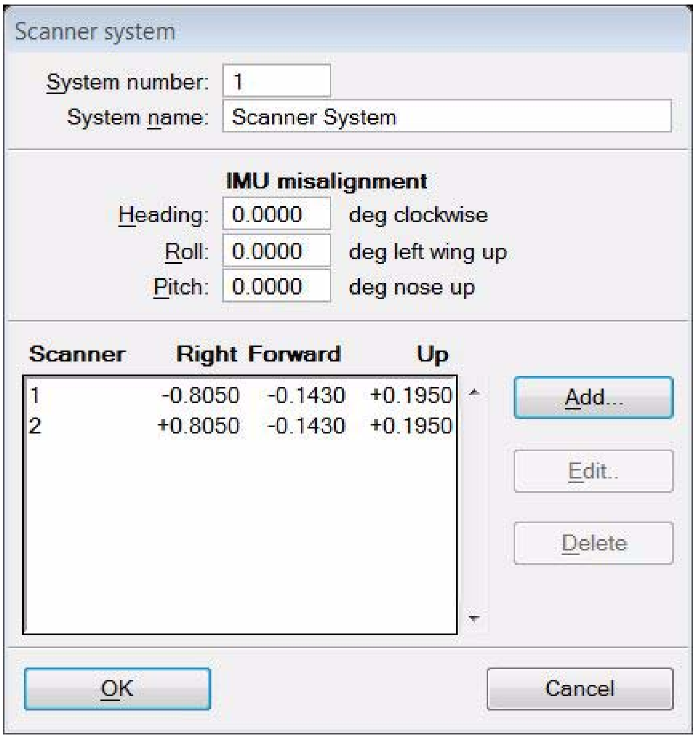
The lever arm of a scanner describes the distance between the IMU and the scanner. It is expressed as X, Y, and Z components of a vector. The direction of the three vector components is as follows:

- **X** - positive values to the right, negative to the left.
- **Y** - positive values forward, negative backward.
- **Z** - positive values up, negative down.

➢ **To add a new scanner system:**

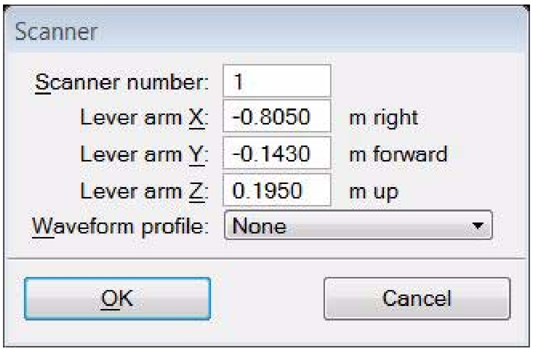
1. Open the **Scanner systems** category.
2. Click **Add** in the **Settings** dialog.

   The **Scanner system** dialog opens:



3. Type a **Scanner number** and a **Scanner name**.
4. If necessary, define values for the misalignment angles **Heading**, **Roll**, and **Pitch** between the scanner system and the IMU.
5. Click **Add** in the **Scanner system** dialog in order to add a new scanner.

   The **Scanner** dialog opens:



6. Define settings and click OK.
7. Add more scanners to the system if necessary.
8. Click OK in the **Scanner system** dialog.
9. Close the **Settings** dialog in order to save the modified settings for TerraScan.

Setting:	Effect:
Scanner number	Number of the scanner. Must be unique within a system.
Lever arm X | Y | Z	X, Y, and Z component of the lever arm vector between IMU and the scanner.
Waveform profile	Waveform profile linked to the scanner.

✍ Scanner systems are stored in a configuration file SCANNER_SYSTEMS.INF in the TerraScan installation folder. You can copy this file to other computers in order to make scanner system definitions available on them.

Scanner waveform profiles

Scanner waveform profiles category shows a list of scanner waveform profile definitions. Scanner waveform profiles provide the reference or standard waveform shape that is typical for a scanner. They are required for waveform processing tasks, such as the extraction of echo properties or the extraction of additional points.

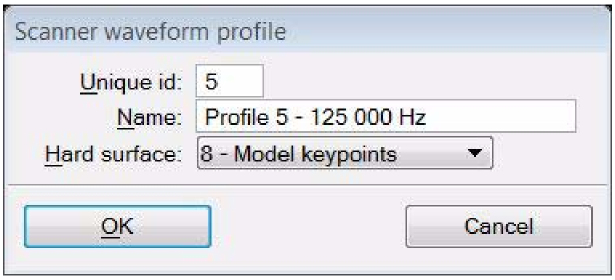
The waveform profile can be created from loaded laser points and trajectory information. The trajectory must include the link to the waveform file. The waveform profile can be best extracted from laser points on open ground, preferable hard surface, where a wider range of intensity values are represented. The sample points should not be too close to flightline edges.

You can **Add**, **Edit**, and **Delete** waveform profiles by using the corresponding buttons in the **Settings** dialog. You can also **Copy** a profile definition and view it in a text editor or **Paste** a profile definition from a text editor in TerraScan.

➢ **To create a new scanner waveform profile:**

1. Use *Manage Trajectories* tool and commands from the **Trajectories** dialog in order to import and manage trajectories, and to link the trajectories with the waveform files.
2. Load points into TerraScan.
3. Classify points in sample areas that are suited for creating the scanner waveform profile.
4. Open the **Scanner waveform profiles** category.
5. Click **Add** in the **Settings** dialog.

   The **Scanner waveform profile** dialog opens:



6. Define settings and click OK.

   The software extracts the waveform profile.

7. Close the **Settings** dialog in order to save the modified settings for TerraScan.

Setting:	Effect:
Unique id	Number of the waveform profile. Must be unique in the list of scanner waveform profiles.
Name	Name of the waveform profile.
Hard surface	Laser point class that the software uses to extract the waveform profile. Uses the active class definitions in TerraScan.

✍ Scanner waveform profiles are stored in a configuration file WAVEFORM_PROFILES.INF in the TerraScan installation folder. You can copy this file to other computers in order to make scanner waveform profiles available on them.

Section templates

Section templates category shows a list of cross section templates. Section templates may represent, for example, clearance areas or tunnel sections. They can be used for the classification of laser points with the **By section template** routine.

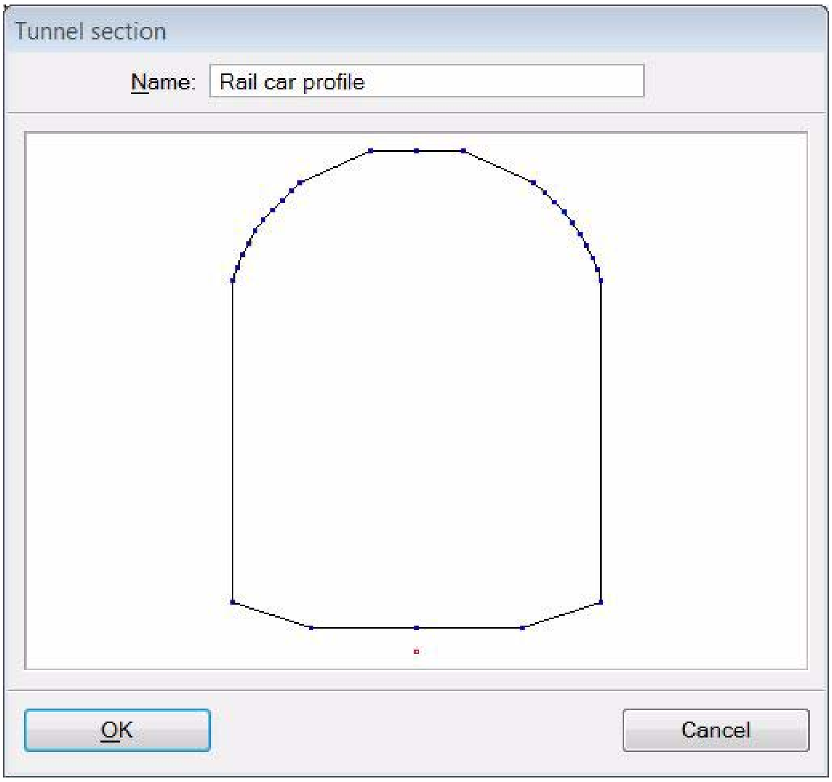
A section template is defined by the outline of the section and it's origin point. The outline must be drawn into a design file top view using correct measures. If the section is used in a classification step, it is aligned to an element by using the origin point.

You can **Add**, **Edit**, and **Delete** section templates by using the corresponding buttons in the **Settings** dialog.

➢ **To create a new section template:**

1. Draw the outline of the cross section as closed shape in a MicroStation top view.
2. (Optional) Draw a line with one end point at the position of the alignment point relative to the cross section shape.
3. Open the **Section templates** category.
4. Click **Add** in the **Settings** dialog.
5. Select the outline of the section with a data click.
6. Define the origin point of the section. If you created an element that represents the origin point, snap to the element in order to define the correct location.

   The **Tunnel section** dialog opens:



7. Type a **Name** for the section template.
8. Click OK in the **Tunnel sections** dialog.
9. Close the **Settings** dialog in order to save the modified settings for TerraScan.

✍ Section templates are stored in a configuration file SECTION_TEMPLATES.INF in the TerraScan installation folder. You can copy this file to other computers in order to make section templates available on them.

Target objects

Target objects category shows a list of target object definitions. Target objects are normally used for matching point clouds of terrestrial laser scanners. Supported shape primitives of target objects include ball, cone, and pyramid.

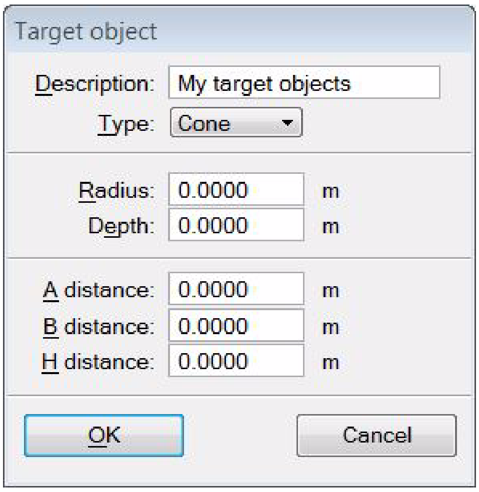
A target object may represent the location of a control point for which the coordinate values are known.

You can **Add**, **Edit**, and **Delete** target objects by using the corresponding buttons in the **Settings** dialog.

➢ **To add a new target object:**

1. Open the **Target objects** category.
2. Click **Add** in the **Settings** dialog.

   The **Target object** dialog opens:



3. Define settings and click OK.
4. Close the **Settings** dialog in order to save the modified settings for TerraScan.

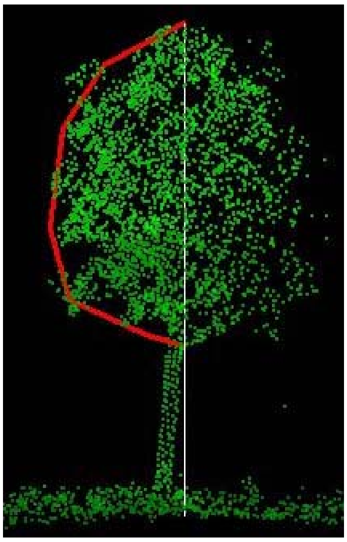
Setting:	Effect:
Description	Description of the target object.
Type	Shape primitives: **Ball**, **Cone**, or **Pyramid**.
Radius	Radius of the ball or cone.
Depth	Depth of a cone or pyramid.
Width	Width of a pyramid.
Height	Height of a pyramid.
A distance	Distance from the target object's center point to the known control point location along the xy line from scanner to object.
B distance	Distance from the target object's center point to the known control point location perpendicular to xy line from scanner to object.
H distance	Elevation difference from the target object's center point to known control point location.

✍ Target objects are stored in a configuration file TARGETS.INF in the TerraScan installation folder. You can copy this file to other computers in order to make target object definitions available on them.

Tree types

Tree types category shows a list of tree types. A tree type is defined by the shape of the tree crown cross section and additional parameters. The tree type definitions are used in automatic tree detection from laser points with the **Detect trees** command.

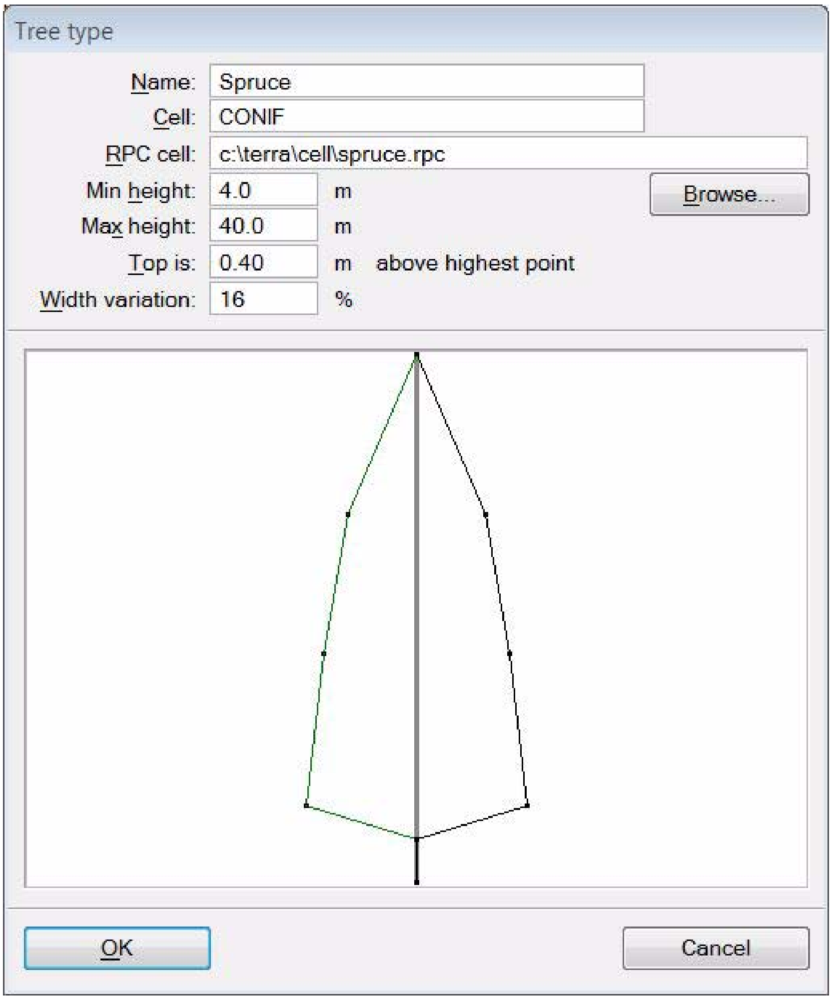
You can **Add**, **Edit**, and **Delete** tree typed by using the corresponding buttons in the **Settings** dialog.



➢ **To create a new tree type:**

1. Draw an outline of one half of a tree crown cross section as a line string in a MicroStation front view. You may use a cross section of a tree in laser points as background for digitizing the tree crown shape. The image on the right shows the approximate tree section centerline in white (used as helping line) and the outline in red.
2. Select the outline.
3. Open the **Tree types** category.
4. Click **Add** in the **Settings** dialog.

   The **Tree type** dialog opens:



5. Define settings and click OK.
6. Close the **Settings** dialog in order to save the modified settings for TerraScan.

Setting:	Effect:
Name	Name of the tree type.
Cell	Name of a MicroStation cell that is drawn into the design file when trees are detected.
RPC cell	Name and location of a RPC cell file that is used in rendered views to replace detected trees.
Min height	Minimum height of a tree.
Max height	Maximum height of a tree.
Top is	Distance between highest hit on the tree in laser points and the tree top of the tree cells.
Width variation	Variation of tree crown width for the tree type. Given in percent.

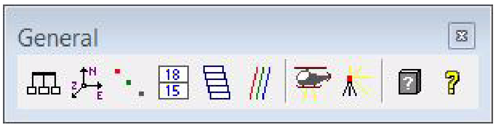
✍ Tree types are stored in a configuration file TREE_TYPES.INF in the TerraScan installation folder. You can copy this file to other computers in order to make tree type definitions available on them.

Undo buffer

Undo buffer category defines how much memory the application allocates for its undo buffer. The setting effects the amount of steps that can be undone with **Undo** or **From list** commands. The recommended value range is 16 - 64 MB.

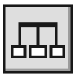
General tool box

The tools in the **General** tool box are used to define user settings, to define point classes, to define project blocks, to manage trajectories, to load points and to access license information and the users' guide document.

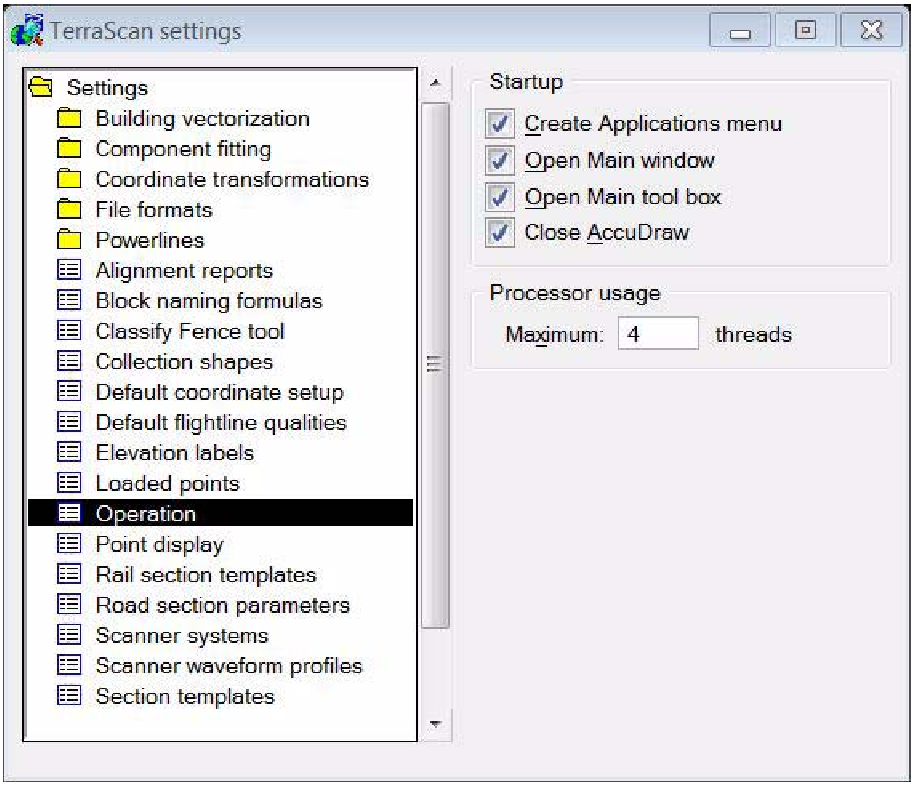


To:	Use:
Change user settings	*Settings*
Define coordinate range and resolution	*Define Coordinate Setup*
Define point classes and drawing symbology	*Define Classes*
Design project block boundaries	*Design Block Boundaries*
Define project and data blocks	*Define Project*
Manage trajectory information	*Manage Trajectories*
Load points from airborne / mobile scanning	*Load Airborne Points*
Load points from static terrestrial scanning	*Load Ground Points*
Show about TerraScan and license information	*About TerraScan*
Open online help	*Help on TerraScan*

Settings



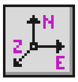
Settings tool lets you change a number of settings that control the way how TerraScan works. Selecting this tool opens the **TerraScan settings** dialog:



The settings are grouped into logical categories. Selecting a category in the list displays the appropriate controls next to the category list.

The different categories and related settings are described in detail in Section **TerraScan Settings**.

Define Coordinate Setup



Define Coordinate Setup tool sets up coordinate system values that a Terra Application uses for laser points and images. It determines the coordinate range inside which all data must be located and the resolution to which coordinate values are rounded. The coordinate setup is stored into the active design file and is used by all Terra Applications.

Terra Applications use signed 32 bit integer values for storing coordinates of laser points and images. This has the advantage of using only 12 bytes of memory for the coordinate information of each point. You can control how accurately coordinate values are stored by defining how big each integer step is.

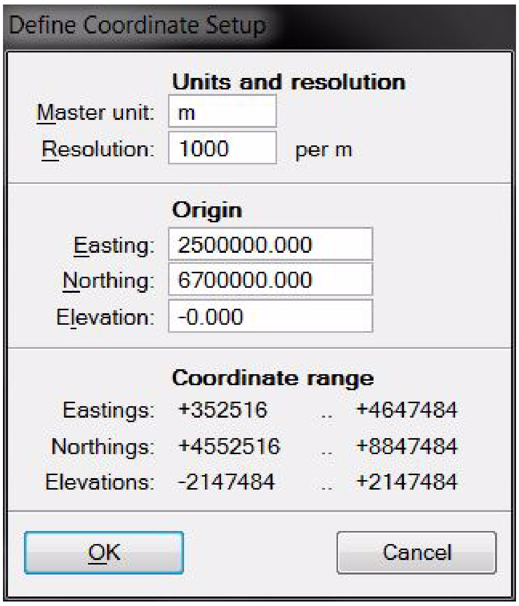
If, for example, one integer step is equal to one millimeter, all coordinate values are rounded to the closest millimeter. At the same time it would impose a limitation on how far apart points can be or how big the coordinate ranges are. Millimeter steps produce a coordinate cube which has a size of 2^{32} millimeters or 4294967.296 meters. If the origin of the coordinate system is at [0.0, 0.0, 0.0], the coordinate ranges are limited to values between -2147483 and +2147483. If necessary, you can fit the coordinate ranges to your data by modifying the Easting and Northing coordinates of the coordinate system origin.

If one integer step is equal to one centimeter, the coordinate values can range from -21 million to +21 million which is large enough for most coordinate systems.

➢ **To define the coordinate setup:**

1. Select the *Define Coordinate Setup* tool.

   This opens the **Define Coordinate Setup** dialog:



2. Define settings and click OK.

   This modifies the coordinate system values used by all Terra Applications in the active design file.

MicroStation SE and MicroStation J

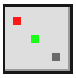
Each design file contains a definition of a 32 bit integer coordinate system which MicroStation uses internally for vector elements. All applications share the same coordinate setup with MicroStation. When you change the coordinate setup with *Define Coordinate Setup* tool, it changes the design file coordinate system.

✍ Since Terra Applications' version 009.00x these MicroStation versions are no longer supported.

MicroStation V8 and V8i

MicroStation V8 uses 64 bit values for storing vector elements. Terra Applications use a coordinate setup which is separate from the design file coordinate system. Their default coordinate setup defines 100 integer steps for each master unit. You can use *Define Coordinate Setup* tool to change the coordinate setup which the application stores in the design file but it does not affect MicroStation itself or the vector elements.

Define Classes



Define Classes tool opens a dialog for managing point classes and related drawing rules.

A point class definition includes descriptive information for the class, such as a unique number, code, and description, and rules for displaying the point on the screen or drawing it into the design file.

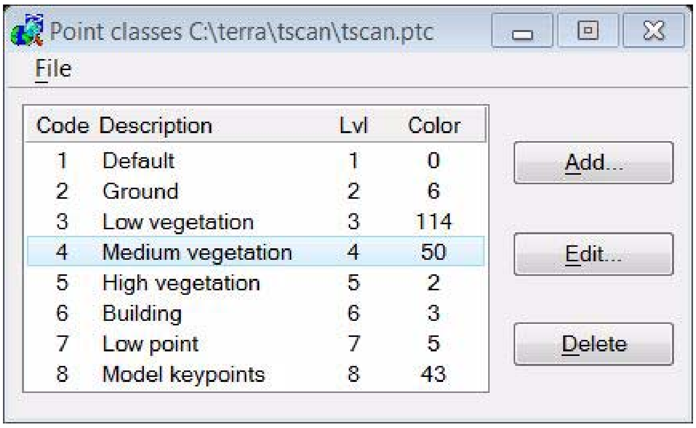
TerraScan provides a default class definition file TSCAN.PTC which is stored in the TerraScan installation folder. You can **Add**, **Edit**, and **Delete** point classes by using the corresponding buttons in the **Point classes** dialog.

You can create a **New** empty class definition file, **Open** an existing file, **Save** changes to an existing file, and **Save** class definitions **as** a new file by using the corresponding commands from the **File** pulldown menu of the **Point classes** dialog. The class definitions are saved into files with the default extension .PTC. Usually, there are different class definition files for different project types. If TerraScan is loaded, the last-used class definition file is still active.

➢ **To add a new class to the active class list:**

1. Select the *Define Classes* tool.

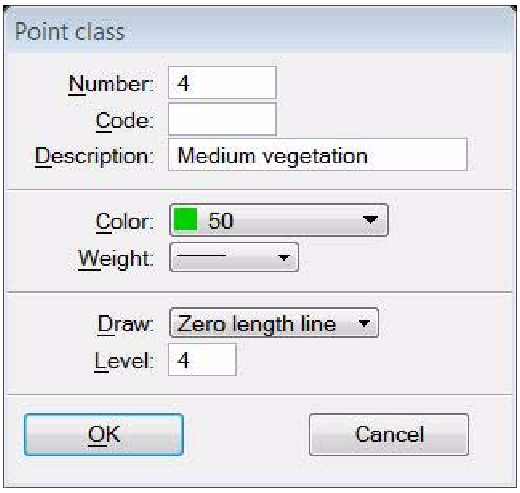
   The **Point classes** dialog opens:



   The dialog contains a list of all point classes in the active class list.

2. Click the **Add** button.

   The **Point class** dialog box opens:



3. Define settings and click OK.

   The class is added to the active class list.

4. Select **Save** or **Save as** from the **File** pulldown menu in order to save the class definitions

into a file.

Setting:	Effect:
Number	Unique number of the point class.
Code	Code of the point class. The code is a text string which can include any kind of characters.
Description	Descriptive name of the class.
Color	Color for displaying points of the class. Uses the active color table of MicroStation.
Weight	Size for displaying points of the class. Uses MicroStation line weights.
Draw	Element type used for drawing points of the class permanently into the design file.
Level	Level on which point of the class are drawn permanently into the design file.
Character	Character used for drawing points permanently into the design file. This is only active if **Draw** is set to **Character**.
Font	Font type used for drawing points permanently into the design file. Uses MicroStation font types. This is only active if **Draw** is set to **Character** or **Elevation text**.
Size	Size of the text used for drawing points permanently into the design file. This is only active if **Draw** is set to **Character** or **Elevation text**.
Justify	Justification of the text relative to the original point. Uses MicroStation justification options for text elements. This is only active if **Draw** is set to **Elevation text**.
Dx	Offset in X direction between the original point and the origin point of the text element. This is only active if **Draw** is set to **Elevation text**.
Dy	Offset in Y direction between the original point and the origin point of the text element. This is only active if **Draw** is set to **Elevation text**.
Diameter	Diameter of a circle used for drawing points permanently into the design file. This is only active if **Draw** is set to **Circle**.
Style	Style of the outline of a circle for drawing points permanently into the design file. Uses MicroStation line styles. This is only active if **Draw** is set to **Circle**.

✍ Since point class definitions are stored in a text file format, you can edit the file in a text editor as well. This may be useful if you need to copy classes and drawing rules from one class list to another.

Design Block Boundaries

Design Block Boundaries tool creates shape elements that can be used as block boundaries for a TerraScan project. The block boundary creation can start from line or shape elements. If points are loaded in TerraScan, the tool can also compute the amount of points inside each block boundary.

The line elements used as starting elements for the tool should cross each other in order to create a closed line work for the shape creation. If shape elements are used as starting elements, the tool does not create new shapes. It only computes the amount of points inside the existing shapes.

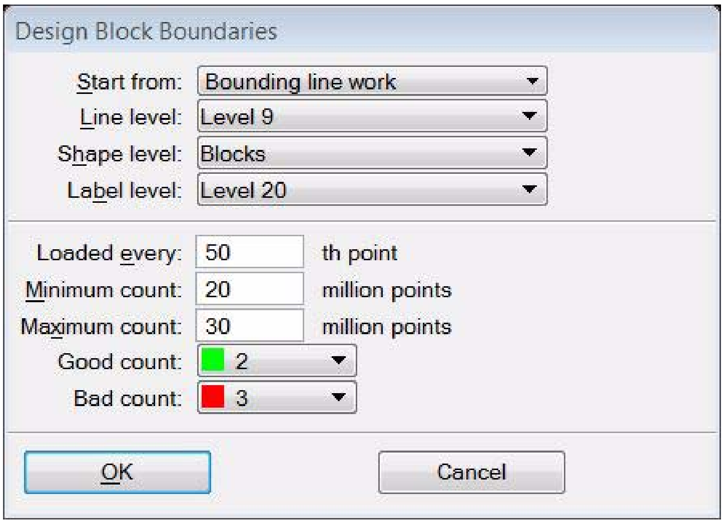
The amount of points inside each block area is shown by text elements, which are drawn into the design file. The color of the label indicates whether the amount of points inside a block is within a given range. The point count is given in values rounded to million points. If no points are loaded in TerraScan, the tool ignores settings related to labels and point counts.

The tool supports significantly the creation of block boundaries for TerraScan projects, especially if the point density is varying in the project area. This is often the case in mobile mapping projects if the driving speed is not constant.

➢ **To design block boundaries:**

1. (Optional) Load points from the whole project area into TerraScan using **Read points** command or *Load Airborne Points* tool. Load only a subset of points if the project area is too big to load all points.
2. Use MicroStation or TerraScan tools to digitize line elements around your project area and to separate the project area into smaller parts.
3. Select the *Design Block Boundaries* tool.

   This opens the **Design Block Boundaries** dialog:

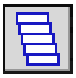


4. Define settings and click OK.

   This creates shapes on the given **Shape level**. If points are loaded in TerraScan, the tool creates text elements on the given **Label level**.
5. If the amount of points per block is not within the given limits, modify the line work. Run the *Design Block Boundaries* tool again in order to update the shapes and labels. Continue until the point counts are within the limits.
6. Continue with the *Define Project* tool in order to add the shapes as block boundaries to a project.

Setting:	Effect:
Start from	Elements that are used to create block boundaries: • **Bounding line work -** line elements that form closed areas. • **Shapes already drawn -** already existing shape elements.
Line level	Design file level on which the line elements are drawn. This is only active if **Start from** is set to **Bounding line work**.
Shape level	Design file level on which the shape elements are drawn. The shapes are created if **Start from** is set to **Bounding line work**.
Label level	Design file level on which text elements are drawn. The texts show the amount of points inside a shape area if points are loaded in TerraScan.
Load every	Indicates the subset of points that is loaded into TerraScan. This is ignored if no points are loaded.
Minimum count	Minimum amount of points accepted in one project block. Rounded to million points.
Maximum count	Maximum amount of points accepted in one project block. Rounded to million points.
Good count	Display color of a label for shape areas, where the amount of points is within the given **Minimum** and **Maximum count** values. Uses the active color table of MicroStation.
Bad count	Display color of a label for shape areas, where the amount of points is outside the given **Minimum** and **Maximum count** values. Uses the active color table of MicroStation.

Define Project



Define Project tool opens the TerraScan **Project** window. The window displays the active project and contains menu commands for handling TerraScan projects.

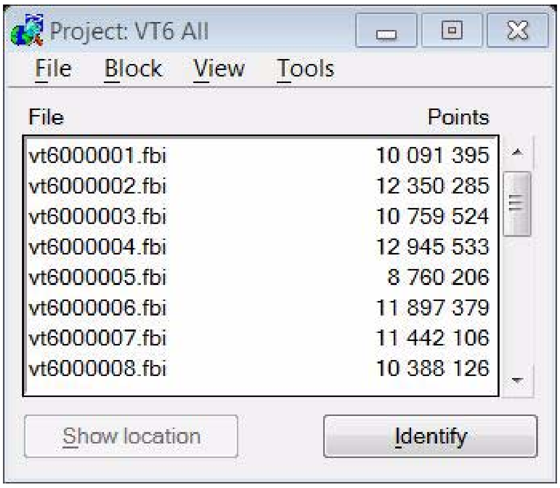
The main benefits of organizing point cloud data in a project are:

- The project definition divides a large data set into smaller parts which are easy to manage. Each part should contain an amount of points that can be loaded into memory and still allow processing of the points.
- When points are imported into a project, the application automatically divides the large point cloud into geographical regions (called ‘blocks’ in TerraScan’s terminology). This is required because raw laser data is often provided in flightline order while some classification routines and other processing steps rely on geographical regions.
- You can run macros that process the data of all or selected blocks of a project. You can also start other processing routines from the **Project** window. This is essential for the automated processing of large point cloud data sets.
- TerraScan projects can be directly used in TerraStereo, Terrasolid’s software for advanced visualization of huge point clouds in mono and stereo mode.

➢ **To view the active project:**

1. Select the *Define Project* tool.

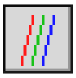
   The **Project** window opens:



If there is an active project in TerraScan, the title bar of the window displays the name of the project. Further, the window shows the list of blocks that belong to the project. For each block, the name and the amount of points in the file are displayed.

The menu commands of the **Project** window are described in detail in Chapter **Working with Projects** on page 317.

Manage Trajectories



Manage Trajectories tool opens the TerraScan **Trajectories** window. The window displays the active trajectories and contains menu commands for handling trajectory information in TerraScan.

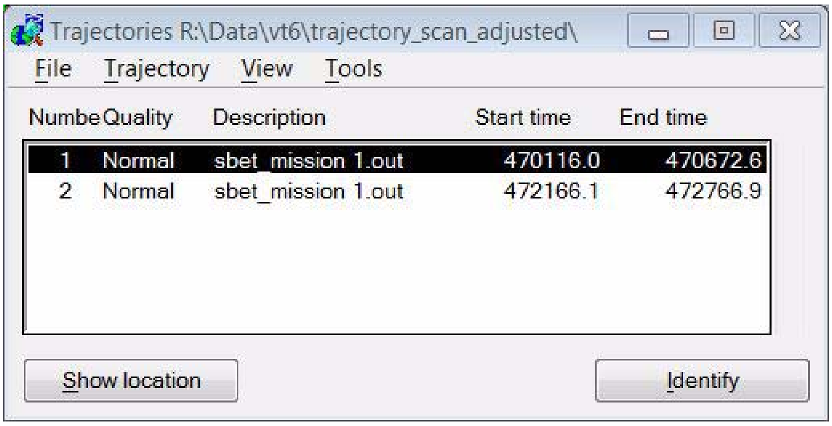
Trajectory information is required by the following processing steps:

- **Cut overlap** menu command for identifying points from overlapping flightlines.
- **Adjust laser angles** menu command for applying heading, roll, and pitch corrections to laser data.
- TerraMatch tools for fixing mismatch in laser data.

➢ **To view information about active trajectories:**

1. Select the *Manage Trajectories* tool.

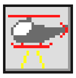
   This opens the **Trajectories** window:



If there are active trajectories in TerraScan, the title bar of the window displays the active trajectory folder. Further, the window shows the list of trajectory files that are stored in the active trajectory folder.

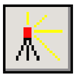
The menu commands of the **Trajectory** window are described in detail in Chapter **Manage Trajectories** on page 356.

Load Airborne Points



Load Airborne Points tool performs exactly the same action as the **Read points** command in the **File** pulldown menu of the **TerraScan Main window**.

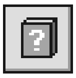
Load Ground Points



Load Ground Points tool is used to load laser points from a static ground-based scanner into TerraScan. The tool works in the same way as the *Load Airborne Points* tool and the **Read points** command.

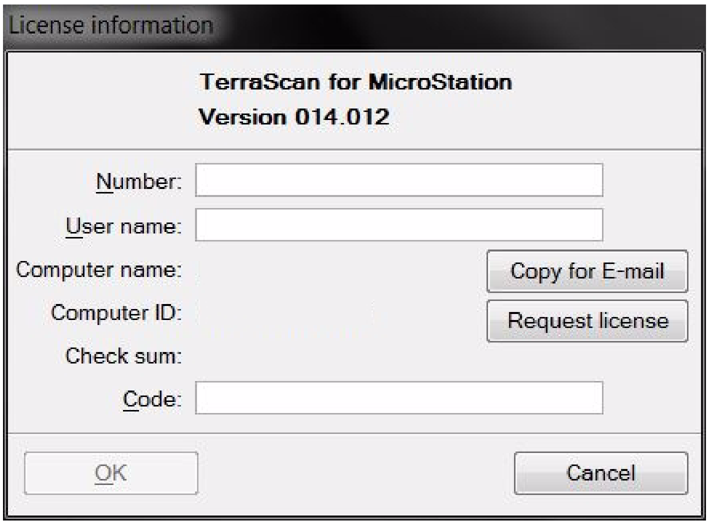
The important difference is that the **Measurement** pulldown menu replaces the **Flightline** pulldown menu in the **TerraScan Main window**. The pulldown menu contains commands tailored for processing data of static ground-based laser scanners.

About TerraScan



About TerraScan tool opens a dialog which shows information about TerraScan and about the license.

From this dialog, you can open the **License information** dialog:



Use the **Request license** button to start the online registration for node-locked licenses.

More information about license registration is available on the Terrasolid web pages: www.terrasolid.com/registration.php.

Help on TerraScan

Help on TerraScan tool launches Acrobat Reader for accessing this User's Guide in PDF format. The PDF must be stored in the /DOCS folder of your Terra Software installation directory.

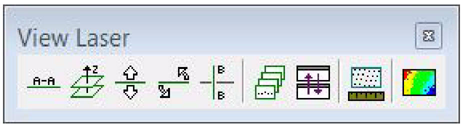
If you installed TerraScan in the default directory C:\TERRA, the User's Guide must be stored in C:\TERRA\DOCS\TSCAN.PDF.

The PDF has hypertext links built in, so you can jump between topics by clicking on the topic names highlighted in green color.

Accessing the PDF also requires that you have the Acrobat Reader installed on your computer. The software looks for a file named ACRORD32.EXE. If the file can not be found, you are asked to locate the file on the hard disk manually.

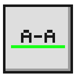
View Laser tool box

The tools in the **View Laser** tool box are used to create and modify section views, create animations along a path, measure point density, and to update distance coloring.



To:	Use:
Rotate view to show vertical cross section	*Draw Vertical Section*
Rotate view to show horizontal cross section	*Draw Horizontal Section*
Move forward or backward in section view	*Move Section*
Rotate a section view around its center	*Rotate Section*
Cut perpendicular section from section view	*Cut Section*
Travel along path and display sections	*Travel Path*
Define automatic synchronization of views	*Synchronize Views*
Measure point density	*Measure Point Density*
Recompute distance colors and update views	*Update Distance Coloring*

Draw Vertical Section



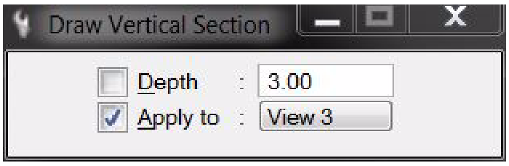
Draw Vertical Section tool creates a 3D section view from a location defined by a center line of the section and its depth.

A vertical section view is simply a rotated MicroStation view which displays all visible design file elements and laser points inside the given slice of space. This makes it well-suited for viewing laser points and for placing 3D vector elements.

➢ **To create a vertical section view:**

1. Select the *Draw Vertical Section* tool.

   The **Draw Vertical Section** dialog opens:

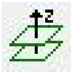


2. Define the start or left point of the section center line with a data click in a top view.
3. Define the end or right point of the section center line with a data click in a top view.
4. Define the section view depth with a data click in a top view or by typing a value in the **Depth** field of the **Draw Vertical Section** dialog.
5. If **Apply to** is not switched on in the **Draw Vertical Section** dialog, select the view for displaying the section with a data click inside this view.

   The selected view is rotated to show the vertical section. The application automatically computes the required elevation range so that all laser points inside the given section space are displayed.

Setting:	**Effect:**
Depth	Display depth of a section on both sides of the center line. If on, the depth is fixed to the given value.
Apply to	If on, the section is automatically displayed in the selected view.

Draw Horizontal Section



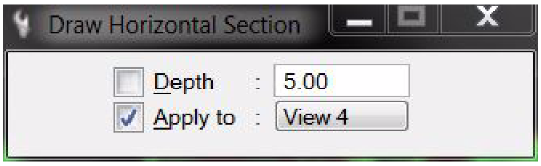
Draw Horizontal Section tool creates a top view which shows laser data and vector elements in a limited elevation range.

Horizontal sections are useful, for example, to display the exact XY location of vertical objects, such as building walls or poles in MLS data sets. You should open at least one top view and one section view before starting to create horizontal section views.

➢ **To create a horizontal section view:**

1. Use the *Draw Vertical Section* tool in order to create a vertical section view.
2. Select the *Draw Horizontal Section* tool.

   The **Draw Horizontal Section** dialog opens:

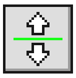
   

3. Define the center elevation of the horizontal section with a data click in the vertical section view.
4. Define the top view depth (= visible elevation range) with a data click in the vertical section view or by typing a value in the **Depth** field of the **Draw Horizontal Section** dialog.
5. If **Apply to** is not switched on in the **Draw Horizontal Section** dialog, identify the view for displaying the top view with a data click.

   The selected view is rotated to a top view and displays the defined elevation range.

Setting:	**Effect:**
Depth	Display depth or visible elevation range of a horizontal section view up and down from the center elevation. If on, the display depth is fixed to the given value.
Apply to	If on, the section is automatically displayed in the selected view.

Move Section



Move Section tool lets you move stepwise forward or backward in section views. The tool is most useful in views created by *Draw Vertical Section* and *Draw Horizontal Section* tools.

➢ **To move sections forward or backward:**

1. Start *Move Section* tool.

   This opens the **Move Section** dialog:

2. Move the mouse pointer into a section view.

   The area covered by a vertical section is highlighted by a rectangle in all top views.

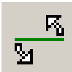
3. Place a data click in order to move the section forward.

   OR

3. Place a reset click in order to move the section backward.

Setting:	Effect:
Move by	Step size: • **Half of view depth -** the section is moved by half of the section’s depth. If the section depth is 1 m, the section is moved 0.5 m with each mouse click. • **Full view depth -** the section is moved by its full depth. If the section depth is 1 m, the section is moved 1 m with each mouse click.

Rotate Section



Rotate Section tool rotates a vertical section view stepwise around its center point.

The direction and angle of rotation can be determined by data clicks inside the section view or by a fixed value in the tool's dialog.

➢ **To rotate a section view:**

1. Select the *Rotate Section* tool.

   This opens the **Rotate Section** dialog:

2. Move the mouse pointer into a section view.

   The area covered by a vertical section is highlighted by a rectangle in all top views.

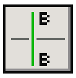
3. Place a data click inside a vertical section view.

   If the data click is placed on the right side of the section view's center, the section is rotated counterclockwise. If the data click is placed on the left side, the rotation direction is clockwise.

   The angle of rotation is determined by the distance of a data click from the center of the section view or by a fixed value in the **Angle** field of the **Rotate Section** dialog.

Setting:	Effect:
Angle	Rotation angle applied to a view with each data click. If on, the rotation is fixed to the given value. Positive values rotate in counterclockwise direction, negative values in clockwise direction.

Cut Section



Cut Section tool creates a vertical section view which is perpendicular to another vertical section view. In addition to just rotating the section by 90 degree, the cut section tool allows you to define another depth for the new section view.

➢ **To cut a perpendicular section view:**

1. Create a vertical section view using the *Draw Vertical Section* tool.
2. Select the *Cut Section* tool.

   This opens the **Cut Section** dialog:

3. Define the position of the new section's center line with a data click in the section view.

   The center line of the new section is defined by the given position perpendicular to the center line direction of the source section.

4. Define the section view depth by placing a data click or by typing a value in the **Depth** field of the **Cut Section** dialog.
5. Identify a view for displaying the new section with a data click inside the view.

   The selected view is rotated to show the new section.

Setting:	Effect:
Depth	Display depth of a section on both sides of the center line. If on, the depth is fixed to the given value.

Travel Path

Travel Path tool lets you view an animation along an alignment element. The tool provides an excellent way for traversing along the survey path and checking the data visually.

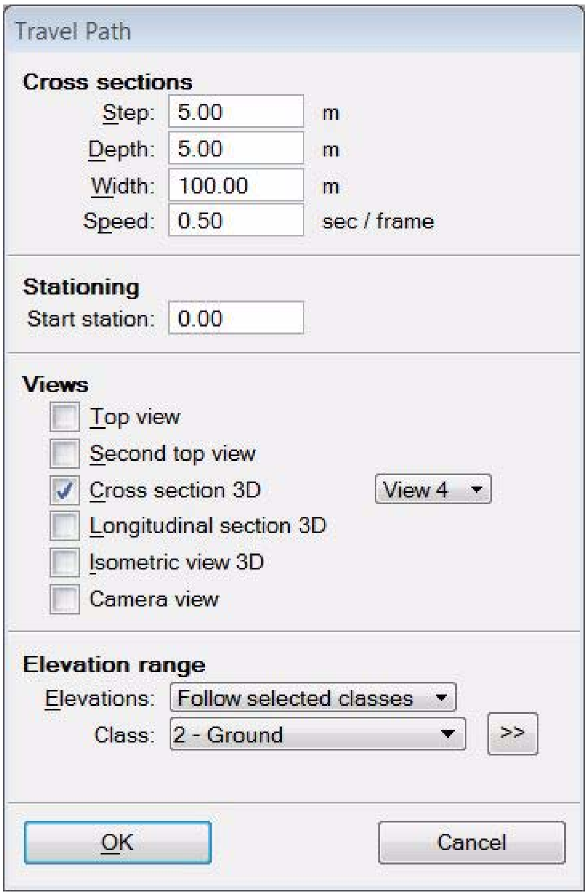
The alignment element can be any linear element. In most cases you create the element manually or draw, for example, a trajectory line into the design file by using the **Draw into design** command. Alternatively, you can use TerraScan's **Draw from points** command which draws an approximate flight path deduced from the order of loaded laser points.

You can define what kind of views you want to see while traveling along the alignment. Supported view types include top, cross section, longitudinal section, and isometric views.

➢ **To create an animation along an alignment:**

1. Draw and select the alignment element.
2. Select the *Travel Path* tool.

   This opens the **Travel Path** dialog:



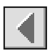
3. Define settings and click OK.

   The application constructs internal tables for the animation and then opens the **Travel Player** dialog.

Setting:	Effect:
Step	Step along alignment between consecutive cross sections.
Depth	Full depth of each cross section. Each cross section covers a rectangular area defined by the **Depth** and **Width** values.
Width	Full width of each cross section. Each cross section covers a rectangular area defined by the **Depth** and **Width** values.
Speed	Speed for automatic animation display.

Setting:	Effect:
Start station	Defines the starting point on the alignment element from which the animation starts.
Views	MicroStation views that are used for displaying the animation: • **Top view -** displays data from the top. • **Second top view -** displays data from the top. • **Cross section 3D -** displays data in a cross section. • **Longitudinal section 3D -** displays data in a longitudinal section. • **Isometric view 3D -** displays data in an isometric view.
Elevations	Method of elevation range computation which defines how the animation follows elevation changes in the data: • **Follow all points -** all points determine the visible elevation range. • **Follow selected classes -** points from selected classes determine the visible elevation range. Select a single class from the **Class** list. Click on the >> button in order to open the list of active classes and select several classes. • **Follow 3D alignment -** the alignment element determines the center elevation and the visible elevation range is determined by the **Minimum dz** and **Maximum dz** values given relative to the alignment element. • **Fixed -** a fixed elevation range defined by **Minimum z** and **Maximum z** values is used for the whole animation.

The **Travel Player** dialog contains the following commands and tools for traveling along the alignment element:

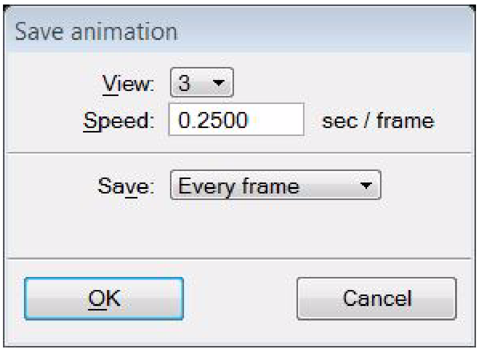
Menu/ Tool:	Command/Tool name:	Effect:
Move	*Using mouse*	Update a cross section view dynamically as you move the mouse pointer along the alignment. If you place a data click, the tool updates all views.
Move	*To start*	Move to the start of the alignment and update all views.
Move	*To end*	Move to the end of the alignment and update all views.
	*Play backward*	Start automatic animation display backward along the alignment.
	*Step backward*	Move stepwise backward along the alignment.
	*Stop*	Stop automatic animation display.
	*Step forward*	Move stepwise forward along the alignment.
	*Play forward*	Start automatic animation display forward along the alignment.

Additionally, an animation can be saved as .AVI file using the **Save animation** command from the **File** pulldown menu.

➢ **To save an animation:**

1. Select **Save animation** command from **File** menu in the **Travel Player**.

This opens the **Save animation** dialog:



2. Define settings and click OK.

   This opens the **Output AVI file** dialog, a standard dialog for saving a file.

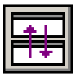
3. Define a location and file name and click **Save**.

   This starts the recording of the animation and saves an avi file.

Setting:	Effect:
View	View number from which the animation is recorded. The view must be open to be recorded.
Speed	Speed of the resulting animation in seconds per frame.
Save	Defines how frames are saved to the avi file: • **Every frame -** every frame is saved. • **Every nth frame -** only every nth frame is saved where n is defined in the **Step** field.

✍ The *Travel Path* tool is able to save very simple animations along an alignment element. It is most useful for saving cross sections in an animation but it has some drawbacks in saving views with a larger number of points, such as isometric views. For the production of more advanced animations, see the *Create Flythru Movie* tool in TerraPhoto.

Synchronize Views



Synchronize Views tool defines dependencies between Microstation views. The display in a dependent view is automatically updated, if the master view display changes. This is useful if you want to view the same location using two different types of content. For example, you may want to see an orthophoto and laser points side by side in two different top views.

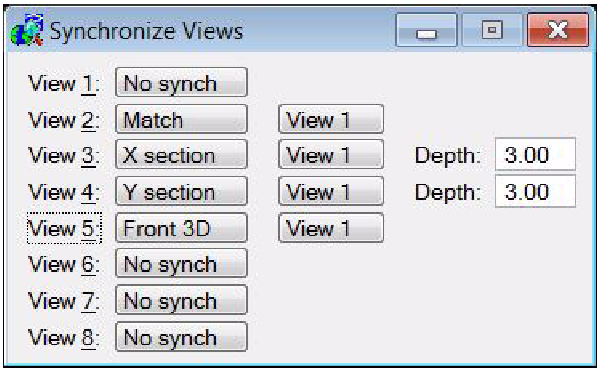
Synchronize Views tool can define the following dependencies:

- **No synch -** view works normally and does not depend on other views. This is the default setting.
- **Match -** dependent view shows the same area using the same rotation as the master view.
- **X section -** dependent view shows a cross section along the screen X axis of the master view.
- **Y section -** dependent view shows a cross section along the screen Y axis of the master view.
- **Front 3D -** dependent view is a 45 degree oblique view looking forward and down to the area displayed by the master view.
- **Side 3D -** dependent view is a 45 degree oblique view looking right and down to the area displayed by the master view.

➢ **To set up synchronized views:**

1. Select the *Synchronize Views* tool.

   This opens the **Synchronize Views** dialog:



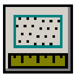
2. Define the dependencies by selecting a dependency type from the lists.
3. Define the master view for each dependency.
4. If required, define additional settings.

   Whenever you move, pan, zoom, or redraw a master view, the dependent views are updated automatically.

Setting:	Effect:
Depth	Depth of a section view.

✍ Synchronization stays active if the **Synchronize Views** dialog is closed. If you want to release the view dependencies and stop synchronization, reopen the dialog and set all views to **No synch**.

Measure Point Density



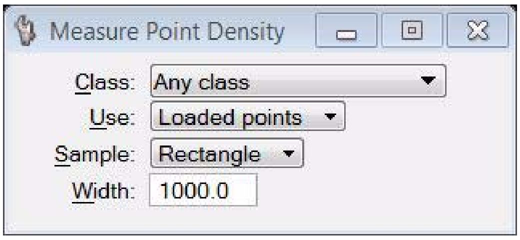
Measure Point Density tool displays the average number points per squared master unit. You can measure the point density in a rectangular or circular area or from the whole data set. The measurement can be based on points loaded into TerraScan or points residing in the active project.

The point density values are displayed in the information bar at the bottom of the MicroStation interface. The values include the amount of points per sample area and the average point density.

➢ **To measure the point density from all loaded points:**

1. Select the *Measure Point Density* tool.

   This opens the **Measure Point Density** dialog:



2. Define settings.
3. If **Sample** is set to **All points**, place a data click anywhere in a MicroStation view (**Loaded points**) or inside the project area (**Project points**).

   This displays the average point density of loaded points or points in the project.

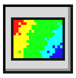
   OR

3. If **Sample** is set to **Rectangle** or **Circle**, define the center point of the sample area with a data click.

   This displays the average point density inside the sample area.

Setting:	Effect:
Class	The point density is computed for points of any class or of a specific class. The list contains the active classe definitions in TerraScan.
Use	Defines which points are used for the density measurement: • **Loaded points -** density is measured from points loaded in TerraScan. • **Project points -** density is measured from binary files referenced by the active project in TerraScan.
Sample	Sample area for the density measurement: • **All points -** area covered by all points. • **Rectangle -** rectangular area. • **Circle -** circular area.
Width	Defines the width of a **Rectangle** or the diameter of a **Circle** depending on the setting in the **Sample** field. Given in master units of the design file.

Update Distance Coloring



Update Distance Coloring tool recomputes distances of laser points to other laser points, surfaces, or design file elements, and updates views in which distance coloring is active.

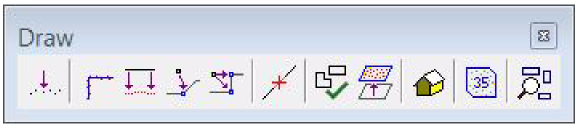
You need to use this tool only if distance coloring is active and:

- you have classified points to or from classes involved in distance computation.
- you have transformed point xy coordinates or elevations.
- you have modified design file elements involved in distance computation.

For more information about distance coloring, see Section **Color by Distance** on page 266.

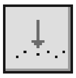
Draw tool box

The tools in the **Draw** tool box are used to fit or adjust vector elements to point cloud data. In addition, there are tools for validating vector elements.



To:	Use:
Adjust mouse clicks to laser point coordinates	*Mouse Point Adjustment*
Fit linear element by laser points	*Fit Linear Element*
Drape linear element to laser surface	*Drape Linear Element*
Find breakline running parallel to an element	*Find Breakline Along Element*
Find curb stone running parallel to an element	*Find Curb Along Element*
Cut linear element with other features closeby	*Cut Linear Element*
Compare footprint polygons and classified roof hits	*Check Footprint Polygons*
Adjust shape element to laser elevation	*Set Polygon Elevation*
Vectorize a building from hits on planar roof surfaces	*Construct Planar Building*
Place shape around a group of laser points	*Place Collection Shape*
Inspect elements one at a time	*Inspect Elements*

Mouse Point Adjustment



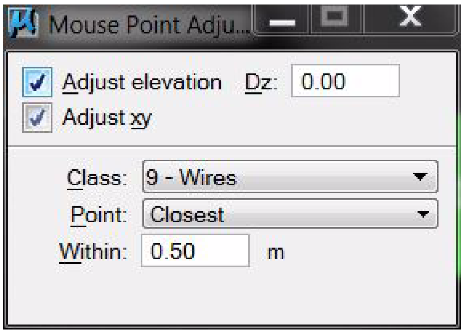
Mouse Point Adjustment tool adjusts vector data to the location and/or the elevation of laser points. You can use this tool with any MicroStation element placement tool. *Mouse Point Adjustment* simply fixes the coordinates of data clicks, which means the vertices of vector elements to laser data coordinates. It might be considered as a “snap to laser points” tool.

You can choose whether the elevation and/or the xy location of the vertices are adjusted. If you want to digitize a linear object, such as a wire, you probably want to adjust both, the xy and the elevation of the vertices to the laser points. On the other hand, if you want to place an object on the ground, you probably want to adjust only the elevation of the vertices.

➢ **To place elements adjusted to laser points:**

1. Select the *Mouse Point Adjustment* tool.

   This opens the **Mouse Point Adjustment** dialog:



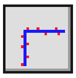
2. Define setting.
3. Start the drawing tool that you want to use and digitize elements.

   As long as the **Mouse Point Adjustment** dialog is open and any adjustment option is switched on, all vertices of elements are adjusted according to the settings.

Setting:	Effect:
Adjust elevation	If on, the elevation of data clicks (vertices) is adjusted.
Dz	Constant offset from the laser data elevation value that is added to the elevation coordinate of element vertices.
Adjust xy	If on, the xy location of data clicks (vertices) is adjusted.
Class	Point class to adjust to. Contains the list of active classes in TerraScan.
Point	Points or surface model from which element vertex coordinates are derived: • **Closest -** point closest to the data click. • **Highest** - highest point within a search area. • **Average** - average xy and/or z of all points within a search area. • **Percentile -** average xy and/or z of a given percentile of points within a search area. • **Lowest** - lowest point within search area. • **TIN model** - elevation of a triangulated surface model and xy from the closest point.
Within	Radius of the search area around the mouse pointer location.

✍ Be sure to always close the **Mouse Point Adjustment** dialog if you do not want to adjust data clicks to laser points. As it effects all data clicks, it may interfere with your normal work if it is active.

Fit Linear Element



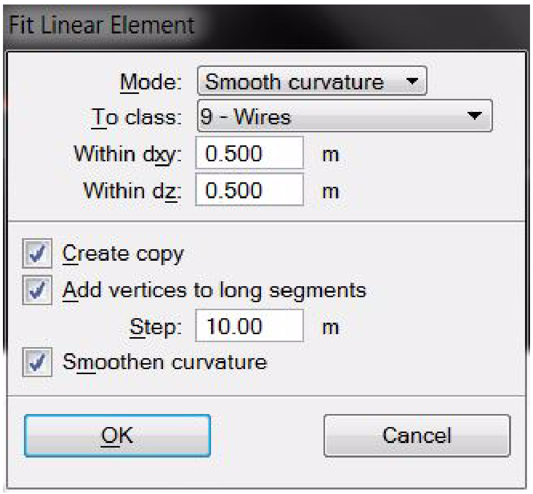
Fit Linear Element tool improves the horizontal accuracy of manually placed linear elements by fitting the xy location of vertices to laser points. The resulting linear element follows laser points more closely.

Valid MicroStation element types for this tool include lines, line strings, shapes, and complex shapes. You can fit several selected elements in a single process.

➢ **To fit linear elements:**

1. (Optional) Select the element(s) that you want to fit.
2. Select the *Fit Linear Element* tool.

   This opens the **Fit Linear Element** dialog:



3. Define settings and click OK.

   If elements have been selected, they are fitted to the laser points.
4. Identify the element to fit with a data click.

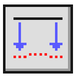
   This highlights the given element.
5. Accept the highlighted element with another data click.

   This fits the selected element to follow laser points more accurately. You can continue to step 4.

Setting:	Effect:
Mode	Defines whether vertices are added to the fitted element or not: • **Sharp vertices** - no additional vertices are added. Suitable for fitting elements with straight line segments between sharp turns (for example overhead wires). • **Smooth curvature** - additional vertices may be added. Suitable for fitting an element which has smooth curvature only (for example paint line on a road).
To class	Point class to fit to. Contains the list of active classes in TerraScan.
Within dxy	Maximum horizontal offset of laser points to be used in the fitting process.
Within dz	Maximum vertical offset of laser points to be used in the fitting process. Enter a large value such as 999.000 if you want to use all laser points regardless of their elevation.

Setting:	**Effect:**
Create copy	If on, a copy of the original element is created and fitted. The new element is placed on the active level using active symbology settings of MicroStation. If off, the original element is fitted.
Ignore points close to vertices	If on, the fitting process ignores points within the distance to element vertices given in the **Within dxy** field. This is only active if **Mode** is set to **Sharp vertices**.
Add vertices to long segments	If on, the fitting process adds intermediate vertices along long segments. The distance between consecutive vertices is given in the **Step** field. This is only active if **Mode** is set to **Smooth curvature**.
Smoothen curvature	If on, the curvature of the fitted element is smoothed by balancing angular direction changes between consecutive vertices. This is only active if **Mode** is set to **Smooth curvature**.

Drape Linear Element



Drape Linear Element tool fits linear elements to the elevation of laser points. The xy position of the elements is not effected.

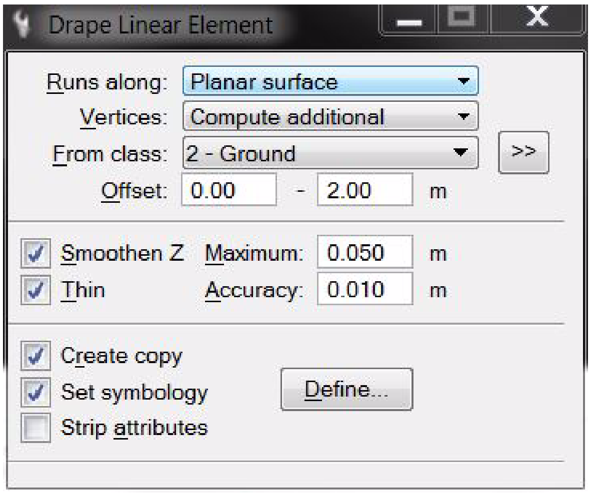
The tool is typically used to drape linear elements which run on a smooth, planar ground surface or along edges of slopes. Valid MicroStation element types for this tool include lines, line strings, shapes, and complex shapes. You can fit several selected elements in a single process.

You can decide whether you want to adjust only existing vertices or add intermediate vertices so that the resulting element follows changes of the surface more closely. The density of automatically added vertices depends on the density of laser points. In addition, smoothing and thinning can be applied to the adjusted element.

➢ **To drape linear elements to laser points:**

1. (Optional) Select the element(s) that you want to drape.
2. Select the *Drape Linear Element* tool.

   This opens the **Drape Linear Element** dialog:



3. Define settings.
4. If elements have been selected, start the draping process with a data click inside the MicroStation view.

   This drapes the selected elements to the laser points.

   OR

4. Identify the element to drape.

   This highlights the given element.

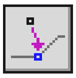
5. Accept the highlighted element with a data click.

   This drapes the linear element to the laser points. You can continue to steps 3 or 4.

Setting:	Effect:
Runs along	Type of surface structure the linear element runs along: • **Planar surface** - smooth planar surface. • **Juncture of surfaces** - intersection of one planar surface on the left side of the element and another planar surface on the right side. • **Edge of surface** - edge of a surface where only points on the left or on the right side of the edge are used for draping. • **Fixed height curb stone** - curb stone of constant height. Two linear elements are generated with a given elevation difference. • **Auto height curb stone** - curb stone of varying height. The average elevation difference is derived from the points on the left and the right side of the curb stone edge. Two linear elements are generated which have the derived elevation difference.
Vertices	Determines the computation of additional vertices: • **Compute additional** - additional vertices are computed, the draped element follows the surface structure more closely. • **Drape original only** - no additional vertices are added.
From class	Point class(es) to drape to. Contains the list of active classes in TerraScan.
>>	Opens the **Select classes** dialog which contains the list of active classes in TerraScan. You can select multiple source classes from the list that are then used in the **From class** field.
Offset	Offset distance range from which to use laser points for computing elevation values for the linear element.
Smoothen Z	If on, smoothing is applied to the elevation of vertices of the draped element. The elevation of vertices can change up to the value given in the **Maximum** field.
Thin	If on, unnecessary vertices are removed from the draped element. The allowed change in position caused by thinning is defined by the value in the **Accuracy** field.
Create copy	If on, a new element is created and draped. If off, the original element is draped.
Set symbology	If on, you can define new symbology settings for the draped element. Click on the **Define** button in order to open the **Draped element symbology** dialog. You can define **Level**, **Color**, **Weight**, and **Style** settings.
Strip attributes	If on, any attribute linkages are removed from the draped element.
Shift	If on, the draped element is created at the given xy **Distance** from the surface edge location. This is only active if **Runs along** is set to **Edge of surface**.
Curb width	Width of a curb stone, xy offset between the upper and lower linear element of a curb stone. This is only active if **Runs along** is set to **Fixed height curb stone** or **Auto height curb stone**.
Curb height	Height of a curb stone, elevation offset between the upper and lower linear element of a curb stone. This is only active if **Runs along** is set to **Fixed height curb stone**.

✍ To drape shapes to a constant elevation derived from laser points inside the shapes, you may also check the *Set Polygon Elevation* tool.

Find Breakline Along Element



Find Breakline Along Element tool creates a linear element which runs along a breakline in the terrain. The search starts with an existing 2D linear element which runs close to the actual breakline location.

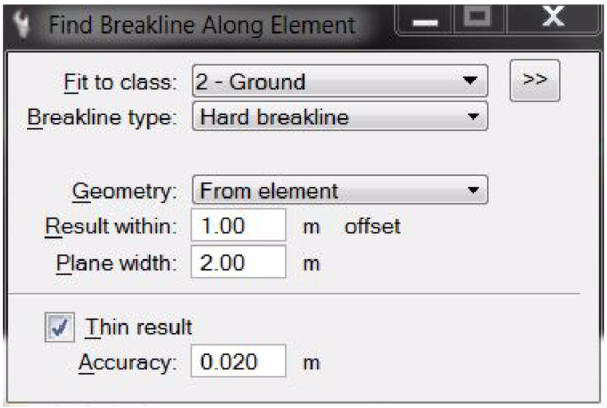
Valid MicroStation element types for this tool include lines, line strings, shapes, and complex shapes. You can fit several selected elements in a single process.

The tool finds the more accurate breakline position if there is a planar surface on both sides of the breakline. It is most useful to create hard breakline elements such as the top of man-made slopes.

➢ **To find a breakline feature:**

1. (Optional) Select the element(s) from which you want to create breaklines.
2. Select the *Find Breakline Along Element* tool.

   This opens the **Find Breakline Along Element** dialog:



3. Define settings.
4. If elements have been selected, start the process with a data click inside the MicroStation view.

   This creates new 3D elements at the most probably position of terrain breaklines close to the selected elements.

   OR

4. Identify the 2D linear element running close to a terrain breakline.

   This highlights the given element.

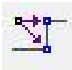
5. Accept the highlighted element with a data click.

   The application determines the most probable position for a breakline using the given parameters and creates a 3D element running along the breakline. You can continue to step 3 or 4.

Setting:	**Effect:**
Fit to class	Point class from which to find the breakline in the terrain. The list contains the active point classes in TerraScan.
	Opens the **Select classes** dialog which contains the list of active classes in TerraScan. You can select multiple source classes from the list that are then used in the **Fit to class** field.

Setting:	Effect:
Breakline type	Type of the breakline: • **Hard breakline** - breakline which forms a sharp corner when viewed in a cross section. • **Soft breakline** - breakline which forms a soft corner when viewed in a cross section. • **Slope change** - terrain slope changes at the breakline. • **Elevation jump top** - top of a breakline feature forming a drop in elevation. • **Elevation jump bottom** - bottom of a breakline feature forming a drop in elevation.
Soft breaks	Distance between single linear elements that form a soft breakline. This is only active if **Breakline type** is set to **Soft breakline**.
Geometry	Geometry type of the resulting element: • **Line** - feature may have sharp turns. • **Curve** - feature has smooth turns only. • **From element** - derived from the original element.
Result within	Determines how close the original 2D element is to the true xy position of the breakline.
Plane width	Width of the planar surfaces on both sides of the breakline.
Thin result	If on, unnecessary vertices are removed from the breakline element. The allowed change in position caused by thinning out vertices is defined by the value in the **Accuracy** field.

Find Curb Along Element



Find Curb Along Element tool is not yet implemented.

Cut Linear Element

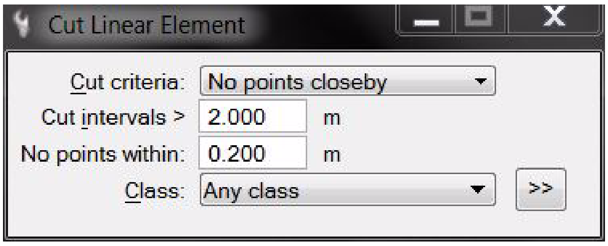
Cut Linear Element tool determines the distance between linear elements and laser points or other linear elements. It removes parts of linear elements for which there are no laser points, no other linear elements, or other linear elements within a certain distance.

It may be used, for example, to mark places where there are elements close to rail tracks or wires.

➢ **To cut linear elements:**

1. (Optional) Select the element(s) that you want to cut.
2. Select the *Cut Linear Element* tool.

   This opens the **Cut Linear Element** dialog:



3. Define settings.
4. If elements have been selected, start the process with a data click inside the MicroStation view.

   This compares the selected elements to laser points or other elements and removes parts if applicable.

   OR

4. Identify the element to cut.

   This highlights the given element.

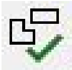
5. Accept the highlighted element with a data click.

   This compares the selected element to the laser points or other elements and removes parts if applicable. You can continue to steps 3 or 4.

Setting:	Effect:
Cut criteria	Defines which element parts are removed: • **No points closeby** - no laser points are close to the element. • **No elements closeby** - no other vector elements are close to the element. • **Another element closeby** - another element is close to the element.
Cut intervals	
No points within	An element (part) is removed if the 3D distance to laser points/other elements is larger than the given value.
Class	Point class(es) considered in the distance computation. This is only active if **Cut criteria** is set to **No points closeby**.

Setting:	**Effect:**
>>	Opens the **Select classes** dialog which contains the list of active classes in TerraScan. You can select multiple source classes from the list that are then used in the **Class** field.
Level	Elements of the given design file level are considered for distance computation. This is only active if **Cut criteria** is set to **No elements closeby** or **Another element closeby**.

Check Footprint Polygons



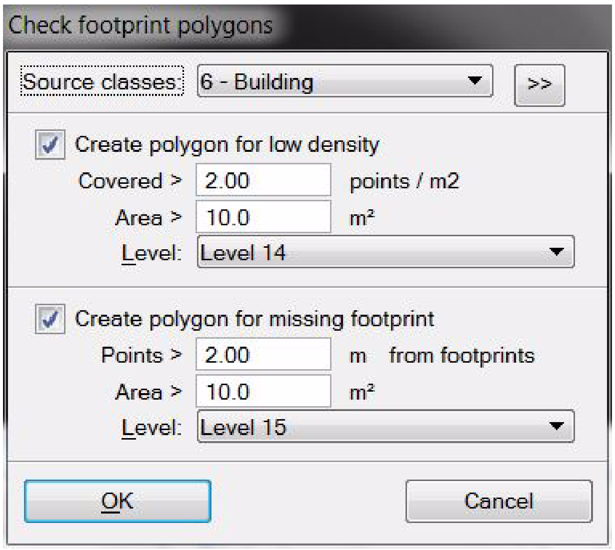
Check Footprint Polygons tool compares building footprint polygons with laser points, usually points classified into building class. It creates polygons at locations where there is a footprint polygon but no laser points in the building class, or where there are laser points but no footprint polygon.

The tool is useful for finding flaws in the building classification, places with no or very sparse laser points on building roofs, and flaws in building footprint vector data.

➢ **To compare footprint polygons and laser data:**

1. Load laser data into TerraScan. Only points in class(es) for the comparison with footprint polygons are required.
2. Select the footprint polygons that you want to include in the comparison.
3. Select the **Check Footprint Polygons** tool.

   This opens the **Check footprint polygons** dialog:



4. Define settings and click OK.

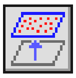
   The comparison starts and the software creates polygons for areas of low point density and/or missing footprints according to the given settings. An information dialog shows the number of created polygons.

Setting:	**Effect:**
Source classes	Point class(es) used for the comparison. The list contains the active classes in TerraScan.
	Opens the **Select classes** dialog which contains the list of active classes in TerraScan. You can select multiple source classes from the list that are then used in the **Source classes** field.
Create polygons for low density	If on, polygons are created on places where there is a selected polygon but no or only sparse laser points in the given **Source class(es)**.
Covered	Defines the minimum point density inside a footprint polygon. If the density is lower, a polygon is created.
Area	Defines the minimum area of a building. Only areas larger than the given value are considered in the comparison.
Level	Polygons marking low density places are drawn on the given level using the active symbology settings of MicroStation.

Setting:	Effect:
Create polygons for missing footprint	If on, polygons are created on places where there are laser points in the given **Source class(es)** but no selected polygon.
Points	Defines the minimum distance between a laser point and a selected polygon. If the distance is larger, a polygon is created.
Area	Defines the minimum area of a building. Only areas larger than the given value are considered in the comparison.
Level	Polygons marking missing footprint places are drawn on the given level using the active symbology settings of MicroStation.

✍ You can check the polygons created by this tool in a structured way with the help of the *Inspect Elements* tool.

Set Polygon Elevation



Set Polygon Elevation tool sets the elevation of a closed element based on laser points inside it.

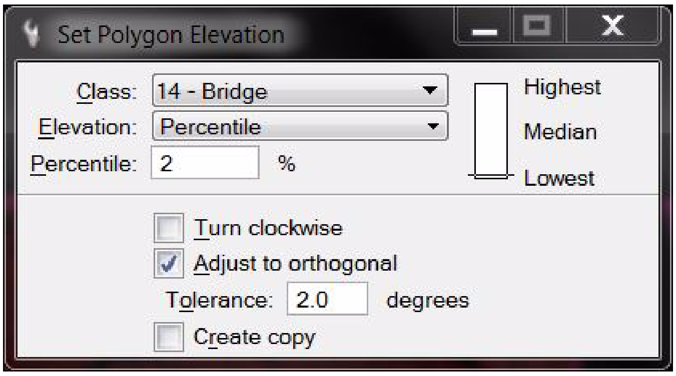
This tool is typically used to drape a digitized 2D shape to the elevation of laser points on a building roof or a bridge. All vertices of the shape are set to the same elevation value derived from laser points.

Valid MicroStation element types include shapes, complex shapes, and ellipses. You can fit several selected elements in a single process.

➢ **To set the elevation of a closed element:**

1. (Optional) Select elements for which you want to set the elevation.
2. Select the *Set Polygon Elevation* tool.

   This opens the **Set Polygon Elevation** dialog:



3. Define settings.
4. If elements have been selected, start the process with a data click inside the MicroStation view.

   This computes an elevation value from laser points for each selected element and adjusts all selected elements.

   OR

4. Identify the element to adjust with a data click.

   This highlights the element.

5. Accept the highlighted element with a data click inside the MicroStation view.

   This computes an elevation value from laser points and adjusts the element. You can continue with steps 3 or 4.

Setting:	Effect:
Class	Point class from which to derive the elevation. Only points inside the shape element are used.
Elevation	Method of elevation computation: • **Percentile** - elevations from the given **Percentile** value of points. The scale shows which percentile of points is used: **Lowest**, **Median**, or **Highest** points, and can be used to set the percentile value with a data click. • **Average** - average of all laser point elevation values inside the shape.
Turn clockwise	If on, the drawing direction of the adjusted shape is forced to run clockwise.

Setting:	Effect:
Adjust to orthogonal	If on, the corner angles of the adjusted shape are fixed to 90 degree turns if the angles of the original element are within **Tolerance** value off from 90 degrees.
Create copy	If on, a new element is created and adjusted using the active level and symbology settings of MicroStation. If off, the original element is modified.

Construct Planar Building

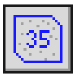
Not Lite

Construct Planar Building tool is used to create a 3D vector model of a building based on laser points on planar surfaces of the roof. With this tool, one building at a time can be vectorized in a half-automatic way.

In fact, the tool opens a kind of special processing environment for the building vectorization task. This includes the arrangement of MicroStation views and the opening of the **Construct Building** dialog and of two new tool sets.

In addition to the *Construct Planar Building* tool, TerraScan offers several tool sets for the automatic creation of building models and for manual improvements. All options, tools, and processing guidelines are described in detail in Chapter **3D Building Models** on page 189.

Place Collection Shape



Place Collection Shape tool lets you create shape elements that are associated with certain thematic types. Collection shapes are used to group laser points together that belong, for example, to a topographic object, such as a building or a road.

Collection shapes can be used later to output groups of laser points.

You need to define collection shape types before you can use the tool. See **Collection shapes** category of TerraScan **Settings** for more information.

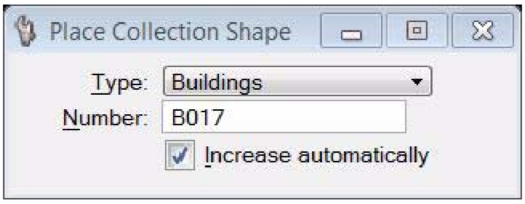
Collection shapes can be created manually by using the digitization function of the *Place Collection Shape* tool. Alternatively, they can be produced automatically from already existing shape elements that have been created using any MicroStation tool for shape drawing.

Collection shapes do not store any attribute information. The application only uses element level, color, line weight, and line style to recognize a shape as a collection shape of a specific type.

➢ **To place a collection shape:**

1. (Optional) Select shape elements that you want to turn into collection shapes.
2. Select the *Place Collection Shape* tool.

   The **Place Collection Shape** dialog opens:



3. Select a collection shape type in the **Type** list.
4. If shape elements have been selected, apply the collection shape symbology with a data click inside the MicroStation view.

   This turns all selected shapes into collection shapes. An information dialog shows the number of effected shape elements.

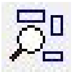
   OR

4. (Optional) Define additional settings in the **Place Collection Shape** dialog.
5. Digitize the shape boundary by placing vertices with data clicks. To close the shape, place a data click close to the first vertex. You can undo a vertex placement with a reset click.

   This creates a shape element on the level and using the symbology specified for the collection shape type. If a **Number** is defined, a text element is drawn inside the shape element using active text settings of MicroStation.

Setting:	Effect:
Type	Collection shape type. The list contains all shape types that are defined in **Collection shapes** category of TerraScan **Settings**.
Number	Text string that is drawn inside the shape element. This works only if collection shapes are digitized manually.
Increase automatically	If on, and if the text string in the **Number** field ends with a number, the number increases by 1 after a collection shape has been created manually.

Inspect Elements



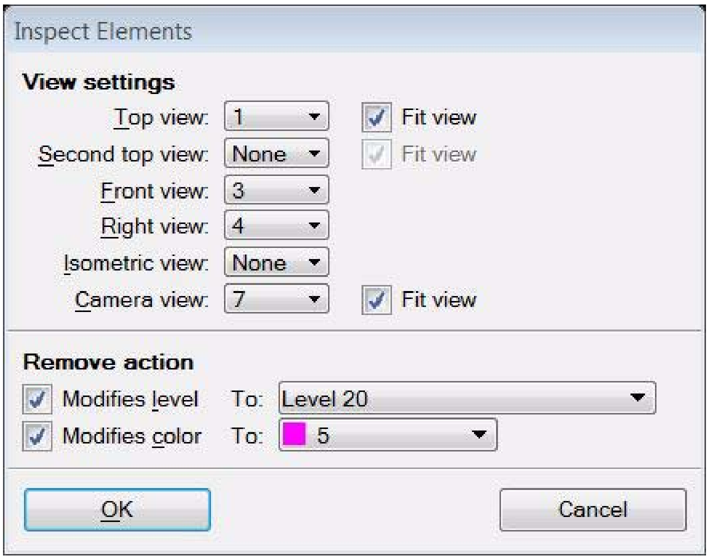
Inspect Elements tool supports the systematic check of vector elements in a design file. It provides a list of elements from which you can select one element after the other.

The tool includes view settings that define MicroStation views displaying the selected element in different view orientations. The selected element is automatically centered in these views.

➢ **To inspect vector elements:**

1. Select the elements you want to check.
2. Select the *Inspect Elements* tool.

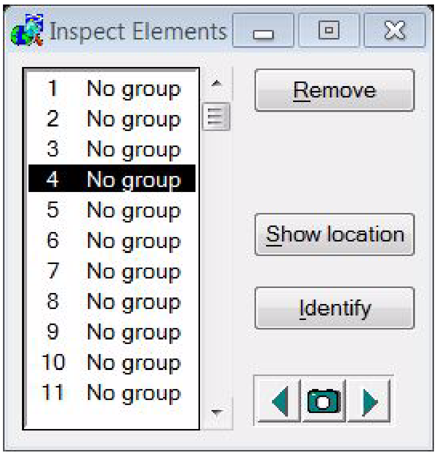
   This opens the **Inspect Elements** dialog:



Setting:	Effect:
Top view	MicroStation view that displays the selected element in a top view.
Second top view	MicroStation view that displays the selected element in a top view.
Front view	MicroStation view that displays the selected element in a front section view.
Right view	MicroStation view that displays the selected element in an isometric view.
Camera view	MicroStation view that displays the selected element in a camera view. The view can display images that are referenced by an active image list in TerraPhoto.
Fit view	If on, the selected element is automatically fitted in the view.
Modifies level	If on, an element removed by the **Remove** button of the **Inspect Elements** dialog is moved to the given MicroStation level.
Modifies color	If on, the selected color is applied to an element removed by the **Remove** button of the **Inspect Elements** dialog. The list contains the active color table of MicroStation.

3. Define settings and click OK.

   This opens another **Inspect Elements** dialog that contains the list of all selected elements:



4. Select a line in the list of elements.

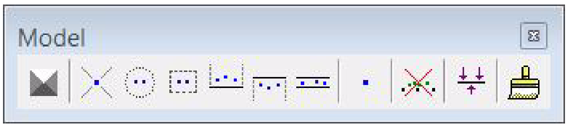
   This centers the selected element in all MicroStation views defined in the tool's **View settings**. You can use the remove button of the dialog to remove an element or take any other appropriate action.

Setting:	Effect:
Remove	Removes the selected element from the list. The element itself is not deleted but it can be moved to another level and/or get another color according to the tool's **Remove action** settings.
Show location	Select a line in the list, click on the button and move the mouse pointer inside a MicroStation view. This highlights the selected element in the view.
Identify	Click on the button and identify an element with a data click in a MicroStation view. This selects the corresponding line in the list.
	Moves one image backward in the active image list and displays the new image in the camera view.
	Click on the button and move the mouse pointer inside a MicroStation view. The image closest to the mouse pointer is highlighted. Select an image for the camera view display with a data click.
	Moves one image forward in the active image list and displays the new image in the camera view.

Model tool box

The tools in the **Model** tool box are used to create an editable surface model, to classify laser points manually, to fix elevations of laser points, and to update the surface model.

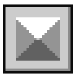
The tools for manual point classification effect points loaded in TerraScan memory. You have to save the points in order to store changes permanently into laser point files.



To:	Use:
To create an editable triangulated model	*Create Editable Model*
Assign class to a laser point	*Assign Point Class*
Classify points using a brush	*Classify Using Brush*
Classify points inside fence	*Classify Fence*
Classify points above line in section view	*Classify Above Line*
Classify points below line in section view	*Classify Below Line*
Classify points close to line in section view	*Classify Close To Line*
Add a synthetic point using mouse click	*Add Synthetic Point*
Classify vegetation points out of ground	*Remove Vegetation*
Set elevation of points inside polygon(s)	*Fix elevation*
Rebuild model after classification	*Rebuild Model*

✍ *Create Editable Model* and *Rebuild Model* tools require TerraModeler to run.

Create Editable Model



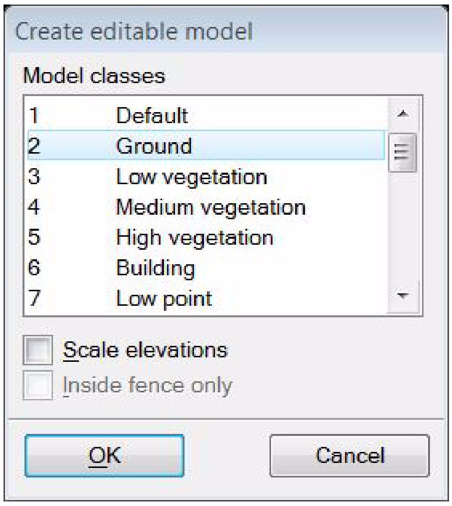
Create Editable Model tool creates a surface model from loaded laser points which can be visualized in TerraModeler. The tool starts TerraModeler automatically if the application is not yet running.

The surface model is actively linked to the loaded laser points, which means that all surface model displays are updated immediately to reflect any change in point classification. The active surface model display is particularly useful for validating ground classification. The best display method for this purpose is a shaded surface drawn by the *Display Shaded Surface* tool in TerraModeler's **Display Surface** tool box.

➢ **To create an editable model and display a shaded surface:**

1. Load laser data into TerraScan.
2. Select the *Create Editable Model* tool.

   This opens the **Create editable model** dialog:



3. Select class(es) which you want to include in the surface model.
4. (Optional) Switch on **Scale elevations** and type a factor by which you want to scale the elevations. A factor > 1.0 results in a model with exaggerated elevation values.
5. Click OK.

   This opens the **Surface settings** dialog in TerraModeler.

6. Enter a descriptive name for the new surface, define other settings if required, and click OK.

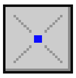
   TerraModeler creates the surface model.

7. Select the *Display Shaded Surface* tool in TerraModeler's **Display Surface** tool box.
8. Define display settings and click OK.

   This displays the shaded surface using the elevation values of the laser points and given lightning conditions.

✓ If TerraModeler is not available, you can use the **Color by Shading** display method of TerraScan to display a shaded surface visualization of the laser points based on class coloring.

Assign Point Class



Assign Point Class tool classifies a single laser point or points that belong to a group of points. It classifies either the closest point to the data click, or the highest or lowest point within a circular search area. For group classification, the point selection method determines which group is classified.

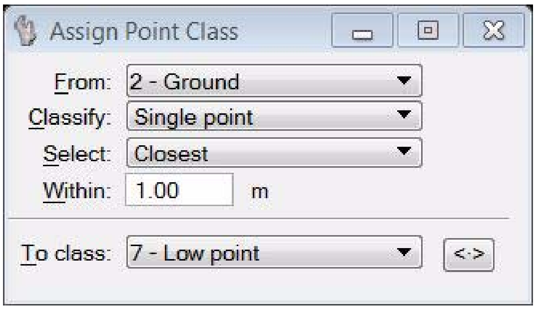
Classifying points of a group requires the assignment of group numbers to laser points. This can be done by using the **Assign groups** command for loaded points or the corresponding macro action for macro processing of project blocks.

The tool works in top views as well as in section views or any rotated views.

➢ **To classify a single point or a group of points:**

1. Select the *Assign Point Class* tool.

   The **Assign Point Class** dialog opens:



2. Define settings.
3. Move the mouse pointer inside a view.

   In a top view, the search area is shown at the mouse pointer position.

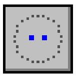
4. Identify the single point or the group of points with a data click.

   This classifies the identified point or point group. You can continue with step 2 if you want to change settings, or with step 3.

5. Use **Save points** or **Save points As** commands in order to save changes to point classes permanently into a laser point file.

Setting:	Effect:
From	Source class; only points from this class are effected. The list contains the active classes in TerraScan. Alternatively, **Any visible point** can be classified.
Classify	Defines which points are classified: **Single point** or **Whole group**. **Whole group** is only active if group numbers are assigned to laser points.
Select	Method how the software selects a point or group for classification: • **Closest -** the point closest to the data click is classified. • **Highest -** the highest point within the search area is classified. • **Lowest -** the lowest point within the search area is classified.
Within	Radius of the search area. Given in master units of the design file.
To class	Target class into which points are classified. The list contains the active classes in TerraScan.
	Switches **From** and **To class** classes. If **From** is set to **Any visible point**, **To class** is switched to the source class with the lowest class number.

Classify Using Brush



Classify Using Brush tool classifies points inside a circular or rectangular brush moved in a MicroStation view.

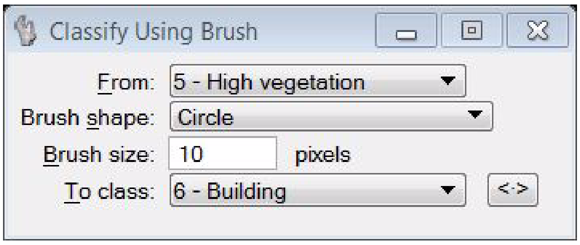
The tool can be utilized with two different kinds of mouse action. You can use it with two separate data clicks and mouse pointer movement in between, or you can keep the data button pressed down while moving the mouse pointer.

The tool works in top views as well as in section views.

➢ **To classify points inside a brush using two mouse clicks:**

1. Select the *Classify Using Brush* tool.

   The **Classify Using Brush** dialog opens:



2. Define settings.
3. Place a data click to start the classification.

   OR

3. Press the data button down to start the classification.

   This classifies points inside the brush area.

4. Move the mouse pointer to classify additional points.
5. Place another data click to stop the classification.

   OR

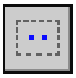
5. Release the data button to stop the classification.

   This classifies all points touched by the mouse pointer. You can continue with step 2 if you want to change settings, or with step 3.

6. Use **Save points** or **Save points As** commands in order to save changes to point classes permanently into a laser point file.

Setting:	Effect:
From	Source class; only points from this class are effected. The list contains the active classes in TerraScan. Alternatively, **Any visible point** can be classified.
Brush shape	Shape of the brush: **Circle** or **Rectangle**.
Brush size	Size of the brush. Given in pixels on the screen.
To class	Target class into which points are classified. The list contains the active classes in TerraScan.
	Switches **From** and **To class** classes. If **From** is set to **Any visible point**, **To class** is switched to the source class with the lowest class number.

Classify Fence



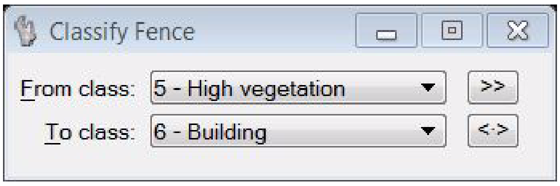
Classify Fence tool classifies points inside a fence area. The fence area can be defined a by MicroStation fence or by a polygon drawn with the *Classify fence* tool.

The tool works in top views as well as in section views.

➢ **To classify points inside a fence:**

1. (Optional) Draw a fence using the *Place Fence* tool of MicroStation.
2. Select the *Classify Fence* tool.

   The **Classify Fence** dialog opens:



3. Define settings.
4. If a fence has been drawn, accept the fence contents with a data click inside the view.

   This classifies the points inside the fence.

   OR

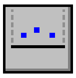
4. Digitize a fence around the points you want to classify by placing data clicks inside a view. The fence is closed if you place a data click close to the first vertex of the fence.
5. Accept the fence contents with a data click.

   This classifies the points inside the fence. You can continue with step 3 if you want to change settings, or with step 4 if you digitize the fence with this tool.

6. Use **Save points** or **Save points As** commands in order to save changes to point classes permanently into a laser point file.

Setting:	**Effect:**
From class	Source class(es); only points from selected class(es) are effected. The list contains the active classes in TerraScan. Alternatively, points from multiple classes or **Any visible point** can be classified.
To class	Target class into which points are classified. The list contains the active classes in TerraScan.
	Opens the **Select classes** dialog which contains the list of active classes in TerraScan. You can select multiple source classes from the list that are then used in the **From class** field.
	Switches **From** and **To class** classes. If **From** is set to multiple classes or **Any visible point**, **To class** is switched to the source class with the lowest class number.

Classify Above Line



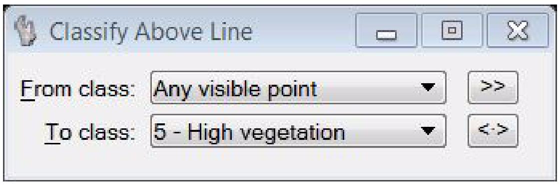
Classify Above Line tool classifies points above a line drawn in a cross section view. The classification effects only points that are inside the extend and display depth of the cross section view.

The tool works in section views.

➢ **To classify points above a line:**

1. Select the *Classify Above Line* tool.

   The **Classify Above Line** dialog opens:

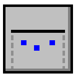
2. Define settings.
3. Draw a line by placing data clicks in a section view. The line and the area effected by the classification are temporarily displayed after placing the start point.

   This classifies points which are above the line, between the line’s start and end point, and within the section view extend and depth. You can continue with step 2 if you want to change settings, or with step 3.

4. Use **Save points** or **Save points As** commands in order to save changes to point classes permanently into a laser point file.

Setting:	Effect:
From class	Source class(es); only points from selected class(es) are effected. The list contains the active classes in TerraScan. Alternatively, points from multiple classes or **Any visible point** can be classified.
To class	Target class into which points are classified. The list contains the active classes in TerraScan.
	Opens the **Select classes** dialog which contains the list of active classes in TerraScan. You can select multiple source classes from the list that are then used in the **From class** field.
	Switches **From** and **To class** classes. If **From** is set to multiple classes or **Any visible point**, **To class** is switched to the source class with the lowest class number.

Classify Below Line



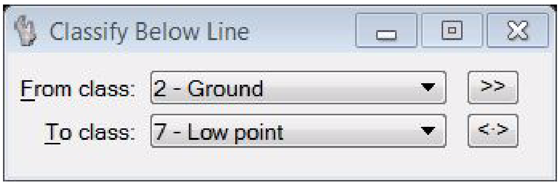
Classify Below Line tool classifies points below a line drawn in a cross section view. The classification effects only points that are inside the extend and display depth of the cross section view.

The tool works in section views.

➢ **To classify points below a line:**

1. Select the *Classify Below Line* tool.

   The **Classify Below Line** dialog opens:



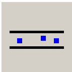
2. Define settings.
3. Draw a line by placing data clicks in a section view. The line and the area effected by the classification are temporarily displayed after placing the start point.

   This classifies points which are below the line, between the line's start and end point, and within the section view extend and depth. You can continue with step 2 if you want to change settings, or with step 3.

4. Use **Save points** or **Save points As** commands in order to save changes to point classes permanently into a laser point file.

Setting:	Effect:
From class	Source class(es); only points from selected class(es) are effected. The list contains the active classes in TerraScan. Alternatively, points from multiple classes or **Any visible point** can be classified.
To class	Target class into which points are classified. The list contains the active classes in TerraScan.
	Opens the **Select classes** dialog which contains the list of active classes in TerraScan. You can select multiple source classes from the list that are then used in the **From class** field.
	Switches **From** and **To class** classes. If **From** is set to multiple classes or **Any visible point**, **To class** is switched to the source class with the lowest class number.

Classify Close To Line



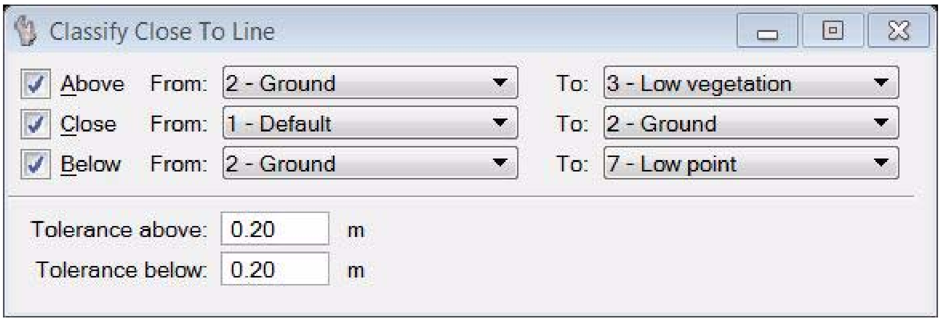
Classify Close To line tool classifies points that are close to a given line in a cross section view. It can combine up to three classification steps, above, close, and below a line. The classification effects only points that are inside the extend and display depth of the cross section view.

The tool works in section views.

➢ **To classify points close to lines:**

1. Select the *Classify Close To Line* tool.

   The **Classify Close To Line** dialog opens:



2. Define settings.
3. Draw a line by placing data clicks in a section view. The line and the area effected by the classification are temporarily displayed after placing the start point.

   This classifies points which are above the line, below the line, and/or close to the line, between the lines' start and end point, and within the section view extend and depth. You can continue with step 2 if you want to change settings, or with step 3.

4. Use **Save points** or **Save points As** commands in order to save changes to point classes permanently into a laser point file.

Setting:	Effect:
Above	If on, above line classification is applied. Source and target classes are selected in the **From** and **To** lists of active classes. Alternatively, **Any visible point** can be selected in the **From** list.
Close	If on, close to line classification is applied. Source and target classes are selected in the **From** and **To** lists of active classes. Alternatively, **Any visible point** can be selected in the **From** list.
Below	If on, below line classification is applied. Source and target classes are selected in the **From** and **To** lists of active classes. Alternatively, **Any visible point** can be selected in the **From** list.
Tolerance above	Distance from the drawn line to the line that defines the above line classification limit. Together with the **Tolerance below** value, this defines the area of close to line classification.
Tolerance below	Distance from the drawn line to the line that defines the below line classification limit. Together with the **Tolerance above** value, this defines the area of close to line classification.

Add Synthetic Point

Add Synthetic Point tool adds synthetic points to a point data set loaded in TerraScan. It creates one point per data click.

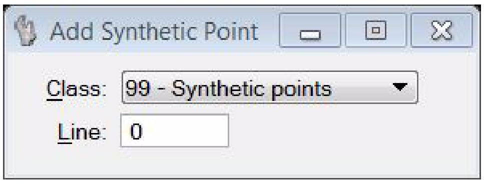
If a point is added in a top view, its xy location is determined by the data click position and its elevation is fixed to the active Z setting in the design file. If a point is added in a section view, its xy location is fixed to the center of the section and its elevation is set by the data click position.

The tool works in top views as well as in section views or any rotated views.

➢ **To add a synthetic point:**

1. Select the *Add Synthetic Point* tool.

   The **Add Synthetic Point** dialog opens:



2. Define settings for the new point.
3. Place a synthetic point with a data click in a view.

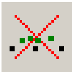
   This adds a new point to the loaded point data set. You can continue with step 2 if you want to change settings, or with step 3.

4. Use **Save points** or **Save points As** commands in order to save added points permanently into a laser point file.

Setting:	Effect:
Above	Target class into which synthetic points are added. The list contains the active classes in TerraScan.
Line	Flightline number assigned to synthetic points.

✍ MicroStation elements can be snapped to place a synthetic point at an exact location. This might be useful, for example, if points have to be added in a regular grid. Also, the *Mouse Point Adjustment* tool can be used to derive elevation values for synthetic points from laser points.

Remove Vegetation



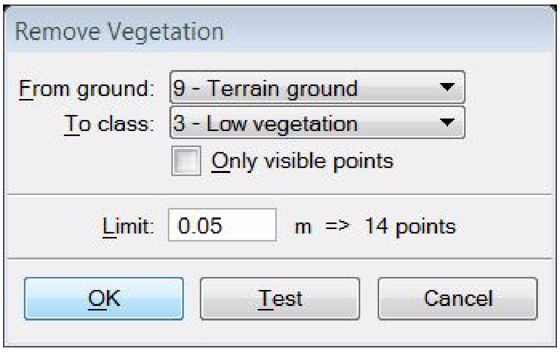
Remove Vegetation tool removes points in low vegetation from points classified as ground. It works within a limited area defined by a MicroStation fence or selected polygon.

The can be used, for example, to smooth the ground surface after automatic ground classification.

➢ **To remove vegetation from ground points:**

1. Use MicroStation tools to draw a fence or polygon around the area for processing. Select the polygon.
2. Select the *Remove Vegetation* tool.

   The **Remove Vegetation** dialog opens:



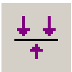
3. Define settings.
4. Click on the **Test** button to see the result of the settings in a preview.
5. Click OK to apply the classification.

   This classifies the points from the ground class to the target class.

6. Use **Save points** or **Save points As** commands in order to save changes to point classes permanently into a laser point file.

Setting:	Effect:
From ground	Source class, usually a ground class; only points from the selected class is effected. The list contains the active classes in TerraScan.
To class	Target class into which points are classified. The list contains the active classes in TerraScan.
Only visible ponts	If on, only points that are visible in the fence are effected. If off, all points in the fence are effected, even if their display is switched off.
Limit	Distance up to which points above a surface defined by the lowest ground points are classified. Given in the master unit of the design file. The amount of effected points is shown next to the input field.

Fix elevation



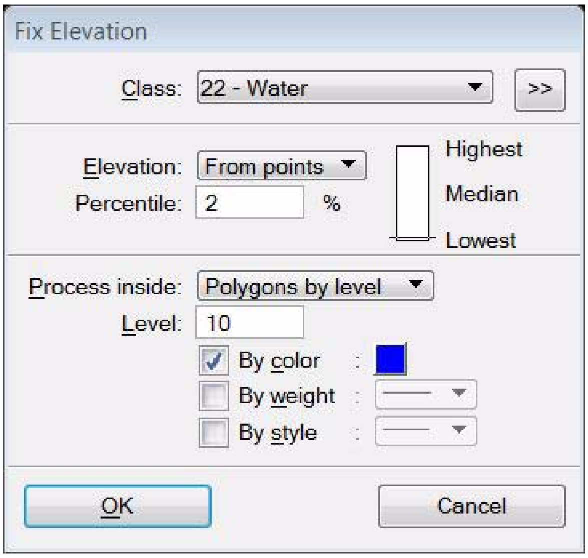
Fix Elevation tool fixes elevation values of points to a constant value. The tool processes data inside limited areas that can be defined by selected polygons, polygons on a given design file level, or a fence.

This tool may be useful, for example, to get a smooth surface from points on water.

➢ **To fix elevations of points:**

1. Use MicroStation tools to draw polygons or a fence around the area(s) for processing. Select polygons, if required.
2. Select the *Fix Elevation* tool.

   The **Fix Elevation** dialog opens:



3. Define settings and click OK.

   This sets the elevation values of the laser points in the selected class(es) and inside the processing area to a constant elevation value.

4. Use **Save points** or **Save points As** commands in order to save changes to point classes permanently into a laser point file.

Setting:	Effect:
Class	Source class(es); only points from selected class(es) are effected. The list contains the active classes in TerraScan. Alternatively, points from multiple classes or **Any class** can be selected.
	Opens the **Select classes** dialog which contains the list of active classes in TerraScan. You can select multiple source classes from the list that are then used in the **Class** field.
Elevation	Method of calculating the constant elevation value: • **From points** - elevation derived from the given **Percentile** value of points. The scale shows which percentile of points is used: **Lowest**, **Median**, or **Highest** points, and can be used to set the percentile value with a data click. • **Keyin value** - absolute elevation value given in the **Value** field.

Setting:	Effect:
Process inside	Determines the processing area: • **Fence -** points inside a MicroStation fence are effected. • **Selected polygons -** points inside selected polygons are effected. • **Polygons by level -** points inside polygons drawn on the given design file **Level** are effected. The polygons can be further specified by selecting **By color**, **By weight**, and/or **By style** options. The selection lists use the active color table, line weights and line styles of MicroStation.

Rebuild Model

Rebuild Model tool runs a complete update of an editable surface model. It effects the surface model that has been created by the *Create Editable Model* tool. The process re-creates the TIN structure from all points in the model class(es) and updates all active displays of the surface model.

You should use this tool if you classify points to/from the surface model class(es) using tools other than those in the **Model** tool box or if the automatic update of the editable surface model does not work correctly.

➢ **To rebuild a model:**

1. Select the *Rebuild Model* tool.

   If an editable model is available, the re-building process starts. A progress bar shows the progress of the process.

   If no editable model is available, an information dialog is shown.

6 Powerlines

TerraScan has a number of tools which are dedicated to powerline processing. These include classification, vectorization and reporting tools.

The general strategy for processing powerline data can be outlined as:

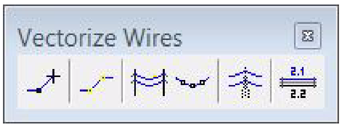
1. Classify ground points using **Ground** routine.
2. Classify high points which may be hits on wires or towers using **By absolute elevation** or **By height from ground** routines.
3. Classify rest of the points (possibly all as vegetation).
4. Manually place a tower string to run from tower to tower using *Place Tower String* tool.
5. Detect wires automatically using *Detect Wires* tool.
6. Manually place wires using *Check Catenary Attachments* tool in places where automatic detection does not work.
7. Validate and adjust catenary curves using *Check Catenary Attachments* tool.
8. Define required types of towers in *Powerlines / Tower types* in TerraScan Settings.
9. Manually place towers using *Place Tower* tool.
10. Create required labels with the help of labeling tools.
11. Output required reports.

Tools for powerline processing are divided into three parts:

- **Vectorize Wires -** tools for automatic detection and manual placement of wires, check and correction of wire attachments as well as assigning wire attributes.
- **Vectorize Towers -** tools for manual placement of towers, manipulation of towers, cross arms and attachments.
- **View Powerline -** tools for labeling powerline parts, finding danger objects and creating text files for catenaries and towers.

Vectorize Wires tool box

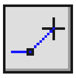
The tools in the **Vectorize Wires** tool box are used to place a powerline centerline, to detect wires automatically, to manually place catenaries, to validate catenary attachment points and to attributes to wires.



To:	Use:
Place a line string from tower to tower	*Place Tower String*
Activate a powerline for viewing and modification	*Activate Powerline*
Detect wires along active powerline	*Detect Wires*
Digitize a catenary line string	*Place Catenary String*
Check catenary attachment points at towers	*Check Catenary Attachments*
Assign number and description to wire	*Assign Wire Attributes*

Place Tower String

Not Lite



Place Tower String tool is used to manually place an approximate centerline from tower to tower. This tower string will be used in later processing steps when detecting wires, validating catenary attachment points, placing towers or producing reports along the powerline.

A tower string is a line string or a complex chain which runs along the powerline with a vertex at each tower.

Place Tower String tool integrates line string placement and view panning in one tool which speeds up the digitization process.

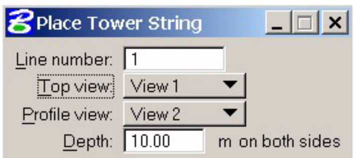
➢ **To prepare for tower string placement:**

1. Classify all higher points into a unique point class using **By absolute elevation** or **By height from ground** routines. This is to separate all points which are high from the ground and which may be hits on towers or wires. The classified high points will also include hits on other high objects such as trees and buildings.
2. Choose a view which you will use as **Top view** for placement. Optionally, choose a second view as profile view where you will see the powerline in a profile while digitizing the tower string.
3. Switch the class of high points on and the classes of all other point classes off for that view in the **Display mode** dialog from **Display mode** command. Set **Color by** to **Elevation**.
4. To emphasize the elevation coloring, sort laser points by increasing Z values using **Sort** command. This improves the visibility of towers and wires within their environment.
5. Zoom in close to a tower so that you can clearly see where the tower is located. Normally this means that you will see only one tower and some wire hits which indicate the direction where the powerline continues.
6. Set active color and active level of the design file in a way that the tower string will be placed on a level with no other elements.

➢ **To place a tower string:**

1. Select the *Place Tower String* tool.

   This opens the **Place Tower String** dialog:



2. Make sure **Top view** is set as the view you have chosen as top view.
3. Enter other settings.

Setting:	**Effect:**
Line number	Defines a number for a tower string. Each tower string should have a unique number to identify a powerline clearly.
Top view	View for digitizing the tower string. In this view the dynamic rectangle will be drawn.
Profile view	An optional view which will display a profile along the direction of tower string you are creating. You will need this if you can not see the locations of towers from a top like view. The profile view will show the curvature of catenary wires to help locate towers.

Setting:	Effect:
Depth	Width of the profile view within which points are displayed. This should include the whole width of a powerline.

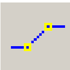
4. Enter a vertex at the approximate center of the first tower.

   This defines the location of the first tower. The application will now draw a dynamic rectangle in the **Top view** whenever you move the mouse inside view windows. If you enter a point outside the rectangle, the application will pan the view in the direction of the mouse click. Entering a point inside the rectangle adds a new vertex to the tower string.

5. Enter a mouse click outside the dynamic rectangle to move the view in the direction of the powerline.
6. Repeat step 5 until you see the next tower inside the dynamic rectangle.
7. Enter a vertex at the approximate center of the next tower.
8. Continue with step 5 until a vertex for each tower of the powerline has been placed.
9. Enter reset to finish the placement of a tower string.

As end result of tower string placement, you should get a single line string or complex chain type element for one powerline. It may happen that you can not place the entire tower string in one operation. In this case you should continue by placing another tower string which runs in the same direction and starts at the exact end point of the previous one. When you have placed a number of shorter tower strings, you can join those using MicroStation's *Create Complex Chain* tool.

Activate Powerline



Activate Powerline tool activates a tower string for further processing steps.

Most of the powerline processing functions such as detection of wires, check of catenary attachments, placement of towers as well as labeling and report tools are applied to the activated powerline.

➢ **To activate a powerline:**

1. Select the *Activate Powerline* tool.
2. Click on the tower string of the powerline to be activated.

   OR

1. Select the tower string of the powerline to be activated using MicroStation *Selection* tool.
2. Select the *Activate Powerline* tool.

   This activates the selected powerline. The tower string is displayed as defined in **Powerlines / Active line** in TerraScan Settings. An information window informs you about the number of the activated powerline.

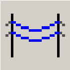
➢ **To deactivate a powerline:**

1. Select the *Activate Powerline* tool.

   This deactivates the currently activated powerline. You can continue with activating another powerline or with selecting any other tool.

Detect Wires

Not Lite



Detect Wires tool finds laser points which form a catenary curve. It will draw detected wires as line strings in the design file and classify matching points in a given class.

This command will search for points which form a straight line and also match the elevation curve of a catenary. This process involves least squares fitting for both the xy line equation and the elevation curve equation of the catenary.

The most important parameter controlling wire detection is **Max gap** which defines the maximum gap between consecutive laser hits on a wire. It is not advisable to run the detection on the whole data set with a long maximum gap because then the chance of finding false catenaries increases. You should normally run the detection first with a relatively short maximum gap which will not necessarily detect all wires. You may then want to process locations with very few hits on the wires one tower segment at a time using a longer maximum gap.

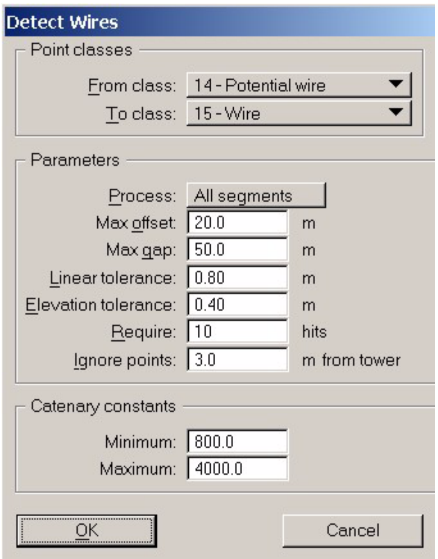
➢ **To prepare for wire detection:**

1. Classify all higher points into a unique point class.
2. Create a tower string using *Place Tower String* tool.
3. Set active color and active level in the design file to place the catenary on a separate level.
4. Activate the tower string using *Activate Powerline* tool.

➢ **To detect wires:**

1. Select the *Detect Wires* tool.

   This opens the **Detect Wires** dialog:



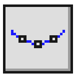
2. Fill in setting values and click OK.

   This will process the points along the activated tower string and generate catenary strings if wires are detected.

Setting:	Effect:
From class	Class from which to search points. This should be the class into which high points have been classified before.
To class	Class into which to classify points matching to detected wires.
Process	Whether to process all tower to tower spans or only a single segment.
Max offset	Maximum offset from tower string to search for points. This should be set to a bit more than half of the approximate width of the powerline.
Max gap	Maximum allowed gap between consecutive hits on a wire.
Linear tolerance	Tolerance for xy line fitting and classification of matching points.
Elevation tolerance	Tolerance for elevation curve fitting and classification of matching points.
Require	Minimum amount of laser hits matching one catenary string required for detection. Values can range from 3 to 999.
Ignore points	Distance from tower within which points are ignored. Points close to the tower can be hits on tower structure and should be ignored when determining the mathematical shape of the wire.
Minimum	Minimum catenary constant to accept as a wire.
Maximum	Maximum catenary constant to accept as a wire.

Place Catenary String

Not Lite



Place Catenary String tool lets you manually place a single catenary curve between two towers. You should use this tool in places where automatic detection of wires does not work. This happens normally when you have a very small number of hits on the wire.

You define the mathematical shape of the catenary curve with three mouse clicks. In most cases you would use **Snap to** lock to use the xyz coordinates of closest laser hits.

You can enter optional mouse clicks to define the start location and the end location of the catenary curve. These affect the length of the catenary only. You would normally use this feature to extend the catenary curve to cover the whole distance from tower to tower. You can snap the start location to an end point of an incoming wire or to a tower string vertex. Similarly, you can snap the end location on to a start point of an outgoing wire or to a tower string vertex.

In order to improve the accuracy of the catenary curve, you can apply least squares fitting. The manually entered three curvature points define the initial shape of the catenary. The application will then search all points which are within a tolerance distance from the manually entered curve and use those points in the fitting process.

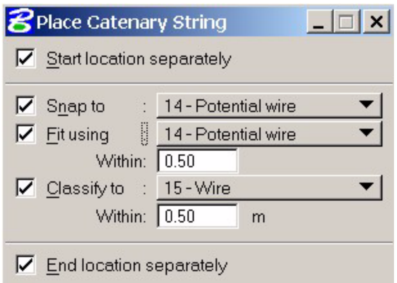
➢ **To prepare for catenary placement:**

1. Use *Draw Vertical Section* tool to create a longitudinal section along the wire from tower to tower.
2. Set active color and active level in the design file to place the catenary on a dedicated level.

➢ **To place a catenary curve manually:**

1. Select the *Place Catenary String* tool.

   This opens the **Place Catenary String** dialog:



2. Select settings.

Setting:	Effect:
Start location separately	If on, first point defines the start location of the catenary.
Snap to	If on, adjust the three curvature points to the coordinates of closest laser points of a given class. This lock is normally on.
Fit using	If on, the catenary curve will be fitted to laser points of the given class within the defined tolerance.
Classify to	If on, classify points to a given class. Affects points within the defined tolerance from the final curve.
End location separately	If on, last point defines the end location of the catenary.

3. (Optional) Enter a start location point at the start tower.
4. Enter the first curvature point.
5. Enter the second curvature point.
6. Enter the third curvature point.

7. (Optional) Enter an end location point at the end tower.

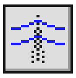
   The application draws the catenary curve defined by the given points.

8. Accept the catenary curve.

   The catenary curve is added to the design file. You can continue to step 3.

Check Catenary Attachments

Not Lite

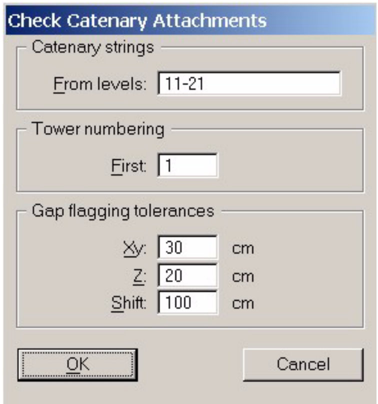


Check Catenary Attachments tool validates and adjusts catenary curves. It checks the gaps between catenary curves at tower locations. Because each catenary curve has been computed using laser points from one tower to tower span only, the incoming curve and the outgoing curve do not meet exactly. The magnitude of the gap gives some indication of how accurately the catenaries have been detected or placed.

➢ **To validate catenary attachment points:**

1. Activate a tower string element using *Activate Powerline* tool.
2. Select the *Check Catenary Attachments* tool.

   This opens the **Check Catenary Attachments** dialog:

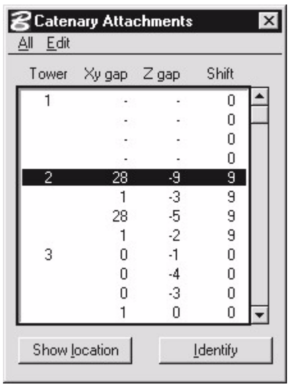


3. Enter setting values and click OK.

   The application searches for catenaries from given levels, finds meeting line strings and computes gaps between the incoming and the outgoing catenaries. The application then opens the **Catenary Attachments** window which displays a list of the attachment points.

Setting:	Effect:
From levels	List of levels from which to search catenaries. For example: • 50 - level 50. • 15,21-24 - levels 15,21,22,23 and 24.
First	Number of the tower at the first vertex of the activated tower string.
Xy	Xy gap flagging limit. Gaps exceeding this value will be drawn in red color in the list.
Z	Z gap flagging limit. Gaps exceeding this value will be drawn in red color in the list.
Shift	Tower shift flagging limit. Gaps exceeding this value will be drawn in red color in the list.

The **Catenary Attachment** window displays a list where each row corresponds to an attachment point or to a catenary end point for which no matching catenary has been found. Each row displays a horizontal gap, an elevation gap and a tower shift in centimeters at the attachment point.



You should now scan through the list and inspect every location where the gap values are unacceptable. If you have given reasonable error flagging tolerances, you will see the large gap values displayed in red. After viewing the problem location, you may choose to improve the catenary curves by using one of the adjustment tools from the menu.

Viewing attachment points

Usually the fastest way to inspect the attachment points is to scroll down the list and stop at each large gap value. You can then use the **Show location** button to view the location from above and the *Draw Vertical Section* tool to see a longitudinal section from the problem location.

➢ **To view attachment point locations:**

1. Select the desired attachment point row in the list.
2. Click on the **Show location** button.
3. Click in a top like view in which you want to see the attachment point location.

   The selected view is centred using the attachment point location.

Improving catenary attachments

The **Catenary Attachments** window offers several menu commands for improving catenary curves.

To:	Choose command:
Shift all tower positions.	**All / Shift all**
Adjust all attachment points to the average.	**All / Adjust all**
Recreate the list by scanning design file elements.	**All / Update list**
View statistics about gaps.	**All / Statistics**
Shift a single tower position.	**Edit / Shift tower**
Set two meeting catenary end points to a given xy location.	**Edit / Set attachment xy**
Adjust a single tower position to the average.	**Edit / Adjust attachment**
Manually enter the location of a catenary end point.	**Edit / Move catenary end**

When you are done improving the catenaries, you can choose **Adjust all** tool from **All** menu to adjust all attachment points to the average of the catenary end points.

All / Shift all

Shift all menu command modifies attachment points as if shifting tower positions. Shifting can be done only at towers where:

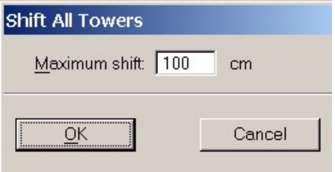
- all incoming wires are higher than corresponding outgoing wires. Shifting attachment points towards the previous tower would lower end points of incoming wires and raise start points of outgoing wires.
- all incoming wires are lower than corresponding outgoing wires. Shifting attachment points towards the next tower would raise end points of incoming wires and lower start points of outgoing wires.

In both cases it is probable that the manually placed tower vertex is not at the correct location.

➢ **To shift all towers:**

1. Select **Shift all** command from **All** menu.

   This opens the **Shift All Towers** dialog:



2. Enter maximum shift distance and click OK.

   The application shifts attachment points at every tower where it is feasible (cases outlined above). If the computed shift distance for a tower exceeds **Maximum shift**, the catenary points will be shifted only **Maximum shift** distance.

   An information window shows the amount of shifted towers.

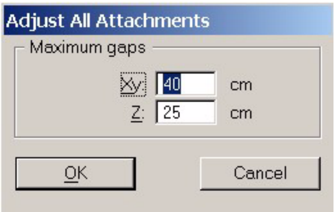
All / Adjust all

Adjust all menu command adjusts all attachment points to the average of the two catenary end points. You should use this command when you have completed inspection and manual improvement of attachment points.

➢ **To adjust all attachment points:**

1. Select **Adjust all** command from **All** menu.

   This opens the **Adjust All Attachments** dialog:



2. Enter maximum gaps for **Xy** and **Z** directions and click OK.

   The application will adjust all attachment points where both the horizontal gap and the vertical gap are within the defined limits. The end point of the incoming catenary and the start point of the outgoing catenary are adjusted to the average xyz position between the two.

All / Update list

Update list menu command re-scan the design file for catenary strings and updates the list. You should use this command whenever you deleted catenary string elements or generated new ones by manual placement or automatic detection, or after an action has been undone using MicroStation's *Undo* tool.

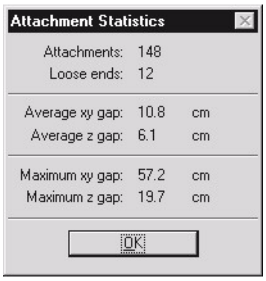
All / Statistics

Statistics menu command displays statistics about attachment points.

➢ **To view attachment point statistics:**

1. Select **Statistics** command from **All** menu.

   This opens the **Attachment Statistics** dialog:



2. View the statistics and click OK.

Value:	Meaning:
Attachments	Number of attachment points (places where two catenary curves meet approximately).
Loose ends	Number of catenary end points for which no matching catenary has been found. These places are: • at the start or at the end of the processed powerline • at a tower span where a catenary has not been detected or has been manually placed • at a tower where an incoming catenary string is too far from the outgoing one

Edit / Shift tower

Shift tower menu command shifts attachment points at a tower. This only works for towers where:

- all incoming wires are higher than corresponding outgoing wires. Shifting attachment points towards the previous tower would lower end points of incoming wires and raise start points of outgoing wires.
- all incoming wires are lower than corresponding outgoing wires. Shifting attachment points towards the next tower would raise end points of incoming wires and lower start points of outgoing wires.

In both cases it is probable that the manually placed tower vertex is not at the correct location.

➢ **To shift a single tower:**

1. Select **Shift tower** command from **Edit** menu.
2. Identify the tower to shift by clicking close to the tower.

Edit / Set attachment xy

Set attachment xy menu command moves a single attachment point to a given xy location.

➢ **To set xy location of an attachment point:**

1. Select **Set attachment xy** command from **Edit** menu.
2. Identify the attachment point to move by clicking close to it.
3. Enter the xy location with a mouse click.

   This moves the start point of the incoming catenary and the end point of the outgoing catenary to the given xy location. This will not modify the catenary constants or the elevation curves of the catenaries.

Edit / Adjust attachment

Adjust attachment menu command adjust a single attachment point to the average of the incoming and the outgoing catenaries.

➢ **To adjust a single attachment:**

1. Select **Adjust attachment** command from **Edit** menu.
2. Identify the attachment point to adjust by clicking close to it.

Edit / Move catenary end

Move catenary end menu command lets you move a catenary end point to a given location. You should use this tool when you can see that one of the catenaries matches laser points better than the other one.

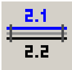
➢ **To move a catenary end or start point:**

1. Choose **Move catenary end** command from **Edit** menu.
2. Identify the start or end point to move by clicking close to it.
3. Enter the new location with a mouse click.

   The catenary curve is recomputed and redrawn.

Assign Wire Attributes

Not Lite

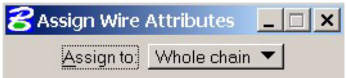


Assign Wire Attributes tool lets you define attributes and set the symbology for whole wires, a certain span range or a single catenary.

➤ **To assign attributes to wires:**

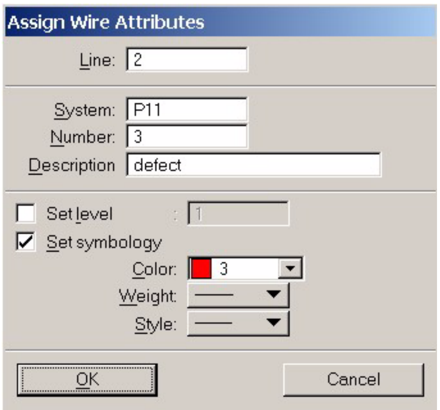
1. Activate a tower string element using *Activate Powerline* tool.
2. Select the *Assign Wire Attributes* tool.

   This opens the **Assign Wire Attributes** dialog:



3. Select, for what to assign attributes: **Whole chain**, **Span range** or **Single span**.
4. If **Assign to** is set to **Span range**, select a start tower and an end tower by clicking near to it to define the range.
5. Select a wire for which to assign attributes.

   This opens the **Assign Wire Attributes** dialog:



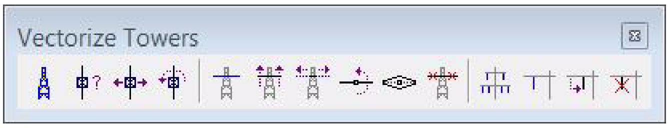
6. Define settings and click OK.

   This assigns the given attributes to the wire.

Setting:	Effect:
Line	Number of the powerline. This is filled automatically from the activated tower string.
System	Text field for entering a free system identifier.
Number	Text field for entering a free wire number.
Description	Text field for entering a free description for the wire.
Set level	If on, the selected wire is moved to the given design file level.
Set symbology	If on, the given color, weight and line style is applied for the selected wire.

Vectorize Towers tool box

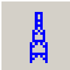
The tools in the **Vectorize Towers** tool box are used to place towers and to manipulate towers, cross arms and attachments.



To:	Use:
Place a tower	 *Place Tower*
Edit tower attributes	 *Edit Tower Information*
Move a tower to another location	 *Move Tower*
Rotate a tower around its base point	 *Rotate Tower*
Add a cross arm to a tower	 *Add Cross Arm*
Set the height of a cross arm	 *Set Cross Arm Elevation*
Set the length of a cross arm	 *Extend Cross Arm*
Rotate a cross arm around the tower	 *Rotate Cross Arm*
Modify a cross arm	 *Modify Cross Arm*
Delete a cross arm	 *Delete Cross Arm*
Create attachments automatically	 *Create Attachments*
Add an attachment manually to a cross arm	 *Add Attachment*
Move an attachment along the cross arm	 *Move Attachment*
Delete an attachment	 *Delete Attachment*

Place Tower

Not Lite



Place Tower tool lets you place a tower. The appearance of the tower has to be defined in **Powerlines / Tower types** in TerraScan Settings before. Each tower is created as MicroStation cell element consisting of a tower body and cross arms as simple lines.

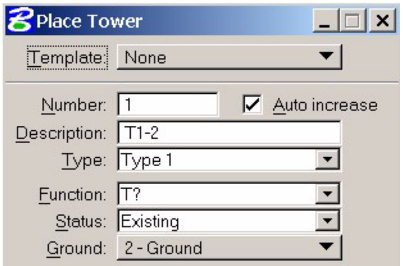
Towers are placed manually based on the tower type definition, an activated tower string and the laser points. Tower vectorization can be supported by view arrangement and display options. One MicroStation view should be defined as top view, another one as section view showing the same tower completely. Set the display of laser points in a way that shows the towers clearly in both views. The *View Tower Spans* tool with settings for viewing tower positions can be helpful to place towers along a powerline in a structured way.

Additionally, a tower template can be defined from an already vectorized tower. If towers are placed using a template, the locations, lengths and shapes of the cross arms are fixed. This might be useful if more complex cross arms have been created manually using the *Add Cross Arm* tool. See **Creating a tower template** on page 134 for more information about how to define a tower template.

➢ **To place a tower:**

1. Activate a tower string element using *Activate Powerline* tool.
2. Select the *Place Tower* tool.

   This opens the **Place Tower** dialog:



3. Enter settings.

Setting:	Effect:
Template	Use of a template for tower placement: • None - no template is used. • Identify - the tower which is selected with the next mouse click is set as active template for further towers. • Active - towers are placed using the active template.
Number	Tower number.
Auto increase	If on, numbers for towers increase automatically for each placed tower.
Description	Text field for entering a free description for the tower.
Type	Type of the tower as defined in TerraScan settings in the **Description** field for a tower type.
Function	Function of the tower defined in TerraScan settings.
Status	Status of the tower defined in TerraScan settings.
Ground	Ground class in laser data to define the base point of the tower.

4. Move the mouse into the top view.

   A line is displayed at the location of the tower that is defined by a vertex of the tower string.
5. Define the base point of the tower with a mouse click in the top view.

The elevation of the base point is set automatically according to the ground laser points.

6. Move the mouse in the section view.

   A line in the center of the tower shows its location.

7. Define the height by a mouse click on the top of the tower as it can be seen in the laser points.

   Next, the software waits for the placement of the cross arms according to the tower type definition.

8. Place the end point of the first cross arm as it can be seen in the laser points in the section view.

   If a template is used for placing the tower, the location and length of the cross arms is defined by the template. Nevertheless, the software waits for the mouse clicks to define the cross arms. Click somewhere in the section view to confirm the cross arm location.

9. Place the end points of all remaining cross arms in the same way.

   After defining the end point of the last cross arm, the placement of the tower is finished. If **Auto increase** is on in the **Place Tower** dialog, the number is increased. You may enter new settings for the next tower and continue with step 4.

✍ While placing one tower, single steps can be redone using the right mouse button.

Creating a tower template

A tower template can be created to place towers with a more complex shape as it can be defined in TerraScan settings. The template defines the location, length and shape of the cross arms of the tower.

➢ **To create a tower template:**

1. Vectorize one tower using *Place Tower*, *Add Cross Arm* or other **Vectorize Towers** tools.
2. Select the *Place Tower* tool.
3. Set **Template** to **Identify**.
4. Select the vectorized tower.

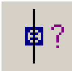
   This sets the selected tower as active template.

You can continue with placing the first tower using the template.

✍ The identification of a template does not influence the tower attributes set in the **Place Tower** dialog.

Edit Tower Information

Not Lite

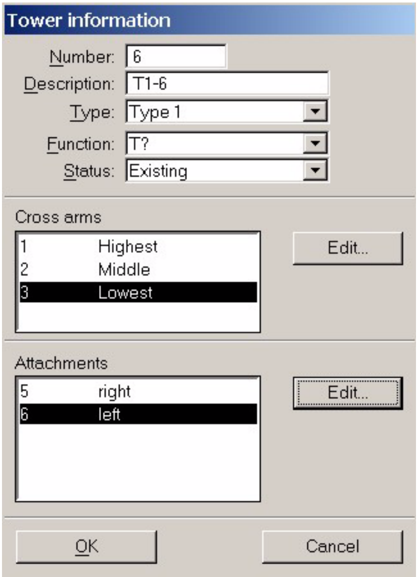


Edit Tower Information tool lets you edit the attributes for a tower. This includes information about tower number, description, type, status and function as well as cross arm and attachment attributes.

➤ **To edit tower information:**

1. Select the *Edit Tower Information* tool.
2. Select a tower for which to edit the attributes.

   This opens the **Tower information** dialog:

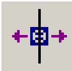


3. Edit tower attributes as desired.
4. To edit the attributes for cross arms, select the line of the cross arm and click the **Edit** button.
5. To edit the attributes for attachments, select the line of an cross arm and of the attachment and click the **Edit** button.
6. Click OK to set the new attributes for the tower.

Setting:	**Effect:**
Number	Tower number.
Description	Text field for entering a free description for the tower.
Type	Type of the tower as defined in TerraScan settings in the **Description** field for a tower type.
Function	Function of the tower defined in TerraScan settings.
Status	Status of the tower defined in TerraScan settings.
Cross arms	Cross arms defined for the tower. Click **Edit** to change cross arm **Number** and **Description**.
Attachments	Attachments defined for the selected cross arm. Click **Edit** to change attachment **Number** and **Description**.

Move Tower

Not Lite



Move Tower tool changes the xy location of a tower. This can be used to correct the location if the placement of the tower string vertex was not accurate enough in the center of the tower. Actually, the tool moves the vertex of the tower string and a vectorized tower if one is placed at the location of the vertex.

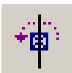
➢ **To move a tower:**

1. Activate a tower string element using *Activate Powerline* tool.
2. Select the *Move Tower* tool.
3. Identify the tower to be moved.
4. Define the new location of the tower with a mouse click.

✍ *Move Tower* tool should not be used after attachments have been placed to prevent an incorrect replacement of attachments and wires.

Rotate Tower

Not Lite



Rotate Tower tool changes the horizontal direction of the tower. This can be used to correct the direction of a tower if the direction of the tower string to the next tower was not accurate enough.

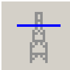
➢ **To rotate a tower:**

1. Activate a tower string element using *Activate Powerline* tool.
2. Select the *Rotate Tower* tool.
3. Identify the tower to be rotated.
4. Define the new direction of the tower with a mouse click.

✍ *Rotate Tower* tool should not be used after attachments have been placed to prevent an incorrect replacement of attachments and wires.

Add Cross Arm

Not Lite

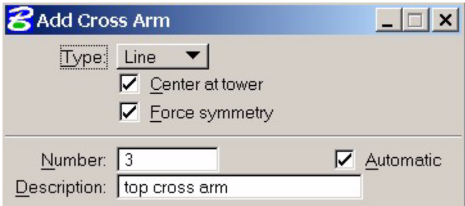


Add Cross Arm tool lets you add a cross arm to a tower. The cross arm can be added as simple line or as shape. This is a way to create more complex cross arms than can be defined in TerraScan **Settings** for a tower type.

➢ **To add a cross arm to a tower:**

1. Select the *Add Cross Arm* tool.

   This opens the **Add Cross Arm** dialog:



2. Define settings.

Setting:	Effect:
Type	Element type of the cross arm: **Line** or **Shape**.
Center at tower	If on, a cross arm is centered at the tower center point.
Force symmetry	If on, a symmetric cross arm is forced. Cross arm shape on the left side of the tower will be a mirror image of the shape on the right side of the tower.
Number	Number of the cross arm.
Automatic	If on, the number is set automatically based on the already existing number of cross arms for the tower.
Description	Text field for entering a free description for the cross arm.

3. Select a tower for which to add a cross arm.
4. Define the height of the cross arm in a section view. A line is shown at the mouse location to indicate the location of the cross arm.
5. Enter the first vertex of the cross arm.
6. Enter the second vertex of the cross arm.

   This finishes the definition of a cross arm of the type **Line**.

7. Enter next vertices for a cross arm of the type **Shape**.

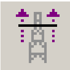
   When the mouse comes close to the first vertex, it snaps to the vertex to close the shape. If **Force symmetry** is switched on, this requires an even amount of already defined shape vertices.

8. Entering the last vertex closes the shape and finishes the creation of a cross arm of the type **Shape**.

✍ When digitizing a cross arm, vertices can be added in a top view as well as in a section view. If entered in a top view, the elevation of the vertex is set by the cross arm height. If entered in a section view, the xy location of the vertex is defined by the section line of the current section.

Set Cross Arm Elevation

Not Lite



Set Cross Arm Elevation tool changes the elevation of a cross arm. This can be used to set a cross arm to a more accurate height as it can be seen in the laser data.

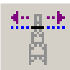
➢ **To set the elevation for a cross arm:**

1. Select the *Set Cross Arm Elevation* tool.
2. Select a cross arm for which to set a new height.
3. Define a new height by a mouse click in a section view.

   This places the cross arm at the new height.

Extend Cross Arm

Not Lite



Extend Cross Arm tool changes the length of a cross arm either on both sides simultaneously or only on one side.

➢ **To extend a cross arm:**

1. Select the *Extend Cross Arm* tool.

   This opens the **Extend Cross Arm** dialog:

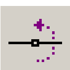
2. Select whether to extend **Both sides** or only **One side** in the **Extend** field.
3. Select the cross arm to be extended with a mouse click near the end point.

   The new extend of the cross arm is indicated by a line when the mouse is moved.

4. Enter a new end point for the cross arm.

Rotate Cross Arm

Not Lite



Rotate Cross Arm tool rotates a cross arm around the center point of the tower.

➢ **To rotate a cross arm:**

1. Select the *Rotate Cross Arm* tool.
2. Select the cross arm to be rotated.

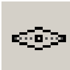
   The new direction is indicated by a line when the mouse is moved.

3. Enter a new direction for the cross arm.

✍ *Rotate Cross Arm* tool should not be used after attachments have been placed to prevent an incorrect replacement of attachments and wires.

Modify Cross Arm

Not Lite

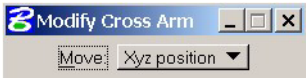


Modify Cross Arm tool modifies a cross arm by moving single vertices of the line or shape. Modification can be made to either the elevation, the xy location or both.

➢ **To modify a cross arm:**

1. Select the *Modify Cross Arm* tool.

   This opens the **Modify Cross Arm** dialog:



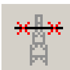
2. Select whether to change the **Elevation**, the **Xy position** or the **Xyz position** of a vertex.
3. Select a vertex with a mouse click near to it.

   The new vertex position is indicated when the mouse is moved.

4. Enter a new position for the vertex.

Delete Cross Arm

Not Lite



Delete Cross Arm tool deletes a cross arm.

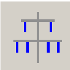
➢ **To delete a cross arm:**

1. Select the *Delete Cross Arm* tool.
2. Select the cross arm to be deleted.
3. Accept the deleted cross arm with a mouse click.

✍ If attachments have been created for a cross arm, they are deleted as well.

Create Attachments

Not Lite



Create Attachments tool creates attachments for a tower. Attachments connect vectorized wires at the end points or meeting points of catenary elements with the cross arms of a tower.

Before the creation of attachments the vectorization of the tower, cross arms and wires finished and checked.

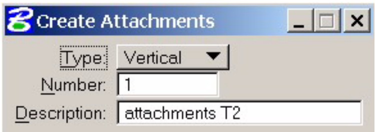
The attachments are created automatically for each wire if a connection to a cross arm can be created. The connection can be either a vertical line or a slope line in forward or backward direction along the tower string. If a vertical line is created the location of the end or meeting point of the catenary elements is moved to the attachment's end point to enable a vertical connection line. A slope line connects the end or meeting points of catenary elements on their original location with the cross arm.

If there is no attachment created at some location, the software can not create a linear connection in vertical or forward/backward direction between the catenaries and the cross arms.

➢ **To create attachments for a tower:**

1. Activate a tower string element using *Activate Powerline* tool.
2. Select the *Create Attachments* tool.

   This opens the **Create Attachments** dialog:



3. Define settings.

Setting:	Effect:
Type	Direction of the attachment: **Vertical** or **Slope 3D**.
Number	Number of the first attachment within that tower.
Description	Text field for entering a free description for all attachments of one tower.

4. Select a tower for which to create attachments.

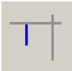
   The attachments that can be created are shown as preview lines.

5. Check attachment lines in a section view.
6. Accept attachments with a mouse click.

   This creates the attachment lines as part of the tower. The creation can be rejected be clicking the right mouse button.

Add Attachment

Not Lite



Add Attachment tool lets you add an attachment manually to a cross arm. Besides the options offered for automatic creation of attachments there are two more possibilities: **Side slope** and **Dual point**. Side slope creates a sloped line in left or right direction from the tower string. Dual point attachments can connect two catenary elements that are not joined.

➢ **To add an attachment to a cross arm:**

1. Select the *Add Attachment* tool.

   This opens the **Add Attachment** dialog:

2. Define settings.

Setting:	Effect:
Type	Direction of the attachment: **Vertical**, **Side slope**, **Slope 3D** or **Dual point**.
Number	Number of the attachment.
Automatic	If on, the number is set automatically based on the already existing number of attachments for the tower.
Description	Text field for entering a free description for the attachment.

3. Select a cross arm for which to add an attachment.
4. Identify the wire or the end point of the first catenary element which to connect with the cross arm.

   This finishes the creation of a **Vertical** attachment.

5. Enter a location on the cross arm where the attachment is placed.

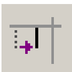
   This finishes the creation of a **Side slope** or a **Slope 3D** attachment.

6. Identify the end point of the second catenary element to connect it with the attachment.

   This finishes the creation of a **Dual point** attachment.

Move Attachment

Not Lite

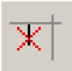


Move Attachment tool moves an attachment along the cross arm.

This is not yet implemented in the software.

Delete Attachment

Not Lite



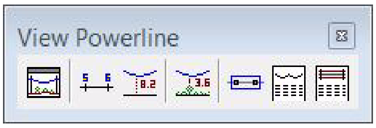
Delete Attachment tool deletes an attachment.

➢ **To delete an attachment:**

1. Select the *Delete Attachment* tool.
2. Select the attachment to be deleted.
3. Accept the deleted attachment with a mouse click.

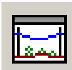
View Powerline

The tools in **View Powerline** tool box are used to view tower spans, to label towers and heights from ground to catenaries, to find danger objects, to create span tiles and to output catenaries and towers into text files.



To:	Use:
View tower spans as profiles and cross sections	*View Tower Spans*
Label tower string with tower numbers	*Label Towers*
Label height from ground to catenary	*Label Catenary Height*
Find points close to the catenaries	*Find Danger Objects*
Create tiles rectangles for powerline spans	*Create Span Tiles*
Output catenary coordinates to text files	*Output Catenary*
Export powerline information to text file	*Export Powerline*

View Tower Spans



View Tower Spans tool makes it easy to traverse through a powerline and view cross sections of towers, profiles of tower spans and top views of either.

Additionally, top like and oblique camera views of towers can be displayed showing the laser data on top of aerial images. This requires TerraPhoto running and the availability of images organized in a TerraPhoto image list. Camera views might be useful in addition to other display options for towers and spans to support classification tasks.

This tool manages the automatic update of certain views as you scroll through a list of towers. Following view types can be automatically updated:

- **Span top** - a top like view showing one tower span.
- **Span profile** - a profile view showing a longitudinal profile along a tower span.
- **Tower top** - a top like view at the start tower.
- **Tower rotated -** a top like view from the start tower rotated into longitudinal direction of the powerline.
- **Tower section** - a section view from the start tower.
- **Tower profile** - a longitudinal section view from the start tower.
- **Camera top -** image and laser data in a top like view showing the start tower.
- **Camera oblique 1 -** image and laser data in an oblique view in backward direction from the start tower along the powerline.
- **Camera oblique 2 -** image and laser data in an oblique view in forward direction from the start tower along the powerline.

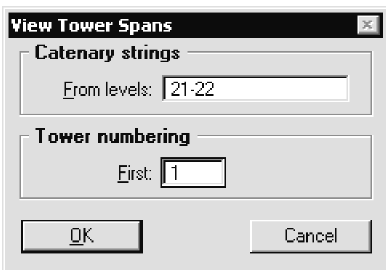
You might use this tool at various stages of powerline processing. For example:

- When manually classifying tree hits which are very close to the wires. You should use **Span top** and **Span profile** views. You can classify vegetation hits using **Below curve** menu command.
- When validating the automatically detected wires. You should use **Span top** and **Span profile** views.
- When classifying towers. You should use **Tower top**, **Tower rotated**, **Tower section** and **Tower profile** views.

➢ **To view tower spans:**

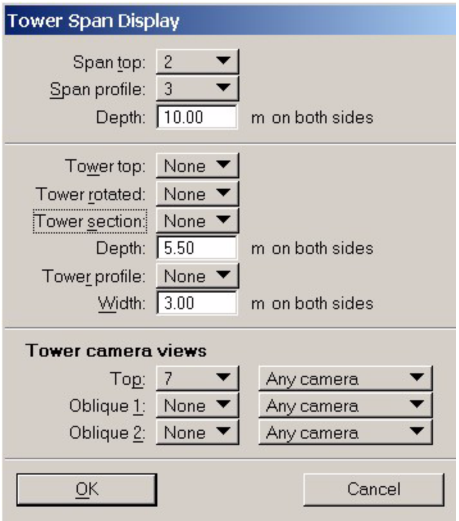
1. Activate a tower string element using the *Activate Powerline* tool.
2. Select the *View Tower Spans* tool.

   This opens the **View Tower Spans** dialog:



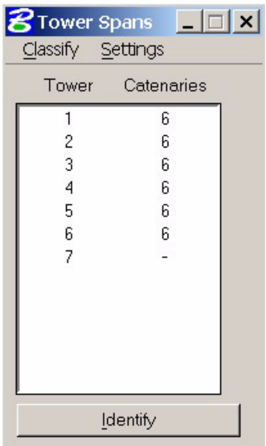
3. Enter a level list to define what levels should be searched for wire elements.
4. Enter a number for the first tower. This is only for display in the list of towers for this tool.
5. Click OK.

This opens the **Tower Span Display** dialog:



6. Select a view to be used as **Span top** view and another view as **Span profile** view.
7. Set a value for **Depth**. This determines the depth of the profile view. It also defines the corridor width within which points are classified using **Above curve** or **Below curve** commands.
8. (Optional) Set **Tower top**, **Tower rotated, Tower section** and **Tower profile** to **None** if you do not want to see views from tower locations.
9. Click OK.

   This opens the **Tower Spans** dialog:



   The list box in the dialog contains one row for each tower span. The tower number is for the start tower of the span and the number of catenaries indicates catenaries starting from this tower.

10. Make sure the views you selected as **Span top** and **Span profile** are open.
11. Select one row at a time in the list.

    This updates the selected views.

➢ **To view tower positions:**

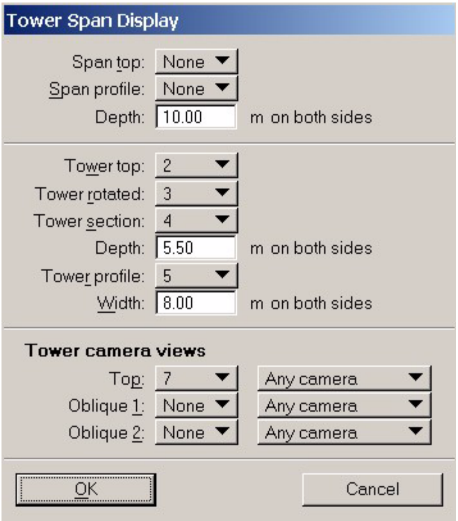
1. Activate a tower string element using the *Activate Powerline* tool.
2. Select the *View Tower Spans* tool.

   This opens the **View Tower Spans** dialog.

3. Enter a level list to define what levels should be searched for wire elements.
4. Enter a number for the first tower. This is only for display in the list of towers for this tool.

5. Click OK.

   This opens the **Tower Span Display** dialog:

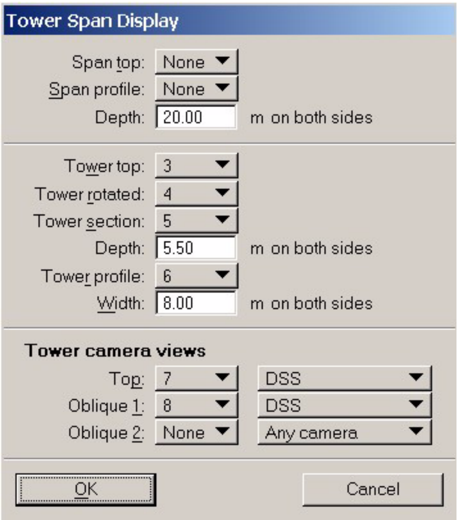


6. (Optional) Set **Span top** and **Span profile** to **None** if you do not want to see views from tower spans.
7. Select views to be used as **Tower top**, **Tower rotated**, **Tower section** and **Tower profile** views.
8. Set a values for **Depth** and **Width**. This determines the depth of the section and profile views.
9. Click OK.
10. Make sure the views you selected in step 7 are open.
11. Select one row at a time in the list in the **Tower Spans** dialog.

    This updates the selected views.

➢ **To view tower camera views:**

1. Open the **Tower Span Display** dialog as described above:



2. Select views to be used as **Top**, **Oblique 1** and **Oblique 2** views.
3. Select a camera from which images are used in the views.
4. Click OK.
5. Make sure the views you selected in step 6 are open.

Above curve

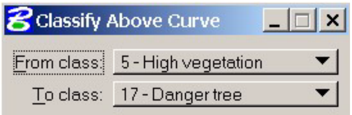
Above curve menu command from **Classify** menu in the **Tower Spans** dialog classifies points above a given curve and within a corridor defined by the **Depth** value for **Span profile** in the **Tower Span Display** dialog. The curve is defined by three points set by mouse clicks.

The command is only active, when display settings for **Span profile** are defined in the **Tower Span Display** dialog.

➢ **To classify points above a curve:**

1. Select **Above curve** command from **Classify** pulldown menu.

   This opens the **Classify Above Curve** dialog:



2. Select a source class in the **From class** field from which to classify points.
3. Select a target class in the **To class** field into which to classify points.
4. Enter the first, second and third point of the curve in the profile view.

   After the third point is placed, the points above the curve and within the corridor defined by the profile depth value are classified from the source class to the target class.

Below curve

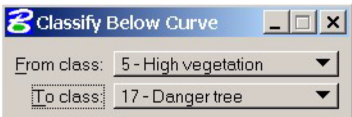
Below curve menu command from **Classify** menu in the **Tower Spans** dialog classifies points below a given curve and within a corridor defined by the **Depth** value for **Span profile** in the **Tower Span Display** dialog. The curve is defined by three points set by mouse clicks.

The command is only active, when display settings for **Span profile** are defined in the **Tower Span Display** dialog.

➢ **To classify points below a curve:**

1. Select **Below curve** command from **Classify** pulldown menu.

   This opens the **Classify Below Curve** dialog:



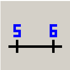
2. Select a source class in the **From class** field from which to classify points.
3. Select a target class in the **To class** field into which to classify points.
4. Enter the first, second and third point of the curve in the profile view.

After the third point is placed, the points below the curve and within the corridor defined by the profile depth value are classified from the source class to the target class.

Settings / Display

Display menu command from **Settings** menu in the **Tower Spans** dialog opens the **Tower Span Display** dialog. The settings can be changed and become active when the dialog is closed with OK.

Label Towers



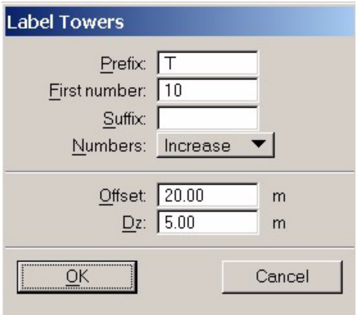
Label Towers tool places a label for towers in the design file. The text is placed as cell element and is always oriented to the viewer no matter how the view is rotated.

The location of the label is determined by the tower string element and offset values from the tower string vertices. Positive offset values place the text on the right side and above of the tower string, negative values on the left side and below. Active text size settings and symbology settings of the design file are used for placing the text.

To place labels for towers:

1. Activate a tower string element using the *Activate Powerline* tool.
2. Define settings for texts using MicroStation’s *Text* tools.
3. Select the *Label Towers* tool.

   This opens the **Label Towers** dialog:



4. Define settings and click OK.

   This places the labels as text cells into the design file.

Setting:	Effect:
Prefix	Free text that is added in front of the tower number.
First number	Number of the first tower.
Suffix	Free text that is added behind the tower number.
Numbers	Method of numbering the towers: **Increase** or **Decrease**.
Offset	Horizontal offset of the label from the tower location. Measured between the tower string vertex and the center point of the text.
Dz	Vertical offset of the label. Measured between the tower string vertex and the center point of the text.

Label Catenary Height

Label Catenary Height tool labels the minimum height from a catenary curve to a point class such as ground.

It searches all the points which are within a given offset limit from the line of the catenary. The minimum height difference from a point to the catenary is labeled with a text element and a vertical line using active text settings and symbology. The text is placed vertically next to the line.

➢ **To label height from catenary to ground:**

1. Select the *Label Catenary Height* tool.

   This opens the **Label Catenary Height** dialog:

2. Select ground class in the **From class** option button.
3. Identify a catenary string element.
4. Accept the catenary string to be labeled.

   The minimum height difference is labeled. You can continue with step 3.

Setting:	**Effect:**
From class	Class to compute the minimum distance of a catenary curve to. Normally set to ground class.
Within	Maximum search offset on both sides of catenary line.
Accuracy	Number of decimals for the measured distance placed as text.

Find Danger Objects

Find Danger Objects tool finds laser points which are within a given three dimensional distance limit from vectorized catenaries.

There are three different methods how danger points can be defined:

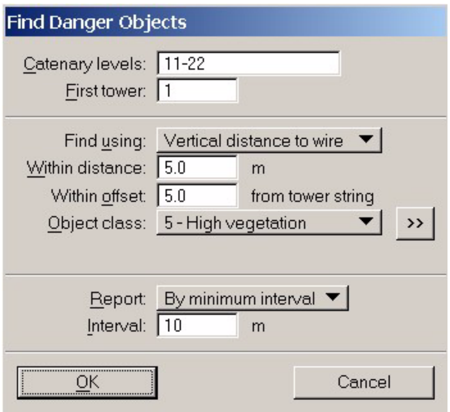
- **Vertical distance to wire -** danger points are searched within a 3D radius around each wire and within a vertical distance from the wire.
- **3D distance to wire -** danger points are searched within a 3D radius around each wire.
- **Falling tree logic -** each vegetation point is considered as a falling tree and classified if the point is travels too close to the wire when falling down as a tree. The falling tree computation treats each vegetation point as a tip of a tree with the trunk at the xy location of that vegetation point.

The tool produces a danger point instance list from all of the locations where there is a point too close to the wires. The list is traversable: you can scroll through the list and check each location.

➢ **To find danger points:**

1. Activate the tower string element for the powerline that you want to process using the *Activate Powerline* tool.
2. Select the *Find Danger Objects* tool.

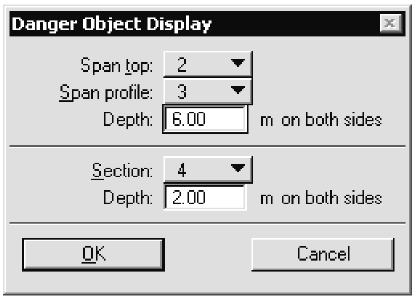
   This opens the **Find Danger Objects** dialog:



Setting:	Effect:
Catenary levels	List of levels from which to search catenaries.
First tower	Number of the tower at first vertex in the selected tower string.
Find using	Method of wire danger point definition: **Vertical distance to wire**, **3D distance to wire** or **Falling tree logic**.
Within distance	3D radius around a wire used as search distance for danger points.
Within offset	Vertical distance from a wire used in **Vertical distance to wire** and **Falling tree logic** methods to search for danger points.
Object class	Point class from which to search for danger objects.
Ground class	Class into which laser points on ground have been classified. This is only active for **Falling tree logic** method.
Report	Reporting frequency: • All - each laser point is reported. This is useful when you want to classify danger points into another laser point class. • One for every span - each wire can have only one report row for one tower span. The reported point is the closest. • By minimum interval - the closest point for each wire within a distance **Interval** is reported.

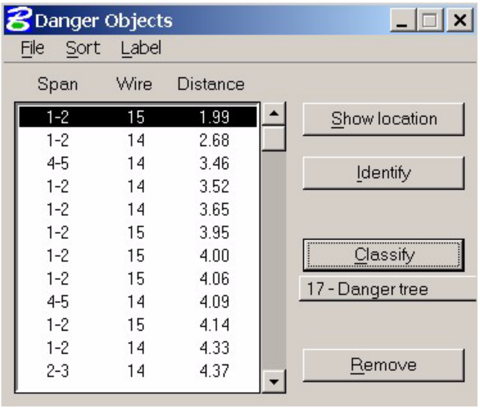
3. Enter settings and click OK.

   The application searches for danger point. When this is complete, the **Danger Object Display** dialog opens:



4. Select views to be used as **Span top**, **Span profile** and **Section** views.
5. Set values for **Depth** fields. This determines the depth of the profile and section views.
6. Click OK.

   The **Danger Objects** dialog is now visible:



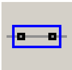
   The list contains a report row for locations where a laser point was close to a wire element. If you click on a row, the application automatically updates selected views to display that danger location.

   The **Danger Objects** window offers several menu and button commands for viewing, classifying, labeling and reporting danger locations.

To:	Choose command:
Output danger locations to a text file	File / Output report
Classify all points in the report into another class	File / Classify all
Change view settings for automatic display update	File / Display settings
Label the selected point in 3D position	Label / In 3d
Label the selected point in a profile drawing	Label / In profile
Label all report points in 3D	Label / All in 3d
Label all report points in a profile drawing	Label / All in profile
Show the location of the selected point in a view	Show location
Identify danger points at a certain location	Identify
Classify the selected point to another class	Classify
Remove the selected row from the report	Remove

✍ The classification of wire danger points can be done automatically using **Wire danger points** routine in a macro.

Create Span Tiles



Create Span Tiles tool creates rectangles for each span. These rectangles can be used for example as tiles for creating orthophotos with TerraPhoto where each resulting image covers one tower-to-tower distance.

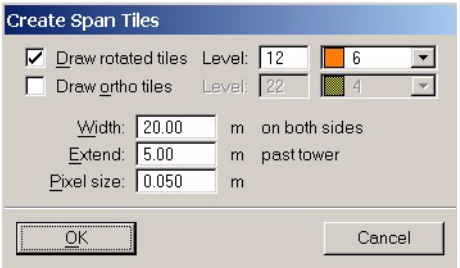
There are two different types of tile rectangles that can be produced:

- **Rotated tiles -** the long tile sides are parallel to the tower string.
- **Ortho tiles -** tiles are drawn as orthogonal bounding box around a span.

➢ **To create span tiles:**

1. Activate the tower string element using the *Activate Powerline* tool.
2. Select the *Create Span Tiles* tool.

   This opens the **Create Span Tiles** dialog:



3. Define settings and click OK.

   This draws the tiles into the design file.

Setting:	Effect:
Draw rotated tiles	If on, rotated tile parallel to the tower string are drawn on the defined Level with the given color.
Draw ortho tiles	If on, orthogonal tiles are drawn around each span on the defined Level with the given color.
Width	Width of the tile measured from the tower string perpendicular to the tile boundary for rotated tiles and from the tower string vertex to the tile boundary for ortho tiles.
Extend	Distance from a tower about which the span’s tower string part is extended before the rectangle tile for the span is created.
Pixel size	Intended pixel size in meters to be used in image rectification. See TerraPhoto documentation for more information.

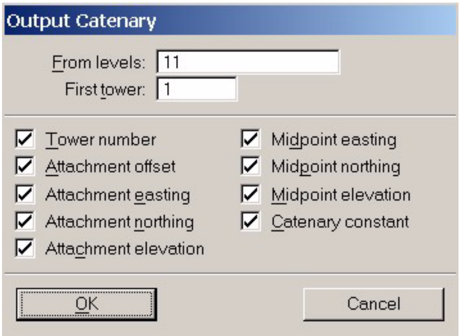
Output Catenary

Output Catenary tool creates a report for the catenaries of an active powerline and on certain design file levels. From a list of all catenaries, a text file can be created for each catenary.

➢ **To output a catenary as text file:**

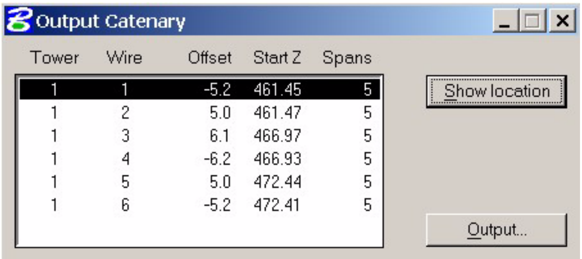
1. Activate the tower string element using the *Activate Powerline* tool.
2. Select the *Output Catenary* tool.

   This opens the **Output Catenary** dialog:



3. Enter design file level(s) from which catenaries are searched for output.
4. Enter a number for the first tower.
5. Select attributes to be included in the report.
6. Click OK.

   The **Output Catenary** dialog opens:



   The list shows rows for each catenary with the number of the start tower, the number of the wire, the attachment offset and elevation at the start tower and the amount of spans that include this catenary.

   The location of a catenary can be shown in a view by selecting the row in the list and using the **Show location** button.

7. Select a catenary row in the list for which to create a text file.
8. Click **Output**.

   This opens a standard Windows dialog to save a file.

9. Select a directory and type a file name for the output file.
10. Click **Save**.

    This saves the catenary into a text file that contains a structured list with a column for each selected attribute. An information dialog informs about the success of the action.

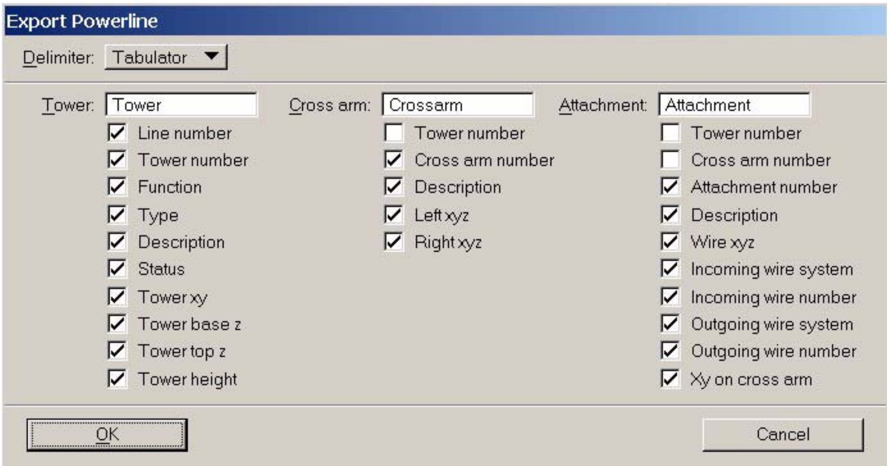
Export Powerline

Export Powerline tool creates a report for the towers of an active powerline. The report is saved as text file.

➢ **To output towers as text file:**

1. Activate the tower string element using the *Activate Powerline* tool.
2. Select the *Export Powerline* tool.

   This opens the **Export Powerline** dialog:



3. Select a delimiter used to separate columns in the text file from each other.
4. Enter texts in the **Tower**, **Cross arm** and **Attachment** fields as to be used in the report.
5. Select attributes to be included in the report.
6. Click OK.

   This opens a standard Windows dialog to save a file.

7. Select a directory and type a file name for the output file.
8. Click **Save**.

   This saves the towers of the powerline into a text file that contains a structured list with a column for each selected attribute. An information dialog informs about the success of the action.

The text file created has the following structure:

Tower	**1**	**1**	**A**	**T1**		**Bestand**	**10321.12**	**735921.11**	**441.12**	**477.34**	**36.22**
Crossarm	***1***		***10321.04***	***735926.33***	***474.88***	***10321.2***	***735915.9***	***474.88***			
Attachment	2		10321.04	735926.29	472.41					10321.04	735926.29
Attachment	5		10321.2	735916.07	472.44					10321.2	735916.07
Crossarm	***2***		***10321.03***	***735927.48***	***469.48***	***10321.22***	***735914.75***	***469.48***			
Attachment	1		10321.03	735927.35	466.93					10321.03	735927.35
Attachment	6		10321.22	735915.01	466.97					10321.22	735915.01
Crossarm	***3***		***10321.04***	***735926.49***	***464***	***10321.21***	***735915.74***	***464***			
Attachment	3		10321.04	735926.27	461.46					10321.04	735926.27
Attachment	4		10321.2	735916.07	461.47					10321.2	735916.07
Tower	**1**	**3**	**A**	**T1**		**Bestand**	**10673**	**735926.52**	**436.97**	**473.06**	**36.09**
Crossarm			***10672.92***	***735931.99***	***470.66***	***10673.08***	***735921.04***	***470.66***			
Attachment	3		10672.93	735931.48	468.09					10672.93	735931.48

The first row shows the tower attributes, followed by the first cross arm attributes in the next row. Then the attributes of the attachments belonging to the first cross arm are listed. This is done for all cross arms for the first tower. In the same way all other towers and tower parts are listed in the text file.

7 Waveform Processing

TerraScan is able to read waveform information from LAS 1.3 files and from TopEye .TEW 1.15 (MarkII) files. It uses the waveform information for processing tasks. It is not possible to write out files that include waveform information.

Waveform capabilities

If waveform data is available, you can perform the following processing steps:

- **View Waveform** for a point in a graph and export waveform information of a point into a text file.
- **Extract echo properties** for laser points:
  - **Echo length -** relative length (millimeter) of a return signal compared to a typical return from a hard surface.
  - **Echo normality -** difference in shape and position of a peak of a return signal compared to a typical return from a hard surface.
  - **Echo position**
- Classify laser points **By echo length**.
- **Extract Echoes** in problem areas using a specific echo extraction logic:
  - **Last possible -** for example in areas with dense low vegetation where the default extraction logic did not provide ground points.
  - **All possible** or **All distinct -** for example in places where points on some feature are missing, such as powerline wires.
  - **First possible**.

Waveform processing principles

If laser data with waveform information is imported into a TerraScan project, the block binary files must be saved as LAS 1.0, 1.1, or 1.2, or as Fast Binary files. Waveform-related attributes, such as echo length, echo normality, and echo position can only be stored in Fast Binary format.

The waveform files are linked to laser points via the trajectory files. The **Trajectory information** dialog contains an input field **Waveform** which defines the file used for reading waveform information. Once a laser point is assigned to a trajectory (by the line number) and the trajectory is linked to a waveform file, the software is able to find the waveform information for any laser point using the time stamp and the echo number stored for the laser point.

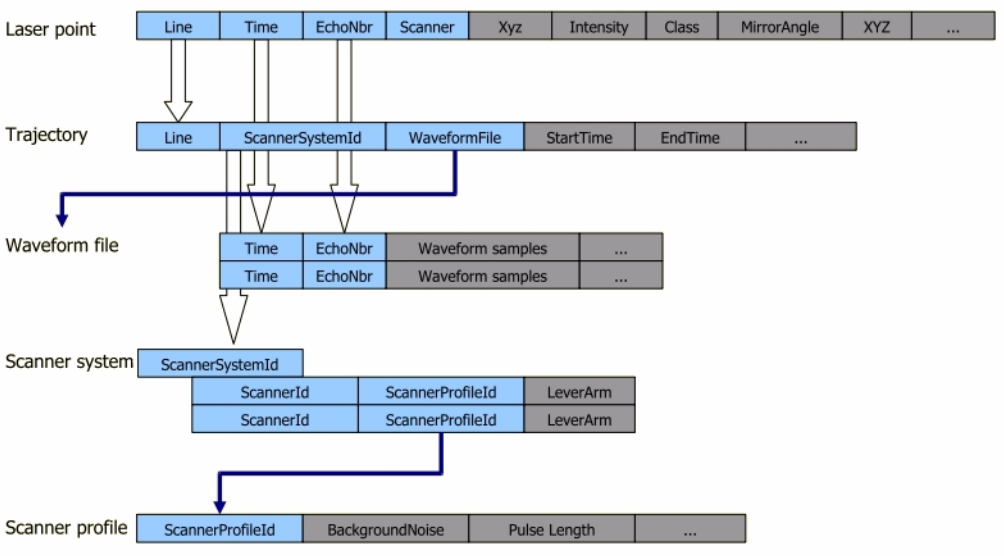
For the extraction of echo properties and of additional points, the software also needs a scanner waveform profile. The profile stores properties of typical returns from a single hard surface. These properties include:

- the background noise level
- the pulse length at 50% of peak strength
- the pulse length at 35% of peak strength
- the shape of the return pulse
- the system-derived point position relative to the return pulse

The scanner waveform profile can be extracted from laser point samples on hard, flat, open ground surfaces. There should be only-echo returns and some intensity variation within the sample area. The sample areas must not be located at the edges of scan lines. The scanner waveform profile is then automatically computed from the sample laser points.

Finally, the scanner waveform profile must be referenced by a scanner system definition which in turn must be linked to the trajectory files.

The following figure illustrates the method how TerraScan finds waveform information for a laser point.



Workflow summary

1. Load trajectories into TerraScan using the *Manage Trajectories* tool.
2. Link trajectories with waveform files using the **Edit information** or **Link to waveform files** commands of the **Trajectories** dialog.
3. Create a TerraScan project using the *Define Project* tool, storage format must be Fast binary for storing echo properties, or LAS.
4. Import points into the project using the **Import points into project** command of the **Project** dialog, deduce flightline numbers from trajectories.

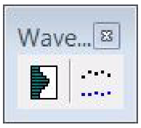
   This enables the display of waveform information using the *View Waveform* tool.
5. Draw polygons around sample areas of single, open, hard surfaces that contain only-echo returns and some variation in intensity values. Sample areas should not be too close to scan corridor edges.
6. Classify points inside the polygons into a separate class using **Inside fence** command or **Inside shapes** classification routine.
7. Create a scanner waveform profile using the user controls in **Scanner waveform profiles** category of TerraScan **Settings**.

   You have to repeat steps 5 to 7 for all scanners or lines collected with different pulse rates.
8. Link the scanner waveform profiles with scanner system definitions using the user controls in **Scanner systems** category of TerraScan **Settings**.
9. Link the trajectories with scanner system definitions using the **Edit information** command of the **Trajectories** dialog.

   This enables the extraction of echo properties using the **Extract echo properties** command of the **Project** dialog or the **Extract echo properties** command for loaded points, and the extraction of additional points using the *Extract Echoes* tool.

Waveform tool box

Tools in the **Waveform** tool box are used to view waveform information and to extract additional echoes from the waveform information.

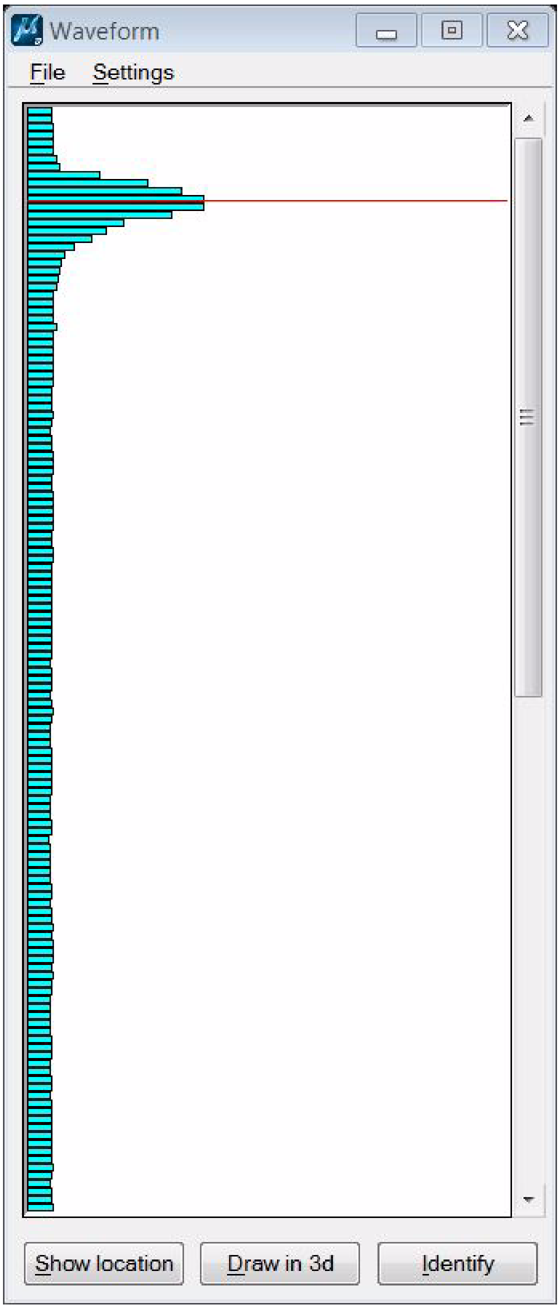


To:	Use:
View waveform data	 *View Waveform*
Extract echos from waveform	 *Extract Echoes*

View Waveform

View Waveform tool opens the **Waveform** dialog that displays the waveform shape of single laser points. The dialog contains commands for identifying a point, showing a point's location, drawing the waveform vector into the design file, saving the waveform as text file, and changing the display settings of the waveform graph.

The waveform graph represents the waveform of a return signal by bars of constant height and varying length. The height of a bar corresponds to a 30 centimeters distance of light travel. The length of a bar indicates how many photons returned to the scanner from an object. Short bars of approximately the same length (= small waveform sample values) represent the background noise, longer bars more or less strong returns from objects. A red line in the graph indicates the location of the selected laser point.



Viewing the waveform requires that trajectories are active and laser points are loaded in TerraScan. The points must be linked to the trajectories and the trajectories must reference the waveform files. See **Waveform processing principles** and **Workflow summary** for more information.

➢ **To view the waveform of a laser point:**

1. Select *View Waveform* tool.

   The **Waveform** dialog opens.

2. Click on the **Identify** button of the dialog.
3. Define a laser point with a data click inside a view.

   This displays the waveform graph for the laser point closest to the data click.

   Click on the **Show location** button and move the mouse pointer inside a MicroStation view in order to highlight the point for which the waveform is show. A data click inside a view centers the highlighted point in the view.

➢ **To draw the waveform vector into the design file:**

1. Identify a laser point for waveform graph display.
2. (Optional) Center the point in a cross section view. This may be best for viewing the waveform vector.
3. Click on the **Draw in 3d** button.

   This draws the waveform vector into the design file. The vector is represented by a cell element that contains lines of different colors. Red color is used for the strongest return, yellow, green, cyan, blue for other returns of decreasing strength, and grey for background noise.

➢ **To save the waveform of a point into a text file:**

1. Identify a laser point for waveform graph display.
2. Select **Save As text** command from the **File** pulldown menu.

   This opens the **Save waveform as text** dialog, a standard dialog for saving files.
3. Define a location and file name for saving the text file and click **Save**.

   This saves the text file.

➢ **To save the waveform of multiple points into a text files:**

1. Draw a fence or polygon around the points for which you want to export the waveform. Select the polygon.
2. Select **Save inside fence** command from the **File** pulldown menu.

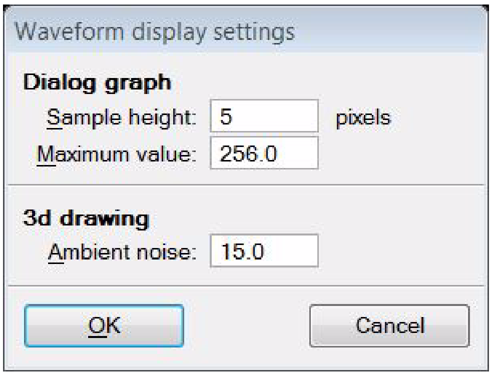
   This opens the **Browse For Folder** dialog, a standard dialog for selecting a storage folder.
3. Select a folder for saving the text files and click **OK**.

   This saves a text file for each point inside the fence/selected polygon. An information dialog shows the number of saved text files out of the number of points. The text files are named automatically as WAVEFORM_<timestamp>_<echo type>.TXT.

➢ **To change the settings for waveform display:**

1. Select **Display settings** command from the **Settings** pulldown menu.

   This opens the **Waveform display settings** dialog:



2. Define settings and click OK.

   This applies the new settings for the display.

Setting:	Effect:
Sample height	Height of a bar in the waveform graph. Given in screen pixels.
Maximum value	Defines the maximum length of a bar that can be displayed in the graph. The value effects the scale of the length of the bars.
Ambient noise	Limit value for background noise. If a waveform sample value is smaller than or equal to the given value, a 3d vector line is drawn in grey.

Extract Echoes

Extract Echoes tool extracts additional points from a return signal.

When scanner system software is generating laser points, it follows a certain logic. It may generate a point from the strongest, first, or last return but usually, it extracts one point from a multiple-return signal. In general, the extraction method of system software works well for laser point clouds.

However, in some places, the generated points might not be optimal. Examples are missing returns from wires or from ground below dense vegetation. In both cases, the system software might extract a point from the return signal, but possibly not the point of biggest interest for certain applications. The *Extract Echoes* tool can be used at such places in order to extract additional points from return signals.

There are several extraction methods available:

- **First possible -** looks only at rising start of the return signal.
- **Last possible -** looks only at trailing end of the return signal. This should be used, for example, to extract additional ground points.
- **All distinct -** constant fraction discriminator, more reliable result than **All possible** method.
- **All possible -** Gaussian decomposition, can generate multiple points from overlapping signals. This should be used, for example, to extract additional points on wires.

The extraction of additional points should only be performed in limited areas where the extraction method of the system software did not provide optimal results. The following methods can be used to limit the processing area for point extraction:

- Place a fence in a section view in order to specify a 3D slice of space where to generate new points. The section depth defines the XY area and the fence the elevation range for point extraction.
- Draw a fence or select polygon(s) to specify a 2D area where to generate points. In this case, the new points can be located at any elevation, only the XY area is defined.

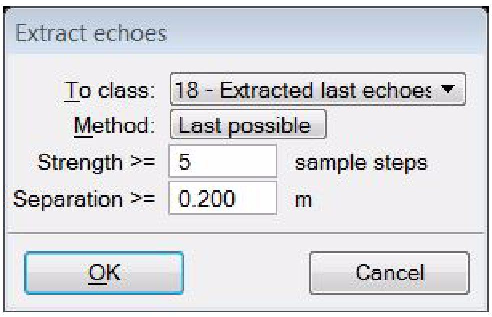
The process creates new points only if it finds returns in the waveform that match the settings for the extraction. For example, if there is no part of the laser beam that penetrated to the ground because of dense vegetation, the **Last possible** method will probably not generate a point on the ground level.

The extraction of additional points requires that trajectories are active and laser points are loaded in TerraScan. The points must be linked to the trajectories and the trajectories must reference the waveform files and the scanner systems. The scanner waveform profile must be available and linked to the scanner system. See **Waveform processing principles** and **Workflow summary** for more information.

➢ **To extract additional points:**

1. Draw a fence or polygon(s) around the area(s) for which you want to extract points as described above. Select the polygon(s).
2. Select *Extract Echoes* tool.

   The **Extract Echoes** dialog opens:



3. Define settings and click OK.

   This generates new points if the software finds return signals in the waveforms that match the settings. The points are added to the laser points in TerraScan memory.

4. Use **Save points As** commands in order to save the laser points into a file.

   The new points are created as inactive points. You must save the points with setting **Points** set to **All points** in the **Save points** dialog. Otherwise, the additional points extracted by *Extract Echoes* tool are not stored.

Setting:	**Effect:**
To class	Target class for extracted points. The list contains the active classes in TerraScan.
Method	Method of point extraction. See explanations above.
Strength	Required number of photons in addition to the background noise. Only if the return signal is stronger than the background noise plus the given value, a points is extracted.
Separation	Minimum distance along the waveform between an existing point and a new point.

8 Roads and Railroads

The tools for processing data of roads and railroads have been developed a lot since more and more data became available from Mobile Mapping System (MMS) surveys. This development is still ongoing, so there will be additions and improvements for the toolsets in the future.

Most of the tools are intended to be used with dense point clouds of high positional accuracy. Such point clouds are usually produced by Mobile Laser Scanning (MLS) systems. However, some of the tool described in this chapter are applicable to Airborne Laser Scanning (ALS) point clouds as well. Some tools benefit from images which are collected by one or several cameras as part of a modern MMS or ALS system.

The processing of point clouds from MLS systems is a complex task if a high accuracy and quality for the end products shall be achieved. This includes the calibration of the scanner system, the matching of drive paths, the classification of the points into classes that support the extraction of the required information, and finally the extraction of the required information itself.

The general workflow of processing MLS data for road and railroad projects can be outlined as follows:

1. **System calibration**: finetuning of the calibration values provided by system manufactors. This is usually done based on laser data that is collected at a specific calibration site. The process is done with TerraScan and TerraMatch and the workflow is described in the TerraMatch Users' Guide.
2. **Project setup**: import and modify raw trajectory information, creation of a TerraScan project, import of raw laser data. This is done with tools of TerraScan.
3. **Drive path matching**: improving the internal and absolute accuracy of the project data. This involves TerraScan and (optionally) TerraPhoto, but mainly tools of TerraMatch are used and the workflow is described in the TerraMatch Users' Guide.
4. **Laser data classification**: cutting off overlap between drive paths, apply classification routines and possibly other automatic/manual processing steps. This depends on the purpose for which the data shall be used.
5. **Extraction of information**: this may include the analysis of the current situation, for example, on a road surface or along a road/rail track; or the detection and/or vectorization of specific features, such as paint markings, road breaklines, rails, overhead wires, or potentially dangerous objects.

The tools described in this chapter are related to the last point in the workflow outline above.

Road data processing

TerraScan provides three options for road breakline extraction. There is the *Find Automatic Breaklines* tool for the automatic extraction of the road crown. The *Find Road Breaklines* tool can be used to extract the crown of a road and road edges semi-automatically based on approximate 2D lines. Finally, there is a special processing workflow that speeds up the digitization of any lines along a corridor. This workflow can be applied to MLS and high-density ALS data, and involves the **Write section points** routine and the *Import Road Breaklines* tool.

Draw Slope Arrows tool and **Color by Slope** display option can be used for water flow analysis on the road surface, for checking the superelevation of road lanes, and for detecting damage on the road surface, such as ruts.

Further functionality is implemented into TerraScan as macro actions, such as **Compute section parameters** and **Find paint lines**.

Road/Railroad data processing

There are some tools in TerraScan that are useful for both application fields, roads and railroads.

One of them is the *Draw Sight Distances* tool that is applicable to ALS and MLS data. Line-of-sight analysis based on laser point clouds has the unique advantage that all objects in the road environment including vegetation are considered.

Fit Geometry Components tool derives geometry components from a surveyed centerline of roads, railroads, and possibly other corridor projects. Geometry components are required for design tasks, especially for the data exchange between different software products.

Railroad data processing

TerraScan has a few tools which are dedicated to railroad processing. They include classification and vectorization tools suited for ALS and/or MLS data.

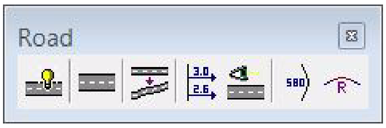
Place Railroad String tool is a useful tool for many purposes. It allows faster digitization of linear features along a corridor as any other MicroStation digitization tool.

Fit Railroad String tool is intended to be used for ALS data of railroads. It fits an approximate rail track centerline to the classified laser points on the rails. The resulting 3D line element follows the rail track centerline more accurately.

There are two tools for the automatic detection and vectorization of rails and overhead wires from MLS data. *Find Rails* tool creates vector lines along rail tracks based on classified laser points, a rail track cross section profile, and an alignment element. *Find Wires* tool is used for the vectorization of all kinds of overhead wires along rail tracks, tram tracks, etc. It creates vector lines and classifies laser points on wires. The automatic wire detection is usually followed by manual improvements of the wire lines which can be done with the *Check Wire Ends* tool.

Road tool box

The tools in the **Road** tool box are used to place breaklines of roads, to analyze the slopes on the road surface and sight conditions, to place labels for curvatures, and to start the fit geometry components module.



To:	Use:
Find road breaklines automatically	 *Find Automatic Breaklines*
Find multiple breaklines along a road	 *Find Road Breaklines*
Import breaklines back to world coordinate system	 *Import Road Breaklines*
Draw slope arrows perpendicular and along alignment	 *Draw Slope Arrows*
Draw sight distance values along road	 *Draw Sight Distances*
Label alignment curvature at regular intervals	 *Label Alignment Curvature*
Fit geometry components to match surveyed alignment	 *Fit Geometry Components*

Find Automatic Breaklines

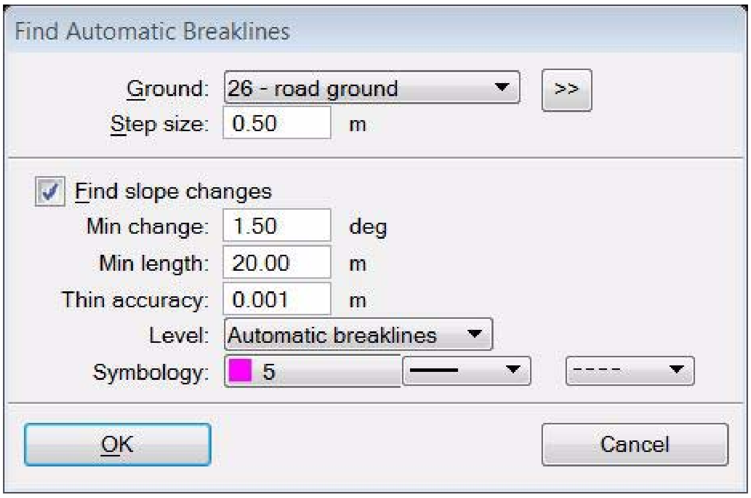
Find Automatic Breaklines tool is used for fully-automatic breakline detection along roads. The tool creates 3D breaklines based on loaded laser points. The laser points on the road surface should be classified into a separate class by using preferably the **Hard surface** classification routine.

The detection works for breaklines along slope changes, for example along the crown of a straight road. After automatic breakline detection, you probably need to check and manipulate the breaklines manually.

➢ **To find road breaklines automatically:**

1. Load laser points into TerraScan. Only points in the class for road breakline detection are required.
2. Select *Find Automatic Breaklines* tool.

   This opens the **Find Automatic Breaklines** dialog:

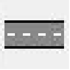


3. Define settings and click OK.

   This starts the breakline detection process. The software draws breakline elements if it finds a slope change in the loaded laser points.

Setting:	Effect:
Ground	Point class that contains points on the road surface. Used for breakline detection. The list contains the active classes in TerraScan.
	Opens the **Select classes** dialog which contains the list of active classes in TerraScan. You can select multiple source classes from the list that are then used in the **Ground** field.
Step size	Distance between locations where the software tries to find a slope change in the laser data in order to insert a vertex for a breakline element.
Find slope changes	If on, the software detects slope changes in the laser data.
Min change	Minimum change in slope gradient. Given in degree.
Min length	Minimum length of a breakline element.
Thin accuracy	Defines the degree of thinning applied to a breakline element. A vertex is removed, if the location of the line does not change more than the given value.
Level	Design file level on which the breakline elements are drawn.
Symbology	Color, line weight, and line style of the breakline elements. Uses the active color table and standard line weights and styles of MicroStation.

Find Road Breaklines



Find Road Breaklines tool is used for semi-automatic breakline detection along roads. The tool requires 2D line elements that run approximately along road breakline locations. Based on that, the software searches the best 3D breakline close by.

The semi-automatic detection works for the following road breakline types:

- edge of pavement - runs along the edge of the road pavement.
- crown of the road - runs along the crown of the road formed by a small slope change.
- planar surface
- section centerline
- section breakline

The parameters for 3D breakline creation are defined for each breakline type.

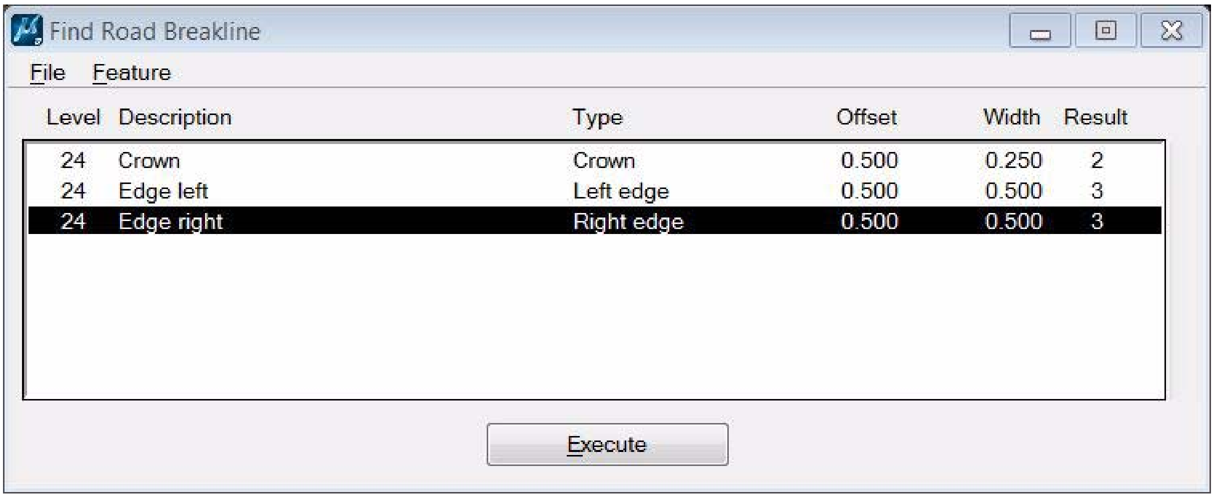
The process requires laser points loaded in TerraScan. The laser points on the road surface should be classified into a separate class by using preferably the **Hard surface** classification routine. Further, the breakline placement along road edges benefits from color values assigned to the laser points.

After automatic breakline detection, you probably need to check and manipulate the breaklines manually.

➢ **To find road breaklines:**

1. Load laser points into TerraScan. Only points in the class for road breakline detection are required.
2. Select *Find Road Breaklines* tool.

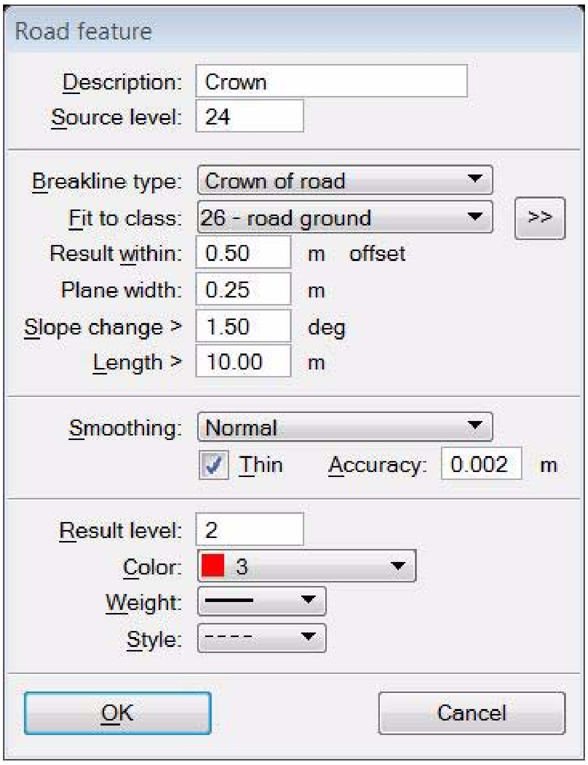
   This opens the **Find Road Breakline** dialog:



3. Select **Add** command from the **Feature** pulldown menu in order to define a new feature for road breakline detection.

   You can modify an existing feature by selecting the feature and using the **Edit** command from the **Feature** pulldown menu. To delete a feature, select the **Delete** command from the **Feature** pulldown menu.

   Add and **Edit** commands open the **Road feature** dialog. The settings in the **Road feature** dialog partly depend on the selected breakline type.



4. Define settings and click OK.

   The feature is added to the list in the **Find Road Breakline** dialog.

5. Repeat steps 3 to 4 for all road breakline features you want to detect.
6. (Optional) Save the road feature definitions into a text file using the **Save as** command from the **File** pulldown menu. You can save changes to an existing text file by selecting the **Save** command from the **File** pulldown menu.
7. Click **Execute** in order to run the breakline detection.

   This starts the breakline detection process. The software draws breakline elements if it finds the 3D location of the breakline features in the loaded laser points.

Setting:	Effect:
Description	Descriptive name of the road feature.
Source level	Design file level, on which the line elements are drawn that define the approximate location of the breakline feature.
Breakline type	Type of the breakline feature: • **Crown of the road** - runs along a small slope change. • **Left edge of pavement** - runs along the edge of pavement on the left side of a road. • **Right edge of pavement** - runs along the edge of pavement on the right side of a road. • **Planar surface** • **Section centerline** • **Section breakline**
Fit to class	Point class that contains points on the road surface. Used for breakline detection. The list contains the active classes in TerraScan.
	Opens the **Select classes** dialog which contains the list of active classes in TerraScan. You can select multiple source classes from the list that are then used in the **Fit to class** field.

Setting:	**Effect:**
Result within	Maximum horizontal offset between the approximate line element and the true breakline location.
Plane width	Width of a plane next to the breakline location. One value applies for the left and right side for **Crown of the road** and **Planar surface** features. There are two values for **Left/Right edge of pavement** features, one for the pavement side and another for the outside-road side. This is not available if **Breakline type** is set to **Section centerline** or **Section breakline**.
Slope change	Minimum change in slope gradient. Given in degree. This is only available if **Breakline type** is set to **Crown of road**.
Length	Minimum length of a breakline element. This is only available if **Breakline type** is set to **Crown of road**.
Use color	If on, the software uses RGB color values assigned to laser points in order to find the breakline location. This is only available if **Breakline type** is set to **Left/Right edge of pavement**.
Fit tolerance	Tolerance value for fitting the breakline element into the laser points. Relates to the noise in the data. This is only available if **Breakline type** is set to **Section centerline**.
Step	This is only available if **Breakline type** is set to **Section centerline**.
Percentile	This is only available if **Breakline type** is set to **Section centerline**.
Max offset	Maximum offset between the approximate line element and the true breakline location.
Smoothing	Defines the degree of smoothing applied to a breakline element: **None**, **Normal**, or **Agressive**.
Thin	If on, a vertex is removed, if the location of the line does not change more than the given value.
Result level	Design file level on which the breakline elements are drawn.
Color	Color of the breakline elements. Uses the active color table of MicroStation.
Weight	Line weight of the breakline elements. Uses the standard line weights of MicroStation.
Style	Line style of the breakline elements. Uses the standard line styles of MicroStation.

Import Road Breaklines

Import Road Breaklines tool converts linear elements from an artificial coordinate system into normal coordinates. It is used in combination digitized lines based on TerraScan *section points* which are produced by the **Write section points** macro action.

The line elements are digitized in an artificial coordinate system in a separate design file. The artificial coordinate system is defined by:

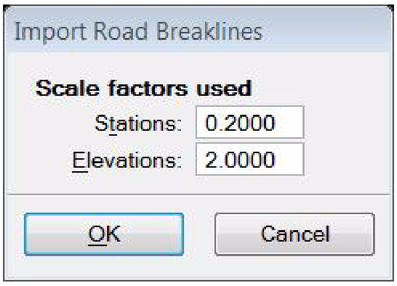
- X axis - scaled stations along an alignment element.
- Y axis - offset from an alignment element.
- Z axis - scaled elevation values of the original point cloud data.

The tool converts these artificial XYZ coordinates back to Easting, Northing, Elevation coordinates. It utilizes the same alignment element and inverse scaling factors as were used for producing the section points.

➢ **To import road breaklines:**

1. Attach the section design file as a reference to the normal project design file.
2. Open two top views, one showing the location of the road and the alignment element used for creating section points, and the other one showing the digitized lines in the attached reference design file.
3. Select the alignment element using MicroStation *Selection* tool.
4. Draw fence around the lines in the reference design file.
5. Select *Import Road Breaklines* tool.

   This opens the **Import Road Breaklines** dialog:

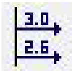


6. Define settings. The values must be the same as used for creating the section points with the **Write section points** macro action in order to compute correct coordinate values for the breakline elements.
7. Click OK.

   This converts the lines inside the fence from the artificial coordinates to the original coordinates and draws them into the master design file.

Setting:	Effect:
Stations	Scale factor along the alignment element. Used for decompressing the digitized lines to their normal length.
Elevations	Scale factor for elevation values. Used for resolve the exaggeration of elevation values of the digitized lines.

Draw Slope Arrows



Draw Slope Arrows tool computes the slope along an alignment element of a road. The computation can be done for the side slope of road lanes (superelevation) or for the longitudinal slope of the road. The tool draws arrows which show the direction of the slope and text labels that show the gradient of the slope.

The tool requires a line string as alignment element. This is usually the approximate center line of the road which can be derived, for example, from the trajectory lines. The alignment element determines the longitudinal direction of the road as well as the horizontal location of the slope arrows. The elevation of the alignment element does not effect the slope arrows.

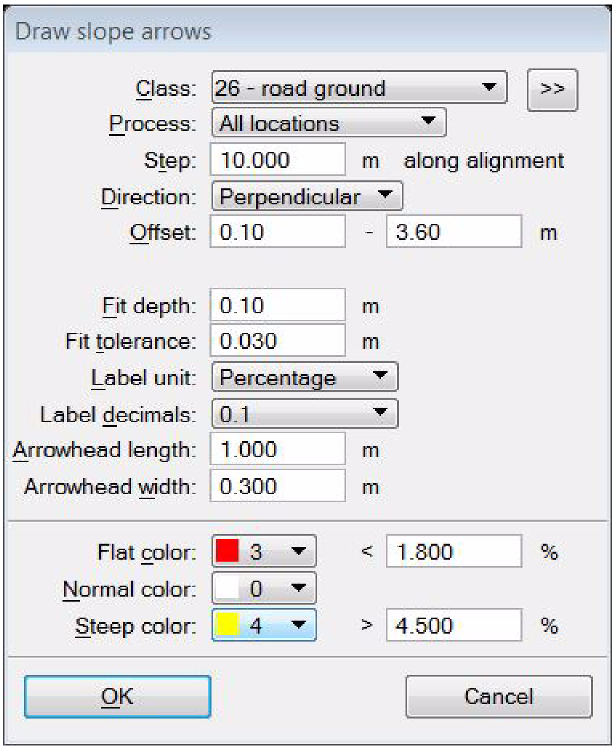
The elevation of the slope arrows is fitted to laser points on the road surface. Thus, these laser points should be classified into a separate class by using preferably the **Hard surface** classification routine. The gradient of a slope arrow is computed from the elevation values of the start and end points of the arrow element.

The tool requires laser points loaded into TerraScan. However, the same process can be performed for a TerraScan project using the **Compute slope arrows** macro action and then, reading the slope arrows from text files using the **Read / Slope arrows** command.

➢ **To draw slope arrows:**

1. Load laser points into TerraScan. Only points on the road surface are required.
2. Select the alignment element with the MicroStation *Selection* tool.
3. Select *Draw Slope Arrows* tool.

   This opens the **Draw slope arrows** dialog:



4. Define settings and click OK.

   This starts the process. The software draws arrows and text elements along the alignment element wherever it finds laser data. The level, line weight, line style, and text size of the arrow and label elements are determined by the active symbology and text size settings in MicroStation.

   An information dialog shows the number of created slope arrows.

Setting:	Effect:
Class	Point class that contains points on the road surface. Used for fitting the elevation of slope arrows. The list contains the active classes in TerraScan.
>>	Opens the **Select classes** dialog which contains the list of active classes in TerraScan. You can select multiple source classes from the list that are then used in the **Class** field.
Process	Area to process: • **All locations -** slope arrows are created wherever there is laser data available. • **Inside active block -** slope arrows are created only inside the active block. This excludes areas covered by neighbour points that are loaded in addition to the points of an active project block.
Step	Distance between locations where the software places a slope arrow.
Direction	Direction of the slope arrows relative to the alignment element or the road direction: • **Longitudinal -** in road direction. • **Perpendicular -** perpendicular to the road direction.
Offset	Defines the horizontal distance of the start and end point of an arrow relative to the alignment element. This is only one value for **Longitudinal** arrows and two values for **Perpendicular** arrows. The two offset values also determine the length of slope arrows with **Perpendicular** direction. Positive offset values create slope arrows to the right side of the alignment element, negative values to the left side.
Length	Length of slope arrows with longitudinal direction. This is only active if **Direction** is set to **Longitudinal**.
Fit depth	Depth of a section in the laser data where the software fits the arrow to the points on the road surface.
Fit tolerance	Tolerance value for fitting the arrow to the laser points. Relates to the noise in the data.
Label unit	Unit for expressing the slope gradient: **Degree** or **Percentage**.
Label decimals	Number of decimals for slope labels.
Arrowhead length	Length of the arrow head as part of the slope arrow.
Arrowhead width	Width of the arrow head.
Flat color	Color of a slope arrow if the slope gradient is less or equal to the given value.
Normal color	Color of a slope arrow if the slope gradient is between the given flat and steep values.
Steep color	Color of a slope arrow if the slope gradient is larger than the given value.

✍ You can undo the creation of slope arrows by using the **Undo** command from the **Edit** pulldown menu of MicroStation.

Draw Sight Distances

Draw Sight Distances tool determines how far a viewer sees along a road, railroad, or other corridor, and produces labels for sight distances. It's basically a tool for line-of-sight analysis based on point clouds.

The path of the viewer along the corridor is defined by a line element. It should run along a lane of the road or a rail track at the elevation of the ground. It can be produced, for example, by drawing the trajectory line into the design file and draping it to the ground points using the *Drape Linear Element* tool. The viewer height is defined a constant value in the tool's settings.

The target positions for the line-of-sight analysis are also defined by a line element draped on the ground elevation. The target line can be a little bit longer than the viewer line. The viewing angle which determines the area for potential obstacle search is defined as a constant value in the tool's settings.

Potential obstacles for the viewer are represented in the laser point cloud. The points on the road surface should be classified into a separate class by using preferably the **Hard surface** classification routine. Point in the close surrounding of the road should be classified into ground and above-ground classes. To get a reliable result from the line-of-sight analysis, any points below the ground, from moving objects, and noise above the road surface should be classified into separate classes. These classes can be excluded from the process.

The process checks if there is any laser point close to a straight line from a viewer position to a target position. If there is a laser point, the closest point to the viewer position determines the sight distance.

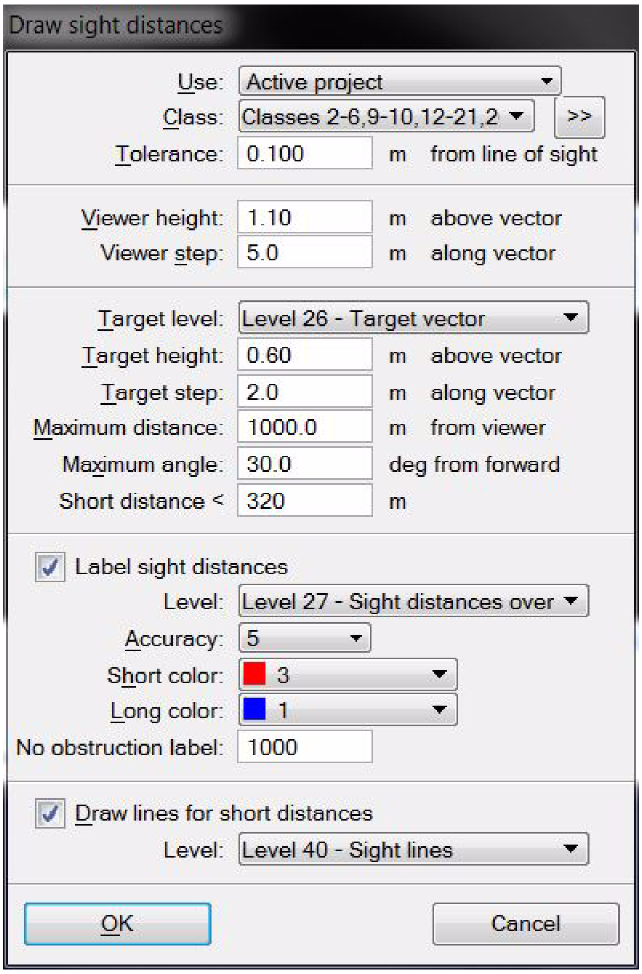
There are rules and regulations for required sight distances along a road. The distances mainly depend on the speed allowance and vary from country to country. There are different sight distance requirements for safely stopping a car or for safely overtaking another car. Example: For safely stopping a car, a viewer with a height of 1.10 m above lane center and a speed of 80 km/h must see a target of 0.40 m above the road surface in a distance of 120 m. For safely overtaking a car, the same viewer must see a target of 0.60 m above the road surface in a distance of 320 m. The viewer path needs to be cut into separate line elements according to speed limits in order to do a precise sight distance analysis.

Draw Sight distances tool can run on loaded points as well as on TerraScan project points. It creates text elements as labels for sight distances. In addition, it can draw line elements for short sight distances into the design file. These line element can then be used to classify points close to them using the **By centerline** classification routine. The classification helps to identify obstacles in the point cloud.

➢ **To draw sight distances:**

1. Create a line element along the lane centerline as viewer traveling path and (optional) another line element as target path.
2. Drape the line element(s) to the ground on the road surface.
3. (Optional) Load points into TerraScan if you want to run the tool on loaded points.
4. Select the line element that defines the viewer traveling path.
5. Select *Draw Sight Distances* tool.

   This opens the **Draw sight distances** dialog:



6. Define settings an click OK.

   This starts the sight distance analysis process. The size of the text elements is determined by the active text size settings in MicroStation. The color, line weight, and line style of lines for short distances are determined by active symbology settings in MicroStation.

Setting:	**Effect:**
Use	Laser points used for the sight distance analysis process: • **Loaded points -** points loaded into TerraScan. • **Active project -** points referenced by the active TerraScan project.
Class	Point classes that are included in the sight analysis. This should include all points on potential obstacles. The list contains the active classes in TerraScan.
	Opens the **Select classes** dialog which contains the list of active classes in TerraScan. You can select multiple source classes from the list that are then used in the **Class** field.
Tolerance	Radius around a straight line between viewer and target position within which a laser point may be considered a sight obstacle.
Viewer height	Height of the viewer above the viewer path element.
Viewer step	Distance between locations along the viewer path element where the software analyses the sight of the viewer.
Target level	Design file level on which the line element is drawn that defines the target path.
Target height	Height of a target object above the target path element.

Setting:	Effect:
Target step	Distance between locations along the target path element where the software analyses the sight towards a target object.
Maximum distance	Maximum distances from the viewer considered in the sight analysis.
Maximum angle	Angle of sight forward from the viewer. Defines the area that is analysed regarding sight obstacles.
Short distance	Defines the maximum value of a critical sight distance. Sight distances smaller or equal to the given value can be labeled with a different color in order to highlight the locations.
Label sight distances	If on, text elements are drawn into the design file that label the sight distance at each viewer position.
Level	Level on which sight distance labels are drawn.
Accuracy	Accuracy of sight distance labels. Values are rounded to the given accuracy, e.g. to the closest 5 m value.
Short color	Color of short sight distance labels. Applied to all distances smaller or equal to the given **Short distance** value.
Long color	Color of sight distance labels if the distance is longer than the given **Short distance** value.
No obstruction label	Label of viewer positions for which no obstruction is found.
Draw lines for short distances	If on, lines are drawn from viewer to target positions if the distance is smaller or equal to the given **Short distance** value.
Level	Level on which lines for short distances are drawn.

✍ You can undo the creation of sight distance labels and lines by using the **Undo** command from the **Edit** pulldown menu of MicroStation.

Label Alignment Curvature

Label Alignment Curvature tool computes the radius of curves of 3D line elements. It places text elements that show the horizontal or vertical curvature radius.

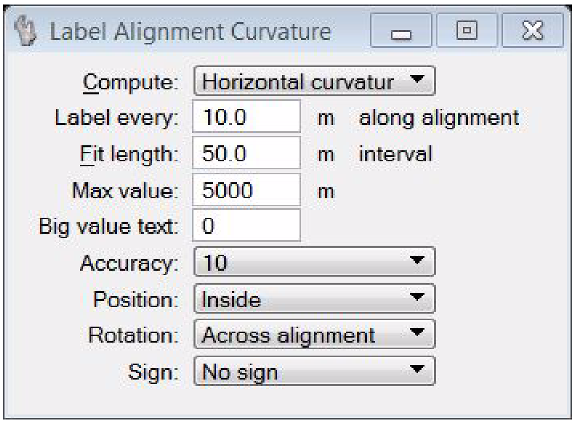
For roads, the curvature radius determines the best value for the side slope inside a curve. Therefore, the tool supports the analysis of road surface properties.

The tool requires an alignment element, which can be the centerline of a road, railroad, etc. derived from trajectory lines or any other representative line element.

➢ **To create labels for the radius of curves:**

1. Select the alignment element with the MicroStation *Selection* tool.
2. Select *Label Alignment Curvature* tool.

   This opens the **Label Alignment Curvature** dialog:



3. Define settings.
4. Place a data click inside the MicroStation view.

   This creates the curvature labels. The software draws text elements along the alignment element. The level, color, and text size of the labels are determined by the active symbology and text size settings in MicroStation.

   An information dialog shows the number of created curvature labels.

Setting:	Effect:
Compute	Curvature to compute: **Horizontal curvature** or **Vertical curvature**.
Label every	Distance between locations along the alignment element where the software places a curvature label.
Fit length	Distance along the alignment element from which the software computes the curvature radius.
Max value	Maximum curvature radius that is labeled.
Big value text	Text used for radius values larger than the given **Max value**.
Accuracy	Accuracy of curvature labels. Values are rounded to the given accuracy, e.g. to the closest 10 m value.
Position	Determines the placement location of the labels relative to the alignment element: • **On alignment** - on the alignment element. • **Left** - left of the alignment according to digitization direction. • **Right** - right of the alignment according to digitization direction. • **Inside** - on the inside of a curve. • **Outside** - on the outside of a curve.

Setting:	Effect:
Rotation	Determines the placement rotation of the labels relative to the alignment element: • **Along alignment -** reading direction is parallel to the alignment. • **Across alignment -** reading direction is perpendicular to the alignment.
Sign	Sign added in front of the curvature radius value: • **No sign -** no sign is added. • **Left negative -** a minus sign is added to left-hand curves. • **Right negative -** a minus sign is added to right-hand curves.

✍ You can undo the creation of curvature labels by using the **Undo** command from the **Edit** pull-down menu of MicroStation.

Fit Geometry Components

Fit Geometry Components tool starts the **Component fitting** module of TerraScan. The module creates design geometry built from the geometry components lines, arcs, and clothoids. The aim is to create a geometry from these components that forms the best match to a surveyed alignment of a road, a railroad, or a pipeline. The module finds the best fit for both horizontal and vertical geometry.

The module and the creation of geometry components serves different purposes:

- **Data exchange -** view the current geometry of a road/railroad/pipeline in design software such as Bentley InRoads, Bentley Track, etc and/or export the geometry into LandXML or Tekla 11/12 format. Design software may only accept certain geometry components for linear features.
- **Road surface analysis -** find long span deformations of road surfaces.
- **Object design comparison -** compare existing object geometry components with design recommendations.

The tool starts from a 3D centerline element that can be created, for example, based on ALS or MMS data.

➢ **To start the component fitting module of TerraScan:**

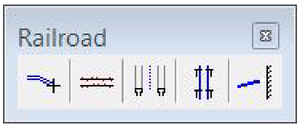
1. Use the MicroStation *Selection* tool in order to select a centerline element that represents the surveyed object.
2. Select the *Fit Geometry Components* tool.

   This starts the module and opens the **Components fitting** dialog.

The processing workflow of component fitting and the commands of the dialog are explained in detail in Chapter **Geometry Component Fitting** on page 390.

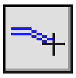
Railroad tool box

The tools in the **Railroad** tool box are used to place lines, to fit lines to classified points along rails, to find rails and overhead wires automatically, and to check end points of overhead wires.



To:	Use:
Place approximate railroad centerline	*Place Railroad String*
Fit railroad centerline to laser points	*Fit Railroad String*
Find rails using a rail section template	*Find Rails*
Find overhead wires	*Find Wires*
Check end points of overhead wires	*Check Wire Ends*

Place Railroad String



Place Railroad String tool can be used for the digitization of line strings. The tool integrates three types of functionality: it draws a line element and allows view panning and rotation. Thus, it enables faster digitization compared with MicroStation tools.

Initially, the tool was implemented for the manual placement of an approximate centerline between two rails based on ALS data or aerial images. The centerline can be used to classify points on rails more accurately. However, the tool is very useful for digitizing any kind of line string.

➢ **To place a line string:**

1. Select the *Place Railroad String* tool.

   The **Place Railroad String** dialog opens:

2. Define the location of the first point on the line string with a data click.

   The application draws a dynamic rectangle whenever you move the mouse pointer inside the view. If you place a data click outside the rectangle, the application pans the view in the direction of the data click. If **Rotate view when panning** is switched on in the tool's dialog, the view is also rotated in the direction of the data click. If you place a point inside the rectangle, you add a new vertex to the line string.

3. Digitize the complete line string.
4. After placing the last vertex, click on the reset button in order to finish the line string.

   The lines string is drawn into the design file on the active level and using the active symbology settings of MicroStation.

➢ **To prepare for railroad line string placement:**

1. Classify potential points on rail into a separate point class using **By intensity** or **Railroad** classification routines.

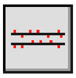
   This initial classification probably includes a number of points which are not points on rails. However, the should provide a visual impression of the railroad track location.

2. Switch on the display of the points on rails and switch off the display of all other point classes in a top view.
3. (Optional) Define a custom line style in MicroStation.

   A recommended line style consists of two lines which are the railroad width apart from each other. This supports the placement of parallel lines for the rails and allows viewing the railroad string as a centerline of the track or as a pair of lines for the rails simply by switching line styles on or off in a view.

4. Select level, color, and the custom line style as the active symbology in MicroStation.
5. Select the *Place Railroad String* tool and digitize the railroad string according to the instructions above.

Fit Railroad String



Fit Railroad String tool can be used to fit a manually placed railroad centerline to classified laser points. It is intended to by used after an approximate railroad centerline has been placed by using, for example, the *Place Railroad String* tool.

After placing an approximate centerline, you can continue as follows:

1. Classify points on rails more accurately by using **Railroad** classification routine with the approximate centerline as alignment element.

   This classifies points with a specific elevation pattern and within a given offset (half of the rail width) from the alignment.

2. Use *Fit Railroad String* tool in order to fit the centerline to the classified points on the rails.

   OR

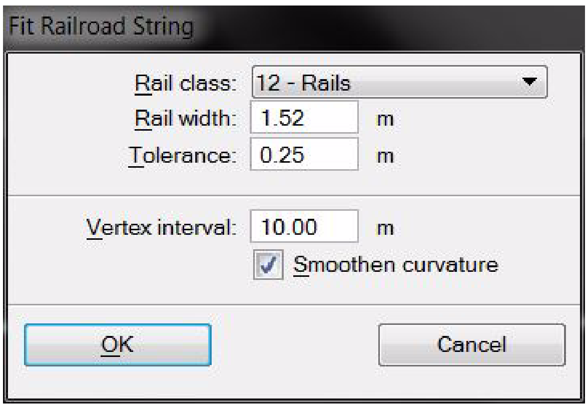
1. Classify ground using the **Ground** classification routine and drape the centerline to the ground elevation using the *Drape Linear Element* tool.
2. Classify points on the rails more accurately by using the **By centerline** classification routine with appropriate offset and elevation difference values.
3. Use *Fit Railroad String* tool in order to fit the centerline to the classified points on the rails.

The tool uses points on rails within an offset distance in order to find the best location for the centerline. The offset depends on the width of the rail track and the initial accuracy of the centerline elements. The offset is defined in the tool's dialog as **(0.5 * Rail width) ± Tolerance**. The elevation of the fitted centerline is derived from the elevation values of the laser points.

➢ **To fit railroad centerlines:**

1. Select the centerline element(s) using MicroStation *Selection* tool.
2. Select the *Fit Railroad String* tool.

   This opens the **Fit Railroad String** dialog:



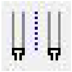
3. Define settings and click OK.

   The application compares each selected line element with points in the given **Rail class** within the given offset from the centerline. It creates a new line string element for the fitted centerline which is drawn on the active level using the active symbology settings of MicroStation.

Setting:	Effect:
Rail class	Point class that contains points on rails.
Rail width	Rail track width, distance from the center of one rail to the center of the other rail.

Setting:	Effect:
Tolerance	Tolerance value for the offset between centerline and rails. This should be big enough to compensate some locational inaccuracy in the inital centerline and in laser points. However, it should be less than half of **Rail width**.
Vertex interval	Maximum distance between vertices of the fitted centerline. Normally between 5.0 and 25.0 meters.
Smoothen curvature	If on, the fitted centerline is smoothened by balancing angular direction changes between consecutive vertices. Normally, this should be switched on.

Find Rails



Find rails tool is used for the automatic vectorization of rails based on MLS point clouds.

The software looks at consecutive cross sections of laser data along an alignment element. For each cross section, it tries to find the position where a user-defined cross section profile of the track matches the best number of laser points.

The vectorization process starts from an alignment element which represents the approximate centerline of a rail track. Any digitized centerline can be used as alignment. You can use, for example, **Draw into design** command for trajectories and apply a lever arm correction in order to derive a centerline from the trajectory. The lever arms are the three components of the vector between the IMU and the center of the rail track.

Alternatively, trajectories can be used directly for the rail detection. They must be imported with the correct system definition values for IMU misalignment. See **Scanner systems** category of TerraScan **Settings** for more information. In addition, they must be projected on the ground and to the center of the rail track. This can be established by applying a lever arm correction to the original trajectories by using the **Add lever arm** command. In the vectorization process based on trajectories, the software uses the roll angle of the trajectory positions as cant (superelevation) angle of the rail track.

The tool further requires a cross section profile defined in TerraScan **Settings**. The profile includes the two rails of a track, possibly places where there are no laser point (shadow parts of rails), and the location of lines that the software creates in the vectorization process. The creation of a rail track cross section is described in **Rail section templates** category of TerraScan **Settings**.

Any overlapping strips in the laser data should be matched and overlap should be cut off before running the rail vectorization. The **By centerline** classification routine should be used for an approximate classification of the points on rails. You can use, for example, the lever arm-corrected trajectory drawn in the design file for the classification by centerline.

The *Find Rails* tool runs on points loaded into TerraScan. It creates line string elements at the location(s) defined in the cross section profile.

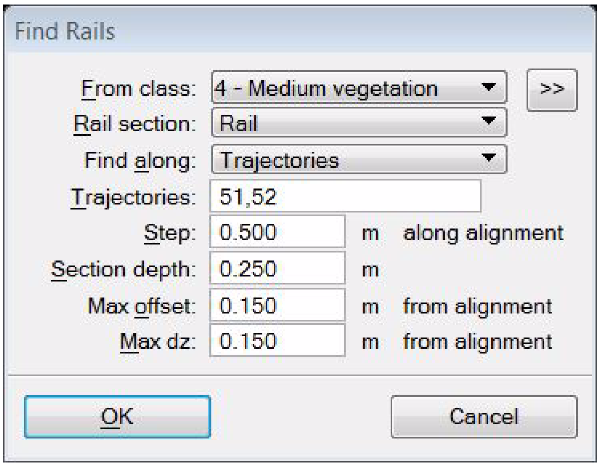
➢ **To vectorize rails automatically:**

1. Load laser points into TerraScan. Only points on and close to the rails are required.
2. Select the alignment element with the MicroStation *Selection* tool if you want to use a selected element as alignment.

   OR

2. Load trajectories into TerraScan using the *Manage Trajectories* tool.
3. Select *Find Rails* tool.

   This opens the **Find Rails** dialog:



4. Define settings and click OK.

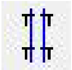
   This starts the vectorization process. The software draws line strings wherever it is able to fit the rail track cross section to the laser data. The level, color, line weight, and line style of the lines are determined by the active level and symbology settings of MicroStation.

   Depending on the amount of laser data, the accuracy of the alignment element, and how well the software can fit the rail track cross section to the laser data, the process may take some time. It is recommended to test the settings for the tool with small data samples.

Setting:	Effect:
From class	Point class that contains points on and close to the rails. Used for fitting the rail track cross section. The list contains the active classes in TerraScan.
>>	Opens the **Select classes** dialog which contains the list of active classes in TerraScan. You can select multiple source classes from the list that are then used in the **From class** field.
Rail section	Name of the rail track cross section. The list contains all sections that are defined in **Rail section templates** category of TerraScan **Settings**.
Find along	Defines the alignment element used for the vectorization: • **Trajectories** - active trajectories in TerraScan. • **Selected vectors** - a selected line string element.
Trajectories	Trajectory numbers that are used for the vectorization. Separate several numbers by comma. Type 0-65535 for using all trajectories. This is only active if **Find along** is set to **Trajectories**.
Max roll	Maximum value of the cant anlge (= rail track superelevation). This is only active if **Find along** is set to **Selected vectors**.
Step	Distance between locations along the alignment where the software tries to fit the rail track cross section to the laser points.
Section depth	Depth of a section in the laser data where the software fits the rail track cross section to the laser points.
Max offset	Maximum horizontal distance between the alignment and a line element that the software should draw as result of the vectorization process (usually a line on the rails or the track centerline).
Max dz	Maximum vertical distance between the alignment and a line element that the software should draw as result of the vectorization process (usually a line on the rails or the track centerline).

✍ You can undo the vectorization of rails by using the **Undo** command from the **Edit** pulldown menu of MicroStation.

Find Wires



Find wires tool is used for the automatic detection of overhead wires based on dense point clouds. The tool can be used for any kind of overhead wires, such as rail or tram wires, in contrast to the *Detect Wires* tools which is exclusively for the detection of powerline wires.

The detection process starts from classified laser points and, optionally, from an alignment element which runs in the direction of the wires. Any overlapping strips in the laser data should be matched and overlap should be cut off before running the wire vectorization.

If data was captured by an MLS system mounted on a survey train, the overlap of parallel strips should be cut off in a way that points from a more distant drive path can be used for the wire detection. This leads to a more reliable result since wires are raised by the survey train in the closest drive path. As an alternative to cutting off overlap, the **By section template** classification routine can be used for classification of points from the closest drive path into a separate class.

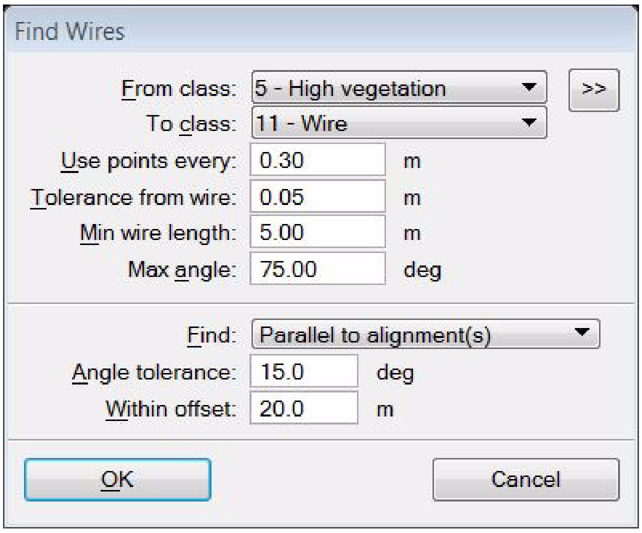
Further, laser points should be classified into ground and above ground points. One of the above-ground point classes should contain the points on wires (e.g. the high vegetation class) and is then used as source class for the wire detection.

The *Find Wires* tool runs on points loaded into TerraScan. It classifies laser points on wires into a separate class and creates line string elements that are fitted to the laser points on wires. The software stops each wire at a small distance from its end points. The wire ends can be placed more accurately by using the *Check Wire Ends* tool.

➢ **To detect wires automatically:**

1. Load laser points into TerraScan. Only points on the wires are required.
2. (Optional) Select an alignment element with the MicroStation *Selection* tool if you want to detect wires running parallel or perpendicular to the alignment.
3. Select *Find Wires* tool.

   This opens the **Find Wires** dialog:



4. Define settings and click OK.

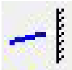
   This starts the detection process. The software draws line strings wherever it is able to fit a line to the laser data. The level, color, line weight, and line style of the lines are determined by the active level and symbology settings of MicroStation.

Setting:	Effect:
From class	Point class that contains points on the wires. Used for fitting the lines. The list contains the active classes in TerraScan.

Setting:	Effect:
>>	Opens the **Select classes** dialog which contains the list of active classes in TerraScan. You can select multiple source classes from the list that are then used in the **From class** field.
To class	Target class into which points on detected wires are classified.
Use points every	Distance between locations along a wire where the software tries to fit the line element to the laser points.
Tolerance from wire	Distance around a wire within which the software uses points for fitting the line element.
Min wire length	Minimum lenght of a line element at a wire location.
Max angle	Maximum vertical angle off from horizontal of a line element at a wire location.
Find	Defines what wires the software is searching for: • **All wires -** wires in all directions. • **Parallel to alignment(s) -** wires that run parallel to selected alignment element(s). • **Perpendicular to alignment(s) -** wires that run perpendicular to selected alignment element(s).
Angle tolerance	Maximum horizontal angular difference between the alignment and a line element at a wire location. This is only active if an alignment element is selected and if **Find** is set to **Parallel to alignment(s)** or **Perpendicular to alignment(s)**.
Within offset	Maximum horizontal distance between the alignment and a line element at a wire location. This is only active if an alignment element is selected and if **Find** is set to **Parallel to alignment(s)** or **Perpendicular to alignment(s)**.

✍ You can undo the detection of wires by using the **Undo** command from the **Edit** pulldown menu of MicroStation (vectorization) and the **Undo** command from the **Point** pulldown menu of TerraScan (classification).

Check Wire Ends



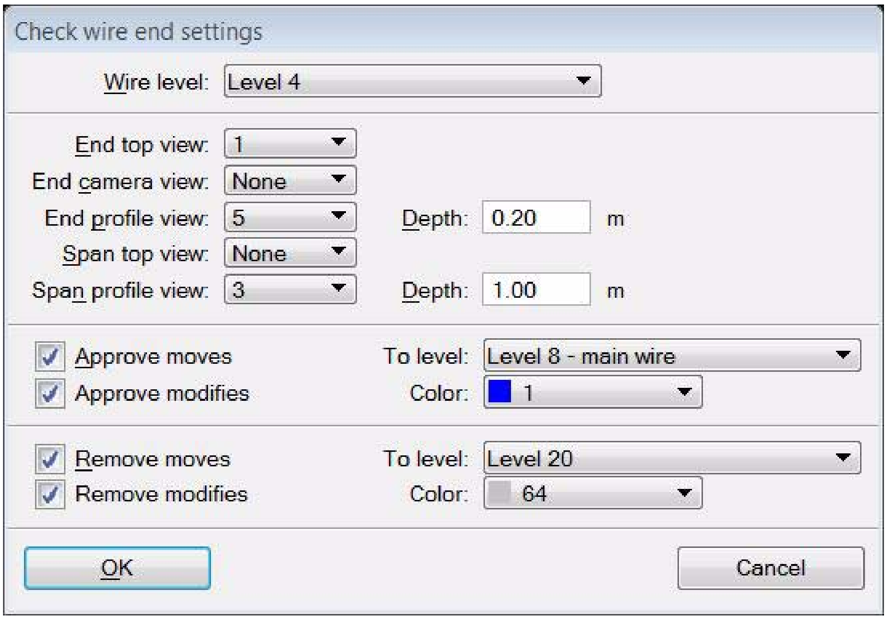
Check Wire Ends tool can be used to check automatically detected wires in an organized way. The tool opens the **Check wire ends** dialog which contains user controls for manipulating wires and wire end points. It is intended to be used after automatic wire detection by *Find Wires* tool.

The dialog shows a list that contains all wires and their end points. If a line in the list is selected, the software updates the display in a number of MicroStation views. The views must be open and defined in the tools settings. The tool can update different view types, such as top, section, and camera views which show either a wire completely or the end point of a wire.

➢ **To check wire end points:**

1. Select *Check Wire Ends* tool.

   This opens the **Check wire end settings** dialog:



2. Define settings and click OK.

   This opens the **Check wire ends** dialog.

Setting:	Effect:
Wire level	Design file level on which the lines for wires have been drawn. All lines on this level are added to the check list.
End top view	A top view showing the end of a wire is displayed in the given view.
End camera view	A camera view showing the end of a wire in displayed in the given view. This view can be used to display images if a mission, camera, and image list are loaded into TerraPhoto.
End profile view	A longitudinal section of the end of a wire is displayed in the given view.
Span top view	A rotated top view showing a wire completely in a horizontal section is displayed in the given view.
Span profile view	A longitudinal section showing a wire completely is displayed in the given view.
Approve moves	If on, the line of an approved wire is moved to the level defined in the **To level** list. If off, the level of approved wires remains unchanged.

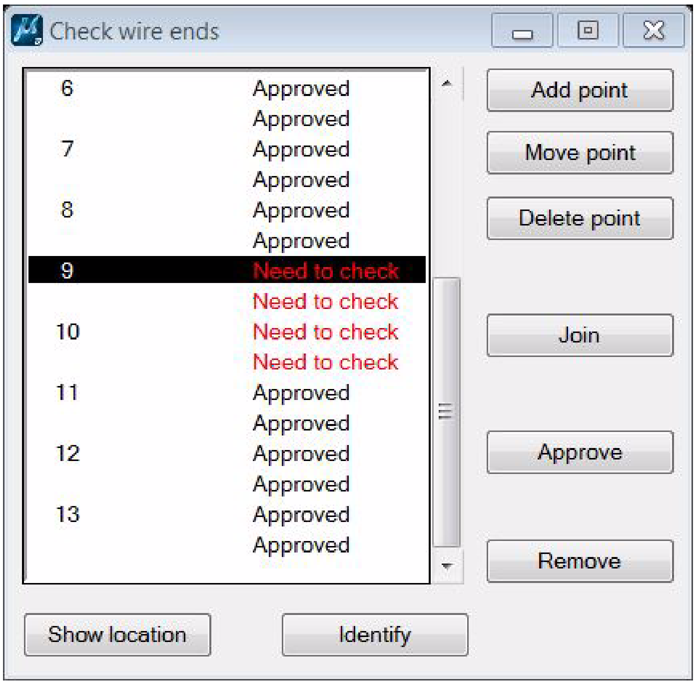
Setting:	Effect:
Approve modifies	If on, the line color of an approved wire is modified to the given **Color**. The list contains the active color table of MicroStation. If off, the color of approved wires remains unchanged.
Remove moves	If on, the line of a removed wire is moved to the level defined in the **To level** list. If off, the level of removed wires remains unchanged.
Remove modifies	If on, the line color of an removed wire is modified to the given **Color**. The list contains the active color table of MicroStation. If off, the color or removed wires remains unchanged.

Check wire ends

The **Check wire ends** dialog shows a list that contains all wires and their end points. For each wire, there is a number and two end points. The status of each wire end point is *Need to check* by default.

Further, the dialog contains buttons that can be used to manipulate wire lines, to change the status of a wire in the list, and to display wire end locations. You can add intermediate vertices to wire lines, move the end points of wire lines, delete points from wire lines, and join wire lines in order to bridge gaps in the automatically detected wires.

If an end point of a wire is moved close to an end point of another wire which is an potential end point for a join, the horizontal and vertical distances between the end points are shown in the dialog.



To show the location of a wire end point, select a line in the **Check wire ends** dialog. Click on the **Show location** button and move the mouse pointer into a view. This highlights the selected wire end point with a cross.

To identify a wire end point, click on the **Identify** button and place a data click close to a wire end point in a view. This selects the corresponding line in the **Check wire ends** dialog.

After checking a wire end point and possibly improving its location, click on the **Approve** button. This changes the status of the selected end point to *Approved*. If both end points of a wire are approved, the wire line is moved to another level and/or the color is changed, if the settings in the **Check wire end settings** dialog are defined accordingly.

If you want to delete a wire, you click on the **Remove** button. This removes the selected wire from the list and the wire line is moved to another level and/or the color is changed, if the settings in the **Check wire end settings** dialog are defined accordingly. The **Remove** button does not delete the wire line from the design file.

To modify wire lines and their end points, you can use the other buttons of the dialog. You can undo the modification of wires by using the **Undo** command from the **Edit** pulldown menu of MicroStation.

➢ **To add a vertex to a wire:**

1. Select the wire in the list.
2. Click on the **Add point** button and move the mouse pointer into a view, preferably a section view.

   This dynamically displays the new vertex and wire line at the mouse pointer location.
3. Define the location of the new vertex by a data click.

   You can continue with step 3. The software lets you place only intermediate vertices for the selected wire.

➢ **To move an end point of a wire:**

1. Select the wire end point in the list.
2. Click on the **Move point** button and move the mouse pointer into a view.

   This dynamically displays the new end point and the wire line at the mouse pointer location.
3. Define the location of the new end point by a data click.

   You can continue with step 3. The software lets you move only the selected wire end point.

➢ **To delete a point of a wire:**

1. Select the wire in the list.
2. Click on the **Delete point** button and move the mouse pointer into a view.

   This dynamically highlights the point on the wire closest to the mouse pointer location.
3. Delete the point by a data click.

   You can continue with step 3. The software lets you delete end points and intermediate vertices of the selected wire.

➢ **To join wires:**

1. Select a wire in the list.
2. Click on the **Join** button and move the mouse pointer into a view.

   This dynamically displays possible connection lines for the selected wire at the mouse pointer location.
3. Move the mouse pointer close to the end point of the wire to which you want to join the selected wire.
4. Confirm a connection line by a data click.

   This joins two wire lines. If the status of the effected wires was already *Approved*, it is set back to *Need to check.*

✍ Manual changes of the wire lines do not effect the classification of laser points. If you want to refine the classification of laser points on wires, you can run a **By centerline** classification using the wire lines as centerlines with offset and elevation difference settings of ± a few centimeters.

9 3D Building Models

There are two approaches in TerraScan for producing 3D vector models of buildings. The main difference between the approaches is their degree of automation and the speed of creating vector models for a large number of buildings.

Approach	Tool sets
Automatic vectorization of multiple buildings with manual improvements	**Buildings tool box** **Building Patches tool box** **Building Edges tool box**
Half-automatic vectorization of single buildings	*Construct Planar Building tool* **Building Planes tool box** **Building Boundaries tool box**

According to the common way for describing building models, the models of TerraScan are at level-of-detail (LOD) 2. In LOD 2, roof shapes and the overall structure of roofs are represented but walls are just plain vertical polygons.

Automatic building vectorization with manual improvements

TerraScan provides a set of tools for automatic building vectorization based on airborne laser scanning (ALS) data. The 3D vector models are created fully-automatically but for higher accuracy, they can be modified manually with dedicated tools. These tools ensure that the topology of a building model is preserved and allow fast and easy editing. The tools can also be used to create non-planar roof shapes.

The automatic vectorization is based on classified laser points of the ground and on building roofs. Building footprints can be used in the vectorization process for placing walls or roof edges. Image data loaded in TerraPhoto supports the automatic vectorization of buildings. For manual editing, images in camera views improve the result essentially, because edges of roofs, roof structures, and smaller details may not be detectable accurately in the laser data.

The automatic building vectorization runs on loaded laser points using the *Vectorize Buildings* tool or for a TerraScan project using the **Vectorize buildings** macro action.

The workflow for automatic building vectorization includes the following steps:

1. Match flightlines and cut off overlap.
2. Classify ground points using the **Ground** classification routine.
3. Classify high points which may be hits on building roofs using the **By height from ground** classification routine. This classification also includes points from high vegetation and other high objects.
4. Classify points on building roofs using the **Buildings** classification routine.
5. (Optional) If images are available, load a mission and an image list into TerraPhoto. The camera parameters of the mission and the image list should be adjusted in order to provide accurately positioned images.
6. Create vector models of buildings using the *Vectorize Buildings* tool for loaded laser points or run a macro including the **Vectorize buildings** macro action on a TerraScan project.
7. Review and improve building models with the help of the *Check Building Models* tool and tools in the **Building Patches tool box** and the **Building Edges tool box**.

The quality of the automatic building vectorization depends on the quality of the laser data processing that is done in preparation of the vectorization, but also on the point density of the data. A higher point density results in more accurate models. The following number may serve as a guideline for estimating the possible results of the automatic vectorization:

- Low density < 2 points / m² - good models of large buildings, more problems with small buildings, loss of small details and roof structures
- Medium density 2-10 points / m² - good models
- High density > 10 points / m² - accurate models with details and roof structures

As alternative to laser data, TerraScan can also utilize line elements for the creation of 3D building models. The line elements must represent different types of roof edges, such as outer edges, internal edges along elevation jumps, and intersection lines, and they must form a closed line work for each building. From the line network, the *Construct Roof Polygons* tool tries to create closed polygons which represent roof planes. Finally, the *Create Buildings from Polygons* can be used to create the 3D vector models from the roof polygons.

The major advantage of this building vectorization approach is the automatic production of 3D building models for large areas in a comparatively short time. The process can also model complex roofs that are non-planar and contain a lot of detailed roof patches. The tools for improving the result of the automatic process are versatile and make the manual work fast and simple.

A disadvantage of the vectorization process is that it fully relies on the quality of the source data, which is either laser data or a line work for building roofs. If, for example, laser data is missing on parts of a building roof, there is no way to create at least an approximate building model based on the represented roof parts.

Single building vectorization

TerraScan's *Construct Planar Building* tool creates a vector model for one building at a time in a half-automatic way. Thus, it requires much more manual effort if several buildings need to be vectorized. On the other hand, it can be used if the automatic vectorization process fails.

The tool requires laser points that are classified into ground and above ground points and loaded into TerraScan. It can also utilize footprint polygons in order to start the detection of roof planes. The process tries to find roof planes in the point cloud and creates polygons for them automatically. Image data loaded in TerraPhoto supports the detection of roof planes. For manual editing, images in a camera view improve the result essentially, because edges of roofs, roof structures, and smaller details may not be detectable accurately in the laser data.

The workflow for single building vectorization includes the following steps:

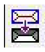
1. Match flightlines and cut off overlap.
2. Classify ground points using the **Ground** classification routine.
3. Classify high points which may be hits on building roofs using the **By height from ground** classification routine. This classification also includes points from high vegetation and other high objects.
4. (Optional) If images are available, load a mission and an image list into TerraPhoto. The camera parameters of the mission and the image list should be adjusted in order to provide accurately positioned images.
5. Create a vector model of a building using the *Construct Planar Building tool*.
6. Review and improve the building model with tools in the **Building Planes tool box** and the **Building Boundaries tool box**.
7. Repeat steps 5 and 6 for all buildings that need to be vectorized.

Although this building vectorization approach requires a lot of manual interaction, is has some advantages compared with the automatic approach described above. There are a few tools which allow the creation of roof planes without the availability of laser data. These tools can derive the plane equation from other sources than laser data and thus, enable the creation of building models in cases of missing laser data on roofs.

Another disadvantage of the approach is that it is impossible to model non-planar roofs or roofs that do not have dominating planar surfaces following a base direction.

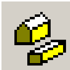
Buildings tool box

The tools in the **Buildings** tool box are used to create 3D building models automatically from point clouds, roof lines or roof polygons, and to check these building models.

To:	Use:	
Vectorize buildings from point cloud		*Vectorize Buildings*
Construct polygons from roof lines		*Construct Roof Polygons*
Create building models from roof polygons		*Create Buildings from Polygons*
Check building models one at a time		*Check Building Models*

Vectorize Buildings

Not Lite



Vectorize Buildings tool creates 3D building models based on loaded laser data. The laser points have to be classified into:

- ground points using the **Ground** classification routine.
- above-ground points which may be hits on building roofs using the **By height from ground** classification routine. This classification also includes points from high vegetation and other high objects.
- points on building roofs using the **Buildings** classification routine.

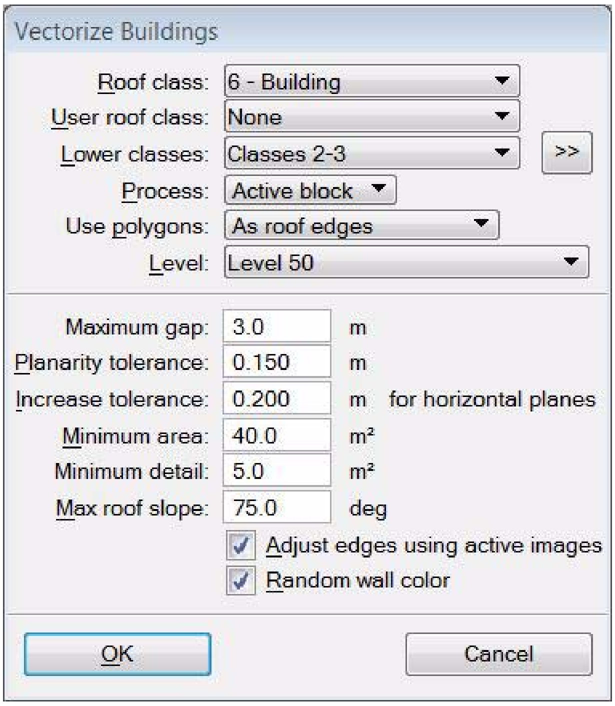
The tool creates MicroStation cell elements that contain shapes for each roof plane, possibly roof sides which determine the roof's thickness according to settings in **Building vectorization / Model**, and wall shapes for each outer roof edge.

Building vectorization can be also performed on project level by using the **Vectorize buildings** macro action.

➢ **To vectorize buildings based on loaded laser points:**

1. Load laser data into TerraScan.
2. Select the *Vectorize Buildings* tool.

   This opens the **Vectorize Buildings** dialog:



3. Define settings and click OK.

   This starts the vectorization process. It may take a while until the first models are created because the routine creates models for large buildings first. The building models are drawn into the design file according to the settings in **Building vectorization / Levels** category of TerraScan **Settings**.

Setting:	**Effect:**
Roof class	Point class consisting points on building roofs.
User roof class	Point class consisting points on building roofs that are ignored in the vectorization process.

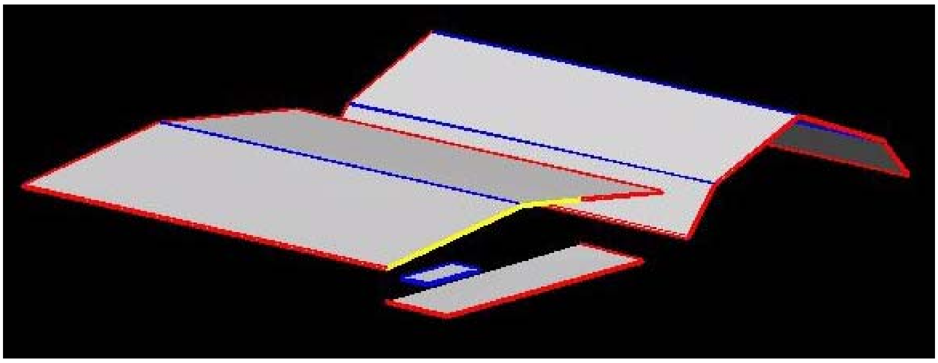
Setting:	Effect:
Lower classes	Point class(es) consisting points next to building roof edges, for example, ground or low vegetation. The points are used to determine the base elevation of building walls and help to place outer roof edges more accurately.
	Opens the **Select classes** dialog which contains the list of active classes in TerraScan. You can select multiple source classes from the list that are then used in the **Lower classes** field.
Process	Area to be processed: • **All points -** all loaded points are processed. This may include points from neighbour blocks. • **Active block -** points of the active block are processed. • **Inside fence -** points inside a fence or selected polygon are processed. This is only active if a MicroStation fence is drawn or a polygon is selected.
Use polygons	Defines how polygons are used in addition to laser data: • **Do not use -** no polygons are used. • **As bounding polygons -** polygons define boundaries that divide large building blocks into separate models. Example: land property polygons. • **As roof edges -** polygons define the XY shape of outer edges of buildings. Examle: footprint polygons.
Level	Design file level on which the polygons are located that are used in the vectorization process. This is only active if **Use polygons** is set to **As bounding polygons** or **As roof edges**.
Maximum gap	Maximum distance between building parts belonging to the same model. If the distance is larger, separate building models are created.
Planarity tolerance	Defines how closely a point must match a plane equation to belong to that roof plane.
Increase tolerance	Additional tolerance for merging close to horizontal planes together.
Minimum area	Minimum size of a building footprint.
Minimum detail	Minimum size of a building part footprint.
Max roof slope	Maximum gradient of a roof plane.
Adjust edges using active images	If on, building edges are adjusted based on images. The images must be referenced by an image list loaded into TerraPhoto.
Random wall color	If on, wall shapes are colored randomly by using a selection of colors from the active color table of MicroStation. If off, the color defined in **Building vectorization / Model** category of TerraScan **Settings** is used for all wall shapes.

Construct Roof Polygons

Construct Roof Polygons tool creates 3D roof polygons from selected 3D roof lines. It tries to determine closed polygons for each roof plane from the line work. The resulting polygons can then be used by *Create Buildings from Polygons* tool in order to create 3D building models.

The line elements must represent different types of roof edges, such as outer edges, internal edges along elevation jumps, and intersection lines, and they must form a closed line work for each building. The tool does not rely on lines being drawn on different levels or using different symbology. It tries to determine which elevations to keep and which to ignore in the polygon-building process only from the geometrical configuration of the line work.

The following figure illustrates the result of the roof polygon construction. Lines are roughly colored according to their roof edge type: outer edges = red, elevation jumps = yellow, intersection lines = blue. The resulting roof polygons are displayed in grey.



➢ **To construct roof polygons from lines:**

1. Select the lines that represent roof edges.
2. Select *Construct Roof Polygons* tool.

   This creates the polygons. The polygons are drawn on the active level using the active symbology settings of MicroStation. An information dialog shows the number of created polygons.

   It is recommended to check the polygons, for example, by using **Smooth** rendering display of MicroStation. This shows gaps or other issues in the roof polygons which may be caused by flaws in the line work. In this case, correct the line work and run the roof polygon construction again.

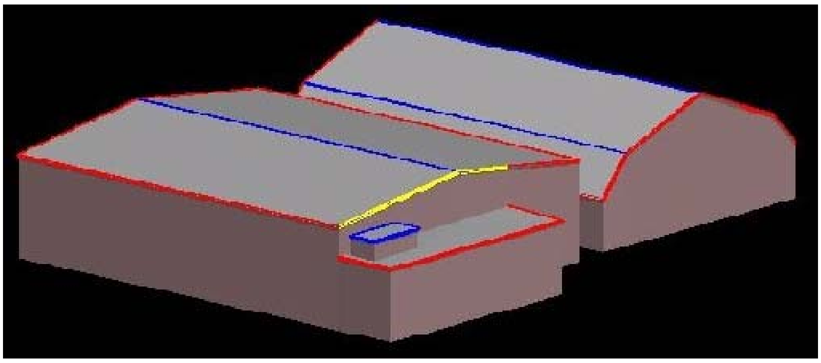
✍ You can undo the creation of roof polygons by using the **Undo** command from the **Edit** pulldown menu of MicroStation.

Create Buildings from Polygons

Create Buildings from Polygons tool creates 3D building models from selected 3D roof polygons. The polygons are usually a result of the *Construct Roof Polygons* tool.

The tools uses laser points to get the base elevation for walls. The building models are created as MicroStation cell elements that contain shapes for each roof plane, possibly roof sides which determine the roof’s thickness according to settings in **Building vectorization / Model**, and wall shapes for each outer roof edge.

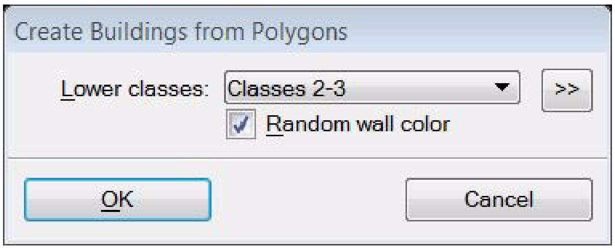
The following figure illustrates the result of the building model creation using the same example as shown for the *Construct Roof Polygons* tool.



➢ **To construct building models from polygons:**

1. Select the polygons that represent roof planes.
2. Select *Create Buildings from Polygons* tool.

   This opens the **Create Building from Polygons** dialog:



3. Define settings and click OK.

   This creates the building models. The building models are drawn into the design file according to the settings in **Building vectorization / Levels** category of TerraScan **Settings**.

Setting:	Effect:
Lower classes	Point class(es) consisting points next to buildings, for example, ground or low vegetation. The points are used to get the base elevation of the building walls.
	Opens the **Select classes** dialog which contains the list of active classes in TerraScan. You can select multiple source classes from the list that are then used in the **Lower classes** field.
Random wall color	If on, wall shapes are colored randomly by using a selection of colors from the active color table of MicroStation. If off, the color defined in **Building vectorization / Model** category of TerraScan **Settings** is used for all wall shapes.

✍ You can undo the creation of building models by using the **Undo** command from the **Edit** pull-down menu of MicroStation.

Check Building Models

Check Building Models tool can be used to check automatically created building models in an organized way. It is intended to be used after building vectorization by *Vectorize Buildings* tool or *Create Buildings from Polygons* tool.

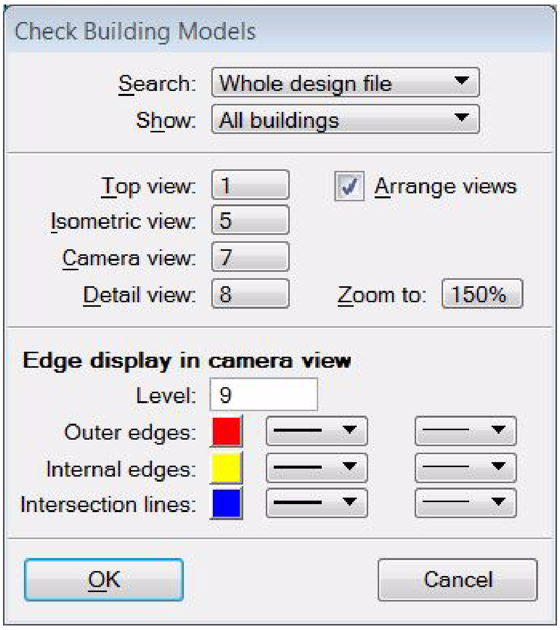
The **Check Building Models** dialog shows a list that contains building models in the design file that are drawn on the levels defined in **Building vectorization / Levels** category of TerraScan **Settings**. If a line in the list is selected, the software updates the display in a number of MicroStation views. The views must be open and defined in the tool settings. The tool can update different view types, such as top, isometric, and camera views which can show the building models on top of laser data and/or images.

With the help of the list, you can start to work with tools of the **Building Patches tool box** and the **Building Edges tool box** in order to improve the accuracy of the building models.

➢ **To check building models:**

1. Select *Check Building Models* tool.

   This opens the **Check Building Models** dialog:



2. Define settings and click OK.

   This opens the **Check Building Models** dialog.

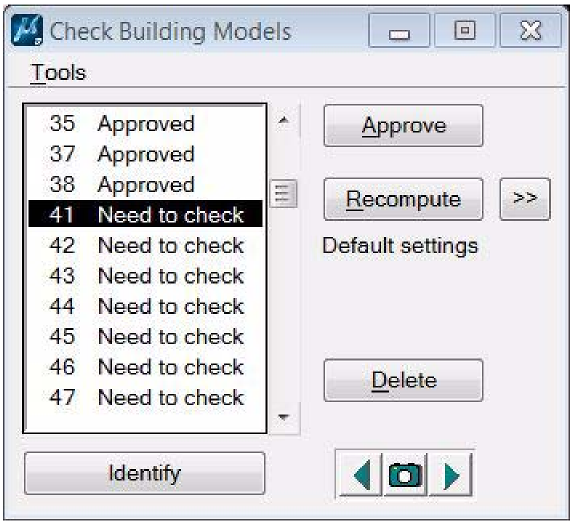
Setting:	Effect:
Search	Area where the software searches for building models: • **Whole design file** - all models in the design file. • **Active block** - all models inside the active project block. This is defined by the TerraScan project block that is loaded or was last loaded into TerraScan.
Show	Building models shown in the list of the **Check Building Models** dialog: • **All buildings** - all building models on levels defined in **Building vectorization / Levels**. • **Unchecked buildings** - building models on **Model to check** levels defined in **Building vectorization / Levels**.
Top view	A top view showing the active building model is displayed in the given view.

Setting:	Effect:
Isometric view	An isometric view showing the active building model. It is recommended to set the display style for this view to **Smooth** rendering in MicroStation view controls.
Camera view	A camera view showing the active building model is displayed in the given view. This view works only if a mission, camera, and image list are loaded into TerraPhoto.
Detail view	A camera view showing the location of an active building model in detail is displayed in the given view. The zoom level is determined by the given **Zoom to** value. This view works only if a mission, camera, and image list are loaded into TerraPhoto and if you select a building edge or corner for modification.
Arrange views	If on, the MicroStation views are arranged on the screen according to the given view settings. The software opens the views and places them within the MicroStation interface without overlap.
Level	Design file level on which the colored edges of an active model are drawn. The level should be switched on in the views that display the active model.
Outer edges	Display color, line weight, and line style of outer edges of the active model. Uses the active color table and standard line weights and styles of MicroStation.
Internal edges	Display color, line weight, and line style of internal edges of the active model. Uses the active color table and standard line weights and styles of MicroStation.
Intersection lines	Display color, line weight, and line style of intersection lines of the active model. Uses the active color table and standard line weights and styles of MicroStation.

Check Building Models

The **Check Building Models** dialog shows a list that contains all building models drawn on the design file levels that are defined in **Building vectorization / Levels** category of TerraScan **Settings**. The status of each model after automatic creation is *Need to check* by default.

Further, the dialog contains buttons that can be used to change the status of a model to *Approved*, to recompute or delete a model, to display a certain model, and to select another image that is displayed in the background of the active model. Menu commands can be used to display the **Vectorize Buildings** dialog and to change the status of all models to *Approved*.



To show the location of a building model, select a line in the **Check Building Models** dialog. This updates the display in all views that are set up for checking building models.

To identify a building model, click on the **Identify** button and place a data click close to a model in a view. This selects the corresponding model in the **Check Building Models** dialog.

After checking a building model and possibly modifying it, click on the **Approve** button. This changes the status of the selected model to *Approved*. It moves the model to the **Approved models** levels defined in **Building vectorization / Levels** category of TerraScan **Settings**.

Use the **Approve all** command from the **Tools** pulldown menu in order to change the status of all models in the list to *Approved*.

You can recompute a model by using the **Recompute** button. This might be necessary, if the settings of the automatic vectorization process did not provide a reasonable model for this building. These settings can be checked by using the **Computation settings** command from the **Tools** pulldown menu. The command opens the **Vectorize Buildings** dialog with the settings used in the automatic vectorization process.

➢ **To recompute a building model:**

1. Load laser data into TerraScan for the location of the building model.
2. Click on the >> button in order to open the **Vectorize Buildings** dialog.

   The settings of the dialog are described for the *Vectorize Buildings* tool.

3. Change settings in the dialog and click OK.
4. Click on the **Recompute** button.

   This recomputes the selected building model by using the tailored settings.

If you want to delete a model, click on the **Delete** button. This deletes the selected model from the list and from the design file. If you undo the delete action, the model is returned into the design file but not into the list of the dialog. You need to re-open the dialog in order to see the model again in the list.

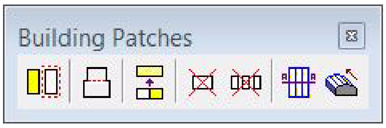
You can use the buttons in the lower right corner in order to select images from the TerraPhoto image list. The image is displayed in the camera views used for checking building models. By default, the software selects the image for display that sees the building (detail) location best.

Click on the camera button in the middle of the button group in order to identify an image for display. Move the mouse pointer into a view. The image footprint closest to the mouse pointer is dynamically displayed. Select an image for display with a data click.

Click on the arrow buttons left and right in the button group in order to select the previous or next image from the currently displayed image in the images list.

Building Patches tool box

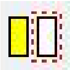
The tools in the **Building Patches** tool box are used to modify roof patches of 3D building models. The term “patch” is used for the single roof planes that form a roof.



To:	Use:
Split building into two separate models	*Split Building*
Split patch into two separate patches	*Split Patch*
Merge two patches into one	*Merge Patches*
Remove a patch by mouse click	*Remove Patch*
Remove small patches	*Remove Details*
Dispaly a building cross section	*Draw Building Section*
Extrude a building model from a cross section	*Extrude Building*

✍ **Building Patches** tools work only when the **Check Building Models** dialog is open. You can undo the actions of the tools by using the **Undo** command from the **Edit** pulldown menu of MicroStation.

Split Building



Split Building tool cuts out a part of a building complex. This part is then treated as own separate building model.

➢ **To split a building into two building models:**

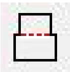
1. Draw a fence around the building part that you want to cut out.
2. Select *Split Building* tool.
3. Move the mouse pointer inside a MicroStation view.

   The two building parts are highlighted with blue and red coloring.

4. Accept the two building parts with a data click inside a view.

   This splits the building. The area outside the fence stays as active building. The area completely inside the fence becomes a new building model that is put at the end of the list in the **Check Building Models** dialog.

Split Patch

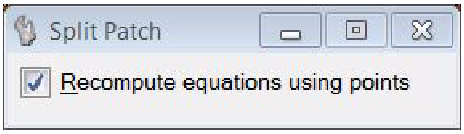


Split Patch tool splits a patch at edge vertices into two separate patches. The process can recompute the plane equations for the two patches if laser points of the roof class are loaded in TerraScan.

➢ **To split a patch:**

1. (Optional) Load laser data into TerraScan. Only points in the building roof class are required.
2. Select *Split Patch* tool.

   The **Split Patch** dialog opens:



3. Define, whether the process should **Recompute** plane **equations using points** or not. The setting is only available if points are loaded in TerraScan.
4. Move the mouse pointer inside a view.

   A potential vertex for splitting is dynamically highlighted if the mouse pointer comes close to it.

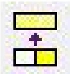
5. Select the first edge vertex with a data click.

   If you move the mouse pointer, the possible split lines are dynamically displayed.

6. Select the second edge vertex with a data click.

   This splits the patch into two patches and recomputes the plane equation of the patches, if applicable. You can continue with step 5.

Merge Patches



Merge Patches tool combines two neighboring patches into one patch. The process recomputes the plane equation for the new patch as average of the two original planes.

➢ **To merge two patches:**

1. Select *Merge Patches* tool.
2. Move the mouse pointer inside a view.

   Potential patches for merging are dynamically highlighted if the mouse pointer is inside a patch.
3. Select the first patch with a data click.

   If you move the mouse pointer, the possible patches for merging are dynamically displayed.
4. Select the second patch with a data click.

   This merges the two patches into one patch and recomputes the plane equation of the new patch. You can continue with step 3.

Remove Patch

Remove Patch tool removes a single building patch.

➢ **To remove a building patch:**

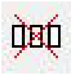
1. Select *Remove Patch* tool.
2. Move the mouse pointer inside a view.

   A patch is dynamically highlighted if the mouse pointer is inside a patch.
3. Select a patch with a data click.

   This removes the patch from the building model. You can continue with step 3.

✍ You can remove all patches of an active building model. Nevertheless, the model still exists in the list and stays active. You can apply additional processing steps, such as recomputing the model using the **Recompute** button of the **Check Building Models** dialog or creating a new model with the help of the *Extrude Building* tool. If you want to delete a model completely, use the **Delete** button of the **Check Building Models** dialog.

Remove Details

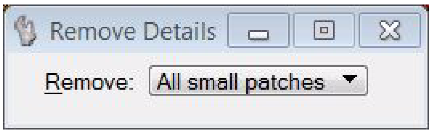


Remove Details tool removes all patches of a building roof which are of the same size or smaller than a patch identified by a data click. This can be used, for example, for removing patches of unnecessary structures on a roof, such as roof windows.

➢ **To remove a building patch:**

1. Select *Remove Details* tool.

   The **Remove Details** dialog opens:



2. Select a method for removing patches.
3. Move the mouse pointer inside a view.

   The patches effected by the removal action are dynamically highlighted if the mouse pointer is inside a patch.

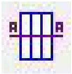
4. Select a patch with a data click.

   This removes all patches from the building model that are defined by the tool's setting. You can continue with step 4.

Setting:	Effect:
Remove	Defines what patches are effected by the removal action: • **All small patches -** all patches that are of the same size or smaller than the patch selected by the data click. • **Internal patches -** only patches that are completely inside a building roof are effected. The same size rules as for **All small patches** apply. • **Outer patches -** only patches that share an outer boundary of the building roof are effected. The same size rules as for **All small patches** apply.

✍ You can remove all patches of an active building model. Nevertheless, the model still exists in the list and stays active. You can apply additional processing steps, such as recomputing the model using the **Recompute** button of the **Check Building Models** dialog or creating a new model with the help of the *Extrude Building* tool. If you want to delete a model completely, use the **Delete** button of the **Check Building Models** dialog.

Draw Building Section



Draw Building Section tool displays a cross section of a building. The tool is intended to be used before *Extrude Building* tool. It provides a cross section view of the laser data that is suited for drawing a building roof profile line.

➢ **To draw a building section:**

1. Load laser data into TerraScan.
2. (Optional) Open an additional MicroStation view that can be used for the cross section display.
3. (Optional) Draw a fence around the area from which you want to create a cross section.
4. Select *Draw Building Section* tool.

   The **Draw Building Section** dialog opens:

5. Select an option for the tool setting.
6. Define a view with a data click.

   This displays the building cross section in the selected view. The software tries to select a cross section location and depth that is well suited for drawing a roof profile line.

Setting:	Effect:
Fit to show	Defines the area that is used to select a location for the cross section: • **Active building points** - area covered by all points that are inside the active building. • **Inside fence** - area covered by a MicroStation fence. This requires a fence element drawn into the design file.

Extrude Building

Extrude Building tool creates a building model by extruding a cross section line. The tool is especially useful for modeling buildings with round roofs. For such roof shapes, the automatic vectorization process usually does not provide a good result.

The cross section line of the building roof needs to be digitized manually based on a vertical section view of the laser data. The line string element should be placed on a design file level that is not used for building models. It can also be deleted after the building model has been created.

➢ **To create a building model from a cross section line:**

1. Remove all patches that you want to replace from the existing model by using *Remove Patch* or *Remove Details* tools.
2. Create a cross section view of the building by using the *Draw Building Section* tool.

   It is recommended to display the section in an additional MicroStation view.

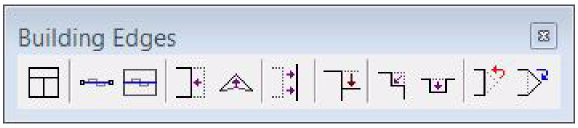
3. Digitize the shape of the cross section based on the laser data that is displayed in the section view. You can use any MicroStation tool for line string placement.
4. Select *Extrude Building* tool.
5. Select the cross section line with a data click.
6. Define the first edge of the building with a data click, preferable in a top view.
7. Define the second edge of the building with a data click, preferable in a top view.

   This creates a building model between the two edges defined by the data clicks. For each intermediate vertex of the cross section line, an intersection line is created in the building model. The outer boundaries of the new model are defined by the first and last vertex of the cross section line.

✍ You can use this tool to add or replace parts of an active building model.

Building Edges tool box

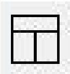
The tools in the **Building Edges** tool box are used to modify roof edges and corners of 3D building models.



To:	Use:
Set all edges of a building model to retangular	*Set All Edges*
Apply a straight line between two vertices	*Apply Straight Line*
Apply an intersection line of two patches	*Apply Intersection Line*
Move an edge of a building	*Modify Edge*
Move an edge vertex	*Move Edge Vertex*
Align an edge segment perpendicular or parallel to another edge	*Align Edge Segment*
Create a step corner	*Build Step Corner*
Cut an edge corner	*Cut Edge Corner*
Cut an edge segment	*Cut Edge Segment*
Delete a vertex from an edge	*Delete Edge Vertex*
Add a new vertex to an edge	*Insert Edge Vertex*

✍ **Building Edges** tools work only when the **Check Building Models** dialog is open. You can undo the actions of the tools by using the **Undo** command from the **Edit** pulldown menu of MicroStation.

Set All Edges



Set all edges tool adjusts all edges of roof patches. As a result, all patches are set to a rectangle or rectangular shape.

➢ **To set all edges:**

1. Select *Set All Edges* tool.

   The **Set All Edges** dialog opens:

2. Select a shape type for the patches.
3. Move the mouse pointer into a view.

   This displays the adjusted shape of patches as preview.

4. Apply the edge adjustment with a data click inside the view.

   This sets the edges and thus, the shape of the patches.

Setting:	Effect:
Set to	Defines the shape of the patches: • **Rectangle -** all patches are set to rectangles. • **Rectangular -** all patches are set to rectangular shapes.

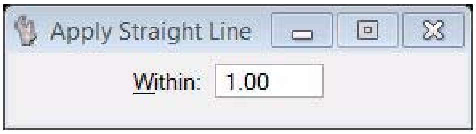
Apply Straight Line

Apply Straight Line tool moves all close by vertices to match a straight line between two selected vertices. Unnecessary vertices are removed from the resulting edge.

➢ **To apply a straight line:**

1. Select *Apply Straight Line* tool.

   The **Apply Straight Line** dialog opens:

2. Define an offset within which the vertices are moved to match the straight line.
3. Move the mouse pointer inside a view.

   The vertex closest to the mouse pointer is dynamically highlighted.

4. Define the first vertex of the straight connection line with a data click.

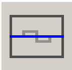
   If you move the mouse pointer, the area within which vertices are effected is dynamically displayed.

5. Define the second vertex of the straight connection line with a data click.

   This moves all vertices within the given offset to the straight line and deletes unnecessary intermediate vertices along the edge. You can continue with steps 2 or 4.

Setting:	**Effect:**
Within	Offset within which vertices are effected. Half of the given offset value applies to the left side and half to the right side of the straight line.

Apply Intersection Line

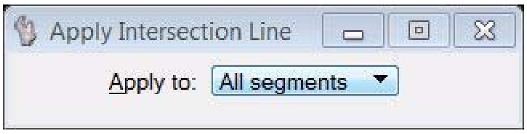


Apply Intersection Line tool replaces edge segment(s) between two planar patches with an intersection line of two planes. This may move the original edge segment(s) and vertices to another location in order to match the exact intersection of the planes. Unnecessary intermediate vertices are removed from the resulting intersection line.

➢ **To apply an intersection line:**

1. Select *Apply Intersection Line* tool.

   The **Apply Intersection Line** dialog opens:



2. Define whether all segments or only one segment is effected by the tool.
3. Move the mouse pointer inside a view.

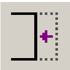
   The location of the intersection line between two patches closest to the mouse pointer is dynamically highlighted.
4. Accept the intersection line with a data click.

   This sets the intersection line, adjusts vertices if necessary, and deletes unnecessary intermediate vertices along the intersection line. You can continue with step 2 or 4.

Setting:	Effect:
Apply to	Defines the edge segments effected by the tool: • **All segments** - all edge segments are replaced by the intersection line. • **One segment** - only one edge segment is replaced by the intersection line.

✍ The tool adjusts vertices in order to apply an intersection line between two patches. If several intersection lines are connected in one vertex, it might be necessary to apply the tool several times to the edges. Then, the location of the vertices is more and more refined until intersection lines can be applied to all edges.

Modify Edge



Modify Edge tool moves an edge vertex or segment. The modification effects all parallel segments of the same edge.

➢ **To modify an edge vertex or segment:**

1. Select *Modify Edge* tool.

   The **Modify Edge** dialog opens:

2. Define setting.
3. Move the mouse pointer inside a view.

   The edge segment or vertex closest to the mouse pointer is dynamically highlighted.
4. Define the edge segment or vertex to move with a data click.

   This updates the **Detail view** and displays the image that sees the selected edge/vertex location best. If you move the mouse pointer, the new edge segment or vertex location is dynamically displayed.
5. Define the new location of the edge segment or vertex with a data click.

   This places the edge segment or vertex at the new location and adjusts all other parallel segments along the same edge accordingly. You can continue with steps 2 or 4.

Setting:	**Effect:**
Vertex	• **Free** - . • **Fixed** - .

Move Edge Vertex

Move Edge Vertex tool moves a vertex. The modification only effects the edge segments that are connected at the vertex but does not move other parallel edge segments.

➢ **To modify an edge vertex or segment:**

1. Select *Move Edge Vertex* tool.

   The **Move Edge Vertex** dialog opens:

2. Define setting.
3. Move the mouse pointer inside a view.

   The vertex closest to the mouse pointer is dynamically highlighted.

4. Define the vertex to move with a data click.

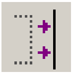
   This updates the **Detail view** and displays the image that sees the selected vertex location best. If you move the mouse pointer, the new vertex location is dynamically displayed.

5. Define the new location of the vertex with a data click.

   This places the vertex at the new location. You can continue with steps 2 or 4.

Setting:	Effect:
Vertex	• **Free** - . • **Fixed** - .

Align Edge Segment



Align Edge Segment tool moves an edge segment. At the same time, it aligns the edge segment according to a base direction defined by a reference edge segment. The alignment is either parallel or perpendicular to the reference edge segment.

➢ **To align an edge segment:**

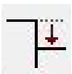
1. Select *Align Edge Segment* tool.
2. Move the mouse pointer inside a view.

   The edge segment closest to the mouse pointer is dynamically highlighted.
3. Define the reference edge segment with a data click. This defines the base direction.
4. Define the edge segment to align with a data click.

   This updates the **Detail view** and displays the image that sees the selected edge segment location best. If you move the mouse pointer, the new edge location is dynamically displayed.
5. Define the new location of the edge segment with a data click.

   This aligns and places the edge segment at the new location. You can continue with step 4. After placing a reset click, you continue with step 3.

Build Step Corner



Build Step Corner tool detaches a vertex and moves it along an incoming/outgoing edge segment. Only the vertex is moved, the effected segment should be aligned in a separate step using the *Align Edge Segment* tool.

➢ **To build a step corner:**

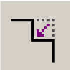
1. Select *Build Step Corner* tool.
2. Move the mouse pointer inside a view.

   The edge segment and vertex to be detached closest to the mouse pointer is dynamically highlighted.
3. Define the vertex to detach and move with a data click.

   This updates the **Detail view** and displays the image that sees the selected vertex location best. If you move the mouse pointer, the new vertex location is dynamically displayed.
4. Define the new location of the vertex with a data click.

   This places the vertex at the new location. You can continue with step 3.

Cut Edge Corner



Cut Edge Corner tool can modify a patch corner in two ways. It cuts off a piece from a corner or it adds a piece to a corner. In any case, the new edges are aligned perpendicular to the edges the form the original corner.

➢ **To add/cut off a piece to/from an edge corner:**

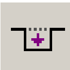
1. Select *Cut Edge Corner* tool.
2. Move the mouse pointer inside a view.

   The edge corner closest to the mouse pointer is dynamically highlighted.
3. Define the edge corner to modify with a data click.

   This updates the **Detail view** and displays the image that sees the selected corner location best. If you move the mouse pointer, the new edge of the corner is dynamically displayed.
4. Define the location of one edge segment with a data click.
5. Define the location of the other edge segment with a data click.

   This places the new corner at the defined location. You can continue with step 3. You can go back from steps 5 to 4 and 4 to 3 by placing a reset click.

Cut Edge Segment



Cut Edge Segment tool can modify an edge segment in two ways. It cuts off a piece from a segment or it adds a piece to a segment. The cut off or added part is formed by three new edge segments of which two are perpendicular and one is parallel to the original edge segment.

➢ **To add/cut off a piece to/from an edge segment:**

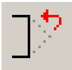
1. Select *Cut Edge Segment* tool.
2. Move the mouse pointer inside a view.

   The edge closest to the mouse pointer is dynamically highlighted.
3. Define the edge to modify with a data click.

   This updates the **Detail view** and displays the image that sees the selected edge location best. If you move the mouse pointer, the new edge is dynamically displayed.
4. Define the location of one perpendicular edge segment with a data click.
5. Define the location of the other perpendicular edge segment with a data click.
6. Define the location of the parallel edge segment with a data click.

   This places the new edge segments at the defined locations. You can continue with step 3. You can go back from steps 6 to 5, 5 to 4, and 4 to 3 by placing a reset click.

Delete Edge Vertex



Delete Edge Vertex tool removes a vertex from an edge. Only vertices that connect two edge segments can be removed.

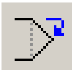
➢ **To delete an edge vertex:**

1. Select *Delete Edge Vertex* tool.
2. Move the mouse pointer inside a view.

   The vertex closest to the mouse pointer is dynamically highlighted.
3. Define the vertex to delete with a data click.

   This removes the vertex. You can continue with steps 3.

Insert Edge Vertex



Insert Edge Vertex tool adds a new vertex to an edge segment. It also defines the location of the new vertex.

➢ **To add an edge vertex:**

1. Select *Move Edge Vertex* tool.
2. Move the mouse pointer inside a view.

   The edge segment closest to the mouse pointer is dynamically highlighted.
3. Define the edge segment to which to add a vertex with a data click.

   This updates the **Detail view** and displays the image that sees the selected edge location best. If you move the mouse pointer, the new vertex location is dynamically displayed.
4. Define the location of the new vertex with a data click.

   This adds the vertex and places the edge segments according to the location of the new vertex. You can continue with step 3.

Construct Planar Building tool

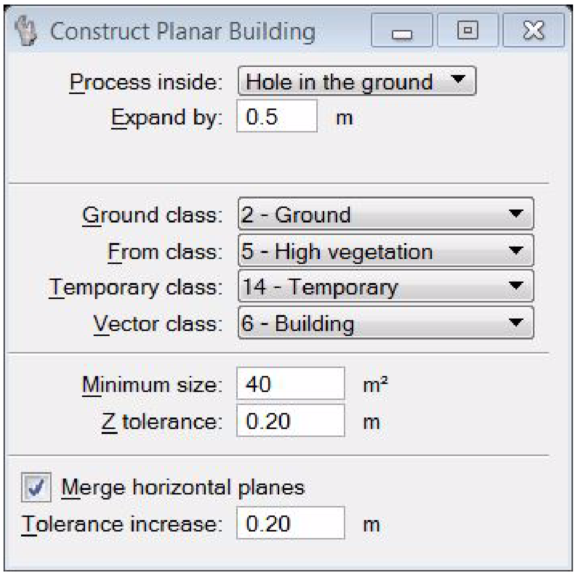
Construct Planar Building tool from the **Draw tool box** is used to create a 3D vector model of a building based on laser points on planar surfaces of the roof. With this tool, one building at a time can be vectorized in a half-automatic way.

The tool requires the classification of ground points and points above the ground which include points on building roofs, usually “high vegetation” points. Starting from a hole in ground points or a footprint element, the tool finds planar surfaces of roofs. For each planar part it creates a roof polygon. The roof polygons can be edited by a set of tools provided for building vectorization. After the roof planes are fixed, the final 3D vector model of a building is created by drawing vertical walls for all outer edges of the roof.

➢ **To construct a building:**

1. Load laser points into TerraScan.
2. Select the *Construct Building* tool.

   This opens the **Construct Planar Building** dialog:



3. Define settings.
4. Identify the building location with a data click. This can be a hole in the ground points or a footprint element.

   The software searches for roof planes and creates temporary polygons for each detected roof plane. The polygons are drawn on the active level. It also classifies points inside the roof planes into the given **Temporary class**.

   It opens the **Construct Building dialog** that shows a list of all detected roof planes. The dialog provides user control elements for modifying the building model and for changing settings for the model display and construction. In addition, the **Building Planes tool box** and **Building Boundaries tool box** are opened for editing the roof planes.

   By default, the MicroStation views are arranged in a way that the building model is shown in a top view, an isometric view, and two section views. If a mission, camera file, and image list are loaded in TerraPhoto, a camera view shows the model on top of an image.

Setting:	Effect:
Process inside	Area to search for potential roof points: • **Hole in the ground** - area without any ground points. • **Footprint element** - inside a polygon that also defines the location of the walls. • **Fence element** - inside a polygon.
Expand by	Distance for expanding the roof planes from the actual building edges in the source data.
Auto direct by footprint	If on, the building model is directed according to the footprint element. This is only active if **Process inside** is set to **Footprint element**.
Ground class	Point class containing points on the ground.
From class	Point class from which building roofs are detected.
Temporary class	Point class into which points are classified after roof detection.
Vector class	Point class into which points are classified after the final building model has been created.
Minimum size	Minimum size of the area covered by a building.
Z tolerance	Elevation tolerance within which points are considered to belong to the same roof plane. Estimated noise level in the laser data.
Merge horizontal planes	If on, separately detected horizontal planes are merged into one plane.
Tolerance increase	Plane fitting tolerance value for merging horizontal planes together.

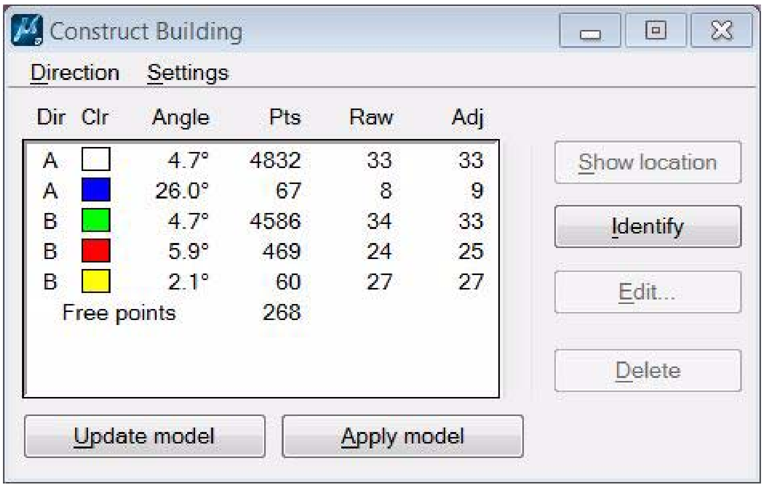
Construct Building dialog

The **Construct Building** dialog shows a list that contains all roof planes of a building model drawn in the design file. At this point, the software has drawn temporary polygons for each roof plane. Only after applying the model, the final 3D model for the building is created.

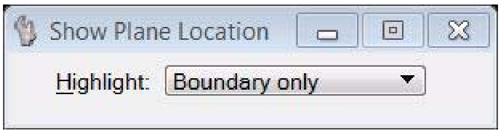
For each plane, the list of the **Construct Building** dialog contains the following information:

- **Dir -** direction group indicated by a letter. If the building has been constructed based on a footprint polygon, the angle between the footprint polygon direction and the plane direction is displayed.
- **Clr -** color of the outline.
- **Angle -** slope angle.
- **Pts -** laser points covered by the plane.
- **Raw -** initial direction angle of roof planes detected from the laser data.
- **Adj -** direction angle of adjusted roof planes.
- **Free points -** number of laser points in the building roof class that are not covered by a roof plane.

Further, the dialog contains buttons that can be used to update, apply, or delete a model, to display or identify a certain roof plane, and to edit the appearance of a plane polygon. Menu commands can be used to apply directions to planes, recompute planes, and to change settings related to the building construction mode.



To show the location of a roof plane, select a line in the **Construct Building** dialog. Click on the **Show location** button, define the setting in the **Show Plane Location** dialog, and move the mouse pointer into a view. This highlights the selected plane according the settings in the dialog:



Setting:	Effect:
Highlight	Defines what elements are highlighted: • **Boundary and Points -** the plane boundary and laser points inside the plane. • **Boundary only -** the plane boundary. • **Points only -** the laser points inside the plane. Highlighting laser points might slow down the display significantly.

To identify a plane, click on the **Identify** button and place a data click inside a plane in a view. This selects the corresponding line in the **Construct Building** dialog.

The **Edit** button opens the **Plane information** dialog which lets you modify a roof plane that is selected in the list. In the dialog, you can change the display color of the temporary plane polygon and its boundary type. The boundary type can be set to **Rectangle**, **Rectangular**, or **Polygon**. If you change the boundary type from a less-restrictive to a more-restrictive type, for example from polygon to rectangular or rectangle, the plane boundaries are adjusted.

If you want to delete a roof plane, click on the **Delete** button. This removes the selected plane from the list. The laser points are not effected by this action.

The **Update model** button creates a 3D model from the roof planes. It constructs polygons for upper roof planes, roof sides (eaves), lower roof planes, and walls. It does not effect the laser points. The temporary plane polygons are still displayed and can be modified as long as the **Construct Building** dialog is open. Use this button if you want to check the 3D model but still do adjustments to the roof planes.

The **Apply model** button creates the final 3D model from the roof planes. It constructs polygons for upper roof planes, roof sides (eaves), lower roof planes, and walls. It classifies the laser points inside the roof planes into the **Vector class** defined in the **Construct Planar Building** dialog. It also closes the **Construct Building** dialog and removes the display of the temporary plane polygons. Thus, the model can no longer be modified with the TerraScan building vectorization tools once it has been applied.

✍ Only **Apply model** classifies laser points into the defined **Vector class**. If you use **Update model** to create the model and then close the **Construct Building** dialog, the vector model is drawn into the design file but the laser points on the roof are still in their original class.

Direction of roof planes

Commands from the **Direction** pulldown menu can be used to apply a new base direction to roof planes or to recompute the boundaries and base direction of roof planes.

Command:	**Effect:**
Apply	Applies a direction to all or selected planes: • **Average direction** - the average direction of planes is applied to all planes. • **Best plane direction** - the optimal direction is applied for each plane. • **Footprint direction** - the direction defined by a footprint polygon is applied to all planes. This is only active if the roof planes have been constructed based on a footprint polygon. • **Two points direction** - the direction of a roof plane is defined by a line. You draw the line by placing two data clicks inside the plane you want to adjust.
Recompute	Recomputes the boundaries and direction of planes: • **All planes** - all planes. • **Selected group(s)** - only planes of the group of which at least one plane is selected in the list. • **Selected plane(s)** - only planes that are selected in the list.

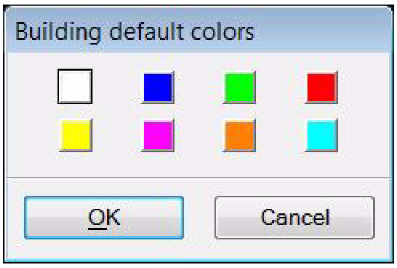
Settings for Construct Building

Commands from the **Settings** pulldown menu can be used to modify settings for the display setup in the building construction mode and for the final 3D building model construction.

➢ **To change the default colors for the display of temporary roof plane polygons:**

1. Select **Default color** command from the **Settings** pulldown menu.

The **Building default colors** dialog opens:



2. Click on a color button in the dialog.

   The active color table of MicroStation is displayed.

3. Select a new color from the table.
4. Click OK.

   The changes become effective for the next building model you create with the *Construct Planar Building* tool.

➢ **To change the view setup for the construct building mode:**

1. Select **View setup** command from the **Settings** pulldown menu.

   The **Construct Building Views** dialog opens:

2. Define settings and click OK.

   This updates the display of the building model in all effected views.

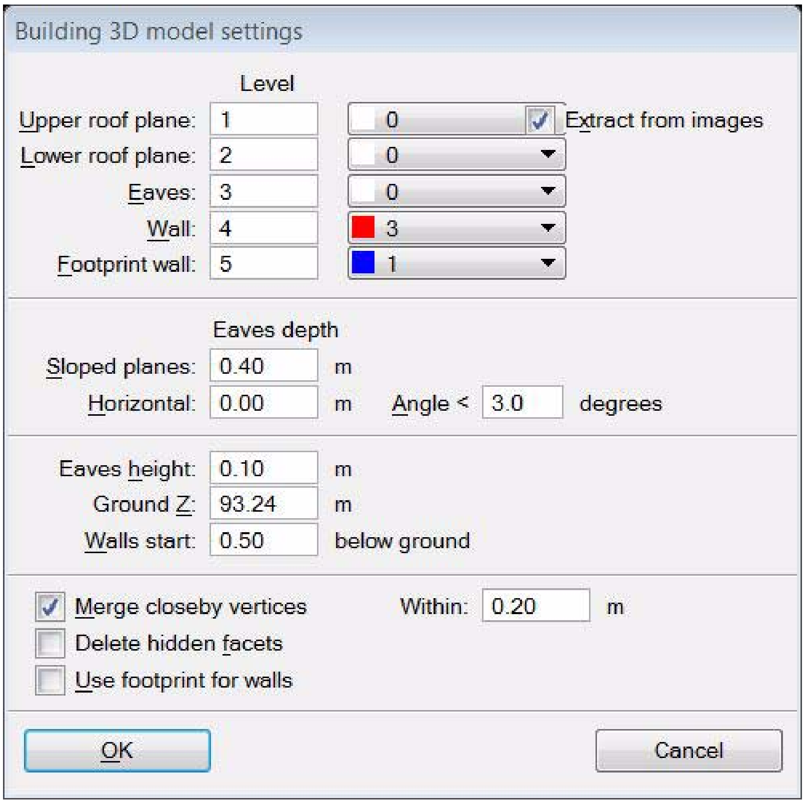
Setting:	Effect:
Top view	Top view that displayes the building model.
Section view 1	Section view that displayes a cross section perpendicular to the building's main direction.
Section view 2	Section view that displays a section along the building's main direction.
Isometric view	Isometric view that displayes the building model. You may set this view to display the model using **Smooth** rendering in MicroStation view controls.
Arrange view	If on, the selected views are opened and arranged in the MicroStation interface.
Display shaded surface	If on, a surface model of the roof planes is created in TerraModeler and displayed in the **Isometric view**. This requires that TerraModeler is available on the computer. It starts TerraModeler automatically if it is not yet loaded.

Setting:	Effect:
Camera view	Camera view that displays the building model on top of an image. This requires that TerraPhoto is available on the computer and loaded. It also requires a mission, camera file, and image list loaded into TerraPhoto.
Rotation	Defines the rotation of the camera view: • **Image direction** - the display is rotated according to the image direction. • **Building direction** - the display is rotated according to the main building direction.
Use images in Auto Align Boundaries	If on, the *Auto Align Boundaries* tool aligns roof plane boundaries to the roof edges visible in images.

➢ **To change the settings for constructing the final 3D building model:**

1. Select **3D Model** command from the **Settings** pulldown menu.

   The **Building 3D model settings** dialog opens:



2. Define settings and click OK.

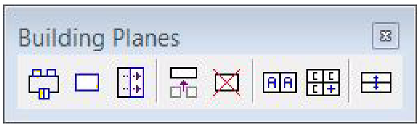
   The changes become effective for the next building model you update or apply with the corresponding buttons of the **Contruct Building** dialog.

Setting:	Effect:
Upper roof plane	Level and color used for drawing the upper roof planes. Uses the active color table of MicroStation.
Extract from images	If on, the color of the upper roof plane is extracted from the image that sees the building location best. The resulting color is the average of all pixels on the roof in the image.
Lower roof plane	Level and color used for drawing the lower roof planes. Uses the active color table of MicroStation.
Eaves	Level and color used for drawing the roof side planes (eaves). Uses the active color table of MicroStation.

Setting:	Effect:
Wall	Level and color used for drawing the walls. Uses the active color table of MicroStation.
Footprint walls	Level and color used for drawing the walls at a location defined by a footprint polygon. Uses the active color table of MicroStation.
Sloped planes	Depth of eaves if the roof planes are sloped. The value determines the distance between the outer roof boundary and the location of walls to the inside of the roof planes.
Horizontal	Depth of eaves if the roof planes are horizontal.
Angle	Determines the maximum slope anlge of a roof plane that is still considered as horizontal.
Eaves height	Height of eaves. The value determines the distance between upper and lower roof planes.
Ground Z	Elevation of the ground level around the building. The default value is determined by the laser points in the ground class.
Walls start	Defines the distance between the **Ground Z** value and the lower edge of walls.
Merge closeby vertices	If on, vertices on roof plane boundaries are merged if they are closer to each other than the distance given in the **Within** field.
Delete hidden facets	If on, hidden facets are not constructed in the final model.
Use footprints for walls	If on, additional wall shapes are drawn at the location of footprint boundaries. This is only applicable if the building model has been construced based on a footprint polygon.

Building Planes tool box

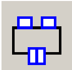
The tools in the **Building Planes** tool box are used to modify roof planes of a 3D building model.



To:	Use:
Add additional roof details	*Find Detail Planes*
Add a new roof plane manually	*Add Building Plane*
Mirrow a roof plane	*Mirror Building Plane*
Merge two roof planes into one plane	*Merge Building Planes*
Remove a roof plane	*Delete Building Plane*
Create a new plane direction group	*Create Direction Group*
Add a plane to a directon group	*Add Plane To Group*
Force planes to symmetric gradients	*Assign Plane Symmetry*

✍ **Building Planes** tools work only when the **Construct Building dialog** is open. You can undo the actions of the tools by using the **Undo** command from the **Edit** pulldown menu of MicroStation. However, this is not recommended because the display of the temporary plane polygons is not updated correctly after using the **Undo** command in MicroStation V8i.

Find Detail Planes



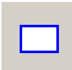
Find Detail Planes tool searches for free points in the source class of roof plane detection and creates additional planes from them.

➢ **To find detail planes:**

1. Select *Find Detail Planes* tool.

   This adds planes for details and updates the list of planes in the **Construct Building** dialog.

Add Building Plane

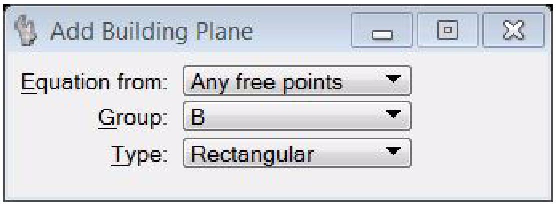


Add Building Plane tool lets you add a new roof plane to a building model. The tool can derive the plane equation for the new plane from different sources, such as laser points, the elevation of a data click, a section line defined by two data clicks, or an existing 3D shape drawn in the design file. Thus, the tool can create roof planes for places where there are no laser points on a building roof part.

➤ **To add a new roof plane:**

1. (Optional) Select a shape element that you want to use as a roof plane.
2. Select *Add Building Plane* tool.

   The **Add Building Plane** dialog opens:

3. Define settings.

   The next steps depend on the source data used to derive the plane equation from.

Equation from Any free points or Fenced points:

4. Define the first vertex of the plane with a data click.
5. Define additional vertices with data clicks.

   If you digitize a plane of type **Rectangle**, the tool lets you place only four vertices.

6. If you digitize a plane of type **Rectangular** or **Polygon**, define the last vertex at the same location as the first vertex. The tool snaps to the first vertex if the mouse pointer comes close to it.

Equation from Elevation click:

4. Define the elevation of the plane with a data click inside a section view.
5. Continue with digitizing the shape of the plane as described above.

Equation from Section line:

4. Define the slope of the plane with two data clicks inside a section view, one data click for the start point of the section line and another data click for the end point.
5. Continue with digitizing the shape of the plane as described above.

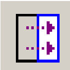
Equation from Element:

4. Accept the shape element selected in step 1 with a data click.

   This adds the new plane to the roof model and to the list of planes in the **Construct Building** dialog. The laser points inside the plane are not effected.

Setting:	Effect:
Equation from	Source data from which the software derives the plane equation: • **Any free points -** laser points of the original roof point class. • **Fenced points -** laser points of the original roof class and inside a MicroStation fence. This is only active if a fence is drawn in the design file. • **Elevation click -** the first data click defines the elevation of the plane. The plane is created as horizontal plane. • **Section line -** the first two data clicks define the slope of the plane in a section. • **Element -** the plane is created from an exising 3D polygon. This requires that the polygon is selected before the tool is started.
Group	Direction group to which the new plane is added.
Type	Shape type of the new plane: **Rectangle**, **Rectangular**, or **Polygon**.

Mirror Building Plane



Mirror Building Plane tool create a new roof plane by mirroring an existing plane. This is useful if one side of a symmetrical roof is represented in the laser data but the other side does not have enough laser points for the automatic creation of a roof plane.

The tool mirrors the shape and the slope of the reference plane. It lets you place the location of the new plane by two data clicks.

➢ **To create a new roof plane by mirroring another plane:**

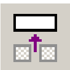
1. Select *Mirror Building Plane* tool.
2. Move the mouse pointer inside a view.

   The roof plane closest to the mouse pointer is highlighted.
3. Define the reference plane with a data click.
4. Define the origin point of the new plane with a data click.

   The new plane is displayed dynamically at the mouse pointer location.
5. Define the destination point of the new plane with a data click.

   This adds the new plane to the roof model and to the list of planes in the **Construct Building** dialog. The laser points inside the plane are not effected.

Merge Building Planes



Merge Building Planes tool combines two neighboring roof planes into one plane. The process recomputes the plane equation for the new plane by giving priority to the first selected plane.

➢ **To merge two planes:**

1. Select *Merge Building Planes* tool.
2. Move the mouse pointer inside a view.

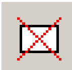
   Potential planes for merging are dynamically highlighted if the mouse pointer is inside a plane.
3. Select the first plane with a data click. This defines the master plane for the merging process which determines the slope and direction of the merged plane.

   If you move the mouse pointer, the possible planes for merging are dynamically displayed.
4. Select the second plane with a data click.

   This opens an alert dialog which informs about the amount of laser points that are outside the planarity tolerance of the merged plane.
5. Decide whether to keep the points in the merged plane or not and click the corresponding button in the alert dialog.

   This merges the two planes into one plane, recomputes the plane equation of the new plane, and updates the list of planes in the **Construct Building** dialog.

Delete Building Plane



Delete Building Plane tool removes a single roof plane. You identify the plane to be deleted by a data click.

You can also delete a plane by selecting it in the list of planes in the **Construct Building** dialog and click on the **Delete** button.

➢ **To delete a roof plane:**

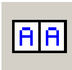
1. Select *Delete Building Plane* tool.
2. Move the mouse pointer inside a view.

   A plane is dynamically highlighted if the mouse pointer is inside the plane.

3. Select a plane with a data click.

   This removes the plane from the roof model and from the list of planes in the **Construct Building** dialog. The laser points inside the plane are not effected. You can continue with step 3.

Create Direction Group



Create Direction Group tool adds planes to a new direction group. The direction of the group can be determined from different sources, such as the planes themselves, a direction line, or a footprint polygon.

➢ **To create a direction group:**

1. Select *Create Direction Group* tool.

   The **Create Direction Group** dialog opens:

2. Define how the software determines the direction of the group.

   The next steps depend on the source data used to derive the direction from.

Direction from Average, First plane, or Footprint:

3. Identify the first plane you want to add to the group with a data click inside the plane. If **Direction** is set to **First plane**, this defines the plane for determining the direction of the group.
4. Identify additional planes with data clicks.
5. Place a data click outside the building area in order to apply the direction group.

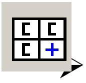
Equation from Two points:

3. Define the direction of the plane group with two data clicks, one data click for the start point of the direction line and another data click for the end point.
4. Continue with identifying the planes you want to add to the group as described above.

   This creates the new direction group, adds the planes to the group, and updates the list of planes in the **Construct Building** dialog.

Setting:	Effect:
Direction	Source data from which the software derives the direction of the group: • **Average -** the average direction of all planes added to the group. • **Two points -** a line defined by two points. • **First plane -** the direction of the first selected plane. • **Footprint -** the direction of the footprint polygon. This is only active if the roof planes have been constructed based on a footprint polygon.

Add Plane To Group



Add Plane To Group tool adds a roof plane to an existing direction group.

➢ **To add a plane to a direction group:**

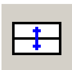
1. Select *Add Plane To Group* tool.
2. Move the mouse pointer inside a view.

   A plane is dynamically highlighted if the mouse pointer is inside the plane. The group letter is displayed as well.

3. Identify a plane with a data click. The plane determines the direction group to which additional planes are added.
4. Identify the plane(s) you want to add to the direction group.
5. Place a data click outside the building model in order to apply the direction group.

   This adds the planes to the group and updates the list of planes in the **Construct Building** dialog. You can continue with step 3.

Assign Plane Symmetry

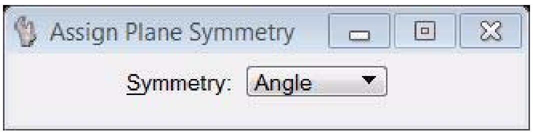


Assign Plane Symmetry tool enforces symmetry between two or more roof planes. The symmetry can effect the slope angle of the planes or the plane equation.

➢ **To enforce symmetry of planes:**

1. Select *Assign Plane Symmetry* tool.

   The **Assign Plane Symmetry** dialog opens:



2. Define the symmetry type you want to assign.
3. Move the mouse pointer inside a view.

   A plane is dynamically highlighted if the mouse pointer is inside the plane.

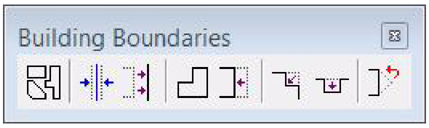
4. Identify a plane with a data click.
5. Identify another plane(s) you want to adjust symmetrically.
6. Place a data click outside the building model in order to apply the symmetry.

   This adjusts the planes and updates the list of planes in the **Construct Building** dialog. You can continue with steps 2 or 4.

Setting:	Effect:
Symmetry	Determines the effect of the symmetric adjustment: • **None -** any symmetry of planes is released. This can be used to undo symmetry assignments. • **Angle -** the slope angle of planes is symmetrically adjusted. • **Equation -** the plane equation is symmetrically adjusted.

Building Boundaries tool box

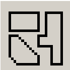
The tools in the **Building Boundaries** tool box are used to modify roof plane boundaries and corners of a 3D building model.



To:	Use:
Set the boundaries of plane to a fixed shape	*Set Boundary Type*
Align boundaries automatically	*Auto Align Boundaries*
Align a boundary segment another boundary	*Align Boundary Segment*
Digitize new boundaries for a plane	*Place Boundary Shape*
Move a boundary segment or a vertex of a plane	*Modify Boundary Shape*
Cut a boundary corner	*Cut Boundary Corner*
Cut a boundary segment	*Cut Boundary Segment*
Delete a vertex from a plane boundary	*Delete Boundary Vertex*

✍ **Building Boundaries** tools work only when the **Construct Building dialog** is open. You can undo the actions of the tools by using the **Undo** command from the **Edit** pulldown menu of MicroStation. However, this is not recommended because the display of the temporary plane polygons is not updated correctly after using the **Undo** command in MicroStation V8i.

Set Boundary Type



Set Boundary Type tool adjusts boundaries of a roof plane. As a result, the plane is set to a rectangle, rectangular or polygon shape. The boundary type of a plane effects, for example, the operation mode of other tools for plane boundary modification.

➢ **To set all edges:**

1. Select *Set Boundary Type* tool.

   The **Set Boundary Type** dialog opens:

2. Select a shape type for the plane.
3. Move the mouse pointer into a view.

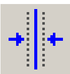
   This highlights a roof plane if the mouse pointer is inside the plane.

4. Apply the boundary adjustment with a data click inside the plane.

   This sets the boundaries and thus, the shape of the plane. You can continue with steps 2 or 4.

Setting:	Effect:
Type	Shape type: • **Rectangle** - the plane is set to a rectangle. • **Rectangular** - the plane is set to a rectangular shape. • **Polygon** - the plane is set to a polygon shape.

Auto Align Boundaries



Auto Align Boundaries tool aligns internal plane boundaries automatically. It fixes their location to an intersection line between two planes or to a shared internal boundary that represents an elevation change between roof planes. Thus, the tool can be used to close gaps between planes of a roof.

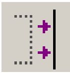
The internal boundaries of different planes should already be close to each other and parallel before the automatic alignment is applied.

➢ **To align boundaries automatically:**

1. Select *Auto Align Boundaries* tool.

   This aligns the internal boundaries.

Align Boundary Segment



Align Boundary Segment tool aligns a boundary segment to the location and direction defined by a reference boundary segment.

➢ **To align a boundary:**

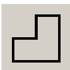
1. Select *Align Boundary Segment* tool.
2. Move the mouse pointer inside a view.

   The boundary segment closest to the mouse pointer is dynamically highlighted.
3. Define the boundary segment to align with a data click.

   If you move the mouse pointer, any potential reference boundaries are dynamically highlighted. The aligned boundary is displayed in a preview.
4. Define the reference boundary segment with a data click.

   This aligns and places the boundary segment at the new location. You can continue with step 4. After placing a reset click, you continue with step 3.

Place Boundary Shape

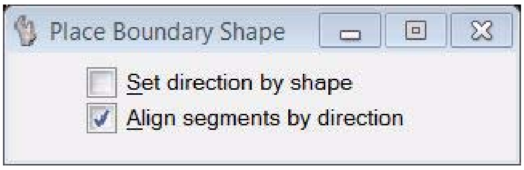


Place Boundary Shape tool lets you replace a roof plane by digitizing a new boundary for the plane. The operation mode of the tool depends on the shape type of the plane.

➢ **To place new boundaries:**

1. Select *Place Boundary Shape* tool.

   The **Place Boundary Shape** dialog opens:



2. Define settings.
3. Move the mouse pointer into a view.

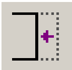
   This highlights a roof plane if the mouse pointer is inside the plane.
4. Identify the plane for which you want to digitize new boundaries with a data click.
5. Digitize a new boundary by placing a data click for each vertex of the plane.

   The amount of vertices you can place are effected by the shape type. Place the last data click close to the first vertex in order to close a rectangular or polygon shape.
6. Apply the boundary of the plane with a another data click.

   This updates the boundaries and thus, sets the new shape of the plane. You can continue with steps 2 or 4.

Setting:	**Effect:**
Set direction by shape	If on, the direction of the plane is set according to the basic direction of the new boundary.
Align segments by direction	If on, the boundary segments are aligned automatically to follow the basic direction of the plane. The direction is set according to the direction group the plane belongs to.

Modify Boundary Shape



Modify Boundary Shape tool moves a boundary vertex or segment. It depends on the shape type how a vertex of segment can be moved.

➢ **To modify a boundary vertex or segment:**

1. Select *Modify Boundary Shape* tool.

   The **Modify Boundary Shape** dialog opens:

2. Define setting.
3. Move the mouse pointer inside a view.

   The boundary segment or vertex closest to the mouse pointer is dynamically highlighted.

4. Define the boundary segment or vertex to move with a data click.

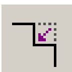
   If you move the mouse pointer, the new boundary segment or vertex location is dynamically displayed.

5. Define the new location of the boundary segment or vertex with a data click.

   This places the boundary segment or vertex at the new location and adjusts all other parallel segments along the same edge accordingly. You can continue with steps 2 or 4.

Setting:	Effect:
Fix segment	

Cut Boundary Corner



Cut Boundary Corner tool can modify a roof plane corner in two ways. It cuts off a piece from a corner or it adds a piece to a corner. In any case, the new boundary segments are aligned perpendicular to the segments that form the original corner.

➢ **To add/cut off a piece to/from a plane corner:**

1. Select *Cut Edge Corner* tool.
2. Move the mouse pointer inside a view.

   The corner closest to the mouse pointer is dynamically highlighted.

3. Define the corner to modify with a data click.

   If you move the mouse pointer, the new boundary segment of the corner is dynamically displayed.

4. Define the location of one boundary segment with a data click.
5. Define the location of the other boundary segment with a data click.

   This places the new corner at the defined location. You can continue with step 3. You can go back from steps 5 to 4 and 4 to 3 by placing a reset click.

Cut Boundary Segment

Cut Boundary Segment tool can modify a boundary segment in two ways. It cuts off a piece from a segment or it adds a piece to a segment. The cut off or added part is formed by three new boundary segments of which two are perpendicular and one is parallel to the original edge segment.

➢ **To add/cut off a piece to/from an edge segment:**

1. Select *Cut Boundary Segment* tool.
2. Move the mouse pointer inside a view.

   The boundary segment closest to the mouse pointer is dynamically highlighted.

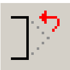
3. Define the segment to modify with a data click.

   If you move the mouse pointer, the new boundary segment is dynamically displayed.

4. Define the location of one perpendicular boundary segment with a data click.
5. Define the location of the other perpendicular boundary segment with a data click.
6. Define the location of the parallel boundary segment with a data click.

   This places the new boundary segments at the defined locations. You can continue with step 3. You can go back from steps 6 to 5, 5 to 4, and 4 to 3 by placing a reset click.

Delete Boundary Vertex



Delete Boundary Vertex tool removes a vertex from a boundary. If a roof plane boundary contains only three vertices, you can no longer delete a vertex.

➢ **To delete a vertex:**

1. Select *Delete Boundary Vertex* tool.
2. Move the mouse pointer inside a view.

   The vertex closest to the mouse pointer is dynamically highlighted.

3. Define the vertex to delete with a data click.

   This removes the vertex. You can continue with steps 3.

10 Main Window Menu Commands

When you load data using *Load Airborne Points* tool or **Open block** menu command, the application reads the laser points into RAM memory and opens the **Main** window. As long as you keep this window open, the points will remain in memory and will be displayed in all open views with appropriate levels on. You can use various tools for viewing, classifying, manipulating or outputting the loaded points.

Typical workflow includes the following steps:

1. Read laser points from raw files generated by system manufacturer's applications.
2. Remove points which are outside the geographical area of interest.
3. Search and delete error points (points high up in the air or below the ground).
4. Classify points into point classes such as ground, vegetation, building or wire.
5. Save modified points into a separate file which you can load if you start another work session later on.
6. Create vector data into the design file based on the points. This might involve automatic detection of some features such as wires, buildings or trees, or manual digitization using the coordinates or the elevations of laser points.

Throughout the work session you will be using various viewing tools to see the data in 3D.

Memory usage of loaded points

The amount of RAM memory in your computer will probably set the limit on how large jobs you can process at one time. Each loaded laser point will take up 20 bytes of memory. You should normally plan your work so that you will have enough physical memory to accommodate the laser points you process. Virtual memory (hard disk) usage will drastically slow TerraScan down.

Basic total memory usage consists roughly of:

- 100-200 MB for Windows operating system
- 20 MB for MicroStation and design file
- 20 MB for every million laser points loaded

You should also note that some classification routines will allocate substantial amounts of memory temporarily. The most demanding routine is probably ground classification which will take up an additional 4 bytes for every point and 80 bytes for each point that ends up in the ground class.

When performing ground or height from ground classifications, you can process up to:

- 2-5 million points with 256 MB RAM
- 4-10 million points with 512 MB RAM
- 8-20 million points with 1024 MB RAM

The low values apply when almost all points are ground hits. The high values apply when majority of the points are hits on vegetation.

For performing other tasks, the maximum point count to process is:

- 8 million points with 256 MB RAM
- 20 million points with 512 MB RAM
- 40 million points with 1024 MB RAM

✍ MDL applications share the same 2 GB address space with MicroStation. The maximum amount of memory that an MDL application can utilize under MicroStation SE is about 1200 MB, under MicroStation J about 700 MB and under MicroStation V8 about 800 MB.

File pulldown menu

Menu commands from **File** pulldown menu in **Main** window are used to open, save and close laser points.

To:	Choose menu command:
Read points of a project block	**Open block**
Read points of a project block inside a fence	**Open inside fence**
Read points from a file	**Read points**
Save modified points	**Save points**
Save or export points to a file	**Save points As**
Close loaded points	**Close points**

Open block

Open block menu command loads points from a project block. You select the project block geographically - that is by clicking inside the boundaries of the desired block.

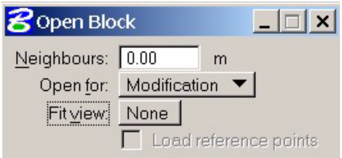
For more information on projects in TerraScan, see chapter **Coordinate Transformations** on page 312.

➢ **To open points from project block:**

1. Select **Open block** command from **File** pulldown menu.
2. Move the mouse inside the desired project block.

   As you move the mouse, the application highlights the boundaries of the block the mouse is inside.
3. Click the mouse inside the desired block.

   This loads the points from the selected block.



Setting:	Effect:
Neighbours	Width of overlap region around the active block boundaries for which the application will load points from neighbouring blocks.
Open for	Mode in which to open the block: **Viewing only** or **Modification**.
Fit view	View(s) to fit to show all loaded points.
Load reference points	If on, points from a reference project are loaded.

✍ **Open block** menu command lets you make a choice whether you are opening a block for viewing or for modification. This setting has an effect if you use project file locking. See **Coordinate Transformations** on page 312 for more information.

Open inside fence

Open inside fence menu command loads laser points from a project inside a fence or selected polygons. The points are opened for read-only access. After modifications, the points can be saved into a new file using **Save points As** command.

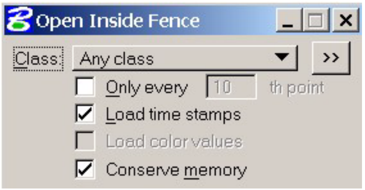
➢ **To open points inside fence:**

1. Draw a fence or select polygon(s) around area(s) for which to load the points.
2. Select **Open inside fence** command from **File** pulldown menu.

   As you move the mouse, the application highlights the boundaries of the areas for which points will be loaded.

3. Click inside a view.

   This loads the points for the selected areas.



Setting:	Effect:
Class	Class(es) that are loaded for the selected areas.
Only every	If on, only every given point is loaded.
Load time stamp	If on, time stamps are loaded. Otherwise, time information is ignored.
Load color values	If on, color values are loaded. This is only active, when color values are attached to the laser points.
Conserve memory	If on, the software first reads through input files to determine how many points would be loaded to be able to make a memory allocation for the exact number of points. This is slower but less likely to run out of memory.

Read points

Read points command loads points from files into TerraScan for visualization or processing tasks. It performs exactly the same action as the *Load Airborne Points* tool.

You can load several files of the same file format together in one reading process. The file format is automatically recognized if it is known by the software. The points of the selected file(s) are loaded into TerraScan memory. You can add more points by loading additional files. If the memory is full, the software shows an error message and the reading process stops.

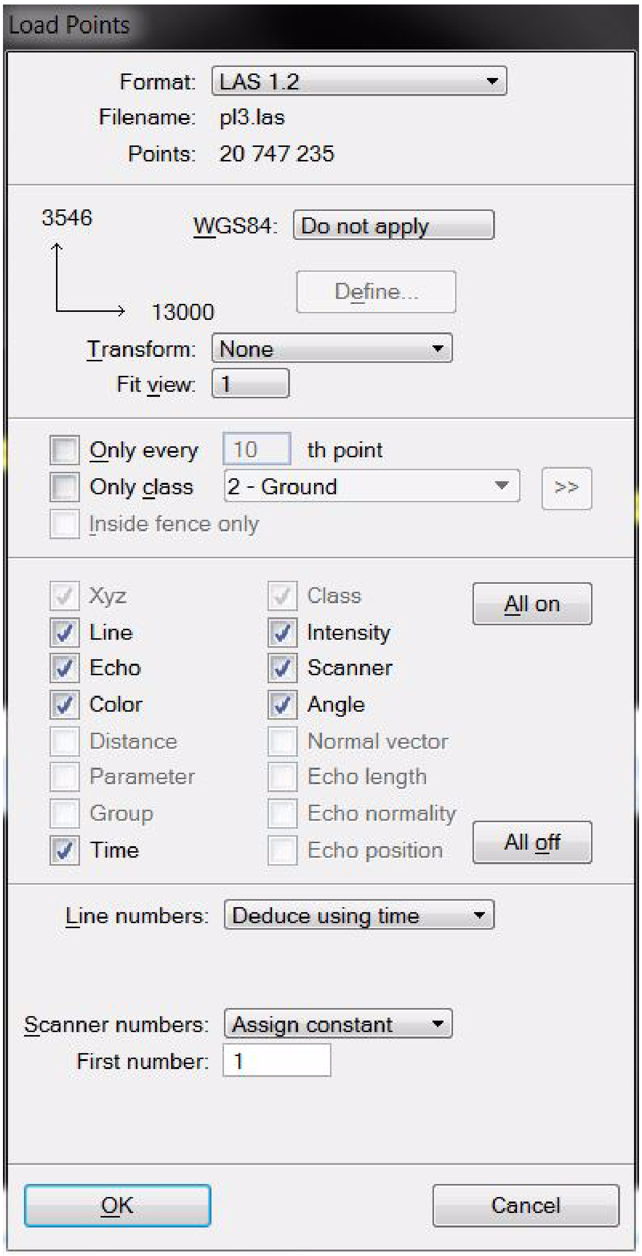
➢ **To load points from files into memory:**

1. Select **Read points** command from the **File** pulldown menu.

   This opens the **Read Points** dialog, a standard dialog for selecting files.

2. Select files and (optional) add them to the list of selected files by clicking on the **Add** button. If all files are selected or added to the list, click **Done**.

   This opens the **Load Points** dialog:



3. Define settings and click OK.

   This loads the points into the memory and displays them on the screen.

Setting:	Effect:
Format	Format of the point file. This is automatically recognized by the software. For user-defined text file formats, it might be necessary to select the correct format.
Filename	Name of the first selected file.
Points	Amount of points in all selected files.
Coordinate preview	Coordinate values of the first point found in the point file. This helps to decide whether a coordinate transformation needs to be applied.
WGS84	Transformation from WGS84 coordinates into projection system coordinates applied during the reading process. The list contains projection systems which are active in **Coordinate transformations / Built-in projection systems**, **Coordinate transformations / US State Planes**, or **Coordinate transformations / User projection systems** categories of TerraScan **Settings**.
Transform	Transformation applied to points during the reading process. The list contains transformations that are defined in **Coordinate transformations / Transformations** category of TerraScan **Settings**.
Fit view	Defines which MicroStation view is fitted to the extent of the loaded points: • **None -** no view is fitted. • **All -** all open views are fitted. • **1...8 -** the selected view is fitted. Only numbers of open views are available.
Only every	If on, only a selection of points is loaded. The software reads every **n**th point from the file(s), where **n** is the given number.
Only class	If on, only points from the selected class(es) are loaded.
	Opens the **Select classes** dialog which contains the list of active classes in TerraScan. You can select multiple source classes from the list that are then used in the **Only class** field.
Inside fence only	If on, only points inside a fence or selected polygon are loaded. This is only active if a MicroStation fence element is drawn in the design file or if a shape element is selected.
Attributes	Attributes that are loaded for laser points. You can decide which attributes you want to load or exclude. Only attributes that are stored in the point file(s) are available for loading. **All on** and **All off** buttons switch the selection of all attributes on or off. Point coordinates and the class number are always required.

Setting:	Effect:
Line numbers	Defines, how line numbers are assigned to the points during the loading process: • **Use from file -** line numbers from source files are used. • **Assign constant -** the number given in the **First number** field is assigned to all points. • **From file name -** the last numerical sequence in a file name is used as line number. • **From folder name -** the last numerical sequence in the name of the folder containing the point files is used as line number. • **Deduce using time -** numbers are assigned based on trajectories loaded into TerraScan. The same process can be performed for by the **Deduce using time** command or the corresponding macro action. • **Increase by xy jump -** the line numbers incease from the given **First number** if the xy distance is bigger than the value given in the **By distance** field. • **Increase by time jump -** the line numbers increase from the given **First number** if a jump in time stamps occurs. This requires that trajectory information is available in TerraScan. • **Increase by file -** the line numbers increase from the given **First number** for each separate file. • **Increase by file name -** the line numbers increase from the given **First number** for each file with another file name. Files with the same name get the same number. • **Increase by directory -** the line numbers increase from the given **First number** for each file stored in another source folder. Files from the same source folder get the same number. The **Line numbers** settings are only available if the **Line** attribute is switched on for loading.
Scanner number	Defines, how scanner numbers are assigned to the points during the loading process: • **Use from file -** scanner numbers from source files are used. • **Assign constant -** the number given in the **First number** field is assigned to all loaded files. • **Increase by file -** the scanner numbers increase from the given **First number** for each separate file. • **From file name -** the first numerical sequence in a file name is used as scanner number. • **From folder name -** the first numerical sequence in the name of the folder containing the point files is used as scanner number. • **From line number -** the line number is used as scanner number. The **Scanner number** settings are only available if the **Scanner** attribute is available and switched on for loading.
Default	Point that is assigned to all points if no class attribute is stored in the point file. This is only active if text file formats are selected for loading.

✍ See **Supported file formats** on page 478 for information about supported file formats and **File formats / User point formats** on page 38 for information on user defined text file formats.

Save points

Save points menu command saves all laser points to the same binary file from which they were read in or into which they were saved to earlier.

✍ To prevent an original file being overwritten by a file which does not include all information, the **Save points** command is disabled if points have been loaded incompletely. This includes the following settings:

- **Load color values** is off and input file had colors
- **Load time stamps** if off and input file had time
- **Only class** is on
- **Inside fence only** is on
- **Only every** is on

Save points As

Save points As menu command writes laser points to a new file which can be a text file or a binary file. You can use this to export points into another format.

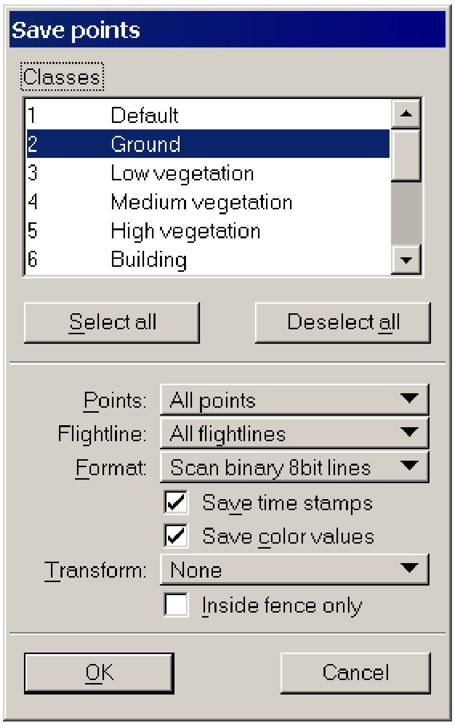
The output file format can be any of the following:

- **Scan binary 8bit lines** - compact and fast file format which includes all essential laser point information. Flightline numbering is limited to range 0 - 255.
- **Scan binary 16bit lines** - same as above but flightline range 0 - 65535.
- **LAS 1.0** - industry standard format for laser data, version 1.0.
- **LAS 1.1** - industry standard format for laser data, version 1.1.
- **LAS 1.2** - industry standard format for laser data, version 1.2.
- **EarthData (E)EBN -** Earth data binary.
- **E N Z** - plain xyz text file.
- **Class E N Z** - text file with class, x, y and z for each point.
- **E N Z Intensity** - text file with x, y, z and intensity for each point.
- **Class E N Z Intensity** - text file with class, x, y, z and intensity for each point.
- **E N Z dZ** - text file with x, y, z and elevation difference to a surface model.
- **Id E N Z -** text file with unique index number, x, y and z for each point.
- **Id E N Z Pulse -** text file with unique index number, x, y, z and echo number for each point.
- User defined point file formats.

➢ **To write points into a file:**

1. Select **Save points As** command from **File** pulldown menu.

   This opens the **Save points** dialog:



2. Select point classes which you want to output.
3. Select settings for the output file.
4. Click OK.

   A standard dialog box for defining an output file name opens.

5. Enter a name for the output file.

A file with the given name is created and points in selected classes are written to it.

Setting:	Effect:
Points	Selection of points that will be saved into the new file: **All points** or **Active block**. The latter is only active if points from neighbour blocks have been loaded using **Open block** command.
Flightline	Selection of flightlines that will be saved into the new file: **All flightlines** or the flightline with the selected number.
Format	Format of the output file.
Save time stamp	If on, time stamps are saved into the new file. This is only available for TerraScan binary files and if loaded points content time information.
Save color values	If on, color values are saved into the new file. This is only available for TerraScan binary files and if the loaded points content color information.
Delimiter	Defines the delimiter for text files: **Space**, **Tabulator** or **Comma**. This is only available if a text file format is selected as output format.
Xyz decimals	Defines the number of decimals for coordinate values. This is only available if a text file format is selected as output format.
Transform	A transformation into a new projection system can be applied for the laser points that are written into a new file.
Inside fence only	Only points that are inside a fence or a selected polygon are written into the new file.
Line numbers	Defines, how flightline numbers are assign to the laser points: • Use from file - flightline number from source files is used. This requires that the text file includes a flightline field. • Assign constant - a constant given in **First number** field is assigned to all loaded files. • Increase by xy jump - the number changes if an jump of the value given in the **By distance** field or more occurs. • Increase by time jump - the number changes if a time jump occurs. • Increase by file - a number per loaded file is assigned. • Increase by file name - a number per file name is assigned. • Increase by directory - a number per source directory is assigned to the loaded files. • Deduce using time - numbers are assigned based on trajectories loaded into TerraScan.

Close points

Close points menu command unloads all currently loaded points from TerraScan memory.

If points have been modified, a window opens to ask if you want to save changes before unloading the points.

- Click **Yes** to save modified points.
- Click **No** to unload points without saving changes.
- Click **Cancel** to close the window without unloading the points.

Output pulldown menu

Menu commands from **Output** pulldown menu in **Main** window are used to create a surface model, to export laser points as lattice or raster files, and to draw points into design.

To:	Choose menu command:
Output report from alignment	**Output alignment report**
Create a surface model	**Create surface model**
Export laser data into a lattice file	**Export lattice model**
Export laser data into a colored image	**Export raster image**
Draw points permanently to design file	**Write to design file**

Output alignment report

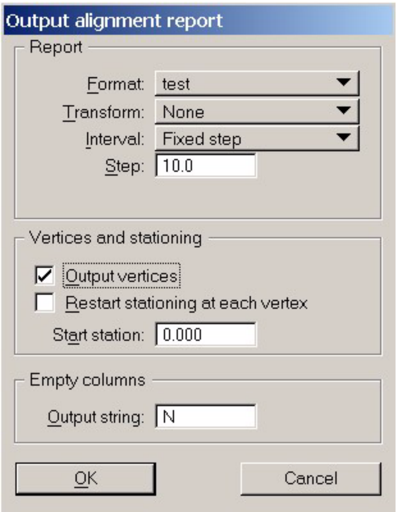
Output alignment report menu command generates a report with information at given intervals along an alignment element. The menu command requires the definition of an alignment report format in TerraScan settings. See **Alignment reports** on page 47 for more information.

Alignment elements can be any MicroStation linear or polygon element like line strings, shapes, circles, etc. The report can be seen as a table where each row corresponds to an alignment location and each column contains a specific type of information.

➢ **To create an alignment report:**

1. Select an alignment element with MicroStation *Selection* tool.
2. Choose **Output alignment report** command from **Output** menu.

   This opens the **Output alignment report** dialog:



3. Enter settings and click OK.

   The application processes information and opens a window which allows you to view the report and to save it to a space or tabulator separated text file.

Setting:	Effect:
Format	Alignment report format to use. Defines the report content.
Transform	Coordinate transformation to convert from design file coordinates to output coordinates.
Interval	Interval definition: • Fixed step - interval length is fixed along the whole alignment element. • Triangle edges - interval length is defined by intersection points between surface triangle edges and the alignment element.
Surface	Surface when **Interval** is set to **Triangle edges**.
Step	Length of a fixed step in design file units.
Min step	Minimum length of a step in design file units, if **Interval** is set to **Triangle edges**.
Output vertices	If on, will generate a row for each alignment vertex.
Restart stationing at each vertex	If on, station values will restart from **Start station** at each vertex.
Start station	Station value at start of the alignment.
Output string	String to output when a column has no valid value.

Create surface model

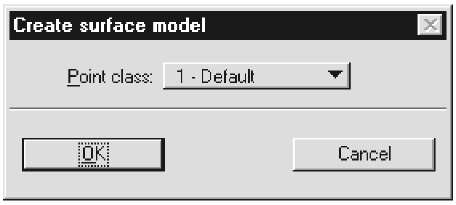
Create surface model menu command passes points in a selected class to TerraModeler which creates a triangulated surface model. You would normally use this to create a ground surface model after having classified and possibly thinned the points.

This menu command requires a valid license for TerraModeler.

➤ **To create a triangulated surface model:**

1. Choose **Create surface model** command from **Output** menu.

   This loads TerraModeler if not loaded already and opens the **Create surface model** dialog:



2. Select point class to triangulate and click OK.

   TerraModeler opens the **Triangulate surface** dialog.

3. Select settings for triangle exclusion, minimum point distance, points along breaklines, and error points. Click OK.

   This opens the TerraModeler **Surface settings** dialog.

4. Select a surface type, and type a name for the surface. You can also change the name for the surface file in the storage field. Click OK.

   This creates a triangulated surface model from the points.

   For more detailed information about surface triangulation in TerraModeler see TerraModeler User’s Guide.

✍ After this operation both TerraScan and TerraModeler will end up keeping copies of the points in memory. If you want to reduce the memory requirement, you can do the following steps:

1. Load, classify and thin the laser points using TerraScan.
2. Write points to an xyz text file using **Save points As** menu command.
3. Unload TerraScan.
4. Load TerraModeler.
5. Import points from text file.

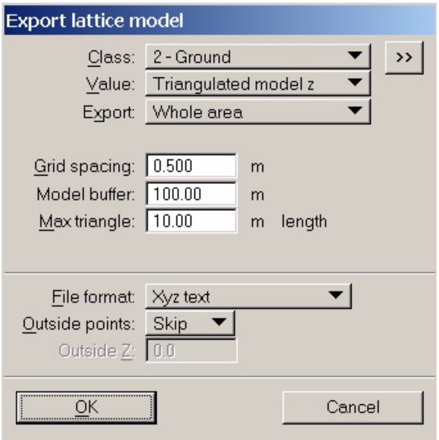
Export lattice model

Export lattice model menu command creates a grid file with uniform distances between points from one or more selected laser point classes. The file stores either elevation values or point count/ density values for each grid cell. There are several formats supported to store the lattice as raster, grid, or text file. The menu command can be used to export the whole area covered by laser data or to export only parts defined by rectangles into separate lattice models. Text strings placed inside the rectangle(s) can be used as file names for the exported lattice files.

➢ **To export a lattice model:**

1. (Optional) Draw rectangle(s) around areas that are to be exported and place text strings inside the rectangles. Select rectangle(s) and text(s).
2. Select **Export lattice model** command from **Output** menu.

   This opens the **Export lattice model** dialog:



3. Select settings and click OK.

   The lattice model is calculated and the **Export lattice model** dialog, a standard Windows dialog to save files, opens.
4. Define a name and location for the lattice file and click Save.

   This saves the lattice model to the given location.

Setting:	**Effect:**
Class	One or more laser point classes as source for the model creation.
Value	Value to be stored for each grid cell: • Triangulated model z - elevation value calculated from a triangulated model of the selected point class(es). • Highest hit z - elevation value defined by the highest hit in the selected point class(es). • Average hit z - average elevation value calculated from hits in the selected point class(es) falling inside the grid cell. • Lowest hit z - elevation value defined by the lowest hit in the selected point class(es). • Point count - amount of points falling inside the cell. • Point density - number of points per squared master unit.

Setting:	Effect:
Export	Area to be exported: • Whole area - area covered by the loaded laser data is exported. • Selected rectangle(s) - only area inside selected rectangle(s) is exported.
Expand by	Distance by which a selected rectangle is expanded for the model export. The area covered by the expanded rectangle is included in the exported model.
Grid spacing	Defines the distance between grid cells.
Model buffer	Buffer area around the model area that is considered for calculating elevation values from a triangulated model.
Max triangle	Maximum length of a triangle edge for calculating elevation values from a triangulated model.
Fill gaps up to	Pixel value for filling gaps. This is only active if **Value** is set to **Highest**, **Average**, or **Lowest hit z**.
File format	Format for exported lattice file: **ArcInfo**, **GeoTIFF**, **Intergraph GRD**, **Raw**, **Surfer ASCII** or **binary**, **Xyz text**.
Z unit	Unit for z values. Relevant for formats storing elevations as integers. This is only active for **Formats GeoTIFF**, **Intergraph GRD**, **Raw**.
Outside points	Defines whether cells that are not covered by laser data are skipped or included in the output. This is only active for **Format Xyz text**.
Outside Z	Defines the value for cells that are inside the model area but not covered by laser data. This is only active for **Formats ArcInfo**, **Surfer ASCII** and **binary**, **Xyz text**.
File naming	File naming setting for exporting selected rectangles: • Enter name for each - a name for each rectangle has to be defined when saving the lattice models. • Selected text elements - selected texts inside the rectangles are used as file names.
Directory	If **File naming** is set to **Selected text elements**, this sets a directory for storing the lattice model files.
Extension	If **File naming** is set to **Selected text elements**, this defines a file extension which is added to the lattice model file names.

✍ **Export lattice model** is a good option to exchange structured elevation models between TerraScan and other software products.

Export raster image

Export raster image menu command generates a raster image colored by elevation, intensity, point color, or point class of laser points.

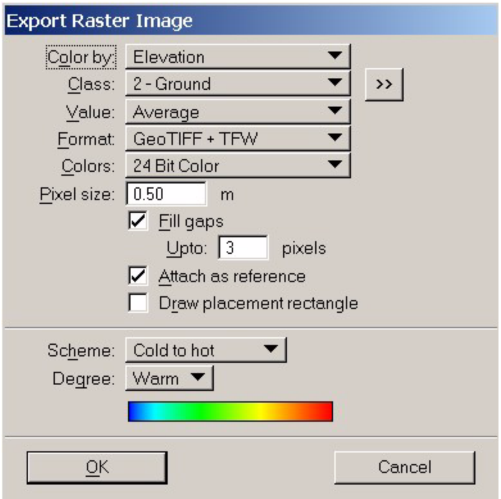
The raster image will be created in Windows bitmap (BMP) or GeoTIFF (TIF) format. The color of each pixel will be determined using laser points whose coordinate values fall inside the pixel. The coloring attribute can be chosen as:

- **Elevation** - Laser point elevation.
- **Elevation difference** - Elevation difference between laser points of two different classes.
- **Intensity hits** - Intensity of laser points with center point inside the pixel.
- **Intensity footprint** - Intensity of laser points with footprint overlapping the pixel.
- **Point color -** Color values stored for laser points.
- **Point class -** Laser point class.

➢ **To create a colored raster image:**

1. Choose **Export raster image** menu command.

   This opens the **Export Raster Image** dialog:



2. Define settings and click OK.

   A standard dialog box for choosing an output file name opens.

3. Enter a name for the output file.

   This creates a raster image with the given name.

Setting:	Effect:
Color by	Coloring attribute.
Class	Point class(es) to use. Several classes can be selected by clicking on the >> button.
Value	Value determination within each pixel: • Lowest - Smallest value of the points. • Average - Average value of the points. • Highest - Highest value of the points. This is only active for elevation and intensity coloring.
Format	File format for the raster file: **Windows BMP**, **GeoTIFF**, or **GeoTIFF + TFW**.

Setting:	Effect:
Colors	Color depth of raster image: • 24 Bit Color - true color image. • 256 Colors - 256 colors. • Grey scale - 8 bit grey scale. This is only active for elevation and intensity coloring.
Pixel size	Size of each pixel in the target raster file.
Fill gaps	If on, small gaps are filled in places where there are no laser hits inside a pixel.
Attach as reference	If on, the image is attached as raster reference to the design file.
Draw placement rectangle	If on, the image position is drawn as a shape. Helps to attach the image at the correct location later on if no TFW file was created.
Scheme	Type of coloring scheme for elevation or intensity coloring: • Cold to hot - varies from blue for low elevation to red for high elevation. This is the common coloring scheme for elevation coloring. • Hot to cold - varies from red for low elevation to blue for high elevation. • Selected colors - a user defined coloring scheme can be created by clicking the **Define** button. See more information below in **Color scheme definition** section. • Black to white - varies from black for low values to white for high values. This is active if **Colors** is set to **Grey scale**. This is the common coloring scheme for intensity coloring. • White to black - varies from white for low values to black for high values. This is active if **Colors** is set to **Grey scale**.
Degree	Determines how the color changes in color schemes are computed.

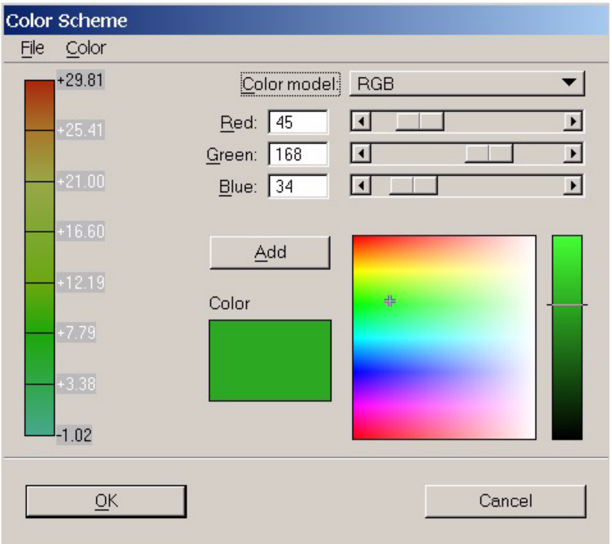
Color scheme definition

For export raster images, a user defined coloring scheme can be applied for elevation or intensity coloring.

➢ **To define a new color scheme:**

1. In the **Raster Image Dialog**, set the **Scheme** to **Selected colors** and click the **Define** button.

   This opens the **Color scheme** dialog:



2. Select a **Color model**: **RGB** or **HSV**.
3. Add colors to the color scheme shown on the left side of the dialog.

   Colors can be selected by typing values in the **Red**, **Green**, **Blue**, or **Hue**, **Saturation**, **Value** fields, by moving the sliders next to the fields, by clicking inside the color window, or by clicking on the color bar right of the color window. The selected color is shown in the **Color** field left of the color window. Click the **Add** button to add the color to the color ramp.

   Colors are always added to the lower end of the color scheme which means that the first added color is for the highest elevation or intensity value.

4. If necessary, the boundary values between colors can be changed by clicking on the value next to the color scheme. This opens the **Color elevation** dialog. Switch **Fixed** on and define a value.
5. To make color changes for the color scheme, select **Remove all** or **Remove last** command from the **Color** pulldown menu.

   This removes all colors or the last added color from the color scheme.

6. A color scheme can be saved into a text file with the extension *.SMC using the **Save As** command from the **File** pulldown menu.

   The file stores the RGB or HSV values for each scheme color in rows. A previously saved color scheme file can be loaded with **Open** command from the **File** pulldown menu. Color values can be also changed by editing the color scheme file in a text editor.

Write to design file

Write to design file menu command draws laser points permanently into the design file using all of the drawing rules assigned to point classes. See **Define Classes** on page 68 for information about class definitions.

MicroStation SE and MicroStation J are limited to a maximum design file size of 32MB. This means that you can draw about half a million laser points into one design file if using **Zero length line** drawing rules. If the design file grows larger, the routine will end the writing process, remove processed points from the table and tell you what has happened. You can then switch to an empty design file and restart the writing process.

MicroStation SE and MicroStation J are no longer supported by TerraScan versions later than V009.001. MicroStation V8 has no effective limit on design file size.

The points drawn into the design file can be colored by the following coloring modes:

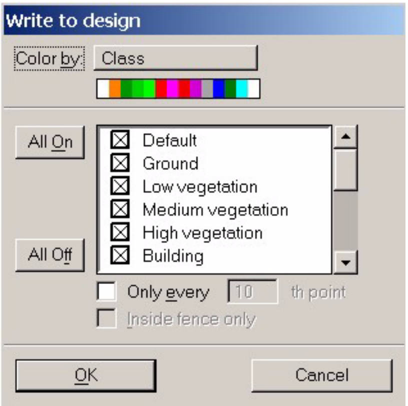
- **Active symbology -** color for all points is defined by the active color set in the design file.
- **Class -** color is defined by class coloring scheme loaded in TerraScan.
- **Elevation -** color is defined by elevation coloring scheme. This can be changed by clicking the **Color** button. See **Color by Elevation** on page 264 for more information.
- **Intensity -** color is defined by intensity coloring scheme. This can be changed by clicking the **Color** button. See **Color by Intensity** on page 266 for more information.
- **Flightline -** color is defined by flightline coloring. This can be changed by clicking the **Color** button. See **Color by Flightline** on page 265 for more information.
- **Point color -** color is defined by color value attributes defined for each laser point.

✍ The color scheme settings for this menu command are defined in the same way and in the same dialogs as for display modes in TerraScan. See detailed descriptions for all coloring settings in section **Display mode** on page 263.

➢ **To write points to design file:**

1. Select **Write to design file** command from **Output** menu.

   This opens the **Write to design** dialog:



2. Select coloring mode which you want to use in the **Color by** field.
3. Select the desired classes by checking the boxes in the list.
4. (Optional) Select **Only every** option if only every nth point is to be drawn in the design file.
5. (Optional) If a fence or selected polygon is defined, points inside can be drawn into the design file by checking the **Inside fence only** option.
6. Click OK.

   This writes the selected points to the design file.

Point pulldown menu

Menu commands from **Point** pulldown menu in **Main** window are used to undo actions on loaded points, to manipulate laser point attributes, and to select points.

To:	Choose menu command:
Undo a classification action	**Undo**
Undo classification actions from a list	**From list**
Edit attributes of one or more laser points	**Edit selected**
Select laser points of a specific class	**Select by class**
Select laser points with specific attribute values	**Find**
Delete points	**Delete**

Undo

Undo menu command lets you undo modifications of points. Actions that can be undone include:

- Manual classification steps
- Automatic classification steps
- Macro actions applied to loaded points
- Smoothen points and thin points menu commands
- Adjust to geoid menu command
- Changes to flightline numbering
- Automatic detection of wires

More than one action can be undone by using the menu command several times. However, a faster way is to use the **From list** menu command instead.

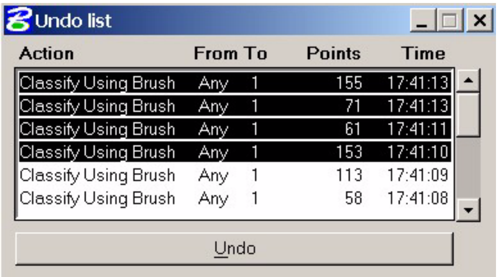
From list

From list menu command lets you undo several actions from a list. This is a faster way to undo more than one action. The amount of actions that can be undone is determined by the memory provided for the undo buffer. See **Undo buffer** on page 63 in TerraScan settings.

➢ **To undo several actions:**

1. Select **From list** command from **Point** pulldown menu.

   This opens the **Undo list**:



2. Select the actions to be undone and click **Undo**.

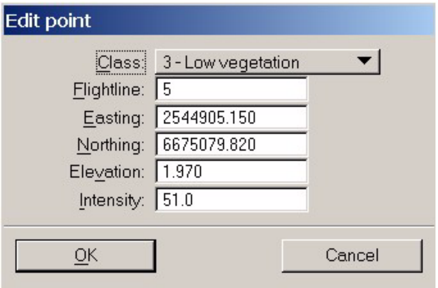
Edit selected

Edit selected menu command lets you modify attributes for one or more laser points selected in the point list of the TerraScan **Main** window. The attributes that can be changed differ depending on the selection of one point or several points.

➢ **To edit attributes of one point:**

1. Select a point in TerraScan **Main** window.
2. Select **Edit selected** command from **Point** pulldown menu.

   This opens the **Edit point** dialog:



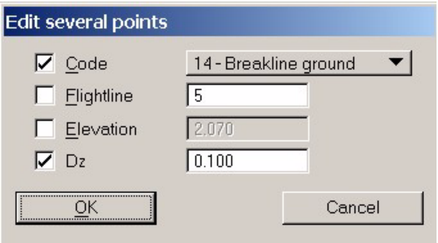
3. Enter new values for attributes that are to be changed.
4. Click OK.

   This changes the attribute values for the selected point.

➢ **To edit attributes of several points:**

1. Select several points in TerraScan **Main** window.
2. Select **Edit selected** command from **Point** pulldown menu.

   This opens the **Edit several points** dialog:



3. Enter new values for attributes that are to be changed.
4. Click OK.

   This changes the attribute values for all selected points.

✍ Edit selected points actions can not be undone.

Select by class

Select by class menu command selects points of the same class. It requires the selection of one example point in the TerraScan **Main** window.

➢ **To select points by class:**

1. Select one example point of the class to be selected in TerraScan **Main** window.
2. Select **by class** command from **Point** pulldown menu.

   This selects all points of the same class as defined by the example points.

✍ A good possibility to select an example point is to use the **Identify** button in TerraScan **Main** window. This lets you select one point by mouse click in a MicroStation view. The identified point is selected in the **Main** window point list. The location of selected points can be shown in a MicroStation view using the **Show location** button in TerraScan **Main** window. If a large amount of points is selected, the display of selected points may take a while.

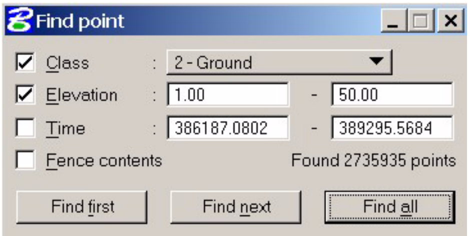
Find

Find menu command selects points with specific attribute values, including class, elevation range and time range. The command finds and selects either all points with the given values or finds the first and next point as they appear in the point list.

➢ **To select points with given attribute values:**

1. Select **Find** command from **Point** pulldown menu.

   This opens the **Find point** dialog:



2. Select attributes and define values for points to be selected.
3. (Optional) Switch **Fence contents** on to limit the search to an area defined by a fence or selected polygon.
4. If you want to select the first point in the point list for which the given values apply, click **Find first**. The next point in the list with corresponding values can be selected by clicking **Find next**.

   OR

5. If you want to select all points with the given values, click **Find all**.

   This selects the points in the point list in TerraScan **Main** window.

✍ The location of selected points can be shown in a MicroStation view using the **Show location** button in TerraScan **Main** window. If a large amount of points is selected, the display of selected points may take a while.

Delete

Delete sub-menu contains commands for deleting laser points:

- **Selected points** - deletes the point(s) selected in the list.
- **By point class** - deletes all points in selected class(es).
- **By flightline -** deletes all points of selected flightline(s).
- **Inside fence** - deletes all points inside a fence or selected polygon and on visible levels.
- **Outside fence** - deletes all points outside a fence or selected polygon and on visible levels.
- **Using centerline** - deletes all points which are more than a given distance away from a selected centerline element. Any MicroStation element can serve as a centerline element.

➢ **To delete selected points:**

1. Select laser points in the point list using one of the point selection options in TerraScan.
2. Select **Selected points** in **Delete** sub-menu from **Point** pulldown menu.

   A message appears to ask if the selected points are to be deleted. Click **Yes** to delete.

➢ **To delete point by point class:**

1. Select **By point class** in **Delete** sub-menu from **Point** pulldown menu.

   This opens the **Delete points by class** dialog that lists all classes included in the loaded points.

2. Select one or more classes for which you want to delete points.
3. Click OK to delete the points.

   A message appears that informs how many points have been deleted.

➢ **To delete point by flightline:**

1. Select **By flightline** in **Delete** sub-menu from **Point** pulldown menu.

   This opens the **Delete by flightline** dialog that lists all flightlines included in the loaded points.

2. Select one or more flightlines for which you want to delete points.
3. Click OK to delete the points.

   A message appears that informs how many points have been deleted.

➢ **To delete points inside or outside a fence:**

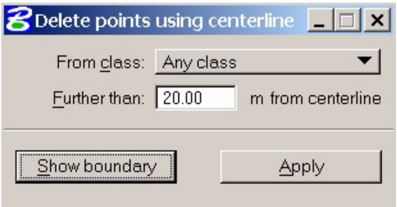
1. Draw a fence or select a polygon around the area for which you want to delete points.
2. Select **Inside fence** or **Outside fence** in **Delete** sub-menu from **Point** pulldown menu.
3. Click inside the MicroStation view where the fence is defined.

   This deletes points inside or outside the fence.

➢ **To delete points using a centerline:**

1. Select a centerline element.
2. Select **Using centerline** in **Delete** sub-menu from **Point** pulldown menu.

   This opens the **Delete points using centerline** dialog:



3. Select a class from which points are to be deleted in the **From class** field.
4. Define a distance for keeping points inside and deleting points outside.
5. Show the current set distance with **Show boundary**. The boundary is displayed when the mouse is moved inside a MicroStation view.
6. Click **Apply**.

   A message appears to ask if the selected points are to be deleted. Click **OK** to delete the points further away than the defined distance.

View pulldown menu

Menu commands from **View** pulldown menu in **Main** window are used to change the appearance of the **Main** window as well as display settings for laser points.

To:	Choose menu command:
Switch main window to small size	**Small dialog**
Switch main window to medium size	**Medium dialog**
Switch main window to large size	**Large dialog**
Switch main window to wide size	**Wide dialog**
Change the display of fields in main window	**Fields**
View header records of a LAS file	**Header records**
Fit a view to display all loaded points	**Fit view**
View points using elevation or intensity coloring	**Display mode**

Small dialog

Small dialog command changes the **TerraScan Main** window to a minimal size which consists of a title bar and pulldown menus only.

Medium dialog

Medium dialog command changes the **TerraScan Main** window to a medium size which consists of a title bar, the pulldown menus, and a medium size list displaying the attributes of loaded points.

Large dialog

Large dialog command changes the **TerraScan Main** window to a large size which consist of a title bar, the pulldown menus, and a large size list displaying the attributes of loaded points.

Wide dialog

Wide dialog command changes the **TerraScan Main** window to a wide size which consist of a title bar, the pulldown menus, and a wide size list displaying the attributes of loaded points.

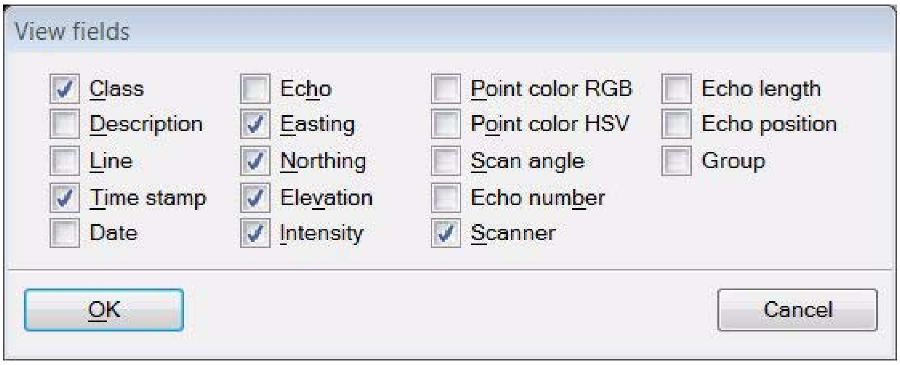
Fields

Fields command lets you select which attributes are displayed in the **TerraScan Main** window if point are loaded in TerraScan. The list of loaded points is only visible if the window is displayed as **Medium dialog**, **Large dialog**, or **Wide dialog**.

➢ **To select visible fields:**

1. Select **Fields** command from the **View** pulldown menu.

   This opens the **View fields** dialog:



2. Select fields you want to see in the list of loaded points and click OK.

Header records

Header records command is not implemented yet.

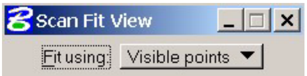
Fit view

Fit view menu command fits a view to display an area covering all of the loaded points or currently visible points.

➢ **To fit a view to display all points:**

1. Choose **Fit view** command from **View** menu.
2. Select setting for **Fit using** field: **All points** or **Visible points**.
3. Select a view by clicking inside the view.

   Application fits the view to all or currently visible loaded points and redraws the view. You can continue to step 2.



See **Draw bounding box** on page 278 for a related tool.

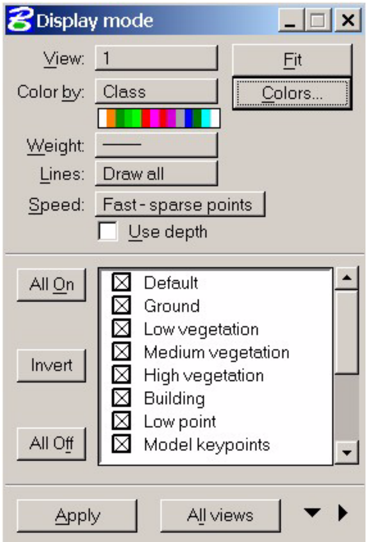
Display mode

Display mode menu command opens the **Display mode dialog** which contents settings for controlling the display of laser points in MicroStation views. This includes settings for laser point coloring, point size, drawing method, and what classes and flightlines are displayed.

➢ **To define the display mode:**

1. Choose **Display mode** command from **View** menu.

   This opens the **Display mode** dialog:



2. Set **View** setting to the view you want to modify.
3. Select visibility and coloring settings as desired.
4. Click on desired rows in the class list to switch the classes on or off.
5. Click on **Apply** to update the selected view.

Setting:	Effect:
View	View for which display settings are applied.
Color by	Coloring attribute: Class, Echo, Elevation, Flightline, Intensity, Distance, Point color, Scanner, Shading, Line & Intensity, Echo length.
Weight	Point size.
Lines	Visibility of flightlines: **Draw all** or **Selected**. Select flightlines by clicking on the **Select** button.
Speed	Speed setting for point display: • Fast - sparse points - amount of points displayed depends on zoom factor, if zoomed out less points are drawn. • Normal - all points - all points are drawn at every zoom level. • Slow - all points are drawn slowly. This is only active if **Color by** is set to **Flightline**.
Use depth	If on, in 3D views the point is displayed which is closest to the viewer compared with all other points falling in the same screen pixel.
Fit	Fits the selected View to the extend of all visible points.
Colors	Opens a window for additional color settings. The content of the window depends on the selected coloring attribute. See the following sections for more information.

Setting:	**Effect:**
All On	Switch all classes on.
Invert	Invert the visibility of classes.
All Off	Switch all classes off.
Apply	Apply current settings to the selected view.
All views	Apply current settings to all views.
	Extend the length and/or width of the **Display mode** dialog.

Point display speed

The term **Sparse points** refers to a display logic which speeds up the display of laser points. If **Speed** is set to **Fast - sparse points**, and you zoom out, TerraScan will draw only every 5th, 10th, 20th or 50th point depending on the zoom factor. In most cases **Fast - sparse points** is the best choice for fast point display.

Normal - all points might is the better solution, if a particular point class which consists of single points spread out in the data set has to be displayed. Points below the ground or other problem points are typical examples.

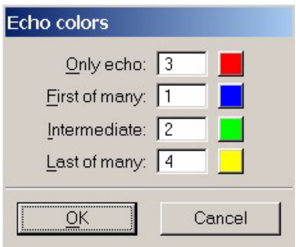
Slow display mode is only active for flightline coloring. This is useful to show the flight direction of flightlines.

Color by Class

Color by Class setting displays the points according to their class. To change the colors for classes, click the **Colors** button next to the **Class by** field. This opens the **Point classes** dialog which is the same as is opened by the *Define Classes* tool. The display color can be changed by selecting a class and clicking the **Edit** button. This opens the **Point class** dialog where the display color is set. See *Define Classes* for more information about settings for point classes.

Color by Echo

Color by Echo setting displays the points according to their return type. To change the colors for echo types, click the **Colors** button next to the **Class by** field. This opens the **Echo colors** dialog:

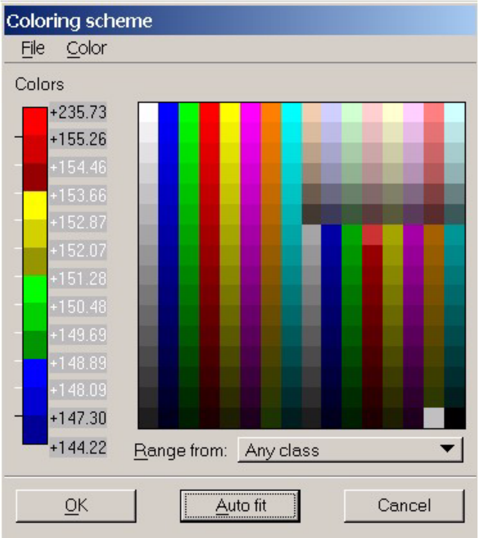


Type a new color number in the echo type field or click on the color button next to the echo number to change the color for the echo type.

Color by Elevation

Color by Elevation setting displays the points according to their z value. A color scheme is applied where each color represents a certain range of elevation values. The default color scheme of TerraScan consists blue colors for low elevation values, green and yellow colors for intermediate values, and red colors for high elevation values.

To change the color scheme, click the **Colors** button next to the **Class by** field. This opens the **Coloring scheme** dialog:



To change a color scheme, select **Remove all** command from **Color** pulldown menu. This removes the current color scheme. Add a new color by selecting it from the color table with a mouse click on the color field. New colors are always added at the lower end of the color scheme which means that the first selected color is for the highest elevation range. If the last added color has to be removed, select **Remove last** from **Color** pulldown menu.

A color scheme can be saved into a text file with the ending .CLR which lists the numbers of the selected colors. To save a color scheme, select **Save as** command from **File** pulldown menu. To load a color scheme, select **Open** from **File** pulldown menu.

By default, the range for the color ramp includes the z values of all loaded laser points. This can be changed by selecting a class in the **Range from** field. The z values from this selected class are applied to the color scheme.

The **Auto fit** button puts outliers in the lowest and highest elevation range. This results in a color scheme where outliers do not affect the color distribution any longer.

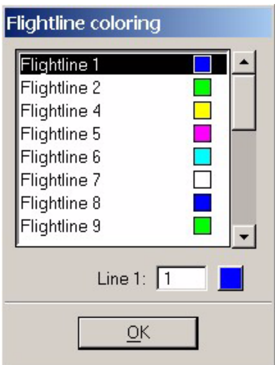
✍ It is recommended to apply **Auto fit** always when the coloring mode for laser points is set to **Elevation**.

The z values defining the boundaries between two colors can be changed by clicking on the color field in the color scheme or on the number. The **Change color at boundary** dialog opens where a new z value can be set. When **Auto fit** is applied, manual changes for boundaries will get lost.

Color by Flightline

Color by Flightline setting displays the points according to their flightline number. The default are seven repeated colors for flightline coloring: blue, green, red, yellow, magenta, cyan, white.

To change the colors for flightlines, click the **Colors** button next to the **Class by** field. This opens the **Flightline colors** dialog:

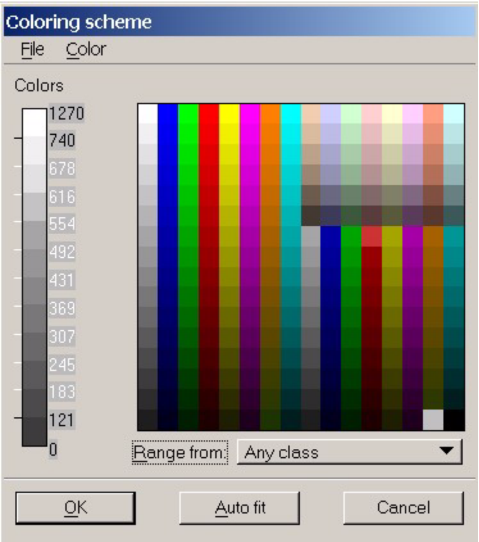


Select a flightline. The color can be changed by typing a new number in the **Line** field or by clicking on the color button next to the field. This opens the MicroStation color table, where a new color can be selected.

Color by Intensity

Color by Intensity setting displays the points according to their intensity value. A grey scale is applied where each grey value represents a certain range of intensity values. The default grey scale of TerraScan stretches from dark grey for low intensity values to white for high intensity values.

To change the grey scale, click the **Colors** button next to the **Class by** field. This opens the **Coloring scheme** dialog:



Modification can be made in the same way as for the elevation color scheme. See description **Color by Elevation** on page 264 for details.

✍ It is recommended to apply **Auto fit** always when the coloring mode for laser points is set to **Intensity** to put outliers in intensity values into the lowest and highest intensity range.

Color by Distance

Color by Distance setting displays points based on the calculated distance of each point against a reference surfaces or MicroStation elements.

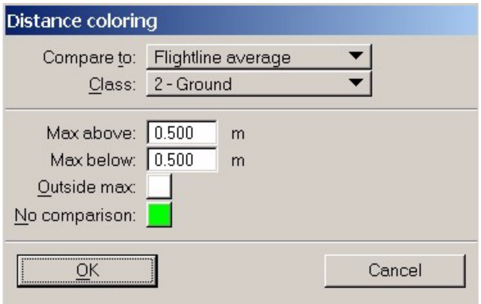
TerraScan can compare laser points against the following references:

- **Elements** - points are compared against vector elements in the design file.
- **Flightline average** - points are compared against the average surface from overlapping flightlines. This can be used to check how well different flightlines match each other. Comparing against the flightline average effectively requires that ground is classified separately for each flightline.
- **Ground class** - points are compared against a surface computed from a ground class. This corresponds to a height from ground coloring.

➢ **To color points by distance**

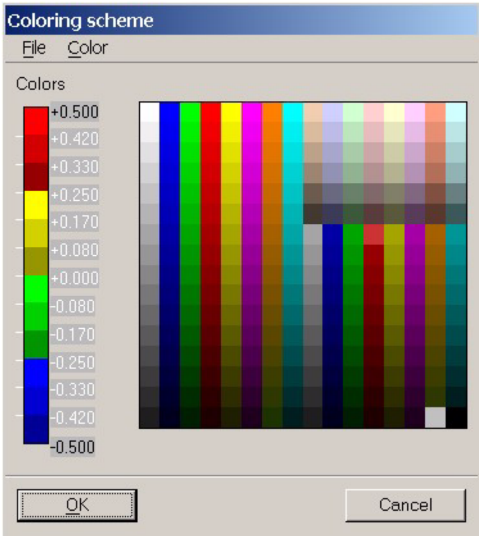
1. Click the **Color** button next to the **Class by** field.

This opens the **Distance coloring** dialog:



2. Select a reference to compare the laser points against.
3. Select a point class for being compared against the reference.
4. Select additional settings dependent on the selected reference.
5. Click OK.

   This opens the **Color scheme** dialog:



   The default color scheme in TerraScan for distance coloring is defined in a way that blue and green color values symbolize negative distances, and yellow and red color values symbolize positive distances.

   The colors can be changed as described in the section **Color by Elevation** on page 264.

6. Click OK in the **Coloring scheme** dialog.

   The software computes the distances.

7. To apply the distance coloring to the point display, click **Apply** in the **Display mode** dialog.

Setting:	Effect:
Compare to	Reference to compare points against.
Class	Point class that is compared against reference.
Max above	Maximum distance above the reference covered by the color scheme.
Max below	Maximum distance below the reference covered by the color scheme.
Outside max	Color for points that are more than the maximum distance above or below the reference.
No comparison	Color for points that cannot be compared against a reference. This is only active if **Compare to** is set to **Flightline average** or **Ground class**.

Setting:	Effect:
Levels	Design file levels on which reference elements are placed. This is only active if **Compare to** is set to **Elements**.
Planar shapes	If on, planar shapes are considered as reference elements.
Tolerance	Defines a stroking tolerance for curved element types.
Above is	Defines which direction is considered positive distance from the shapes: • Upwards - points above a shape has a positive distance. • Clockwise side - point on clockwise side of a shape has a positive distance.
Linear elements	If on, compares against linear vector elements.
Surface elements	If on, compares against surface elements.

✍ Distance coloring can be updated in a fast way by using the *Update Distance Coloring* tool.

Color by Point color

Color by Point color setting displays the points with their RGB values. This requires that RGB values are stored as attribute for the laser points.

Color values for laser points can be extracted from raw images or raster attachments loaded into TerraPhoto, or assigned per class by setting values for red, green, and blue. See **Extract color from images** on page 293 and **Assign color to points** on page 291 for corresponding menu commands in TerraScan.

The **Point color** option in the **Display mode** dialog is only active, if the laser points have color attributes.

Color by Scanner

Color by Scanner setting displays the points according to their scanner number. This is useful to distinguish points from different scanners of a ground-based mobile system. At the moment, the scanner number is only stored in LAS files.

To change the colors for scanners, click the **Colors** button next to the **Class by** field. This opens the **Scanner coloring** dialog which is the same as for **Color by Flightline**.

Color by Shading

Color by Shading setting displays a triangulated surface of the points colored by class and shaded by triangle slope. This is useful to check for example ground classification results because error ground points show up clearly in the shaded surface display.

When **Color by Shading** is selected, two more fields are added to the **Display mode** dialog:

- **Azimuth -** direction of the light source. Zero is north.
- **Angle -** height above horizon of the light source.

Color by Line & Intensity

Color by Line & Intensity setting combines flightline and intensity coloring in on display. The color value is determined by the flightline while the brightness of the color indicates the intensity value. This can be used for example to check the horizontal accuracy of overlapping flightlines.

To change the colors for flightlines, click the **Colors** button next to the **Class by** field. This opens the **Flightline coloring** dialog which is the same as for **Color by Flightline**. The brightness values for intensity cannot be changed.

Color by Echo length

Color by Echo length setting displays points according to the length of the return pulse. The value is relative to a typical return pulse length on a hard surface. The echo length has to be extracted from a waveform file delivered from the system.

Color by Slope

Classify pulldown menu

Commands from **Classify** pulldown menu are used to classify laser points and to detect planes or trees from laser points.

To:	Choose menu command:
Start an automatic classification routine for points	**Routine**
Start an automatic classification routine for point groups	**Groups**
Classify points inside a fence	**Inside fence**
Classify points inside a 3D fence	**3D fence**
Detect plane areas from laser points	**Detect plane**
Detect trees from laser points	**Detect trees**
Assign a new class to a laser point	**Assign**
Add points within a specific area to ground class	**Add point to ground**

Routine

Routine sub-menu contains commands for calling automatic classification routines. They can be used to classify points loaded in TerraScan. Most of the classification routines are also available as macro actions in order to use them in batch processing.

The different routines are explained in detail in Chapter **Classification Routines** on page 397.

Groups

Groups sub-menu contains commands for calling automatic classification routines for point groups. They can be used to classify points loaded in TerraScan. Most of the classification routines are also available as macro actions in order to use them in batch processing.

Classifying groups requires that a group number is assigned to the points. See **Assign groups** command for more information.

The different routines are explained in detail in Chapter **Classification Routines** on page 397.

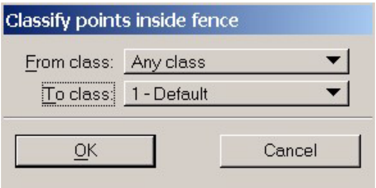
Inside fence

Inside fence menu command classifies points inside a fence or selected polygon from one class into another class. Only points that are displayed on screen are affected by the classification.

➢ **To classify points inside a fence:**

1. Draw a fence or select a polygon.
2. Select **Inside fence** command from **Classify** pulldown menu.

   This opens the **Classify points inside fence** dialog:



3. Select classes in the **From class** and **To class** fields and click OK.

   This classifies the points that are displayed and fall inside the fence.

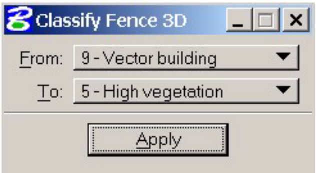
3D fence

3D fence menu command lets you classify points inside a 3D fence from one class into another class. Only points that are displayed on screen are affected by the classification. The 3D fence is defined in two steps. Usually, a fence or selected polygon is drawn in a top view first. Then the tool waits for the selection of a second view, which is automatically turned into a front view covering the same area as defined by the fence. In this front view a second fence can be drawn to define the final 3D fence content.

➢ **To classify points inside a 3D fence:**

1. Draw a fence or select a polygon using MicroStation tools.
2. Select **3D fence** command from **Classify** pulldown menu.
3. Select a second view.

This turns the second view into a front view, and opens the **Classify Fence 3D** dialog:



4. Select classes in the **From** and **To** fields.
5. Draw a fence in the second view. The fence tool is already started by the command.
6. When at least 3 vertices for the fence are defined, the **Apply** button in the **Classify Fence 3D** dialog becomes active. Click **Apply** to finish the fence.

   This classifies the points that are displayed in the first view and fall inside the fence.

Detect plane

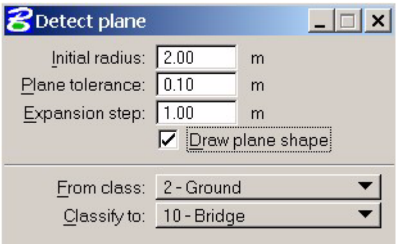
Not Lite

Detect plane menu command detects points on a plane inside a fence or selected polygon from one point class. It classifies the points into another class and optionally draws a polygon around the points on the plane.

➢ **To detect a plane from laser points:**

1. Draw a fence or select a polygon around the area where the plane(s) are located.
2. Select **Detect plane** command from **Classify** pulldown menu.

   This opens the **Detect plane** dialog:



3. Define settings.
4. Click inside the fence to define a start point for the plane detection.

   The software highlights points that are found on a plane.

5. Accept highlighted points to classify points on the plane and to draw a plane shape (optionally).

Setting:	Effect:
Initial radius	Start radius for plane detection.
Plane tolerance	Tolerance how close points must match a fitted plane equation.
Expansion step	Maximum gap among points belonging to the same plane.
Draw plane shape	If on, a 3D polygon is drawn around the points on the detected plane.
From class	Source class from which points are used for plane detection.
Classify to	Target class for points on the detected plane.

Detect trees

Not Lite

Detect trees menu command detects trees from the laser point cloud automatically based on tree shape definitions. This requires the classification of the laser points into ground, vegetation and preferably building points as well as the definition of tree types in TerraScan settings. See **Tree types** on page 61 for information on how to define tree types.

For detected trees, either MicroStation cells or RPC cells can be placed to represent trees in 3D visualizations. MicroStation cells have to be defined in the **MicroStation Cell Library** to be placed correctly. RPC cells are replaced by RPC files when a view is rendered and if the software finds the RPC file at the given location. Settings for cell names and RPC files can be found in **Tree types** as well.

RPC files are purchased by Archvision (www.archvision.com). For more information about RPC cells and visualization options, see TerraPhoto User's Guide, Chapter 6, or the MicroStation On-line Help.

➢ **To detect trees from laser points:**

1. Select **Detect trees** command from **Classify** pulldown menu.

   This opens the **Detect trees** dialog:

2. Select one or more tree types listed in the upper part of the dialog.
3. Define settings for tree detection.
4. Click OK.

   The software starts the detection process. It classifies points from detected trees into the defined class and places cells and/or RPC cells on each tree location (optionally). A process window shows the progress of the detection. Depending on the amount of ground and vegetation points loaded into TerraScan and given settings the process might take some time.

Setting:	Effect:
Ground class	Class representing the ground level.
From class	Source class from which trees are detected.
To class	Target class for points from detected trees.
Find	Determines how many trees are detected: • More trees - higher amount of trees is detected. • Normal level - normal amount of trees is detected. • Fewer trees - lower amount of trees is detected.
Tolerance	Positional tolerance for laser points.

Setting:	**Effect:**
Use echo information	If on, echo information is considered when determining what is likely to be a tree.
Inside fence only	If on, the detection area is limited to a fence area. Requires the definition of a fence or a selected polygon.
Place cell	If on, MicroStation cells are places at detected tree locations.
Place RPC cells	If on, RPC cells are placed at detected tree locations.

✍ RPC cells can be also placed manually based on laser points and aerial images using the *Place Rpc Tree* tool in TerraPhoto. See TerraPhoto User’s Guide for more information.

Assign

Assign menu command changes the class of one laser point.

It performs exactly the same action as the *Assign Point Class* tool.

Add point to ground

Add point to ground menu command lets you classify points inside a certain area to the ground class. This might be useful to correct classification errors effectively in areas where the automatic ground classification does not provide a good result.

The command is used most likely after an automatic ground classification has been performed. It is described in connection with the **Ground** classification routine. See **Add point to ground** on page 403 for more information.

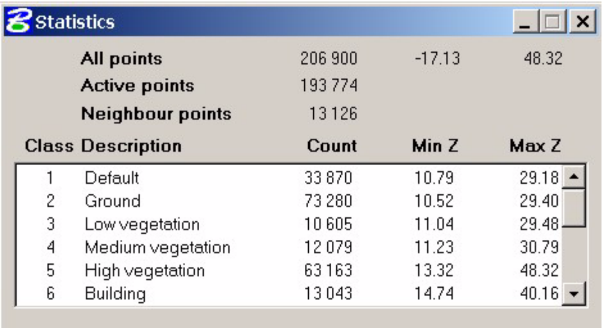
Tools pulldown menu

Menu commands from **File** pulldown menu in **Main** window are used to perform different actions to laser points or based on laser points.

To:	Choose menu command:
View statistics about points	**Show statistics**
Open the **Macro** window	**Macro**
Start an addon tool	**Addon**
Draw bounding box for fitting views	**Draw bounding box**
Draw points into a profile drawing	**Draw into profile**
Draw points into alignment cross sections	**Draw into sections**
Draw polygons around groups of points	**Draw polygons**
Smooth points	**Smoothen points**
Remove unnecessary point density	**Thin points**
Adjust the elevations of points to a geoid model	**Adjust to geoid**
Extract a local geoid model from a geoid file	**Convert geoid model**
Transform loaded points into a new projection system	**Transform loaded points**
Transform a known points file into a new projection system	**Transform known points**
Check the z accuracy of the laser data	**Output control report**
Assign a group value to laser ponts	**Assign groups**
Assign color values to laser points	**Assign color to points**
Assign a normal vector value to laser points	**Compute normal vectors**
Assign color values from images to laser points	**Extract color from images**
Assign echo properties to laser points	**Extract echo properties**
Compare laser points with points from a reference project	**Compare with reference**
Sort laser points	**Sort**
Draw building models from text files	**Read / Building models**
Draw paint markings from text files	**Read / Paint lines**
Draw section parameter values from text files	**Read / Section parameters**
Draw slope arrows from text files	**Read / Slope arrows**

Show statistics

Show statistics menu command displays basic statistics information about laser points. In the upper part of the **Statistics** window, the amount of all points, active points and neighbor points are listed, as well as the elevation range for all points. In the lower part, the separate classes are listed with the point count, minimum elevation and maximum elevation values for each class.



Macro

Not Lite

Macro menu command opens the **Macro** window which lets you create a macro for automating processing steps. The creation and use of macros in TerraScan is described in detail in chapter **Macros** on page 429.

Addon

Addon sub-menu consists of commands that call custom functions in TerraScan. Commands can be added to TerraScan using a custom DLL. An example is given by the **View histogram** function, that opens the **Laser intensity histogram** window. It displays the distribution of intensity values of all loaded points or selected classes.

See Chapter **DLL Interface** on page 457 for more information about custom DLLs in TerraScan.

Draw bounding box

Draw bounding box menu command draws a bounding box around all of the loaded points. This is useful if you want to fit rotated views to display all laser points. Because laser points are kept in TerraScan's memory, MicroStation is not aware of those and will use design file elements only when fitting a view.

The command draws a graphical group of twelve line elements which enclose all laser points. You can then use MicroStation *Fit View* tool to fit rotated views.

As an alternative, the TerraScan command **Fit view** or the **Fit** button in the **Display mode** dialog can be used to fit views.

Draw into profile

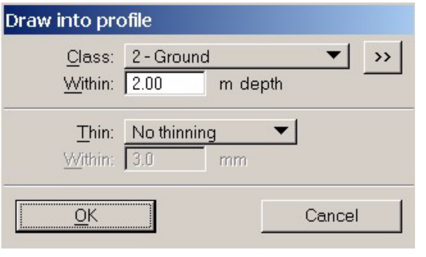
Draw into profile menu command draws laser points into a profile or a cross section as permanent elements. This command will draw all points from one or more given classes which are within the given offset limit from the profile alignment into the profile.

You have to create the profile or the cross sections using TerraModeler before the tool can be used. See TerraModeler User’s Guide for more information.

➢ **To draw laser points into a profile:**

1. Choose **Draw into profile** command from **Tools** menu.

   This opens the **Draw into profile** dialog:



2. Enter settings and click OK.
3. Identify the profile cell element.

   This draws the points as permanent MicroStation elements into the profile.

Setting:	Effect:
Class	Source class(es) from which points are drawn into the profile.
Within	Distance from the alignment element.
Thin	Determines if points are thinned for being drawn into the profile: • No thinning - points are not thinned. • As point cloud - appropriate thinning when drawing 3D objects such as trees or powerline towers. • As terrain surface - appropriate thinning when drawing a ground surface.
Within	Determines how close a point must be to the profile alignment to be drawn into the profile.

Draw into sections

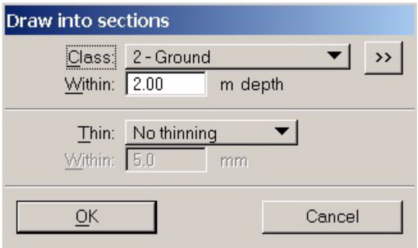
Draw into sections menu command draws laser points into alignment cross sections as permanent elements. This command will draw all points from one or more given classes which are within the given offset limit from the section alignment into the sections.

You have to create the alignment cross sections using TerraModeler before the tool can be used. See TerraModeler User's Guide for more information.

➢ **To draw laser points into sections:**

1. Choose **Draw into sections** command from **Tools** menu.

   This opens the **Draw into sections** dialog:



2. Enter settings and click OK.
3. Identify one of the alignment section cell elements.

   This draws the points as permanent MicroStation elements into the alignment cross sections.

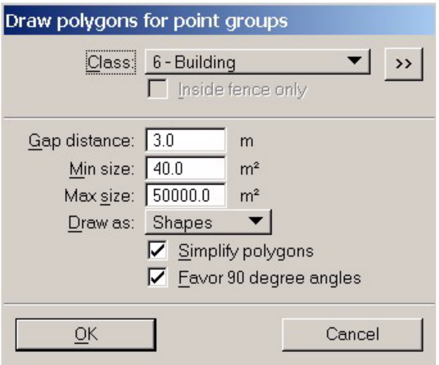
Draw polygons

Draw polygons menu command draws 3D shapes or line strings around groups of points within one or more point classes. The elevation of the vertices for the elements is defined by the laser points.

➢ **To draw polygons around groups of points:**

1. Select **Draw polygons** command from **Tools** pulldown menu.

   This opens the **Draw polygons for point groups** dialog:



2. Define settings.
3. Click OK.

   This draws the polygons around point groups into the design file.

Setting:	Effect:
Class	Source class(es) for drawing polygons around point groups.
Inside fence only	If on, only the area inside a fence or selected polygon is considered for processing.
Gap distance	Maximum distance of gaps up to which points are considered to belong to one point group.
Min size	Minimum size of a polygon to be drawn.
Max size	Maximum size of a polygon to be drawn.
Draw as	Definition of the element type that is created: **Shapes** or **Line strings**.
Simplify polygons	If on, unnecessary vertices of the polygons are removed.
Favor 90 degree angles	If on, 90 degree angles are created whenever it comes close to.

Smoothen points

Smoothen points menu command adjusts elevations of laser point so that a smoother surface is created. You would normally run this on ground point class in order to:

- Remove random variation in laser point elevations and produce a more accurate model.
- Produce a smoother surface so that contours look nicer.
- Produce a smoother surface so that profile drawings look nicer.

Smoothing is an iterative process where every point is compared with close-by points. A best fit plane equation is derived for this group of points and the elevation of the center point is adjusted to better match the plane equation.

During this iteration elevations of laser points are adjusted so that a smoother surface is created. At the end of the iteration the application will try to decide what areas have become smooth and what areas did not result in a smooth surface. Only the smooth areas will be adjusted. Areas which have significant elevation changes will be restored to the original elevation values.

You should not use smoothing if:

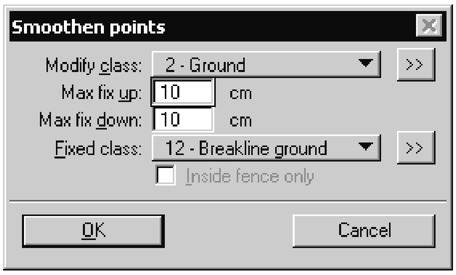
- All of the terrain is covered with low vegetation and there are no smooth surfaces.
- You intend to extract features which have only a small elevation change such as road curb stone lines. Those would be completely smoothed out.

You can incorporate a point class into smoothing which will be used as part of the surface but which will not be modified. You should use this capability if you have classify points close to breakline features into a separate class.

➢ **To smoothen points:**

1. Select **Smoothen points** command from **Tools** menu.

   This opens the **Smoothen points** dialog:



2. Enter setting values and click OK.

   This starts the iteration which will adjust laser point elevations.

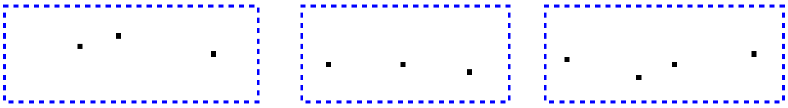
Setting:	Effect:
Modify class	Point class(es) to smoothen and to compare against.
Max fix up	Maximum elevation change upwards to apply to a point.
Max fix down	Maximum elevation change downwards to apply to a point.
Fixed class	Point class(es) to compare against but points in these classes will not be modified.

✍ Smoothing uses a large amount of memory. Approximately 92 bytes is allocated for every point to be smoothed. This means that you can run smoothing on about 4 million points on a computer with 512 MB of RAM.

Thin points

Thin points menu command removes unnecessary point density by removing some of the points which are close to each other. You would often use this to thin ground points before creating a triangulated surface model. See **Model keypoints** on page 424 for an alternative method for thinning points.

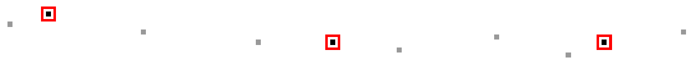
The thinning routine tries to find groups of points where all the points are within the given horizontal distance and the given elevation difference from a central point in the group. The **Keep** setting determines which of the points in each group will be kept.



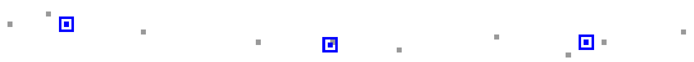
Groups of points

Highest point in each group

Lowest point in each group



Central point in each group



Created average for each group

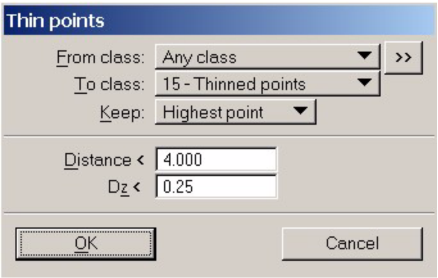
There is also the option to keep points from one class.

The removed points can be either deleted or classified into another point class.

➢ **To thin points:**

1. Select **Thin points** command from **Tools** menu.

This opens the **Thin points** dialog:



2. Enter setting values and click OK.

This removes some of the points from the given class.

Setting:	**Effect:**
From class	Class from which to remove unnecessary points.
To class	Class into which removed points are classified.
Keep	Which point to keep in a group of close points: • Highest point - point with highest elevation. • Lowest point - point with lowest elevation. • Central point - point in the middle of the group. • Create average - substitute group by creating an average. • Point class
Distance	Horizontal distance limit between two points.
Dz	Elevation difference limit between two points.

Adjust to geoid

Adjust to geoid menu command adjusts the elevation values of a laser data set to a local elevation model defined by a text file, a TerraModeler surface or a selected linear chain.

The theory of geoid adjustment and the use of the command in TerraScan are explained in detail in section **Geoid adjustment** on page 313.

➢ **To run elevation adjustment on loaded points:**

1. Choose **Adjust to geoid** command from **Tools** pulldown menu in the **Main** window.

   This opens the **Adjust to geoid** dialog:

2. Select input model type and click OK.

   If you selected **Points from file** as the **Dz model**, the application will ask you to select the input text file.

Convert geoid model

Convert geoid model menu command converts a source geoid model into a new geoid definition. Supported source geoid models are:

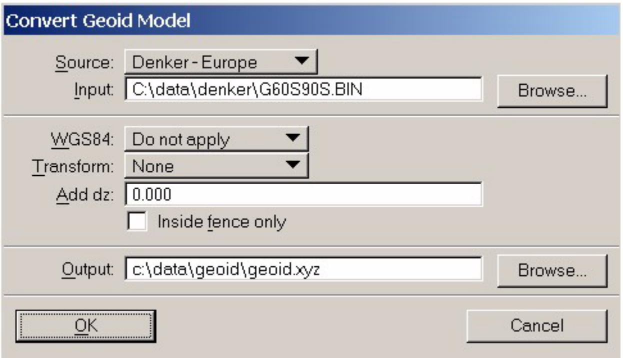
- **Denker -** European geoid model
- **EGM96 -** global geoid model
- **EGM2008 -** global geoid model
- **GSIGEOME -** Japanese geoid model
- **RAF98 -** French geoid model

The process can include a conversion from WGS84 into a projection system, a transformation between projection systems as well as the addition of an elevation difference. A projection system has to be activated and a transformation has to be defined in TerraScan settings before they can be used in this command. See **Coordinate transformations / Transformations** and **Coordinate transformations / User projection systems** for more information.

➢ **To convert a geoid model:**

1. Select **Convert geoid model** command from **Tools** menu.

   This opens the **Convert Geoid Model** dialog:



2. Enter settings and click OK.

   This converts the source geoid model into the output file.

Setting:	Effect:
Source	Source geoid model
Input	Name and location of the source geoid model file.
WGS84	Conversion from WGS84 into a given projection system.
Transform	Transformation into the given projection system.
Add dz	Addition of an elevation difference.
Output	Name and location of the target file.

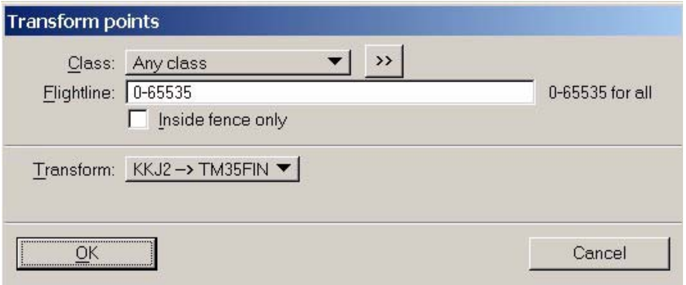
Transform loaded points

Transform loaded points menu command applies a coordinate transformation to all points or to points in selected class(es) and to all or selected flightlines. The transformation can be defined either by an elevation difference value, a text file storing difference values for easting, northing and elevation, or a projection change. A transformation between projection systems must be defined in user settings before selecting this menu command. See for more information on defining coordinate transformations in **Coordinate transformations / Transformations**.

➢ **To transform loaded points:**

1. Select **Transform loaded points** command from **Tools** menu.

   This opens the **Transform points** dialog:



2. Enter settings and click OK.

   This modifies the laser point coordinates using the selected transformation.

Setting:	Effect:
Class	Point class(es) for which the transformation is applied.
Flightline	Flightline number for which the transformation is applied.
Transform	Type of transformation: • Dz - elevation values are transformed by the given value. • Dxyz - coordinate values are transformed by values given in a text file. • User defined projection change.
Dz	If **Transform** is **Dz**, value that is added to the original elevation values of the laser points.
File	If **Transform** is **Dxyz**, name and location of the file storing the transformation values.

✍ Both, the original and the final coordinates must fit inside the design file coordinate system.

Transform known points

Transform known points menu command transforms coordinate values in a text file from one projection system into another. A projection system has to be activated and a transformation has to be defined in TerraScan settings before they can be used in this command. See **Coordinate transformations / Transformations** and **Coordinate transformations / User projection systems** for more information.

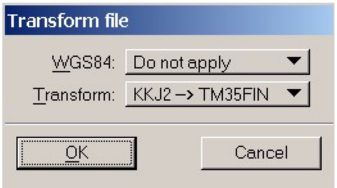
➢ **To transform a known points file:**

1. Select **Transform known points** command from **Tools** menu.

   This opens the **Source point file** dialog, a standard dialog to select a file.

2. Select the file that stores the known points coordinates and click **Open**.

   This opens the **Transform file** dialog:



3. Select desired transformation settings and click OK.

   This opens the **Save transformed points** dialog, a standard dialog to save a file.

4. Define a name and location for the transformed file and click **Save**.

   This saves the transformed known points into a new text file.

Output control report

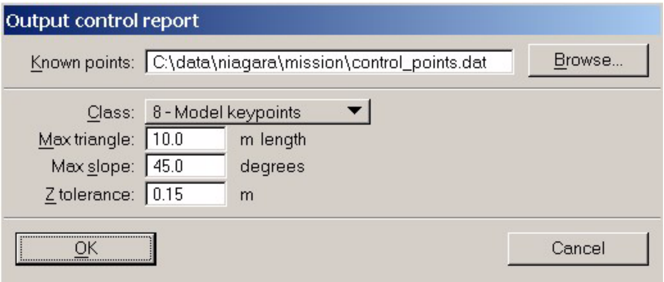
Output control report menu command creates a report of elevation differences between laser points and control points. This can be used to check the elevation accuracy of a laser data set and to calculate a correction value for improving the elevation accuracy of the laser points.

The control points have to be stored in a space delimited text file in which each row has four fields: identifier, easting, northing and elevation. the identifier field is normally a number but it may include non-numeric characters as well.

➢ **To create a control report:**

1. Select **Output control report** command from **Tools** menu.

   This opens the **Output control report** dialog:



2. Select a file that stores the control points.
3. Define settings and click OK.

   This calculates the elevation differences and opens the **Control report** window which is described in detail in Section **Systematic elevation correction** on page 314.

Setting:	Effect:
Known points	Name and location of the file that stores the coordinates of the control points.
Class	Point class used for comparison with the control points.
Max triangle	Search radius around each known point.
Max slope	Maximum terrain slope for which an elevation difference will be computed.
Z tolerance	Normal elevation variation of laser points. This value is used only when computing the terrain slope so that small triangles will not exceed **Max slope**.

Assign groups

Assign groups command assigns a group number to laser points of one or more classes. The software tries to find groups of points based on geometrical characteristics, such as the amount of points close to each other and the distance to other points.

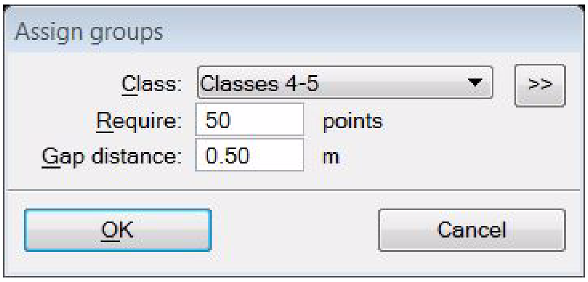
The group number can be used for the visualization of points and for classifying points. For instance, points on moving objects, such as cars on a road in an MLS data set, can by classified into a separate class by first assigning a group number and then using the **Groups** command.

The group number can be stored in TerraScan **Fast binary** files.

➢ **To assign a group number to points:**

1. Select **Assign groups** command from the **Tools** pulldown menu.

   This opens the **Assign groups** dialog:



2. Define settings and click OK.

   This assigns a group number to points in the selected class(es) that fit the requirements. All other points get group number 0.

Setting:	**Effect:**
Class	Point class(es) included in the search for groups.
	Opens the **Select classes** dialog which contains the list of active classes in TerraScan. You can select multiple source classes from the list that are then used in the **Class** field.
Require	Minimum number of points that form a single group.
Gap distance	Minimum distance between two single groups. If the distance between a group and a point islarger than the given value, the point either belongs to another group or to no specific group at all.

Assign color to points

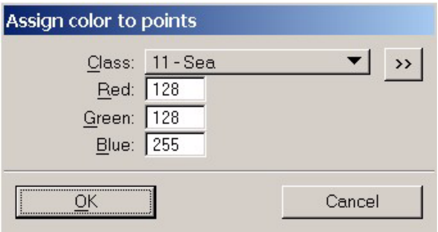
Assign color to points menu command assigns a fixed RGB color value to laser points of one or more classes. This might be useful for laser point classes where color extraction from images results in poor coloring (e.g. thin features like wires) or to emphasize certain classes.

Color values can be stored in TerraScan binary files and LAS 1.2 files.

➢ **To attach a color value to laser points:**

1. Select **Assign color to points** from **Tools** pulldown menu.

   This opens the **Assign color to points** dialog:

2. Select a laser point class in the **Class** list for which to assign color values. Click the >> button to select several classes.
3. Define values for **Red**, **Green** and **Blue** color channel. Values have to be between 0 and 255. A value of 255 for all three channels corresponds to white.
4. Click OK.

   This assigns the defined RGB color to the points in the selected class(es).

Compute normal vectors

Compute normal vectors command can be used to compute and store two additional attributes for laser points, a dimension and a normal vector.

The software determines the dimension of each point by analyzing the point and its closest neighbor points. There are three types of dimensions:

- **Linear -** points form a linear feature.
- **Planar -** points form a planar surface.
- **Complex -** random group of points.

The normal vector is computed for points of planar dimension.

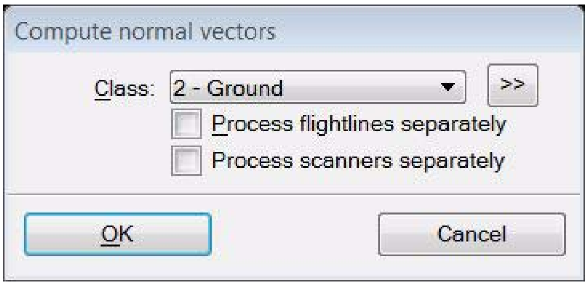
The attributes can be used for the visualization of points and for classifying points. For instance, points on a road surface in an MLS data set can be classified **By normal vector** in order to detect locally flat places. In ALS data, the normal vector can be utilized to analyze roof structures.

The dimension and the three components (XYZ) of the normal vector can be stored in TerraScan **Fast binary** files.

➢ **To compute normal vectors:**

1. Select **Compute normal vectors** command from the **Tools** pulldown menu.

   This opens the **Compute normal vectors** dialog:



2. Define settings and click OK.

   This starts the computation process. It assigns a dimension value to all laser points and a normal vector to all points of planar dimension. An information dialog shows the number of points for which a normal vector has been computed.

Setting:	Effect:
Class	Point class(es) for which dimensions and normal vectors are computed.
	Opens the **Select classes** dialog which contains the list of active classes in TerraScan. You can select multiple source classes from the list that are then used in the **Class** field.
Process flightlines separately	If on, the dimension and normal vector computation is done for each strip separately. This is recommended if the data of different strips do not match to each other.
Process scanners separately	If on, the dimension and normal vector computation is done for each scanner separately. This is recommended if the data of different scanners do not match to each other.

Extract color from images

Extract color from images menu command extracts RGB color values for laser points from raster images. The color source can be orthophotos attached as references or raw images from an image list. Also color points from a color point file can be used to balance colors of the raw images before the colors are assigned to the laser points. All options require TerraPhoto or TerraPhoto Lite.

The color value is derived by sampling all the pixels inside a circular footprint area of each laser point.

The extracted color values can be used to classify points or to make visualization images. Color values can be stored in TerraScan binary files and LAS 1.2 files.

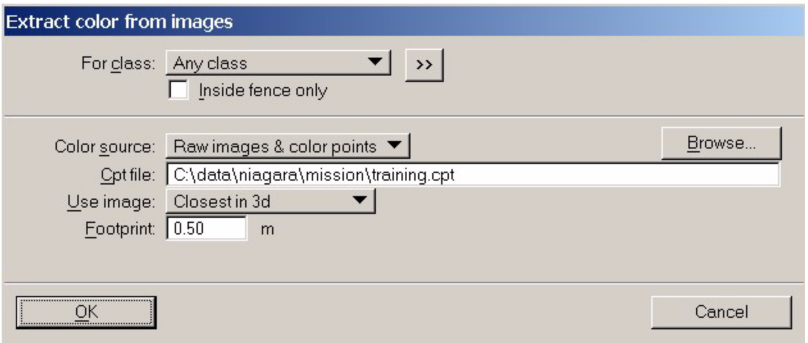
➢ **To extract color from attached or raw images:**

1. Attach reference images with TerraPhoto's *Manage Raster References* tool.

   OR

1. Create a camera calibration file, a mission definition and an image list in TerraPhoto.
2. Select **Extract color from images** command from **Tools** pulldown menu.

   This opens the **Extract color from images** dialog:



3. Select settings and click OK.

   This derives color values for the laser points from the defined source images.

Setting:	Effect:
For class	Laser point class(es) for which colors are extracted.
Inside fence only	If on, color values are extracted for laser points inside a fence or selected polygon.
Color source	Source files for color extraction: • Ortho images - colors are extracted from attached raster images. • Raw images - colors are extracted from raw images defined in an image list in TerraPhoto. • Raw images & color points - colors are extracted from raw images and from a color point file.
Cpt file	Location and name of a color point file. This is only active if **Color source** is set to **Raw images & color points**.

Setting:	Effect:
Use image	Method how the software calculates the closest raw image: • Closest in 3D - each laser point gets color value from the raw image that has the closest camera xyz position to the laser point. • Closest in xy - each laser point gets color value from the raw image that has the closest camera xy position to the laser point. • Closest in time - each laser point gets color value from the raw image that has the closest time stamp to the laser point. • Closest - Mobile logic - each laser point gets color value from the raw image that is closest in time to the laser point. This method is optimized for mobile data sets.
Footprint	Radius of a circular area around each laser point within which pixel color values are resampled.

Extract echo properties

Extract echo properties command extracts information from waveform data and assigns it as attributes to the laser points. The command requires that waveform data and a scanner waveform profile are available. The processing steps for preparing the extraction of waveform-related information are described in detail in Chapter **Waveform Processing** on page 155.

The command can extract the following attributes:

- **Echo length** - relative length (millimeter) of a return signal compared to a typical return from a hard surface.
- **Echo normality** - difference in shape and position of a peak of a return signal compared to a typical return from a hard surface.
- **Echo position**

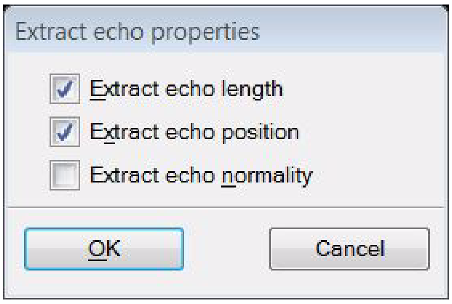
The echo length can be used for the visualization of points and for classifying points. For instance, a classification **By echo length** prior to ground classification can improve the result of the **Ground** routine especially in areas of low vegetation.

The echo properties can be stored in TerraScan **Fast binary** files.

➢ **To extract echo properties:**

1. Select **Extract echo properties** command from the **Tools** pulldown menu.

   This opens the **Extract echo properties** dialog:



2. Select what properties you want to extract by switching the corresponding options on.
3. Click OK.

   This starts the extraction process. It assigns the extracted attributes to all laser points for which waveform information is available. Depending on the amount of points, the process may take some time. An information dialog shows the number of effected points.

Compare with reference

Compare with reference menu command compares two laser data sets from the same location. It classifies locations where the two data sets differ from each other. This is useful to locate places where buildings or other objects have been built or destroyed, trees have grown or ground has changed.

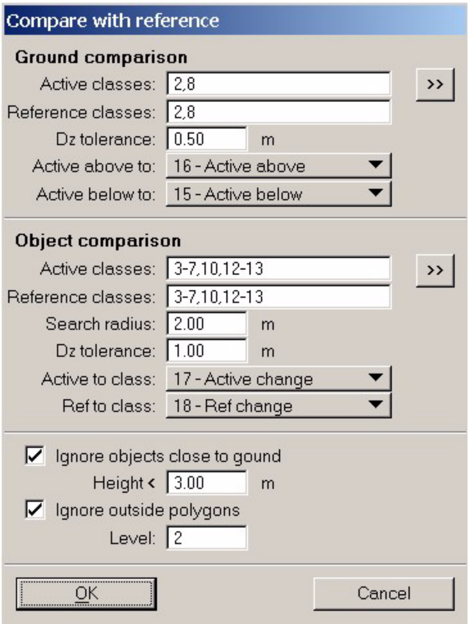
The older data set has to be defined as a reference project. See **New project** on page 320 for information about defining a reference project.

The user can define classes for ground and object comparison. Ground comparison is based on a triangulated ground surface calculated from the given classes. For objects a search radius is defined within which the software expects a corresponding point on the object.

➢ **To compare laser points with a reference project:**

1. Select **Compare with reference** command from **Tools** pulldown menu.

   This opens the **Compare with reference** dialog:



2. Define settings and click OK.

   This classifies points for which there are no corresponding points in the other project.

Setting:	Effect:
Active classes	List of classes for ground comparison in active laser point file.
Reference classes	List of classes for ground comparison in reference project.
Dz tolerance	Elevation tolerance in comparison. Points within this elevation tolerance are considered as corresponding points.
Active above to	Points in the active file that are above the ground in the reference file are classified into this class.
Active below to	Points in the active file that are below the ground in the reference file are classified into this class.
Active classes	List of classes for object comparison in active laser point file.
Reference classes	List of classes for object comparison in reference project.
Search radius	Xy radius around each point to search for corresponding points.
Dz tolerance	Elevation tolerance in comparison. Points within this elevation tolerance are considered as corresponding points.
Active to classes	Points in the active file that differ from corresponding points in the reference file are classified into this class.

Setting:	Effect:
Ref to classes	Points in the reference project that differ from corresponding points in the active project are classified into this class.
Ignore objects close to ground	If on, points close to the ground smaller than the given Height are ignored for object comparison.
Ignore outside polygons	If on, points outside polygons on the given Level are ignored for comparison.

Sort

Sort command sorts loaded laser points according to the selected attribute. The sub-menu includes the following options for sorting points:

- **By time stamp -** points are sorted according to their time stamp.
- **By increasing X -** points are sorted according to increasing easting coordinate values.
- **By decreasing X -** points are sorted according to decreasing easting coordinate values.
- **By increasing Y -** points are sorted according to increasing northing coordinate values.
- **By decreasing Y -** points are sorted according to decreasing northing coordinate values.
- **By increasing Z -** points are sorted according to increasing elevation coordinate values.
- **By decreasing Z -** points are sorted according to decreasing elevation coordinate values.

➢ **To sort laser points:**

1. Select an option from the **Sort** sub-menu from the **Tools** pulldown menu.

   This sort the points according to the selected attribute.

Read / Building models

Read / Building models command reads text files that have been created in an automatic building vectorization process. It is used to draw the buildings as 3D vector models into the design file. See **Vectorize buildings** macro action for more information about the creation of building text files.

The building models are drawn as MicroStation cell elements into the design file. The settings in **Building vectorization / Levels** and **Building vectorization / Model** categories of TerraScan **Settings** determine level, color, and layout definitions of the models.

After drawing the building models into the design file, they can be checked and modified using dedicated tools of TerraScan. They are described in detail in Chapter **3D Building Models** on page 189.

➢ **To read building models into the design file:**

1. Select **Read / Building models** command from the **Tools** pulldown menu.

   This opens the **Read building models** dialog, a standard dialog for opening files.

2. Select building text files and click **Done**.

   This opens the **Read building models** dialog:

3. Select a wall coloring option and click OK.

   This reads the text files and draws the building models into the design file.

Setting:	**Effect:**
Random wall color	If on, the walls are drawn by using colors chosen randomly from the active color table of MicroStation. If off, the color defined in **Building vectorization / Model** category of TerraScan **Settings** is used for all walls.

✍ You can undo the action by using the **Undo** command from the **Edit** pulldown menu of MicroStation.

Read / Paint lines

Read / Paint lines command reads text files that have been created in an automatic vectorization process for linear paint markings on a road surface. It is used to draw the paint markings as 3D line string elements into the design file. See **Find paint lines** macro action for more information about the creation of paint line text files.

The line string elements are drawn on the active level using the active symbology settings of MicroStation.

After drawing the paint lines into the design file, they should be checked with the help of, for example, the *Inspect Elements* tool of TerraScan or the *Validate linear elements* tool of TerraModeler.

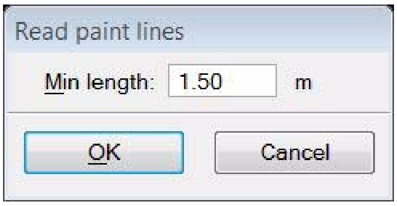
➢ **To read lines for paint markings into the design file:**

1. Select **Read / Paint lines** command from the **Tools** pulldown menu.

   This opens the **Paint line files** dialog, a standard dialog for opening files.

2. Select paint line text files and click **Done**.

   This opens the **Read paint lines** dialog:



3. Define a minimum length value and click OK.

   This reads the text files and draws the paint lines into the design file.

Setting:	Effect:
Min lenght	Minimum length of a paint line that is drawn into the design file. Any shorter line elements are ignored.

✍ You can undo the action by using the **Undo** command from the **Edit** pulldown menu of MicroStation.

Read / Section parameters

Read / Section parameters command reads text files that have been created in an automatic process for extracting road section parameters. It is used to draw the section parameter values into the design file. See **Compute section parameters** macro action for more information about section parameters and the creation of section parameter text files.

The section parameter values are drawn as MicroStation text and linear elements into the design file. The settings in **Road section parameters** category of TerraScan **Settings** determine level, color, text size, and unit definitions of the parameters.

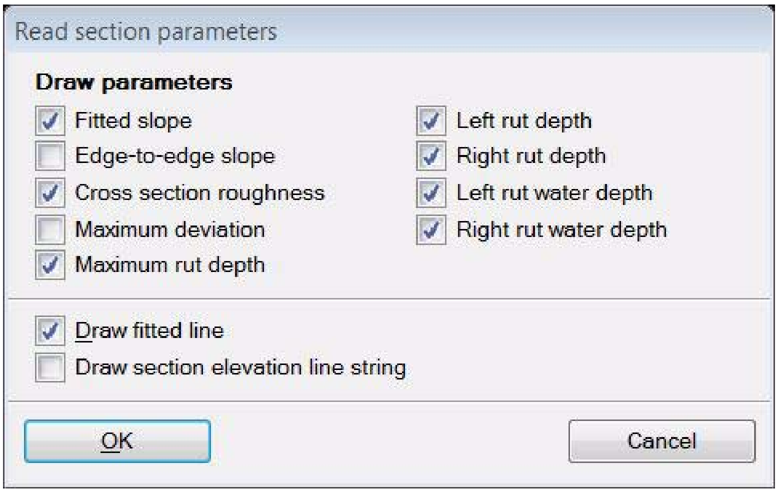
➢ **To read section parameters into the design file:**

1. Select **Read / Section parameters** command from the **Tools** pulldown menu.

   This opens the **Section parameter files** dialog, a standard dialog for opening files.

2. Select section parameters text files and click **Done**.

   This opens the **Read section parameters** dialog:



3. Define parameters that you want to draw and click OK.

   This reads the text files and draws the section parameter values into the design file.

Setting:	Effect:
Fitted slope	If on, the fitted slope value of a section is drawn as text element.
Edge to edge slope	If on, the edge-to-edge slope value of a section is drawn as text element.
Cross section roughness	If on, the roughness value of a section is drawn as text element.
Maximum deviation	If on, the maximum deviation value of a section is drawn as text element.
Maximum rut depth	If on, the maximum depth value of ruts at a section location is drawn as text element.
Left rut depth	If on, the depth value of the left rut at a section location is drawn as text element.
Right rut depth	If on, the depth value of the right rut at a section location is drawn as text element.
Left water depth	If on, the water depth value of the left rut at a section location is drawn as text element.
Right water depth	If on, the water depth value of the right rut at a section location is drawn as text element.
Draw fitted line	If on, the line of the fitted slope of a section is drawn as line element.

Setting:	Effect:
Draw section elevation line string	If on, the line following the elevation variation of a section is drawn as line string element.

✍ You can undo the action by using the **Undo** command from the **Edit** pulldown menu of MicroStation.

Read / Slope arrows

Read / Slope arrows command reads text files that have been created in an automatic process for extracting the superelevation of road lanes. It is used to draw arrows and labels into the design file. The arrows point into the direction of the slope and the labels show the gradient of the slope. See **Compute slope arrows** macro action for more information about slope arrows and the creation of slope arrow text files.

The slope arrows are drawn as 3D line string elements and the gradient values as text elements into the design file. The elements are drawn on the active level using the active line width, line style, and text size settings of MicroStation. The color is determined by settings in the **Read slope arrows** dialog.

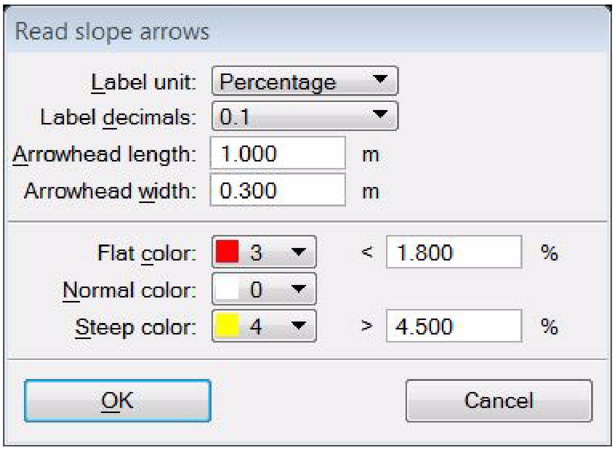
➢ **To read slope arrows into the design file:**

1. Select **Read / Slope arrows** command from the **Tools** pulldown menu.

   This opens the **Slope arrow files** dialog, a standard dialog for opening files.

2. Select slope arrow text files and click **Done**.

   This opens the **Read slope arrows** dialog:



3. Define settings and click OK.

   This reads the text files and draws the slope arrows an values into the design file.

Setting:	Effect:
Label unit	Unit for expressing the slope gradient: **Degree** or **Percentage**.
Label decimals	Number of decimals used in slope gradient text elements.
Arrowhead length	Lenght of the arrowhead in the arrow drawing. Given in design file units.
Arrowhead width	Width of the arrowhead in the arrow drawing. Given in design file units.
Flat color	Color of slope arrows and labels if the gradient is smaller than or equal to the given value.
Normal color	Color of slope arrows and labels if the gradient value is larger than the value defined for **Flat color** and smaller than or equal to the value defined for **Steep color**.
Steep color	Color of slope arrows and labels if the gradient is larger than the given value.

✍ You can undo the action by using the **Undo** command from the **Edit** pulldown menu of MicroStation.

Flightline pulldown menu

Menu commands from **Flightline** pulldown menu in **Main** window are used to manipulate laser points based on flightline information or to create and modify flightlines.

To:	Choose menu command:
Assign flightline number to laser points from trajectory information	**Deduce using time**
Assign flightline number to laser points from scan pattern	**Deduce from order**
Start a new flightline from a selected laser point	**Start new at selection**
Modify the numbering of flightlines	**Modify numbering**
Draw approximate flight path	**Draw from points**
Draw a flight path from a raw trajectory file	**Draw from file**
Apply a boresight angle correction	**Adjust laser angles**
Remove points from overlapping flightlines	**Cut overlap**

The **Flightline** pulldown menu is not available if laser data is loaded into TerraScan using the *Load Ground Points* tool. In this case it is replaces by the **Measurement** pulldown menu which offers special commands for laser data from static terrestrial scanners.

Deduce using time

Deduce using time command assigns flightline numbers to laser points based on time stamps and imported trajectory information. It essentially looks at the time stamp of each laser point, finds a trajectory which covers that time and assigns that trajectory number to the laser point.

This command if often the easiest way to make sure that laser point flightline numbering matches trajectory numbering.

This command should be used only when **GPS second-of-week** time stamps on laser points are unique **--** it should not be used if you have loaded multiple flight sessions from the same week day but different weeks.

If both, laser points and trajectory store **GPS standard time** stamps, the time information is unique and there are no roll-over problems between different weeks.

Deduce from order

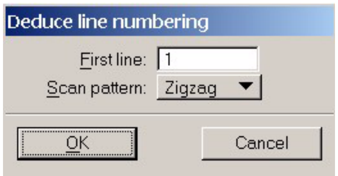
Deduce from order command assigns a flightline number to the laser points according to the scan pattern of the scanner system. There are two scan pattern supported:

- **Zigzag -** the scanner system produces points in a linear zigzag pattern for each flightline.
- **Rotating** - the scanner system produces points in a circular pattern for each flightline.

➢ **To deduce flightlines from order:**

1. Select **Deduce from order** command from **Flightline** pulldown menu.

   This opens the **Deduce from order** dialog:



2. Type a number for the first flightline in the **First line** field.
3. Select a **Scan pattern** type.
4. Click OK.

   This attaches flightline numbers to the laser points according to the scan pattern. A message window informs about the number of resulting flightlines.

Start new at selection

Start new at selection command starts a new flightline at the location of a selected laser point. From a selected laser point, a new flightline number is attached to all laser points that recorded after the selected points. The numbering of all following flightlines is increased by 1.

➢ **To start a new flightline numbering at a selection:**

1. Select a laser point in TerraScan **Main** window using the **Identify** button or by selecting a row in the point list.
2. Select **Start new at selection** command from **Flightline** pulldown menu.

   This attaches new flightline numbers to all following laser points.

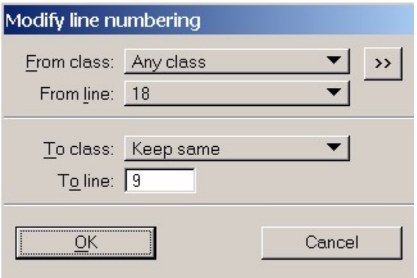
Modify numbering

Modify numbering command modifies the number of single flightlines. The modification can be done for all or a selection of laser point classes. Additionally, points of the renumbered flightline can be reclassified to another class at the same time.

➢ **To modify flightline numbering:**

1. Select **Modify numbering** command from **Flightline** pulldown menu.

   This opens the **Modify line numbering** dialog:



2. Select settings and click OK.

   This modifies the numbering for the laser points in the selected classes.

Setting:	Effect:
From class	Laser point class for which the new flightline number is attached.
From line	Flightline number to be changed.
To class	Target class for laser points of the renumbered flightline.
To line	New flightline number.

Draw from points

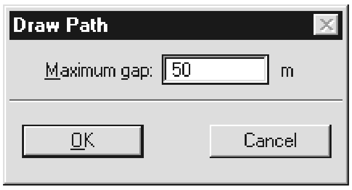
Draw from points menu command draws an approximate flight path into the design file as a linear element. This will work only if the laser points are in original scan order as the command deduces the flight path from laser points only.

The created linear element can be used as an alignment for drawing profiles, deleting points by centerline or outputting alignment reports.

➢ **To draw an approximate flight path:**

1. Choose **Draw path** command from **Flightline** menu.

   This opens the **Draw Path** dialog:



2. Enter maximum gap and click OK.

   This computes the approximate flight path and draws it as one or several linear elements.

Setting:	Effect:
Maximum gap	Starts a new path element whenever the distance between two consecutive laser points exceeds this value.

Draw from file

Draw from file menu command draws the flight path into the design file using the original trajectory file from the post-processing software. A linear element with the active symbology is created.

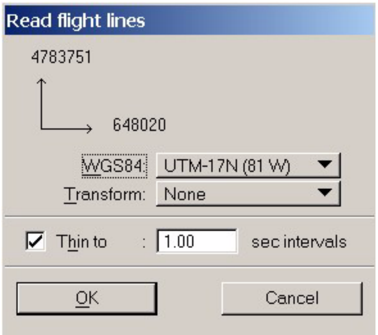
➢ **To draw the flight path into the design file:**

1. Select **Draw from file** command from **Flightline** pulldown menu.

   This opens the **Read flight lines** dialog, a standard dialog for selecting files.

2. Select the raw trajectory file(s), **Add** them to the list and click **Done**, when all files are added to the list.

   This opens the **Read flight lines** dialog:



   The first coordinate values of the selected trajectory are displayed in the upper part of the dialog. This helps to select the correct projection system for drawing the trajectory.

3. Select a projection system in the **WGS84** selection list.
4. (Optional) Select a transformation from the **Transform** list that is applied to the trajectory.
5. (Optional) Select **Thin to** and define a value for thinning the drawn trajectory line. This reduces the amount of vertices of the resulting line element in the design file.
6. Click OK.

   This draws the flight path into the design file.

✍ For more information about projection systems and transformations see **Coordinate transformations / Built-in projection systems** and **Coordinate transformations / Transformations** in TerraScan settings.

Adjust laser angles

Adjust laser angles menu command applies a positional correction to laser points. This command is normally used to fix for a laser misalignment by applying heading, roll or pitch corrections to the points.

Adjusting laser angles is possible only when the application can determine where each laser point was fired from. The application can determine this only if trajectories have been imported and the flightline numbering on laser points matches trajectory numbering. See **Deduce using time** on page 304 for a description how to match flightline and trajectory numbering.

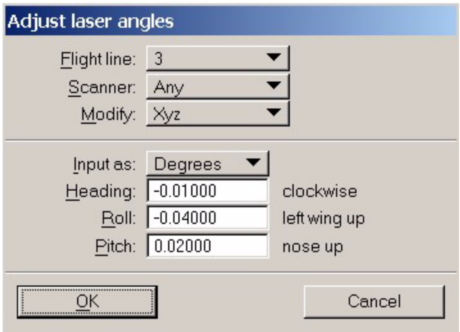
Time stamps for laser points are not mandatory. If time stamps are present, the application can derive the laser scanner position for each laser point more accurately. If time stamps are not present, the application will find a perpendicular projection from each laser point to the trajectory.

Modify setting allows you to select what coordinate axis the adjustment will be applied to: **Xyz**, **Xy** or **Z**. In all cases the application computes an xyz correction as a true angular rotation of the vector from laser scanner to the point. This setting only affects which coordinate axis the derived corrections will be applied to.

➢ **To adjust laser angles:**

1. Select **Adjust laser angles** command from **Flightline** pulldown menu.

   This opens the **Adjust laser angles** dialog:



2. Select settings and click OK.

   The application derives the laser scanner position for each point and applies a positional correction according to the settings.

Setting:	Effect:
Flightline	Flightline to which to apply the correction. Choose **Any** to apply the correction to all loaded points.
Scanner	Scanner to which to apply the corrections. This is mainly used for mobile ground-based systems if more than one scanner is included.
Modify	Coordinates to modify: • Xyz - applies a true angular correction. • Xy - applies a horizontal shift relative to trajectory and flight direction. • Z - applies a change to elevation only.
Input as	Unit of angle values: • Degrees - angles given in decimal degrees. • Radians - angles given in radians. • Ratio - angles given as a ratio over a value corresponding to a full circle.
Heading	Heading correction, increases clockwise.
Roll	Roll correction, increases left wing up.
Pitch	Pitch correction, increases nose up.

Cut overlap

Not Lite

Cut overlap menu command removes laser points from locations where laser data from multiple flightlines overlap. You can select if overlapping points will be classified to a specific class or if they will be deleted from the data set.

This command can perform cutting using three distinct principles: **Cut by quality**, **Cut by offset** and **Cut by range**.

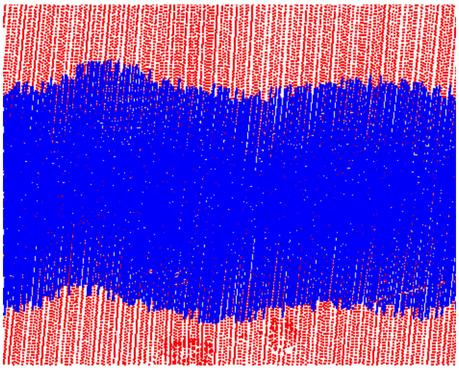
Cut by quality

Cut by quality option removes lower quality laser points from locations where there is laser data from a better flightline. This action makes sense only if there is a clear reason why one flightline is better than another one. You might use this if:

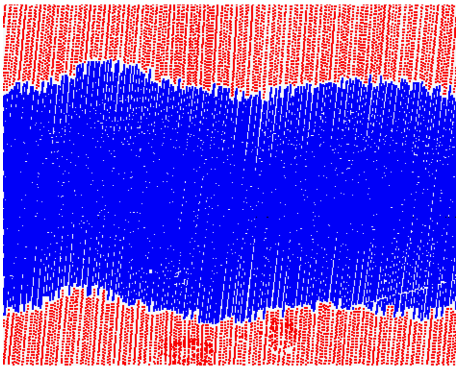
- You have laser data from two different flying altitudes. A lower flight pass is normally more accurate than a higher one.
- Hardware was not working at the best level for some flightlines.
- GPS trajectory is weak for some flightlines.

You can define the quality of different flightlines as an attribute of the imported trajectories using *Manage Trajectories* tool. If you do not have trajectory information, you can assign quality values to different flightline number sequences in the **Default flightline qualities** category of TerraScan settings.

Cut by quality option does not use time stamps.



Before cut: red is lower quality



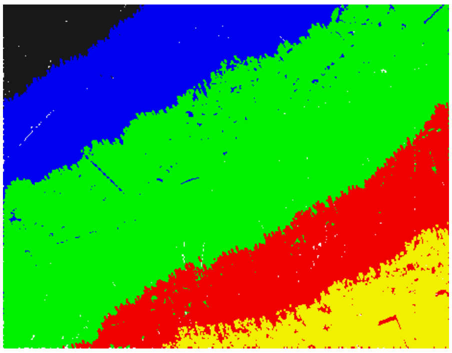
After cut by quality

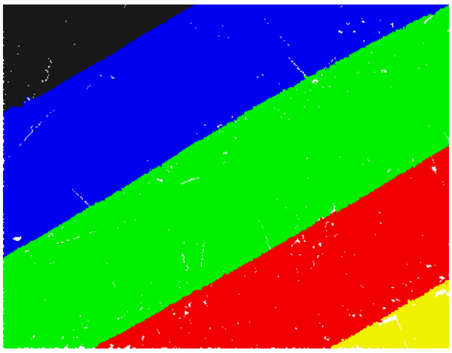
Cut by offset

Cut by offset option removes laser points from the edges of the corridor if that location has been covered more vertically from another flightline. This action serves two purposes:

- Removing edges of the corridors produces a more uniform data density and point pattern.
- Magnitude of error sources grows as scan angle increases. Removing edges of the corridors gets rid of less accurate data and keeps the more accurate central part.

Cut by offset option requires trajectory information. If laser point time stamps are not available, it will assume a perpendicular projection to the trajectory.


Before cut

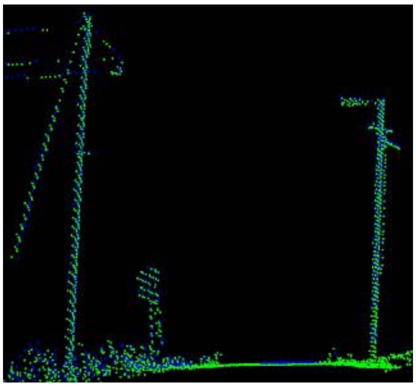

After cut by offset

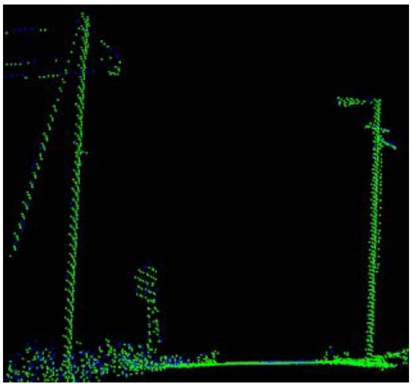
Cut by range

Cut by range option removes laser points from the edges of the corridor if that location has been covered by points from a shorter measurement distance. This option is designed for laser data from mobile ground-based systems.

The software searches for points inside a sphere and cuts the points resulting from long measurements if points from a shorter range at the same location within the search radius are present.

Cut by range option requires trajectory information.

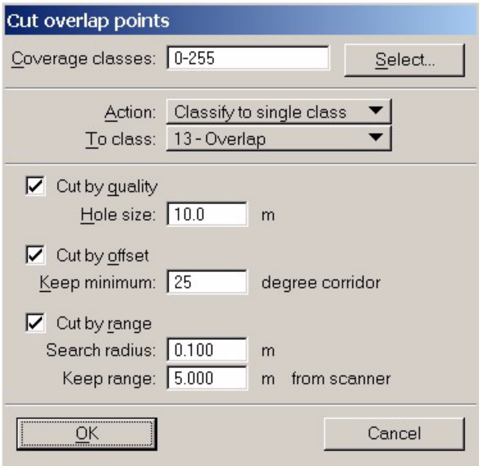

Before cut


After cut by range

➢ **To cut overlap:**

1. Select **Cut overlap** command from **Flightline** pulldown menu.

   This opens the **Cut overlap points** dialog:



2. Enter settings and click OK.

   This removes overlap points according to settings.

Setting:	**Effect:**
Coverage classes	List of point classes to consider when determining if a flightline covers an area. Normally 0-255 to use all classes.
Action	Action to perform on points to cut: • Add constant to class - add given value to class code. • Classify to single class - classify to one target class. • Delete - remove points from the data set.
To class	Target class when **Action** is **Classify to single class**.
Add	Value to add to classes when **Action** is **Add constant to class**.
Cut by quality	If on, cut points from lower quality flightlines.
Hole size	Approximate diameter of maximum acceptable hole.
Cut by offset	If on, cut points by scan angle.
Keep minimum	Minimum central part of the corridor to keep as degrees. Setting this to 20 would keep a +10 .. -10 degree corridor.
Cut by range	If on, cut points from longer measurement distance.
Search radius	Radius of a sphere within which the software searches for closer range points from other drive passes.
Keep range	Range within which all points are kept.

✓ You can run cut overlap on the project blocks as a macro step. When you do this, you should use some overlap from neighboring blocks to avoid ill effects at block boundaries.

11 Coordinate Transformations

TerraScan can apply a coordinate transformation to point clouds at different steps of the processing workflow, for example, when loading or importing points, working with the points in RAM, processing points in batch mode in a TerraScan project or with a macro step, or writing points to output files. Coordinate transformations may also be applied to trajectories.

TerraScan divides coordinate transformations into several categories:

- **Projection system transformations** - used to transform coordinates from one coordinate system to another.
- **User-defined transformations** - coordinate transformations which can be defined by a number of parameters or equations.
- **Geoid adjustment** - used to transform elevation values from one height model to another.
- **Systematic elevation correction** - applies a single correction value to elevation values.

Projection system transformations

The transformation of coordinates from one coordinate system to another is a common task. Usually, the coordinates of raw laser data or trajectories are given in WGS84 or some UTM projections system values. For data processing and/or delivery, it is often necessary to transform these coordinates into another (national) projection system.

Coordinates in WGS84 system can be provided as longitude, latitude, ellipsoidal elevation values or geocentric XYZ values. TerraScan automatically recognizes the coordinate value format when it reads the points or trajectories.

The transformation into the destination coordinate system is usually done when point cloud data or trajectories are imported into TerraScan, for example, at the beginning of the processing workflow, or when data is prepared for delivery.

A projection system has to be activated in **Coordinate transformations / Built-in projection systems** or **Coordinate transformations / US State Planes** categories of TerraScan **Settings**. Only active projection systems are available for transformations. If your local projection system is not implemented in TerraScan, you can define it in **Coordinate transformations / User projection systems** category of TerraScan **Settings**.

After activating the projection system(s), you can define a transformation of type P**rojection change** in **Coordinate transformations / Transformations** category of TerraScan **Settings**. See also **Projection change transformation** for more information.

User-defined transformations

There are different types of transformations that can be used to manipulate the coordinate values of point cloud data and trajectories in TerraScan. The implemented transformation types are:

- **Linear transformation**
- **Equation transformation**
- **Known points transformation**
- **Xy multiply transformation**
- **3D translate & rotate transformation**
- **3D Affine transformation**

You can define the values for the transformation parameters in **Coordinate transformations / Transformations** category of TerraScan **Settings**.

Geoid adjustment

The elevation values of raw laser data and trajectories are often provided as ellipsoidal height values. Usually, these values need to be transformed into orthometric values of a local height system.

For larger areas, the adjustment from ellipsoidal to orthometric height values can not be defined as one mathematical formula. Therefore, the elevation adjustment model needs to be defined by using local points for which the elevation difference between the height systems is known.

In TerraScan, the elevation adjustment can be performed for loaded points or for project blocks in batch mode. See **Adjust to geoid** command for loaded points, **Adjust to geoid** command for project blocks, and **Adjust to geoid** command for trajectories for more information.

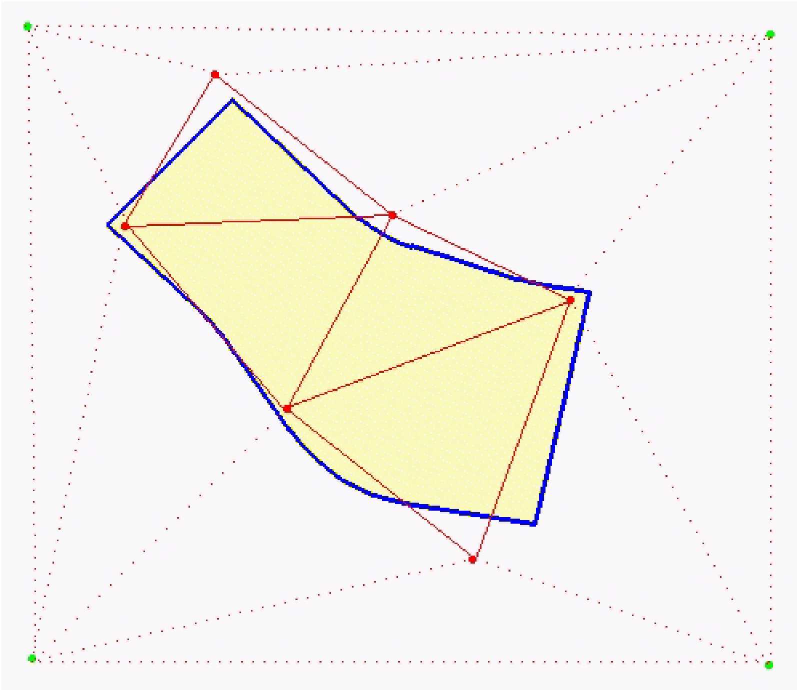
Elevation adjustment model

The input model for geoid adjustment must be provided in one of the following formats:

- **Points from file -** text file containing space-delimited X Y dZ- points.
- **TerraModeler surface -** triangulated surface model created from X Y dZ - points. The surface model in TerraModeler has the advantage that you can visualize the shape of the adjustment model.
- **Selected linear chain -** linear element of which the vertices represent the X Y dZ - points.

XY are the easting and northing coordinates of the geoid model points, dZ is the elevation difference between ellipsoidal and local heights at the location of each geoid model point. Intermediate adjustment values of the model are derived by aerial (text file or surface model as input) or linear (linear element as input) interpolation between the known geoid model points.

The figure below illustrates the aerial interpolation method. The yellow shape represents a project area covered by laser data, the red points symbolize known X Y dZ - points and the green points interpolated X Y dZ - points. The red (dotted) lines show the triangulated model.



The six known points in the illustration above do not create a model that completely encloses the laser data area. If the model does not provide any additional information, TerraScan automatically adds four corner points (green points in the illustration) to expand the elevation adjustment model. Each added corner point has the same dz value as the closest known point.

It is recommended to use an adjustment model that contains the complete project area and thus, provides more accurate elevation information for project boundaries.

Systematic elevation correction

A systematic elevation correction needs to be applied if the point cloud data are systematically shifted in elevation. The systematic shift can be detected by comparing the point cloud with ground control points (GCPs).

TerraScan can do the comparison automatically. The GCPs must be provided in a text file which stores an identifier, X, Y, and Z coordinates in space-delimited fields, one line for each control point. The identifier is normally a number but it may include non-numeric characters as well.

In the point cloud, at least the points on the ground around the GCP locations should be classified into a separate class. In practice, the check of a systematic elevation shift is often done after the ground points have been classified in the point cloud.

In TerraScan, the check of a systematic elevation shift can be performed for loaded points or for project blocks in batch mode. See **Output control report** command for loaded points and **Output control report** command for project blocks for more information.

TerraScan performs the following steps:

1. Read the GCPs from a text file.
2. Scan through project blocks and load laser points from a given class within a given search radius around each GCP.

   OR

2. Scan through loaded points.
3. Create a small triangulated surface model (TIN) from the laser points within a given search radius around each GCP.
4. Compute a laser data elevation for each GCP XY location from the TIN. This effectively interpolates an elevation value from three laser points which are closest to a GCP.
5. Output a **Control point report** that lists all GCPs, the laser elevation value, and the difference between laser elevation and GCP elevation.

The report of the elevation comparison shows, among other things, an **Average dz** value which represents the average elevation shift of the point cloud from the GCP elevations. This value gives some indication about the elevation accuracy of the point cloud. Furthermore, the value can be used in a **Linear transformation** in order to improve the elevation match between point cloud data and GCPs.

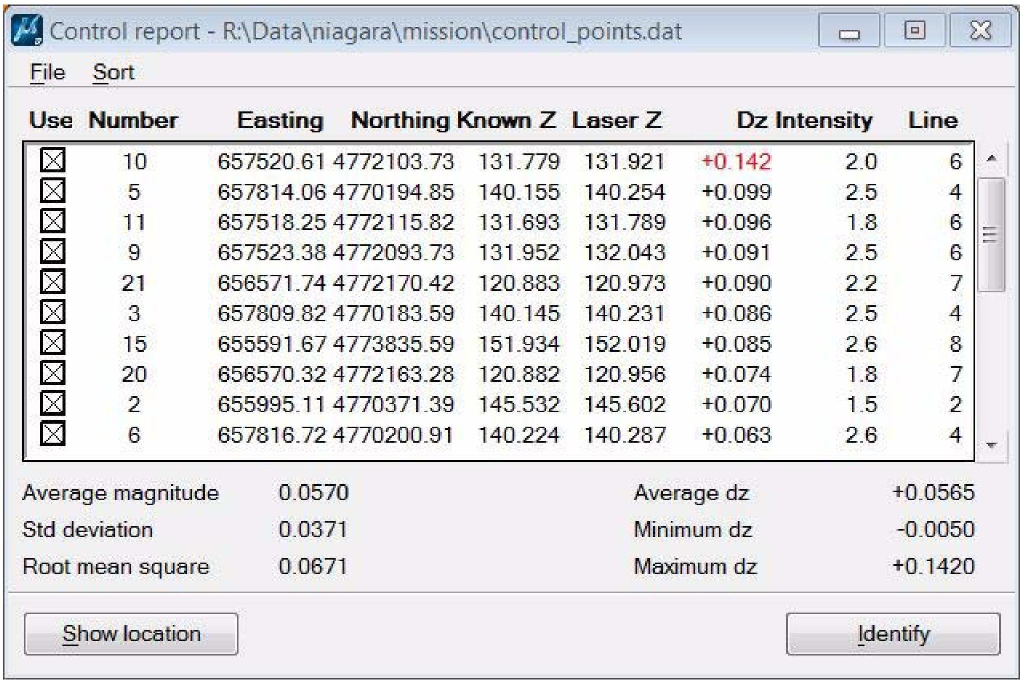
To apply a systematic elevation correction to a point cloud, proceed as follows:

1. Create a control report using either **Output control report** command for loaded points or **Output control report** command for project blocks.
2. Define a **Linear transformation** in TerraScan **Settings**. Use the given **Average dz** value from the report with the inverse sign as **Add constant Z** value in the transformation definition.
3. Apply the transformation using **Transform loaded points** command for loaded points or the **Transform points** action of a macro for project blocks or multiple files.

✍ You can type the dz value directly into the **Transform loaded points** dialog if you want to apply the elevation adjustment to loaded points only. In this case, you do not need to define a transformation in TerraScan **Settings**.

Control point report

The control point report is shown in the **Control report** window:



The window contains the list of all GCPs in the input text file. For each point, the following information is shown:

- **Use** - determines whether a GCP is used in the comparison or not. Switch control points on or off by clicking on the square.
- **Number** - identifier of the GCP.
- **Easting** - easting coordinate of the GCP.
- **Northing** - northing coordinate of the GCP.
- **Known Z** - elevation coordinate of the GCP.
- **Laser Z** - elevation value derived from the laser points at the GCP’s XY location.
- **Dz** - difference between Known Z and Laser Z. If the value exceeds a limit defined in the **Control report settings**, the value is displayed in red.
- **Intensity** - average of intensity values of the laser points at the GCP’s XY location. This is displayed if the option in the **Control report settings** is switched on.
- **Line** - flightline number assigned to the laser points at the GCP’s XY location. This is displayed if the option in the **Control report settings** is switched on.

Below the GCP list, some statistical information computed from the elevation difference values is provided. This includes average magnitude, standard deviation, and root mean square. Additionally, the average, minimum, and maximum value of elevation differences is displayed.

If a line in the list is selected, the MicroStation views defined in the **Control report settings** are centered at the location of the corresponding GCP.

To show the location of a GCP, select a line in the list. Click on the **Show location** button and move the mouse pointer into a view. This highlights the selected GCP with a square.

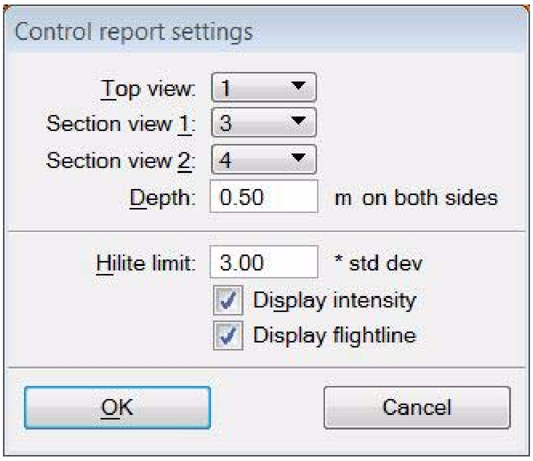
To identify a GCP, click on the **Identify** button and place a data click close to a GCP in a view. This selects the corresponding line in the list.

The GCPs in the list can be sorted in different ways using the commands from the **Sort** pulldown menu.

The report can be saved into a text file or sent to a printer using **Save as text** or **Print** commands from the **File** pulldown menu.

Control report settings

The display settings for the control point report can be changed using **Settings** command from the **File** pulldown menu. This opens the **Control report settings** dialog:



Setting:	Effect:
Top view	Top view that is updated if a GCP is selected.
Section view 1	First section view that is updated if a GCP is selected. The section is drawn in east-west direction.
Section view 2	Second section view that is updated if a GCP is selected. The section is drawn in north-east direction.
Depth	Depth of a section in the section views. The actual depth shown in a section view is the given value * 2.
Hilite limit	Determines the limit for displaying elevation difference values in red in the report.
Display intensity	If on, the average intensity value of the laser points at the GCP location is displayed in the report.
Display flightline	If on, the flightline number of the laser points at the GCP location is displayed in the report.

12 Working with Projects

A project definition in TerraScan helps to organize the work with a huge amount of laser data and to automate processing tasks. Basically, it is a method of dividing the whole data set into smaller, better manageable parts. These smaller geographical regions or blocks should be sized so that the laser data referenced by one block fits into the computer's RAM. There must be some space left in RAM for processing routines. See **Memory usage of loaded points** on page 235 for more information.

A project definition is saved in a TerraScan project file with the extension *.PRJ. It is an ASCII file including:

- **header** - project settings.
- **block names** - laser file name and extension as link between the project and the referenced binary files storing the points.
- **block boundaries** - coordinates of the vertices of each block boundary.

Typical steps for creating a project are:

1. Use *Load Airborne Points* tool or **Read points** command to load a subset of points (every 10th, 50th, 100th, ...) from all input files that belong to a project. This shows the geographical coverage and location of laser points without providing the full point density.
2. Create block boundaries. There are several tools and methods that support the creation of block boundaries:
   - *Measure Point Density* tool - estimate the amount of points inside a certain area by using a rectangle as sample area.
   - MicroStation *Place Shape* tool - place boundary shapes for the blocks. Each block should enclose a manageable number of points. You should use MicroStation snapping tools to avoid gaps and overlap between neighboring block boundaries.
   - *Design Block Boundaries* tool - create shapes from a closed line work and get an approximate number how many points are inside each shape.
   - **Create along centerline** command - create blocks along a linear element.
   - **Create along tower string** command - create blocks along a tower string element for a powerline corridor.
   - Create grid block boundaries automatically during the import of point files. See **Import points into project** command.

   The result of block boundary creation is always a set of shape elements which include the project area for processing. Each shape should enclose a part of the project area with a number of points that easily fit into the computer's memory.
3. Close the points loaded in step 1 using **Close points** command.
4. Select *Define Project* tool. This opens the **TerraScan Project window**.
5. Define a new project using **New project** command.
6. Select all block boundary shapes created in step 3.
7. Add blocks to the project using **Add by boundaries** command.
8. Save the project definition using **Save project as** command.
9. Import points from all input files into the project using **Import points into project** command.

Once the points are imported into the project, you can process the data of the referenced files in batch mode. You can run macros on the project which may include several processing steps. Macros can run in TerraScan but most of the macro actions can also be performed in TerraSlave. See Chapter **Macros** on page 429 for detailed information.

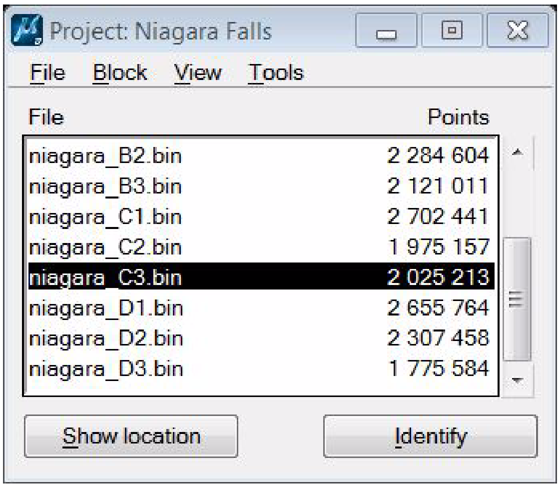
Before you process project data in batch mode, you would test the settings of macro actions based on loaded points in order to find the optimal parameter values for the project area. It is also recommended to test a macro on several blocks before you run it on the whole project.

Besides running macros on a project, there are several processing tasks that can be performed on project level. The corresponding commands are included in the pulldown menus of the **TerraScan Project window** and described in the following sections.

TerraScan Project window

The **Project** window contains all menu commands for creating and modifying project definitions, managing block definitions, and for running processing steps on project level.

Select the *Define Project* tool to open the **Project** window:



In the file list of the project window, all blocks that belong to the project are listed with the amount of points in the referenced laser binary file. If **Project file locking** is active for a project, the list also includes information about file locking.

To select a block, click on the name in the list. Press the <**Ctrl-key**> to select several blocks.

To show the location of a block, select a line in the **Project** window. Click on the **Show location** button and move the mouse pointer into a view. This highlights the boundary of the selected block.

To identify a block, click on the **Identify** button and place a data click inside a block boundary in a view. This selects the corresponding line in the **Project** window. Several blocks can be identified if the <**Ctrl-key**> is pressed while selecting block locations in the view.

If the **Project** window is displaying a **Long list**, you can use the **Select all** and **Deselect all** buttons in order to select and deselect all blocks. The **Invert** button selects all blocks that are previously not selected and deselects previously selected blocks.

File pulldown menu

Menu commands from **File** pulldown menu in **Project** window are used to create a project, edit project information and to import points into a project.

To:	Choose menu command:
Create a new project definition	**New project**
Open an existing project definition	**Open project**
Save changes to an existing project definition	**Save project**
Save a new project definition	**Save project as**
Edit project information and settings	**Edit project information**
Import laser files into the project	**Import points into project**
Import all laser files from a directory	**Import directory**

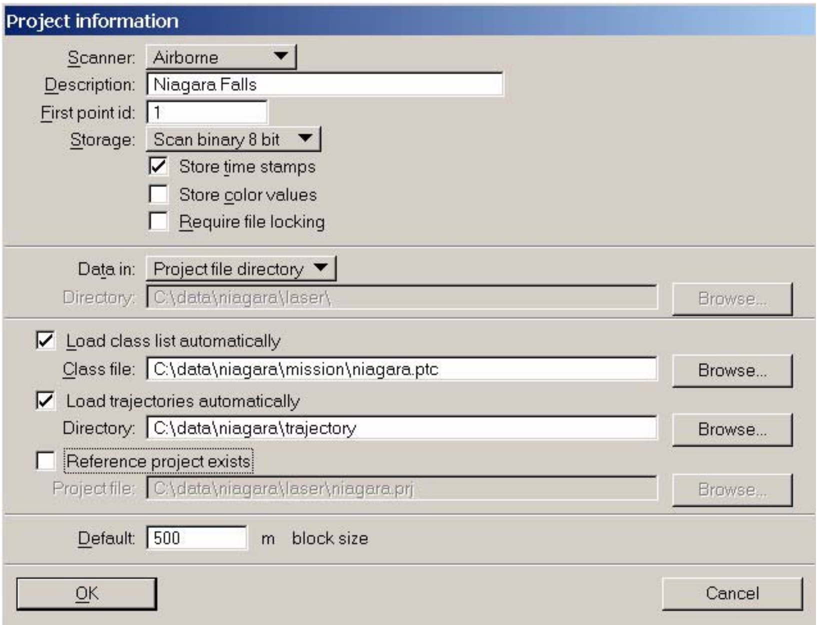
New project

New project command creates a new project definition. The complete project definition includes some descriptive information and a list of block boundaries. For the definition of block boundaries see **Add by boundaries** on page 328.

➢ **To create a project definition:**

1. Choose **New project** command from **File** pulldown menu.

   This opens the **Project information** dialog:



2. Enter settings and click OK.

Setting:	Effect:
Scanner	Scanner type: **Airborne**, **Mobile** or **Ground based**.
Description	Descriptive text for the project.
First point id	Start ID number for the laser data file.
Storage	Block binary file format: **Scan binary 8 bit**, **Scan binary 16 bit**, **EarthData EEBN**, **EarthData EBN** or **LAS binary**. For more information see **Coordinate Transformations** on page 312.
Store time stamps	If on, a time stamp will be stored for each laser point.
Store color values	If on, a RGB color value will be stored for each laser point.
Require file locking	If on, a project block file will be marked as locked when a user opens it for modification with the **Open block** menu command. For more information see **Project file locking** on page 321.
Data in	Defines how the directory for binary files is determined: • Project file directory - same directory where project file is placed. This setting is independent of the absolute path of the data and is therefore good in a network environment or when moving the data set from one computer to another. • Separate directory - complete path is given in the **Directory** field.
Load classes automatically	If on, the defined **Class file** is automatically loaded with the project into TerraScan.
Load trajectories automatically	If on, trajectories from the given **Directory** are loaded automatically with the project into TerraScan.
Reference project exists	If on, the given **Project file** is defined as reference project to be used in corresponding tools.

Setting:	Effect:
Default	Size of rectangular blocks if points are encountered which are outside all of the user block boundaries.

Choosing a project storage format

TerraScan supports multiple file formats from which you need to choose one to be used for the project block binary files. The best choice depends on a few factors:

- If your project may have more than 255 flightlines, you should choose **Scan binary 16 bit** instead of **Scan binary 8 bit**. If your project has fewer than 255 flightlines, **Scan binary 8 bit** is the most compact format.
- If your project is ground based, you must choose **Scan binary 8 bit** or **Scan binary 16 bit**.
- If you wish to have other applications working on the same block binary files, your best choice is **LAS binary** which is an open industry standard format.

Project file locking

In a network work group environment, it may be desirable to implement a file locking scheme which would prevent two people from modifying the same data at the same time. TerraScan supports a relaxed form of such file locking for project block binary files.

Project file locking is active if **Require file locking** setting is switched on in the project definition. This feature is needed only if you have multiple people working on the same project data set through a network.

TerraScan implements file locking in a simple manner. It does not lock the block binary file itself but instead creates a temporary file which informs that the block binary file is undergoing modification action. The temporary file has the same name and location as the block binary file but has extension .LCK. This temporary file only contains the name of the computer which has opened the block binary file for modification. The creation time stamp of the temporary file tells you when the block binary file was opened.

This relaxed locking scheme does not completely prevent modification of a file that has been locked. It only causes that most TerraScan tools refuse to carry out modification action on that block. Other applications do probably not acknowledge or recognize the fact that the block file is locked.

Example case

When file locking is active, your project storage directory might contain the following files:

Block binary file	Temporary lock file
h:\otaniemi\ota000317.bin	h:\otaniemi\ota000317.lck
h:\otaniemi\ota000318.bin	
h:\otaniemi\ota000319.bin	h:\otaniemi\ota000319.lck
h:\otaniemi\ota000320.bin	

The above files would indicate that blocks 000317 and 000319 are undergoing modification and are locked.

Operations supporting locking

The following actions lock a block binary file:

- **Open block** menu command if **Open for setting** is set to **Modification**.
- **Read points** menu command when opening a single block binary file.
- *Load Airborne Points* tool when opening a single block binary file.
- *Load Ground Points* tool when opening a single block binary file.
- Executing **Adjust to geoid** from project window.
- Executing **Run macro** from project window with **Save points at the end** setting on.

The same actions also check whether a file is locked.

Releasing a lock on a block

The project window has a **Release lock** command in the **Block** pulldown menu for releasing a locked block. The user can release a block only if it has been locked by that same workstation or if the locking was done more than 24 hours ago.

As a last precaution, a lock can be released simply by deleting the '.lck' file.

Open project

Open project command opens an existing project definition.

➢ **To open a project:**

1. Select **Open project** command from **File** pulldown menu.

   This opens the **Open project** dialog, a standard Windows dialog to open files.

2. Select a project file and click **Open**.

   This loads the selected project into TerraScan.

Save project

Save project command saves changes of the project definition to an existing project file. This can be used after changing project settings or block definitions.

➢ **To save changes to a project:**

1. Select **Save project** command from **File** pulldown menu.

   This saves the project.

Save project as

Save project as command saves a project by creating a new project definition file with the extension .PRJ. This can be used after creating a new project definition.

➢ **To save a project into a new project file:**

1. Select **Save project as** command from **File** pulldown menu.

   This opens the **Save project** dialog, a standard Windows dialog to save files.

2. Select a location and type a name for the project file. Click **Save**.

   This saves the project file.

Edit project information

Edit project information command lets you edit project definition settings of a loaded project. The project has to be saved to make the changes permanent.

➢ **To edit project information:**

1. Select **Edit project information** command from **File** pulldown menu.

   This opens the **Project information** dialog, which is the same as for defining a new project. See **New project** on page 320 for a description of the settings.

Import points into project

Import points into project command imports point files into a project. The process reads the points from the input files and stores them into new files according to the block definitions. As a result, there is a new point file for each block of the project.

The file names and the format are derived from the block names which contain the name itself and the extension depending to the selected storage format. Thus, a file that is created in the import process is linked to a project by its name and extension.

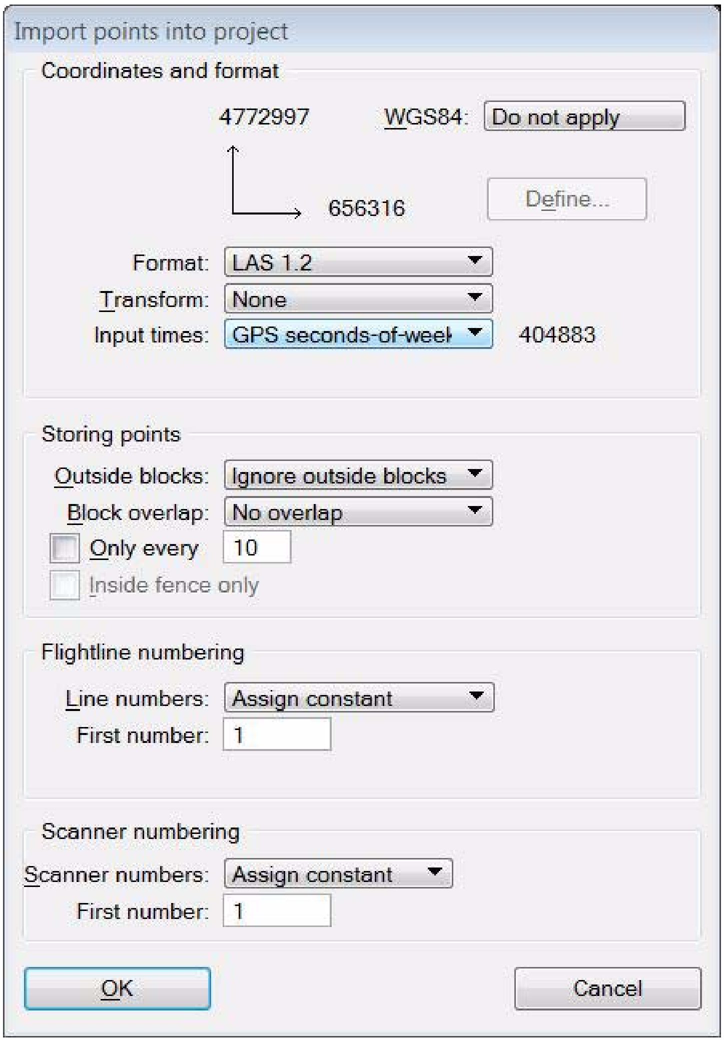
➢ **To import points into a project:**

1. Select **Import points into project** command from the **File** pulldown menu.

   This opens the **Import points into project** dialog, a standard dialog for open files.

2. Select input files and add them to the file list using the **Add** button. If all files are added to the list, click **Done**.

   This opens the **Import points into project** dialog:



3. Define settings and click OK.

   This starts the import of the input files. A progress bar shows the progress of the process.

   After all files are imported, a report is displayed. The report lists the imported files, the amounts of imported and ignored points for each file, and the overall amounts of imported and ignored points. It can be saved as a text file or sent to a printer by using commands from the **File** pulldown menu of the report window.

Setting:	Effect:
Coordinates	The coordinate axes show the coordinate values of the first point in the laser file. This helps to decide if the points are in the correct coordinate system or if an coordinate transformation has to be applied.
WGS84	If necessary, a transformation from WGS84 into a projection system can be applied.
Define	Opens the **Transformation** dialog which lets you define a transformation. See **Coordinate transformations / Transformations** category for more information. This is only active if **Transform** is set to **Define now**.
Format	Format of the input files. This is automatically recognized by the software. For ASCII files, there might be more than one option.
Transform	Transformation applied to the points during the import process. The list contains transformation that are defined in **Coordinate transformations / Transformations** category of TerraScan **Settings**. Select **Define now** in order to define a new transformation.
Input times	GPS time format of the time stamps in the input files.
Survey date	Date when the data was captured. The format is day/month/year (dd/mm/yyyy). This is only active if **Input time** is set to **GPS seconds-of-week** and the time stamp format for the project is set to **GPS standard time**.
Outside blocks	Defines how points outside the block boundaries are handled: • **Ignore outside blocks -** points outside boundaries are ignored. • **Ignore outside selected -** points outside selected block boundaries are ignored. This is only active if blocks are selected in the list of blocks in the **Project** window. • **Create grid blocks -** the software creates new blocks for the points outside boundaries. The size of the new blocks is defined by the **Default** setting in the **Project information** dialog.
Block overlap	Defines how points in overlapping block areas are handled: • **No overlap -** points in the overlapping area are loaded only in the first of the overlapping blocks. The area is empty in all other overlapping blocks. • **Duplicate points -** points in the overlapping area are loaded into all blocks that overlap each other.
Only every	If on, only every **n**th point of the input files is imported where **n** is the given value.
Default	Point class that is assigned to all imported points if no class attribute is stored in the input files. This is only active if **Format** is set to any text file format that does not include the class attribute.

Setting:	Effect:
Line numbers	Defines, how line numbers are assigned to the points during the import process: • **Use from file -** line numbers from source files are used. • **Assign constant -** the number given in the **First number** field is assigned to all points. • **From file name -** the last numerical sequence in a file name is used as line number. • **From folder name -** the last numerical sequence in the name of the folder containing the input files is used as line number. • **Deduce using time -** numbers are assigned based on trajectories loaded into TerraScan. The same process can be performed for by the **Deduce using time** command or the corresponding macro action. • **Increase by xy jump -** the line numbers incease from the given **First number** if the xy distance is bigger than the value given in the **By distance** field. • **Increase by time jump -** the line numbers increase from the given **First number** if a jump in time stamps occurs. This requires that trajectory information is available in TerraScan. • **Increase by file -** the line numbers increase from the given **First number** for each separate file. • **Increase by file name -** the line numbers increase from the given **First number** for each file with another file name. Files with the same name get the same number. • **Increase by directory -** the line numbers increase from the given **First number** for each file stored in another source folder. Files from the same source folder get the same number. The availability of options depends on the number of input files and the attributes stored for points in the input files.
Scanner number	Defines, how scanner numbers are assigned to the points during the loading process: • **Use from file -** scanner numbers from source files are used. • **Assign constant -** the number given in the **First number** field is assigned to all points. • **Increase by file -** the scanner numbers increase from the given **First number** for each separate file. • **From file name -** the first numerical sequence in a file name is used as scanner number. • **From folder name -** the first numerical sequence in the name of the folder containing the point files is used as scanner number. • **From line number -** the line number is used as scanner number. The availability of options depends on the number of input files and the attributes stored for points in the input files.

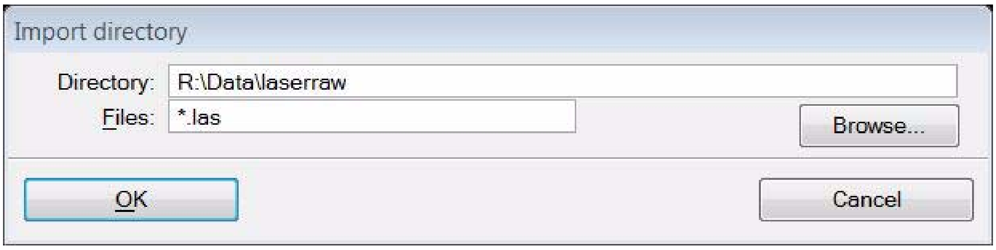
Import directory

Import directory command imports point files into the project. All files of the same format in a directory are imported. The import process itself works in the same way as described for the **Import points into project** command above.

➢ **To import all files in a directory into a project:**

1. Select **Import directory** command from the **File** pulldown menu.

   This opens the **Import directory** dialog:



2. Define settings and click OK.

   This opens the **Import points into project dialog**. Follow the steps of **Import points into project** procedure in order to import the files.

Setting:	Effect:
Directory	Folder from which to import files. Click on the **Browse** button in order to select a folder in the **Browse for Folder** dialog.
Files	Defines the files that are imported. You can use the * character as placeholder for any number of characters in a file name or extension. For example, *.las imports all LAS files from a folder, *.* imports all files from a folder.

Block pulldown menu

Menu commands from **Block** pulldown menu in **Project** window are used to create and edit block boundaries.

To:	Choose menu command:
Add block definitions to a project	**Add by boundaries**
Edit a block definition	**Edit definition**
Delete a block definition	**Delete definition**
Lock selected block files	**Lock selected**
Release the lock of a block file	**Release lock**
Draw block boundaries into the design file	**Draw boundaries**
Create block boundaries along a centerline	**Create along centerline**
Create block boundaries along a tower string	**Create along tower string**
Transform block boundaries	**Transform boundaries**

Add by boundaries

Add by boundaries command adds the block boundaries to a project definition. This is usually the second step of the project creation after defining settings of the project itself.

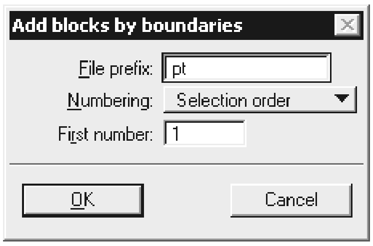
Project boundaries can be created using MicroStation drawing tools for polygons or shapes, or using one of the tools provided by TerraScan. See **Create along centerline** on page 332 or **Create along tower string** on page 333 for more information.

Block file names include a prefix, a block number of six digits or a text string and the extension *.bin for TerraScan binary files or *.las for LAS files.

➢ **To add boundary elements to the project definition:**

1. Select boundary elements and (optional) text inside the boundaries using MicroStation *Selection* tool.
2. Choose **Add by boundaries** command from **Block** pulldown menu.

   This opens the **Add blocks by boundaries** dialog:



3. Enter settings and click OK.

Setting:	Effect:
File prefix	Prefix for the block binary file.
Numbering	Order in which numbering is assigned to blocks: • Selection order - number increases in the same order as boundary shapes are selected. • Selected numbers - the number is defined by a unique numerical text that is placed inside the block boundary. • Selected strings - the file name is defined by a text string that is placed inside the block boundaries. • North to south - numbering increases north to south and secondarily west to east. • South to north - numbering increases south to north and secondarily west to east. • West to East - numbering increases west to east and secondarily south to north. • East to West - numbering increases east to west and secondarily south to north.
First number	Number of the first block to add.

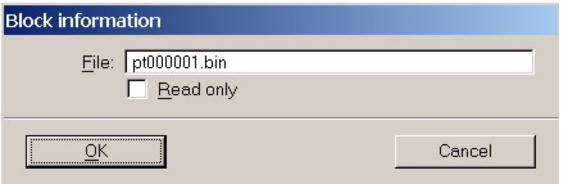
Edit definition

Edit definition command lets you edit a block definition.

➢ **To edit a block definition:**

1. Select **Edit definition** command from **Block** pulldown menu.

   This opens the **Block information** dialog:



2. Enter new settings and click OK.

Setting:	Effect:
File	Block file name definition.
Read only	If on, the block file can be opened only for reading and modifications are not allowed.

✍ If the file name in the block definition is changed after points have been imported to the project, the link between the file name in the project definition and the laser data file is lost unless the laser data file is renamed accordingly.

Delete definition

Delete definition command deletes one or more block definitions from a project.

➢ **To delete a block definition:**

1. Select one or more blocks in the **Project** window file list.
2. Select **Delete definition** command from **Block** pulldown menu.

   This deletes the selected blocks. If more than one block is selected, a message appears that asks you to confirm the deletion.

Lock selected

Lock selected command locks the selected block binary files and thus, disables processing of the files on any other computer than the locking computer. See **Project file locking** for more information about the file locking concept of TerraScan.

The command is only available if **Require file locking** is active in the **Project information** dialog. See **New project** for a description of the dialog settings.

➢ **To lock selected block binary files:**

1. Select one or more block names in the **Project** window.
2. Select **Lock selected** command from the **Block** pulldown menu.

   This locks the selected block. The name of the locking computer and the time when the lock was established are displayed in the **Project** window.

Release lock

Release lock command releases the lock for block binary files. The lock can be released only on the same workstation that locked the file or if the file is already locked for more than 24 hours. See **Project file locking** for more information about the file locking concept of TerraScan.

The command is only available if **Require file locking** is active in the **Project information** dialog. See **New project** for a description of the dialog settings.

➢ **To release a lock for a block binary file:**

1. Select one or more locked blocks in the **Project** window.
2. Select **Release lock** command from the **Block** pulldown menu.

   This releases the locking of the selected block. A message appears that informs about the success of the release.

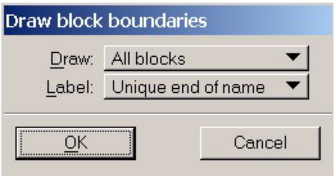
Draw boundaries

Draw boundaries command draws block boundaries of already defined blocks in the design file. This can be used to draw boundaries if points have been imported into the project without defining block boundaries beforehand or after block files have been transformed.

➢ **To draw block boundaries into the design file:'**

1. (Optional) Select block definitions in the **Project** window to be drawn into the design file.
2. Select **Draw boundaries** command from **Block** pulldown menu.

   This opens the **Draw block boundaries** dialog:



3. Select settings and click OK.

   The boundaries and, if selected, block labels are drawn into the design file. The active symbology settings in MicroStation are used to draw the block boundaries and labels.

Setting:	Effect:
Draw	Blocks to be draws: **All blocks** or **Selected** blocks.
Label	Defines the way of labeling the blocks: • None - no labels are drawn. • Block number - the complete block number is drawn. • Full file name - the complete file name without extension is drawn. • Unique end of name - only the last unique number or text string of the block names is drawn. This works only if more than one block is drawn into the design file.

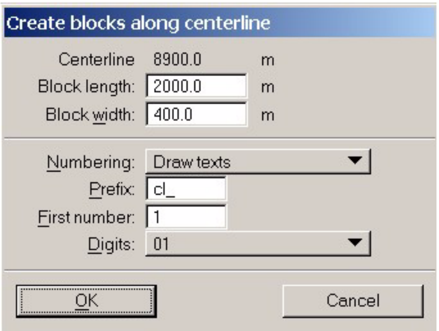
Create along centerline

Create along centerline command creates block boundaries along a centerline element. The block length is measured along the centerline element and defined exactly by the tool's settings. The block boundaries can be used to define blocks for the project with **Add by boundaries** command.

➢ **To create blocks along a centerline element:**

1. Draw a centerline element using MicroStation or TerraScan drawing tools.
2. Select the centerline element.
3. Select **Create along centerline** command from **Block** pulldown menu.

   This opens the **Create blocks along centerline** dialog:



4. Define settings and click OK.

   This draws the block boundaries into the design file. The active symbology settings in MicroStation are used to draw the block boundaries.

Setting:	Effect:
Centerline	The length of the centerline element is displayed.
Block length	Length of a block measured along the centerline element.
Block width	Width of the blocks.
Numbering	Defines labeling of the blocks: **None** or **Draw texts**.
Prefix	Prefix for the text string drawn as label for each block. This is followed by a number.
First number	Number of the first block.
Digits	Defines the amount of digits for block numbering.

Create along tower string

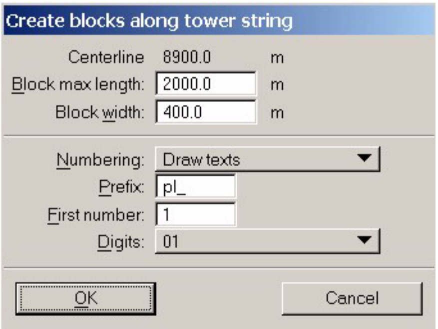
Create along tower string command creates vector elements for block boundaries along a tower string. Any linear element can serve as a tower string for this command. The block length is defined by the distance between vertices of the linear element. On a tower string, each vertex defines the location of a tower which means that a block is usually drawn between two towers. This is most useful for powerline processing which is described in detail in chapter **Coordinate Transformations** on page 312.

The created elements can be used in **Add by boundaries** command to create a block boundaries.

➢ **To create blocks along a tower string:**

1. Draw a tower string using TerraScan *Place Tower String* tool or MicroStation drawing tools.
2. Select the tower string using MicroStation *Selection* tools.
3. Select **Create along tower string** command from **Block** pulldown menu.

   This opens the **Create blocks along tower string** dialog:



4. Define settings and click OK.

   This draws the block boundaries into the design file. The active symbology settings in MicroStation are used to draw the block boundaries and labels.

Setting:	Effect:
Centerline	The length of the tower string is displayed.
Block max length	Maximum length of a block. Routine combines several tower spans into one block if the combined length is shorter than this value.
Block width	Width of the blocks.
Numbering	Defines labeling of the blocks: **None** or **Draw texts**.
Prefix	Prefix for text string drawn as label for each block. This is followed by a number.
First number	Number of the first block.
Digits	Defines the amount of digits for block numbering.

Transform boundaries

Transform boundaries command transforms the block boundaries. This affects the coordinates of the block boundary vertices stored in the project definition file. The command can be used to update a project if the laser data files have been transformed into a new projection system. After transforming also the block boundaries, the location of the laser data and the blocks fit again to each other.

➢ **To transform block boundaries:**

1. Select **Transform boundaries** command from **Block** pulldown menu.

   This opens the **Transform boundaries** dialog:

2. Select a transformation in the **Transform** field and click OK.

   This transforms the block boundaries into the new projections system.

Use **Draw boundaries** command to draw the transformed boundaries into the design file.

✍ Information about the definition of transformations in TerraScan settings can be found in **Coordinate transformations / Transformations** on page 27.

View pulldown menu

Menu commands from **View** pulldown menu in **Project** window are used to set the display of the **Project** window.

To:	Choose menu command:
Display a short list of project block files	**Short list**
Display a medium list of project block files	**Medium list**
Display a long list of project block files	**Long list**
Sort project block files	**Sort**

Short list

Short list command changes the **Project** window to a small size which consists of a title bar, the pulldown menus, **Show location** and **Identify** buttons, and a small size list displaying the project blocks.

Medium list

Medium list command changes the **Project** window to a medium size which consists of a title bar, the pulldown menus, **Show location** and **Identify** buttons, and a medium size list displaying the project blocks.

Long list

Long list command changes the **Project** window to a large size which consists of a title bar, the pulldown menus, **Show location**, **Identify**, **Select all**, **Invert**, and **Deselect all** buttons, and a long list displaying the project blocks.

Sort

Sort command displays the block definitions in the **Project** window sorted by a specific attribute. The sub-menu of the command includes the following options for sorting:

- **By name -** alphabetically ascending by block name.
- **By number -** ascending by last number in the block name.
- **By point count -** ascending by amount of points in the block binary files.
- **North to south -** by geographical location, north to south and secondarily west to east.
- **South to north -** by geographical location, south to north and secondarily west to east.
- **West to east -** by geographical location, west to east and secondarily south to north.
- **East to west -** by geographical location, east to west and secondarily south to north.

Tools pulldown menu

Commands from the **Tools** pulldown menu in the **Project** window are used to perform different actions on block binary files or based on block binary files on project level.

To:	Choose menu command:
Run a macro on a project	**Run macro**
Adjust the elevation of block files to a geoid model	**Adjust to geoid**
Adjust xyz coordinates of block files	**Adjust xyz**
Check the z accuracy of block files	**Output control report**
View statistics about points in block files	**Show statistics**
Check the coverage of block files	**Check coverage**
Validate block boundaries	**Validate blocks**
Copy points into block files from a reference project	**Copy from reference**
Assign color values from images to laser points in block files	**Extract color from images**
Assign echo properties to laser points	**Extract echo properties**
Export laser data from block files into lattice files	**Export lattice models**
Export laser data from block files into raster images	**Export raster images**
Export a 3D point cloud from ortho images	**Export 3D ortho**
Output collections from block files	**Output collections**

Run macro

The main benefit from defining a project is the ability to perform batch processing on blocks. **Run macro** command lets you run a TerraScan macro on project level. This requires that you first define a macro which includes all the processing steps to perform. You can then execute that macro on all or on selected blocks.

A macro can run on project level either by using TerraScan or by using TerraSlave. Using TerraSlave has the advantage that TerraScan and MicroStation are not blocked when a macro is processed.

For detailed information about macros and TerraSlave, see chapter **Macros** on page 429.

Adjust to geoid

Adjust to geoid command adjusts the elevation values of the block files to either a local elevation model defined by a text file, a TerraModeler surface or a selected linear chain.

The theory of geoid adjustment and the use of the command for TerraScan projects are explained in detail in section **Geoid adjustment** on page 313.

➢ **To run elevation adjustment on project blocks:**

1. Select the *Define Project* tool.

   This opens the **Project** window.
2. (Optional) Select the desired rows if you want to adjust only selected blocks.
3. Choose **Adjust to geoid** command from **Tools** pulldown menu.

   This opens the **Adjust blocks to geoid** dialog.
4. Select input model type and click OK.

Adjust xyz

Adjust xyz command applies a varying xyz correction to the block files. The correction model is defined by a text file containing rows with five fields: X Y dX dY dZ.

The command performs the same action on project level as the menu command **Transform loaded points** from TerraScan **Main** window with setting **Transform = Dxyz** on loaded points.

➢ **To adjust block file coordinates to a varying xyz correction model:**

1. (Optional) Select desired block files in the **Project** window's file list to manipulate only selected block files.
2. Select **Adjust xyz** command from **Tools** pulldown menu.

   This opens the **Adjust xyz** dialog:

3. Select whether to adjust **All blocks** or **Selected blocks** in the **Process** field.
4. Define a correction model file in the **File** field.
5. Click OK to apply the adjustment to the block files.

Output control report

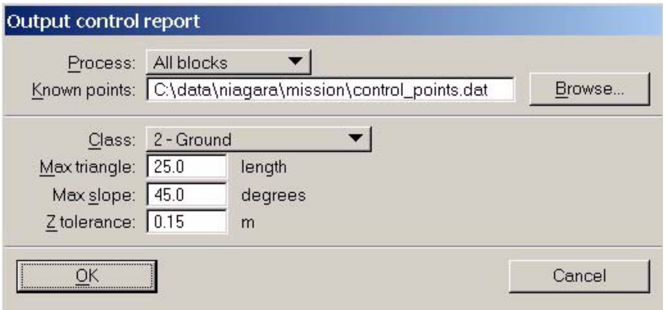
Output control report command creates a report of elevation differences between laser points and control points. This can be used to check the elevation accuracy of a laser data set and to calculate a correction value for improving the elevation accuracy of the laser points.

The control points have to be stored in a space delimited text file in which each row has four fields: identifier, easting, northing and elevation. the identifier field is normally a number but it may include non-numeric characters as well.

➢ **To output a control report:**

1. (Optional) Select the desired block files in the **Project** window's file list if you want to compare against selected blocks only.
2. Select **Output control report** command from **Tools** pulldown menu.

   This opens the **Output control report** dialog:



3. Click on **Browse** and locate the text file which contains control points.
4. Enter other settings and click OK.

   This performs the comparison and opens a report window which shows the elevation differences between known points and laser data. The report is described in detail in Section **Systematic elevation correction** on page 314.

Setting:	Effect:
Process	Blocks to process: **All blocks** or **Selected blocks**.
Known points	Text file containing control points in rows with four space delimited fields: identifier, easting, northing and elevation.
Class	Point class to compare against.
Max triangle	Search radius around each known point.
Max slope	Maximum terrain slope for which an elevation difference will be computed.
Z tolerance	Normal elevation variation of laser points. This value is used only when computing the terrain slope so that small triangles will not exceed **Max slope**.

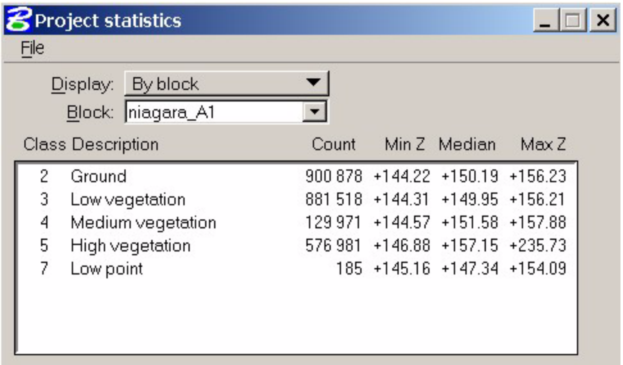
Show statistics

Show statistics command calculates statistics for the project blocks. The output dialog includes information about classes, point count as well as minimum, maximum and median elevation values. Besides for the whole project, these values are also calculated for each block and flightline. The statistics can be saved into a text file.

➢ **To calculate statistics for a project:**

1. Select **Show statistics** command from **Tools** pulldown menu.

   The calculation process starts and an information window is displayed that shows the progress of the process. After finishing the calculation, the **Project statistics** dialog opens:



2. Check the statistics for the project, different blocks or flightlines by selecting the corresponding setting in the **Display** field.

   The statistic values for all classes occurring in the data are listed in the lower part of the **Project statistics** dialog.

3. (Optional) Save the statistics into a text file using **Save as** command from **File** menu in the **Project statistics** dialog.

   The text file stores the point count for each class and the elevation values if **Display** is set to **Project total**, and the point count for each class per block or flightline, if **Display** is set to **By block** or **By flightline**.

Setting:	**Effect:**
Display	Content of statistics display: • Project total - values for the whole project are displayed. • By block - values are shown for the selected block. • By flightline - values are shown for the selected flightline.
Block	Name of the block, for which statistic is shown. This is only active if **Display** is set to **By block**.
Flightline	Number of the flightline, for which statistic is shown. This is only active if **Display** is set to **By flightline**.

Check coverage

Check coverage command finds holes and areas of low point density in the laser data of a project. The area to be covered can be either defined by one or more selected polygons or by all block boundaries of the project. The application calculates the point density within sample areas and decides to which coverage level an area belongs.

The command can create a few different output products to illustrate coverage. The options include the creation of a density raster image showing the point coverage in a TIFF file, the creation of points that fill the holes and areas of low point density, as well as the output of the density calculation results in a report that can be saved as a text file.

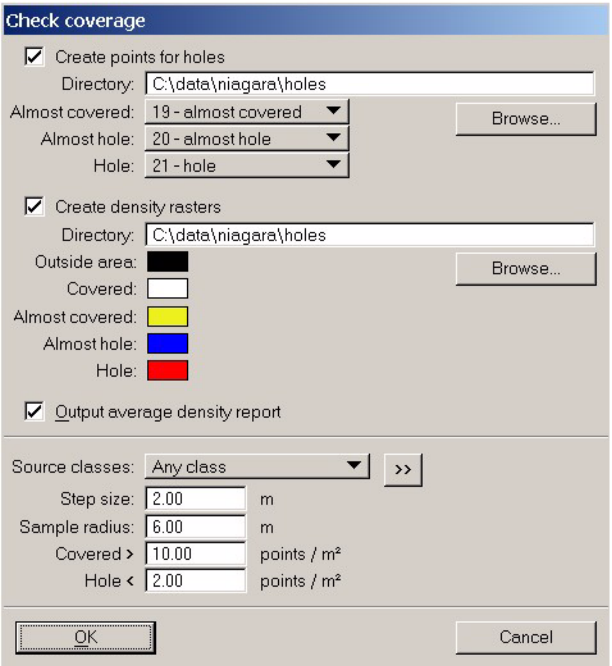
The point coverage is defined in four levels: **Covered**, **Almost covered**, **Almost hole** and **Hole**. User settings define the point densities which are interpreted as being holes or as being fully covered. If points are created to fill the holes and low density areas, TerraScan binary files are created that store the artificial laser points in separate classes according to the coverage level. The points are created in a uniform grid with a distance between points that is defined in the command's settings as well. The elevation of the points is set to be equal to the number of points found in the sampling area.

Additionally to the *.bin files, a TerraScan project file is created that includes all binary files with the artificial points. The name of the project file is 'density.prj'. The binary and TIFF files are named with the original block name.

➢ **To check the laser point coverage in a project area:**

1. (Optional) Draw and select one or more polygons around areas for which to check the coverage.
2. Select **Check coverage** command from **Tools** pulldown menu.
3. If no polygon is selected, the application informs you that it will use all block boundaries for the check. Click OK.

   This opens the **Check coverage** dialog:



4. Define settings and click OK.

   This starts the calculation of the point densities and the coverage. The software creates the files according to the settings in the given directories. An information window shows the progress of the process.

Setting:	Effect:
Create points for holes	If on, holes are filled with artificial laser points that are stored in TerraScan binary files for each block.
Directory	Directory for storing the created *.bin files and the *.prj file.
Almost covered	Class for artificial points in almost covered areas.
Almost hole	Class for artificial points in almost hole areas.
Hole	Class for artificial points in holes.
Create density raster	If on, a density raster image in TIFF format is created for each block.
Directory	Directory for storing the created *.tif files.
Outside area	Color in density raster image for areas outside the covered area.
Covered	Color in density raster image for covered areas.
Almost covered	Color in density raster image for almost covered areas.
Almost hole	Color in density raster image for almost hole areas.
Hole	Color in density raster image for holes.
Output average density report	If on, a report about the point densities per block is shown after the calculation process.
Source classes	Laser point class(es) that are included in the coverage calculation.
Step size	Distance between the artificial points that are created to fill holes and areas of low density.
Sample radius	Radius of sampling area from which to calculate local point density.
Covered >	Point density required for an area being considered as covered.
Hole <	Point density for an area that will be considered as hole.

✍ Since the file names of the created files are fixed for this command, existing files are overwritten without warning if the command is performed a second time on a project with the same directory settings.

Validate blocks

Validate blocks command checks block definitions of a project regarding duplicated block names, small area blocks, and overlap between block boundaries. This helps to analyze automatically created block boundaries and names before the block binary files are created.

➢ **To validate blocks:**

1. Select **Validate blocks** command from the **Tools** pulldown menu.

   This opens the **Block validity check** window that displays the results of the validation in a report.

   The report can be saved into a text file or sent to a printer using **Save as text** or **Print** commands from the **File** pulldown menu. The size of the report window can be changed with commands from the **View** pulldown menu.

Copy from reference

Copy from reference command copies attributes for laser points from another project that stores for example the same points at an earlier processing status. An example case for using this command could be:

- Laser data has been imported into a project \laser1.
- Heading, Roll and Pitch misalignment has been solved and applied to another project \laser2.
- Automatic and manual classification steps have been performed.
- After the classification it has been realized that HRP correction was wrong but the classification is good and has taken a lot effort.
- As a solution, the \laser1 project is defined as reference project in the information settings for \laser 2 project.
- The attributes xy and z are copied from \laser1 into \laser2 using the **Copy from reference** command.

This restores the coordinate values from the status before the HRP correction but preserves the classification that was done after the HRP correction.

The command requires the definition of a reference project in the project information settings of the current project. See **Edit project information** on page 322 for information about defining a reference project.

To ensure that the correct attributes are attached to the laser points, there has to be a possibility to define a laser point clearly. Therefore, there have to be attributes which are equal in reference and current project and unique for each laser point.

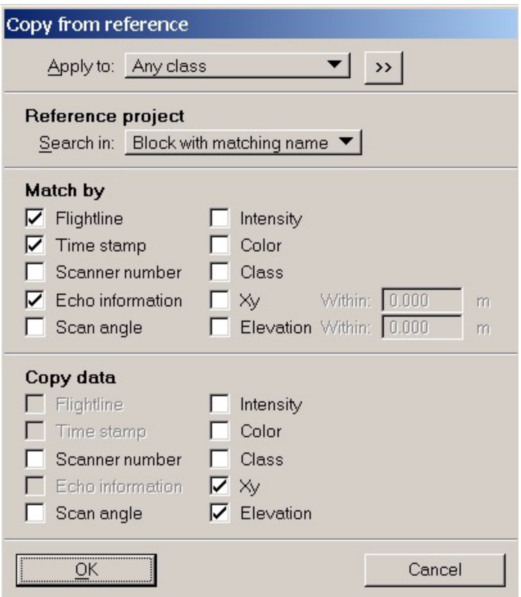
For **LAS** files, the combination of time stamp + echo information is unique for each laser point if all the data are from one GPS week or if GPS standard time stamps are used. If the data are from more than one GPS week, the combination flightline + time stamp + echo information defines a laser point clearly.

For **TerraScan Binary** files, the combination time stamp + echo information is not unique for each laser point because of the resolution with which time stamps are stored in this format. Therefore, additional attributes that do not have changed between reference and current project have to be selected to identify laser points clearly.

➢ **To copy attributes from a reference project:**

1. Select **Copy from reference** command from **Tools** pulldown menu.

   This opens the **Copy from reference** dialog:



2. Select one or more classes for which to apply the copied attributes.

3. Select a method how corresponding blocks are searched in the reference project.
4. Define settings for matching laser points between the reference and the current project.
5. Define laser point attributes that are copied from the reference project to the current project.
6. Click OK.

   This copies the selected attributes to the laser points of the current project. An information window shows the progress of the process. After finished processing, a report is displayed that lists all blocks and the number of points that have been changed. The report can be saved as a text file or printed out directly using commands from the **File** pulldown menu in the report window.

Setting:	**Effect:**
Apply to	One or more laser point classes for which the attributes are copied. Click the >> button to select multiple source classes.
Search in	Method how corresponding blocks are searched in the reference project: • Block with matching name - the block names in reference and current project have to be the same. • Blocks close in xy - the blocks are at the approximately same location.
Match by	Laser point attributes that are unchanged and unique for each laser point and thus define a point clearly in reference and current project.
Copy data	Attributes that are copied from the laser points in the reference project to the corresponding laser points in the current project.

Extract color from images

Extract color from images command extracts RGB color values for laser points from raster images. The color source can be orthophotos attached as references or raw images from an image list. In addition, color points from a color point file can be used to balance colors of the raw images before the colors are assigned to the laser points. All options require TerraPhoto or TerraPhoto Lite. The color value is derived by sampling all the pixels inside a circular footprint area of each laser point.

The command performs the same action on project blocks as the **Extract color from images** command for loaded points.

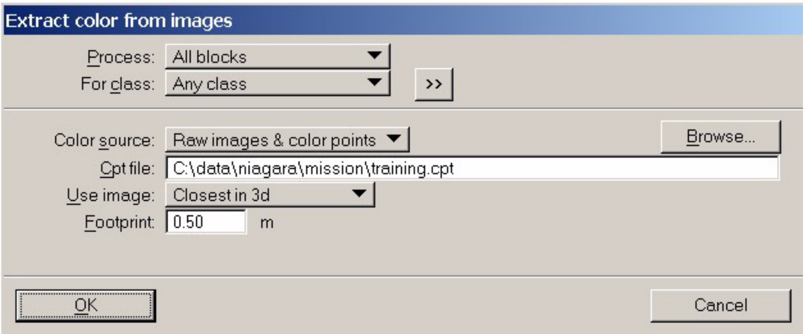
➢ **To extract color from attached or raw images:**

1. Attach reference images with TerraPhoto's *Manage Raster References* tool.

   OR

1. Create a camera calibration file, a mission definition and an image list in TerraPhoto.
2. (Optional) Select the desired block files in the **Project** window's file list if you want to attach color values for selected blocks only.
3. Select **Extract color from images** command from **Tools** pulldown menu.

   This opens the **Extract color from images** dialog:



4. Select settings and click OK.

   This derives color values for the laser points in block files from the defined source images.

Setting:	Effect:
Process	Blocks for processing: **All blocks** or **Selected blocks**.
For class	Laser point class(es) for which colors are extracted.
Color source	Source files for color extraction: • Ortho images - colors are extracted from attached raster images. • Raw images - colors are extracted from raw images defined in an image list in TerraPhoto. • Raw images & color points - colors are extracted from raw images and from a color point file.
Cpt file	Location and name of a color point file. This is only active if **Color source** is set to **Raw images & color points**.

Setting:	Effect:
Use image	Method how the software defines the closest raw image to a laser point: • Closest in 3D - raw image that has the closest camera xyz position to the laser point. • Closest in xy - raw image that has the closest camera xy position to the laser point. • Closest in time - raw image that has the closest time stamp to the laser point. • Closest - Mobile logic - raw image that has the closest time stamp to the laser point. This is only for laser points and images from mobile systems. This is only active if **Color source** is set to **Raw images** or **Raw images & color points**.
Footprint	Radius of a circular area around each laser point within which pixel color values are resampled.

Extract echo properties

Extract echo properties command extracts information from waveform data and assigns it as attributes to the laser points. The command requires that waveform data and a scanner waveform profile are available. The processing steps for preparing the extraction of waveform-related information are described in detail in Chapter **Waveform Processing** on page 155.

The command can extract the following attributes:

- **Echo length -** relative length (millimeter) of a return signal compared to a typical return from a hard surface.
- **Echo normality -** difference in shape and position of a peak of a return signal compared to a typical return from a hard surface.
- **Echo position**

The echo length can be used for the visualization of points and for classifying points. For instance, a classification **By echo length** prior to ground classification can improve the result of the **Ground** routine especially in areas of low vegetation.

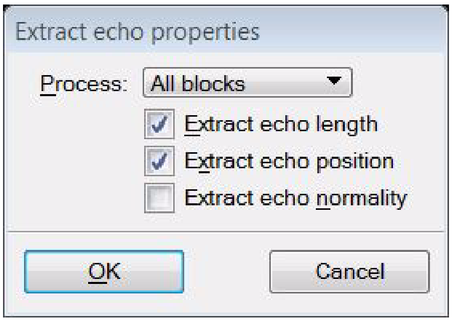
The echo properties can be stored in TerraScan **Fast binary** files. The command can be used to extract echo properties for all blocks or a selection of blocks defined in a project.

The command performs the same action on block binary files as the **Extract echo properties** command on loaded points.

➢ **To extract echo properties:**

1. Select **Extract echo properties** command from the **Tools** pulldown menu.

   This opens the **Extract echo properties** dialog:



2. Select which blocks to process: **All blocks** or **Selected blocks**.
3. Select what properties you want to extract by switching the corresponding options on.
4. Click OK.

   This starts the extraction process. It assigns the extracted attributes to all laser points of the processed block binary files for which waveform information is available. Depending on the amount of points, the process may take some time.

Export lattice models

Export lattice models command creates grid files with uniform distances between points from one or more selected laser point classes. The files store either elevation values or point count/density values for each grid cell. There are several formats supported to store the lattice as raster, grid, or text file.

The menu command can be used to export all or selected project block files into separate lattice models. The block boundaries for blocks to be exported have to be selected in the design file. Text strings placed inside the block boundaries can be used as file names for the exported lattice files.

The command performs the same action on project block files as the **Export lattice model** command on loaded points.

✍ If **Conserve memory** setting for export lattice models on project level is on, the software first reads through input files to determine how many points would be loaded to be able to make a memory allocation for the exact number of points. This is slower but less likely to run out of memory.

Export raster images

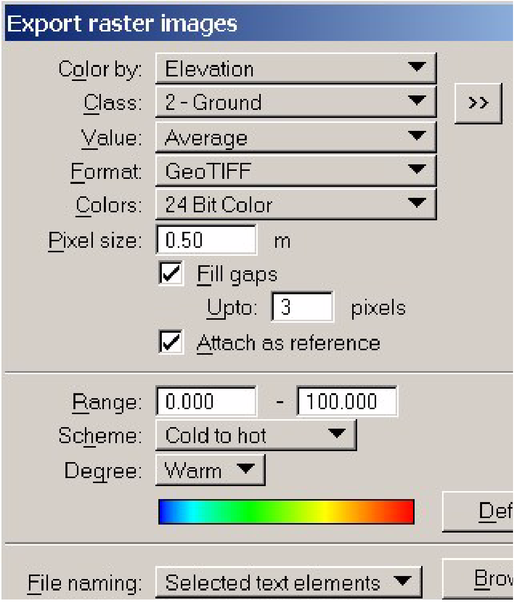
Export raster image command generates a raster image colored by elevation, intensity, point color or point class of the laser points in the block files. The block boundaries for blocks to be exported have to be selected in the design file. Text strings placed inside the block boundaries can be used as file names for the exported raster files.

The command performs approximately the same action on project block files as the **Export raster image** command in the TerraScan **Main** window on loaded points.

➢ **To create a colored raster image of project block files:**

1. Select block boundaries and (optional) text strings inside the boundaries.
2. Choose **Export raster image** command from **Tools** pulldown menu.

   This opens the **Export raster image** dialog:



3. Define settings and click OK.

   This calculates the images and creates the raster files.

   If **File naming** is set to **Enter name for each**, a standard dialog box for choosing an output file name opens for each exported file.
4. Enter a name for the output file and click **Save**.

   This creates a raster image with the given name.

Setting:	Effect:
Color by	Coloring attribute: • **Elevation** - laser point elevation. • **Elevation difference** - elevation difference between laser points of two different classes. • **Intensity hits** - intensity of laser points with center point inside the pixel. • **Intensity footprint** - intensity of laser points with footprint overlapping the pixel. • **Point color** - color values stored for laser points. • **Point class** - laser point class.
Class	Point class(es) to use for image creation. Several classes can be selected by clicking on the >> button.

Setting:	Effect:
Value	Value determination within each pixel: • Lowest - Smallest value of the points. • Average - Average value of the points. • Highest - Highest value of the points. This is only active for elevation and intensity coloring.
Format	File format for the raster file: **Windows BMP**, **GeoTIFF**, or **GeoTIFF + TFW**.
Colors	Color depth of raster image: • 24 Bit Color - true color image. • 256 Colors - 256 colors. • Grey scale - 8 bit grey scale. This is only active for elevation and intensity coloring.
Pixel size	Size of each pixel in the target raster file.
Fill gaps	If on, small gaps are filled in places where there are no laser hits inside a pixel.
Attach as reference	If on, the image is attached as raster reference to the design file.
Range	Defines the value range that is covered by the color scheme for elevation and intensity coloring. Should be set to the general elevation or intensity range covered in the laser data to ensure that all values are represented by the complete color scheme.
Scheme	Type of coloring scheme for elevation or intensity coloring: • Cold to hot - varies from blue for low elevation to red for high elevation. This is the common coloring scheme for elevation coloring. • Hot to cold - varies from red for low elevation to blue for high elevation. • Selected colors - a user defined coloring scheme can be created by clicking the **Define** button. See more information in section **Color scheme definition** on page 251. • Black to white - varies from black for low values to white for high values. This is active if **Colors** is set to **Grey scale**. This is the common coloring scheme for intensity coloring. • White to black - varies from white for low values to black for high values. This is active if **Colors** is set to **Grey scale**.
Degree	Determines how the color changes in color schemes are computed.
File naming	File naming setting for exporting selected rectangles: • Enter name for each - a name for each rectangle has to be defined when saving the lattice models. • Selected text elements - selected texts inside the rectangles are used as file names.
Directory	If **File naming** is set to **Selected text elements**, this sets a directory for storing the raster image files.

Export 3D ortho

Export 3D ortho command generates a point cloud from orthophotos. The process combines orthophoto pixel location and color, and laser point coordinates in order to create a high-density colored point cloud. The resulting point cloud contains one point for each orthophoto pixel with the following attributes:

- **XY coordinates -** computed from the center of orthophoto pixels or used from the original laser points.
- **Z coordinate -** computed from a TIN generated from laser data or used from the original laser points.
- **Class -** defined by the source class in the laser data.
- **RGB color values -** determined by the pixel color, the laser point color, or a fixed color.

The command requires TerraScan and TerraPhoto running on the same computer. The orthophotos must be attached as TerraPhoto raster references in order to create the 3D point cloud. In addition, the laser data in the block binary files should be classified in order to distinguish point elevations on the ground, vegetation, building roofs, etc. This allows you to define different rules for the point cloud generation depending on the object types.

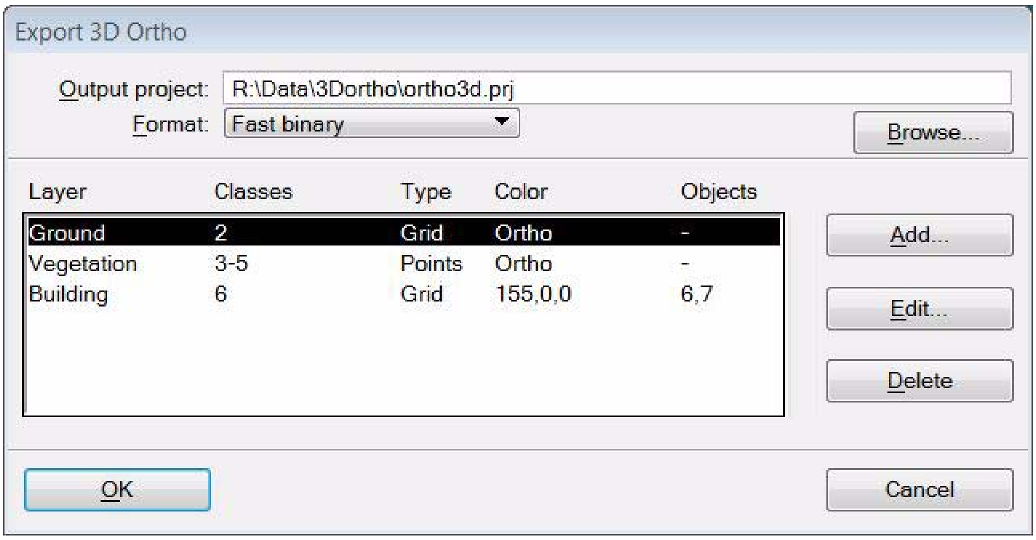
The process can also include vector elements in the point cloud computation, such as 3D building models or other 3D shapes.

The process creates a new TerraScan project file and block binary files for each orthophoto inside the area covered by blocks of the original laser project. The names of the blocks and binary files are determined by the names of the orthophotos. The point density and the amount of points per block binary file of the generated point cloud is determined by the pixel resolution of the orthophotos but also by the rules for the point cloud creation.

➢ **To export a 3D ortho point cloud:**

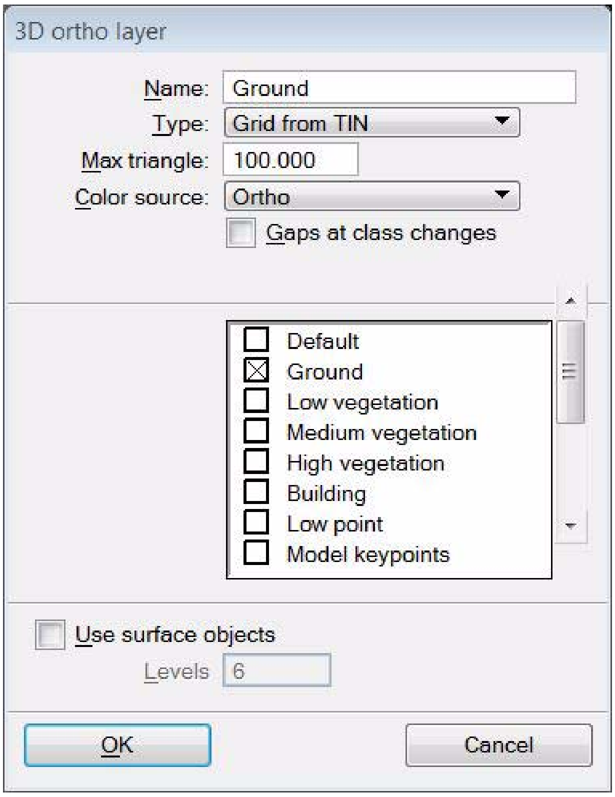
1. Attach orthophotos by using the *Manage Raster References* tool of TerraPhoto.
2. Open the TerraScan project that references the classified laser binary files.
3. Select **Export 3D ortho** command from the **Tools** pulldown menu.

   This opens the **Export 3D Ortho** dialog:



4. Define a storage directory and name for the **Output project**. The block binary files are stored in the same folder as the project file. Click on the **Browse** button in order to define the output project in a standard dialog.
5. Select a file format for the block binary files: **Fast binary** or **LAS 1.2**.
6. Click on the **Add** button in order to define rules for the export of different layers, such as ground, vegetation, or buildings.

This opens the **3D ortho layer** dialog:



7. Define settings and click OK.

   You can **Edit** and **Delete** layers by using the corresponding buttons in the **Export 3D ortho** dialog.

8. Click OK in the **Export 3D Ortho** dialog.

   This starts the point cloud generation process. A progress window displays the progress of the process.

Setting:	Effect:
Name	Descriptive name of the layer.
Type	Defines how points are extracted for the export: • **Grid from TIN -** a grid is extracted from a TIN created from the laser points. The XY location of a grid point is determined by the pixel center, the elevation by the TIN. • **Points directly -** the original laser points are exported.
Max trianlge	Maximum length of a triangle edge in the TIN. Determines how big gaps are filled with points generated by aerial triangulation.This is only active if **Type** is set to **Grid from TIN**.
Color source	Defines the source for extracting RGB values for the points: • **Ortho -** each point gets the color of the closest pixel in the orthophotos. • **Point color -** each point gets the color assigned to laser points. • **Fixed color -** each point gets a fixed color value.
Gaps at class changes	If on, a gap is enforced at boundaries between different classes in the source laser data. This avoids that point are generated by aerial triangulation between different point classes.
R G B	RGB color values of a fixed color assigned to points. This is only active if **Color source** is set to **Fixed color**.

Setting:	Effect:
List of classes	Select the source class(es) for this layer. The list contains the active classes in TerraScan.
User surface objects	If on, shape elements on the given **Levels** are used for determining the elevation of exported points inside a shape area.

✍ The TerraScan project file for the exported point cloud is created with the **Color** attribute inactive. Therefore, you have to switch the attribute on and save the project before you can see the points displayed by color correctly. See **Edit project information** for instructions how to edit a project.

Output collections

TerraScan has the capability for creating logical groups of laser points which are hits on the same object such as a building. You can perform the grouping by placing collections shapes around the objects using the *Place Collection Shape* tool.

The collection shapes can be used to produce output files where logical groups of laser points are grouped together. This output action starts from the **Project** window and automatically gathers all the necessary laser points from the project blocks.

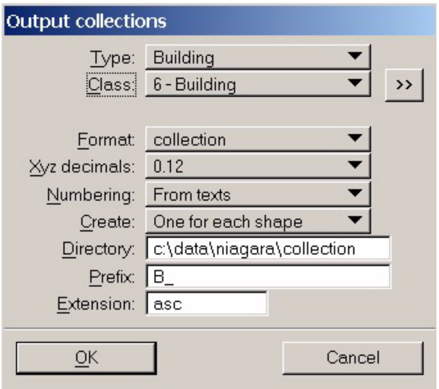
The output files can be created in two logical ways:

- Each collection is written to its own output file. The output file format can be any output format supported by TerraScan. The output file includes a collection number which comes from a text element inside the collection shape or is generated automatically by the application of an increasing number.
- All collections are written to the same output file. The output file format should be a user defined file format which includes **Collection** field so that points belonging to different collections can be distinguished from each other. See **File formats / User point formats** on page 38 for more information about how to define user file formats in TerraScan **Settings**.

➢ **To output collection shapes:**

1. Choose **Output collections** command from **Tools** pulldown menu.

   This opens the **Output collections** dialog:



2. Select type of objects to output in the **Type** field.
3. Define other settings and click OK.

   The application processes each collection shape of the given type found in the active design file. It searches all overlapping project blocks for laser points and writes those into output files.

Setting:	**Effect:**
Type	Collection shape type to output.
Class	Point class(es) to output. Click the >> button to select multiple source classes.
Format	Output file format.
Xyz decimals	Number of decimals with which the coordinate values are stored in the output files. This is only active if a text file format is chosen as output format.
Numbering	Source of collection numbers: • Automatic - application assigns increasing numbers to the file names. • From texts - uses text elements inside collection shapes.
Create	How to generate files: **Single file** or **One for each shape**.

Setting:	Effect:
Directory	Directory for storing output files.
Prefix	File name prefix before the collection number is added.
Extension	File name extension.

13 Manage Trajectories

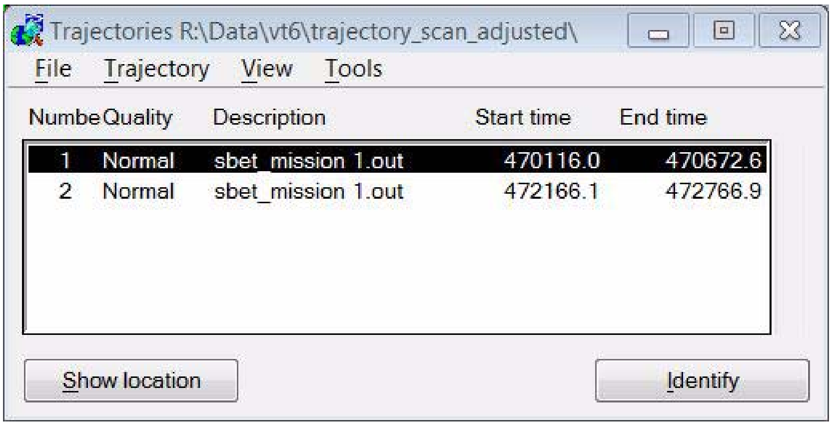
Trajectories are required for many processing steps in TerraScan and TerraMatch. They provide positional and, usually, attitude information of the scanner system for each point of time during the data collection.

Normally, the raw trajectory is produced by so-called post-processing software that combines the input of GPS and IMU sensors. The raw trajectory may be provided in a binary or ascii file format. TerraScan is able to import common binary formats of post-processing software as well as a number of ASCII formats. Additional text file input formats for trajectories can be defined in **File formats / User trajectory formats** category of TerraScan **Settings**. All imported trajectories are converted into the TerraScan trajectory binary format (*.TRJ). See **Trajectory file formats** for more information.

All commands related to trajectories is combined in the TerraScan **Trajectories** window which is opened by the *Manage Trajectories* tool.

TerraScan Trajectories window

The **Trajectories** window contains pulldown menu commands for importing, modifying, and managing trajectory information.



The list in the window shows all TerraScan trajectory files that are stored in the active trajectory folder. The active directory is shown in the title bar of the window.

To select a trajectory, click on the line in the list. Press the <**Ctrl-key**> to select several trajectories.

To show the location of a trajectory, select a line in the list. Click on the **Show location** button and move the mouse pointer into a view. This displays the selected trajectory. With a data click inside the view you can center the selected trajectory in the view.

To identify a trajectory, click on the **Identify** button and place a data click close to a trajectory in a view. This selects the corresponding line in the **Trajectories** window.

File pulldown menu

Commands from the **File** pulldown menu in the **Trajectories** window are used to import trajectory information into TerraScan and to export trajectory information into text files.

To:	Choose menu command:
Set the active trajectory folder	**Set directory**
Import trajectory files	**Import files**
Import trajectory files from a folder and its subfolders	**Import directory**
Import separate text files from GPS and INS sensors	**Merge from GPS and INS**
Import accuracy files for trajectories	**Import accuracy files**
Export trajectory information into text files	**Output positions**

Set directory

Set directory command is used to define the active trajectory directory. The software writes trajectory files into this folder during the import process. It loads TerraScan trajectory files from a folder if it is set as active directory and files do already exist. Usually, this is the first command you use when you start working with trajectories.

It is good practice to reserve a folder in your project directory structure for storing trajectories imported into TerraScan. In some cases, it might be advisable to save a new copy of TerraScan trajectories. Then, you would have multiple trajectory directories in a project and change the active directory whenever needed in order to access the correct set of trajectory files.

➢ **To set the active trajectory directory:**

1. Select **Set directory** command from the **File** pulldown menu.

   This opens the standard dialog for selecting a folder.

2. Select a folder and click OK.

   This sets the active directory to the given folder. TerraScan scans the directory. If there are TerraScan trajectory files in the folder, it reads the header information from each file into memory and displays them in the list.

Import files

Import files command is used to import raw trajectories into TerraScan. During the import, trajectory information is converted into **TerraScan trajectory binary files** (*.TRJ).

The input files must contain at least time-stamped position and, for most processing tasks, attitude information. The input files can be:

- text files in one of the implemented ASCII formats, see **Supported file formats**
- binary files from Applanix or Riegl software, see **Supported file formats**
- text files in a user-defined file format, see **File formats / User trajectory formats**

During the import, the software assigns some attributes to the trajectories and can apply coordinate transformations and/or a time stamp format conversion. Most of the settings defined in the import process can be changed later for the converted trajectory files by using the **Edit information** command or commands from the **Tools pulldown menu**.

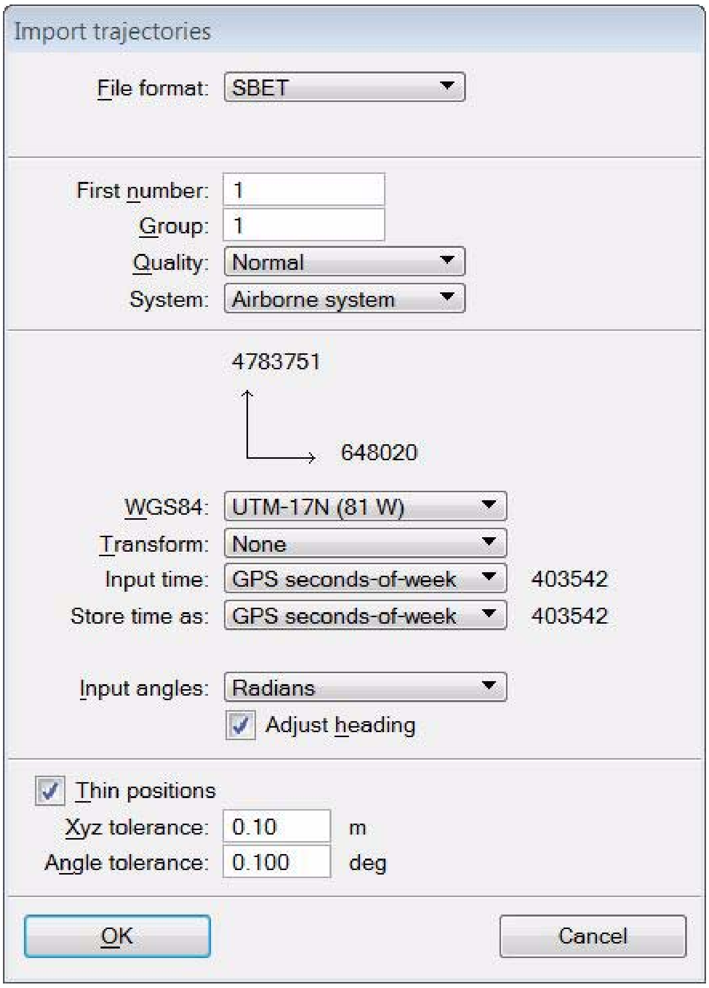
➢ **To import a raw trajectory:**

1. Select **Import trajectories** command from the **File** pulldown menu.

   This opens the **Import trajectories** dialog, a standard dialog for selecting files.

2. Select the raw trajectory file(s) and click **Done**.

   The **Import trajectories** dialog opens:



3. Define settings and click OK.

   This imports the trajectory file(s) and stores them as TerraScan trajectory binary file(s) into the active trajectory directory. The name of a file is determined by the seconds values of the first and last position in a trajectory file separated by an underline character.

Setting:	Effect:
File format	File format of the input file(s). This is usually recognized automatically for implemented input formats.
Attitude format	Format of the INS file. This is only active if **Merge from GPS and INS** command is used to import trajectory information.
First number	Number assigned to the first trajectory file. If more than one file is imported, the files are numbered incrementally.
Group	Group number assigned to trajectory file(s). Groups may indicate, for example, different flight sessions and can be used by TerraMatch processes.
Quality	Quality attribute assigned to trajectory file(s). Quality may indicate, for example, the accuracy of trajectories and can be used for TerraMatch and TerraScan processes.
System	Scanner system used for data collection. This may add lever arm corrections to trajectory positions and thus, effect the computation of the scanner location at the moment of measuring a laser point.
WGS84	Transformation from WGS84 coordinates to another projection system applied during the import. The list contains projection systems that are active in **Coordinate transformations / Built-in projection systems**, **Coordinate transformations / US State Planes**, and **Coordinate transformations / User projection systems** categories of TerraScan **Settings**.
Transform	Transformation applied during the import. The list contains transformations defined in **Coordinate transformations / Transformations** category of TerraScan **Settings**.
Input time	Format of time stamps in the raw trajectory file(s): **GPS seconds-of-week**, **GPS standard time**, or **Unix time**.
Store time as	Format of time stamps in the converted files: **GPS seconds-of-week**, or **GPS standard time**. If the format is different from the **Input time** format, time stamps are converted.
Survey data	Date when the trajectory data was captured. The format is day/month/year (dd/mm/yyyy). This is required for the conversion of time stamps from **GPS seconds-of-week** to **GPS standard time** and is only active if **Input time** and **Store time as** are set accordingly.
Input angles	Format of angle values in the raw trajectory file(s): **Degrees**, **Radians**, or **TopEye radians**. This is usually set automatically for implemented input formats.
Adjust heading	If on, the software applies a meridian convergence correction to heading values. The correction is based on the projection system set for **WGS84** or the coordinate transformation set for **Transform**.
Thin positions	If on, intermediate positions are skipped as long as the trajectory accuracy stays within the given tolerances.
Xyz tolerance	Maximum allowed locational change of the trajectory caused by thinning. This is only active if **Thin positions** is switched on.
Angle tolerance	Maximum allowed angular change of the trajectory caused by thinning. This is only active if **Thin positions** is switched on.

✍ TerraScan and TerraMatch do not need highly accurate trajectory information. It is beneficial to remove unnecessary positions with **Thin positions** setting when importing a raw trajectory. This reduces the amount of memory consumed by trajectory information and speeds up processes.

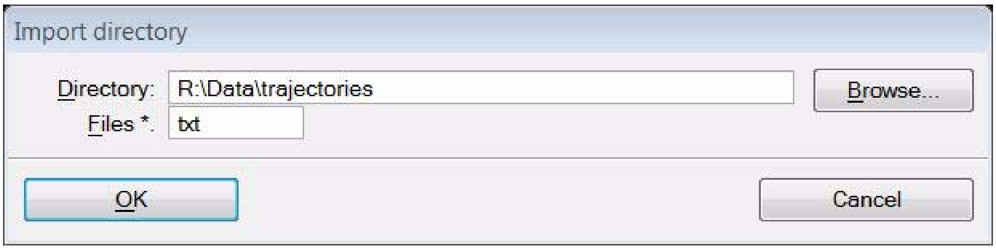
Import directory

Import directory command imports trajectory files into TerraScan. All files of the same format in a directory are imported. The import process itself works in the same way as described for the **Import files** command above.

➢ **To import all trajectory files in a directory:**

1. Select **Import directory** command from the **File** pulldown menu.

   This opens the **Import directory** dialog:



2. Define settings and click OK.

   This opens the **Import trajectories** dialog. Follow the steps of **Import files** procedure in order to import the files.

Setting:	Effect:
Directory	Folder from which to import files. Click on the **Browse** button in order to select a folder in the **Browse for Folder** dialog.
Files	Defines the extension of files that are imported. You can use the * character as placeholder for any file extension or type a specific extension.

Merge from GPS and INS

Merge from GPS and INS command creates a trajectory binary file for TerraScan from separate GPS and INS files. The GPS file contains time stamps and coordinates for the trajectory positions, while the INS file includes time stamps and attitude angle values for the same trajectory positions. The software combines the two input files using the time stamps.

The GPS and INS files are usually text files. The format of the files can be defined in **File formats / User trajectory formats** of TerraScan **Settings**.

➢ **To create a trajectory from GPS and INS files:**

1. Select **Merge from GPS and INS** command from the **File** pulldown menu.

   This opens the **GPS positions files** dialog, a standard dialog for opening files.

2. Open the file that contains the positional information.

   This opens the **INS attitude files** dialog, a standard dialog for opening files.

3. Open the file that contains the attitude information.

   The **Import trajectories** dialog opens. See **Import files** for a description of the settings in the dialog.

4. Define settings and click OK.

   The software combines the two input files and creates the binary trajectory file in the active trajectory directory.

Import accuracy files

Import accuracy files command imports an output file from post-processing software that contains accuracy estimates for each trajectory position. The file includes the RMS values for xyz positions as well as for heading, roll, and pitch angles. It is connected to the trajectory file by the time stamps.

TerraScan can import the accuracy files from Applanix and IPAS SOL software. The RMS values are stored in the binary trajectory files. TerraScan stores only four RMS values for each trajectory position: x/y, z, heading, roll/pitch.

The information from the accuracy files is used for strip matching computations in TerraMatch and for drawing trajectories into the design file.

➢ **To import an accuracy file:**

1. Import the trajectory file(s) as described in **Import files** or **Import directory**.
2. Select **Import accuracy files** command from the **File** pulldown menu.

   This opens the **Import accuracy files** dialog, a standard dialog for opening files.
3. Open the accuracy file delivered by the post-processing software.

   This reads the file and connects the RMS values to the trajectory positions. The values are saved automatically to the binary trajectory files in the active trajectory directory. A dialog informs about the number of positions for which RMS values are available.

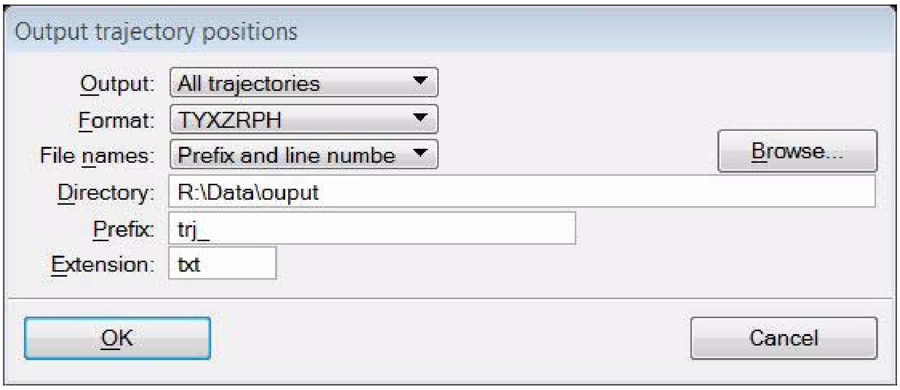
Output positions

Output positions command saves trajectory positions into text files. It creates a separate text file for each trajectory file. The format of the output files can be defined in **File formats / User trajectory formats** of TerraScan **Settings**. There are also two implemented output formats. The software writes one line for each trajectory position into an output file.

➢ **To create text files for trajectory positions:**

1. (Optional) Select the trajectories in the list of the **Trajectories** window for which you want to save positions into text files.
2. Select **Output positions** command from the **File** pulldown menu.

   This opens the **Output trajectory positions** dialog:



3. Define settings and click OK.

   This writes the trajectory positions into text files.

Setting:	**Effect:**
Output	Trajectories for which text files are created: **All trajectories** or **Selected only**.
Format	Text file format, defines which attributes are stored in the columns of the text file. The list contains two implemented formats: • **TYXZRPH -** time northing easting elevation roll pitch heading • **TXYZ -** time easting northing elevation and any formats defined for output in **File formats / User trajectory formats**.
File names	Method of naming the output files: • **Prefix and line number -** name contains a prefix and the trajectory number. • **Same as trj file -** name of the trajectory binary file is used.
Directory	Folder into which the output files are written. Click on the **Browse** button in order to select a folder in the **Browse for Folder** dialog.
Prefix	Text string added in the beginning of an output file name. This is only active if **File names** is set to **Prefix and line number**.
Extension	File name extension.

Trajectory pulldown menu

Commands from the **Trajectory** pulldown menu are used to modify information of a trajectory, to assign a new trajectory number, to delete trajectory files, and to view the positions of a trajectory.

To:	Use:
Modify trajectory information	**Edit information**
Assign a number by identifying a laser point	**Assign number**
Delete selected trajectories	**Delete**
View positions of a selected trajectory	**View positions**

The commands of the pulldown menu are only available if at least one trajectory is selected in the **Trajectories** window.

Edit information

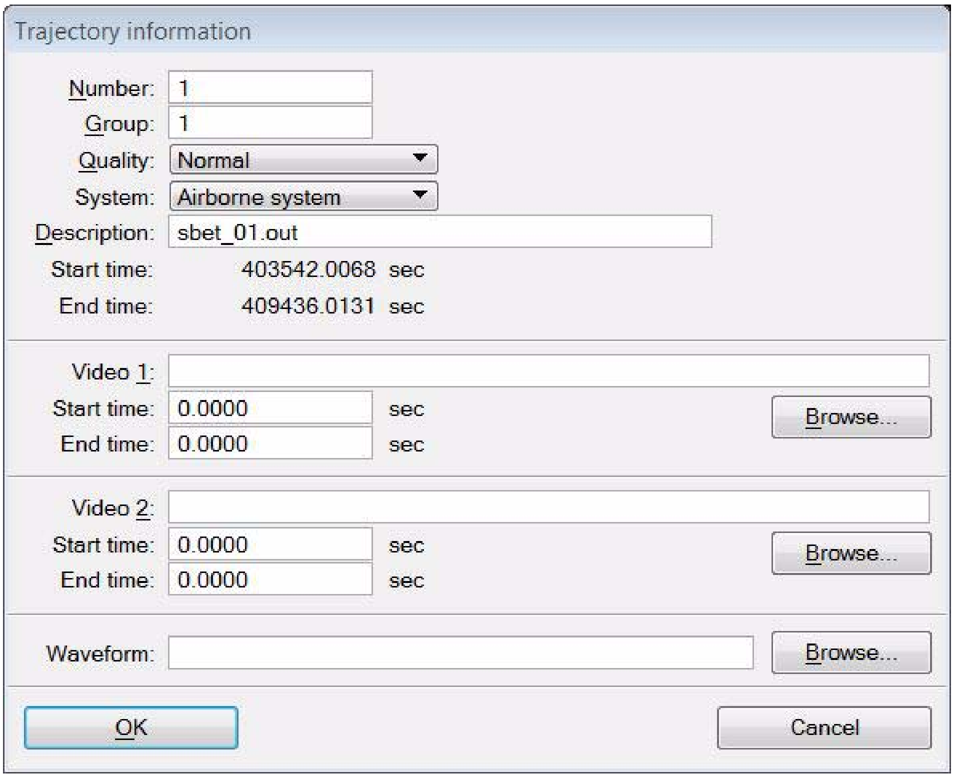
Edit information command opens a dialog that contains basic information and attributes stored for a selected trajectory. The attributes can be modified. Modifications are immediately stored in the binary trajectory file.

In addition, up to two video files and a waveform file can be linked to a trajectory. The video file settings are not actively used by TerraScan but required for the compatibility of trajectories with TerraPhoto.

➢ **To modify trajectory information:**

1. Select a trajectory in the list of the **Trajectories** window.
2. Select **Edit information** command from the **Trajectory** pulldown menu.

   This opens the **Trajectory information** dialog:



3. Define settings and click OK.

   This modifies the information in the header of the corresponding .TRJ file.

Setting:	**Effect:**
Number	Number of the trajectory.
Group	Group number of the trajectory. Group numbers may indicate, for example, different flight sessions and are used for TerraMatch processes.
Quality	Quality attribute of the trajectory. Quality may indicate, for example, the accuracy of trajectories and can be used for TerraMatch and TerraScan processes.
System	Scanner system used for data collection. This determines lever arm corrections that are added to trajectory positions and thus, effects the computation of the scanner location at the moment of measuring a laser point.
Description	Text that describes the trajectory. By default, the name of the raw trajectory file is used as descriptive text.

Setting:	Effect:
Video 1	Primary video file linked to the trajectory. This is not actively used by TerraScan but reqired for the compatibility of trajectories with TerraPhoto.
Start time	GPS time stamp of the start position of **Video 1**.
End time	GPS time stamp of the end position of **Video 1**.
Video 2	Secondary video file linked to the trajectory. This is not actively used by TerraScan but reqired for the compatibility of trajectories with TerraPhoto.
Start time	GPS time stamp of the start position of **Video 2**.
End time	GPS time stamp of the end position of **Video 2**.
Waveform	Waveform data file linked to the trajectory. See **Link to waveform files** command and Chapter **Waveform Processing** on page 155 for more information.

✍ If you select several trajectories in the **Trajectories** window, the **Edit information** command opens the **Edit several trajectories** dialog. This dialog allows you to modify only settings which may apply for several trajectories, such as **Group**, **Quality**, and **System** settings.

Assign number

Assign number command lets you modify the trajectory number based on laser points loaded in TerraScan. It assigns the line number of the laser point closest to a data click as trajectory number to the selected trajectory.

The command applies the number only to one trajectory at a time. If several trajectories are selected, only the first one is effected by the number assignment.

➢ **To assign a trajectory number from laser points:**

1. Load laser data into TerraScan.
2. Select a trajectory in the list of the **Trajectories** window.
3. Select **Assign number** command from the **Trajectory** pulldown menu.
4. Place a data click inside a view.

   This assigns the line number of the point closest to the data click location as number to the trajectory.

Delete

Delete command deletes one or more selected trajectory files. The entries for the files are removed from the list and the binary files are deleted from the hard disc.

➢ **To delete trajectories:**

1. Select the trajectory file(s) in the list of the **Trajectories** window.
2. Select **Delete** command from the **Trajectory** pulldown menu.

   A dialog asks to confirm the removal of the file(s).

3. Click **Yes** in order to delete the selected file(s).

   A dialog informs about the deletion process.

View positions

View positions command can be used to display the single positions of a trajectory file. The command opens a window that shows the list of positions and for each position the attributes stored in the trajectory file. This may include the time stamp, coordinate values, heading, roll, and pitch values, as well as RMS values.

➢ **To view trajectory positions:**

1. Select a trajectory file in the list of the **Trajectories** window.
2. Select **View positions** command from the **Trajectory** pulldown menu.

   This opens the **View trajectory positions** dialog which contains the list of trajectory positions.

To show the location of a trajectory position, select a line in the list of positions. Click on the **Show location** button and move the mouse pointer into a view. This highlights the selected position with a cross. Place a data click inside a view in order to center the display at the selected position.

To identify a position, click on the **Identify** button and place a data click close to a trajectory in a view. This selects the line of the position closest to the data click in the **View trajectory positions** dialog.

View pulldown menu

Commands of the **View** pulldown menu are used to change the size of the **Trajectories** window, to sort trajectory files in the list, and to select attribute fields for being displayed in the window.

To:	Use:
Display a small window	**Small dialog**
Display a larger window	**Large dialog**
Sort trajectories according to specific criteria	**Sort**
Select fields to be displayed in the window	**Fields**

Small dialog

Small dialog command changes the size of the **Trajectories** window to be a small window.

Large dialog

Large dialog command changes the size of the **Trajectories** window to be a large window.

Sort

Sort command defines the display order of trajectory files in the list. The trajectories can be sorted by up to two attributes.

➢ **To sort trajectory files:**

1. Select **Sort** command from the **View** pulldown menu.

   This opens the **Sort trajectories** dialog:

2. Select a **Primary key** and **Secondary key** for sorting.
3. Click OK.

   The display order of the trajectory files in the list is changed according to the settings.

Setting:	Effect:
Primary key	Attribute used first for sorting the trajectories: • **Number -** increasing trajectory numbers. • **Group -** increasing group numbers. • **Time -** increasing time stamps.
Secondary key	Attribute used second for sorting the images: • See **Primary key** attributes. • **None -** no secondary key is used for sorting.

Fields

Fields command lets you select which attributes are displayed for each trajectory in the **Trajectories** window.

➢ **To select visible fields:**

1. Select **Fields** command from the **View** pulldown menu.

   This opens the **View trajectory fields** dialog:

2. Select fields and click OK.

Field:	Description:
Number	Trajectory number.
Group	Group number of the trajectory.
Quality	Attribute that indicates the quality of the trajectory.
System	Scanner system assigned to the trajectory.
File	Name of the trajectory binary file on the hard disk.
Description	Description of the trajectory given in the **Trajectory information** dialog.
Vertical video	Name of the video file defined as **Video 1** in the **Trajectory information** dialog. See **Edit information** command.
Forward video	Name of the video file defined as **Video 2** in the **Trajectory information** dialog. See **Edit information** command.
Waveform file	Path and name of a waveform file linked to the trajectory.
Start time	Time stamp at the start of the trajectory.
End time	Time stamp at the end of the trajectory.
Duration	Length of the trajectory in seconds.

Tools pulldown menu

Commands in the **Tools** pulldown menu are used to manipulate trajectories and to create macros automatically based on trajectory information.

To:	Use:
Split a trajectory manually	**Split**
Split trajectories automatically at turnarounds	**Cut turnarounds**
Split trajectories and keep only parts inside a polygon	**Delete outside polygons**
Split trajectories at gaps in laser data	**Split at laser gaps**
Link trajectories to waveform files	**Link to waveform files**
Apply new numbers to trajectories	**Renumber trajectories**
Remove unnecessary trajectory positions	**Thin positions**
Transform trajectory coordinates	**Transform**
Add lever arms to trajectory positions	**Add lever arm**
Adjust trajectory elevations to a geoid model	**Adjust to geoid**
Convert trajectory angle values	**Convert angles**
Convert trajectory time stamps	**Convert time stamps**
Create macros automatically based on trajectory information	**Create macro / For stops and turns**
	Create macro / For poor accuracy
	Create macro / For repeated passes
Draw trajectories into the design file	**Draw into design**

Split

Split command can be used to split a trajectory manually into smaller parts. This is useful for some tools that require that one trajectory does not contain turnarounds and overlaps itself.

The command lets you define the location for splitting the trajectory with a data click.

➢ **To split a trajectory:**

1. Select **Split** command from the **Tools** pulldown menu.

   If the mouse pointer is moved inside a MicroStation view, the closest trajectory is highlighted.

2. Identify the trajectory to split with a data click.

   A red cross shows dynamically the split location.

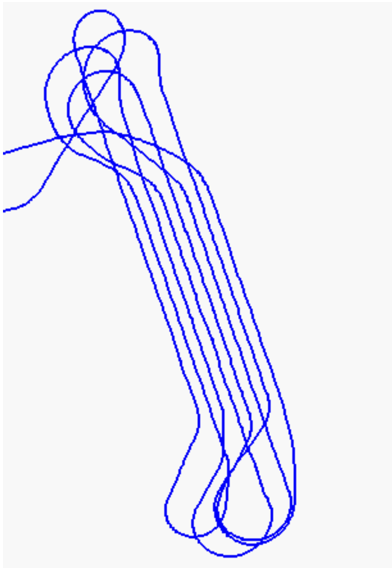
3. Define the position at which to split the trajectory with a data click.

   This splits the trajectory at the given position. The application deletes the old trajectory file and creates two new files in the active trajectory directory.

✍ There are also automatic ways to split a trajectory. See **Cut turnarounds**, **Delete outside polygons**, and **Split at laser gaps** commands for more information.

Cut turnarounds

Cut turnarounds command splits a trajectory into several trajectories that do not overlap themselves anymore. It does not remove any parts of the original trajectory. The following figure illustrates the method.



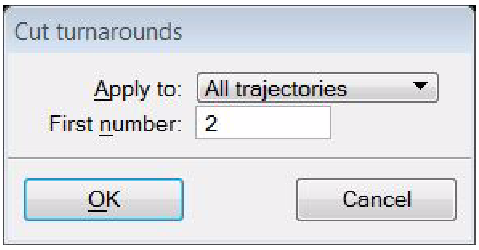
Original trajectory

Resulting trajectories

➢ **To cut trajectories at turnarounds:**

1. (Optional) Select the trajectory file(s) you want to cut in the **Trajectories** window.
2. Select **Cut turnarounds** command from the **Tools** pulldown menu.

   This opens the **Cut turnarounds** dialog:



3. Define settings and click OK.

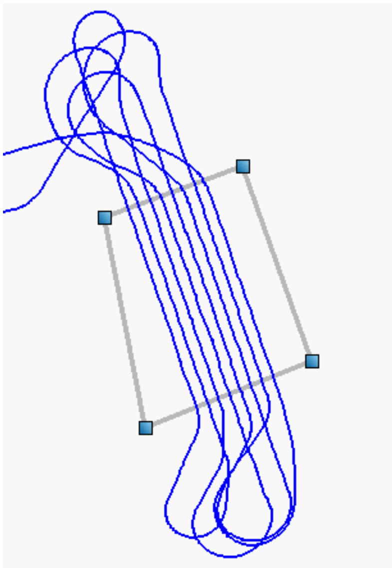
   This splits trajectories whenever there is a close to 180 degree turn. The application deletes the old trajectory file(s) and creates new files in the active trajectory directory.

Field:	Description:
Apply to	Trajectories effected by the process: **All trajectories** or **Selected only**.
First number	Number of the first additional trajectory that is created by the process.

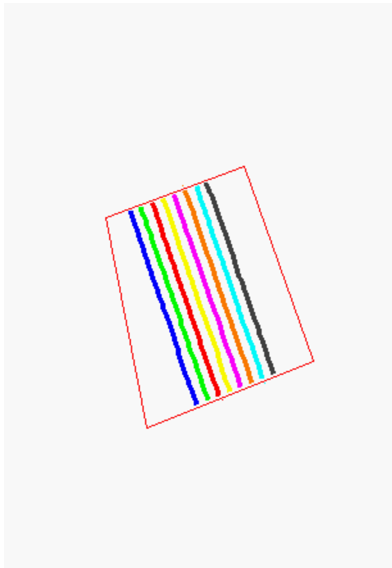
Delete outside polygons

Delete outside polygons command cuts trajectories at the boundary of a shape element. It keeps only trajectory lines inside the shape and deletes all parts outside.

This is often the easiest way to split trajectories of aerial projects. It requires one or multiple MicroStation shape elements that enclose areas for which to keep trajectory information. The figure below illustrates the method.



Original trajectory

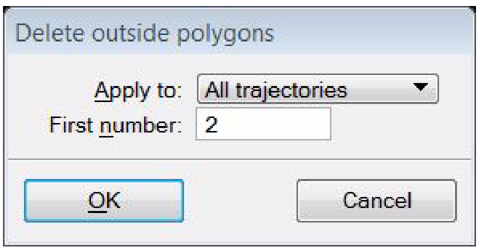


Resulting trajectories

➢ **To delete trajectories outside polygons:**

1. Use MicroStation tools to draw polygon(s) around areas where you want to keep trajectory information. *S*elect the polygon(s).
2. (Optional) Select the trajectory file(s) you want to cut in the **Trajectories** window.
3. Select **Delete outside polygons** command from the **Tools** pulldown menu.

   This opens the **Delete outside polygons** dialog:



4. Define settings and click OK.

   This deletes all trajectory parts outside the selected polygon(s). The application deletes the old trajectory file(s) and creates new files in the active trajectory directory. An information dialog shows the result of the process.

Field:	Description:
Apply to	Trajectories effected by the process: **All trajectories** or **Selected only**.
First number	Number of the first additional trajectory that is created by the process.

Split at laser gaps

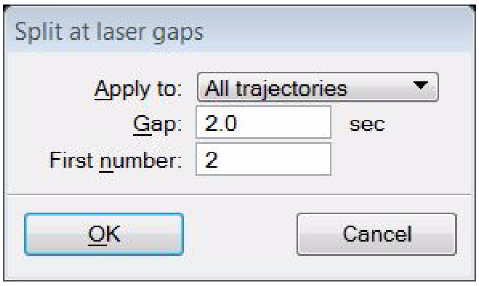
Split at laser gaps command cuts trajectories if there is a gap in laser data. It removes part of trajectories where there is no laser data available.

The command requires a project loaded into TerraScan which references the laser binary files for detecting gaps. See *Define Project* tool and Chapter **Working with Projects** on page 317 for more information about projects in TerraScan.

➢ **To split trajectories at laser gaps:**

1. (Optional) Select the trajectory file(s) you want to cut in the **Trajectories** window.
2. Select **Split at laser gaps** command from the **Tools** pulldown menu.

   This opens the **Split at laser gaps** dialog:



3. Define settings and click OK.

   This deletes all trajectory parts outside the area covered by laser data. The application deletes the old trajectory file(s) and creates new files in the active trajectory directory. An information dialog shows the result of the process.

Field:	Description:
Apply to	Trajectories effected by the process: **All trajectories** or **Selected only**.
Gap	Minimum time interval that defines a gap in the laser data. If there is data missing for a longer time, the trajectory is splitted.
First number	Number of the first additional trajectory that is created by the process.

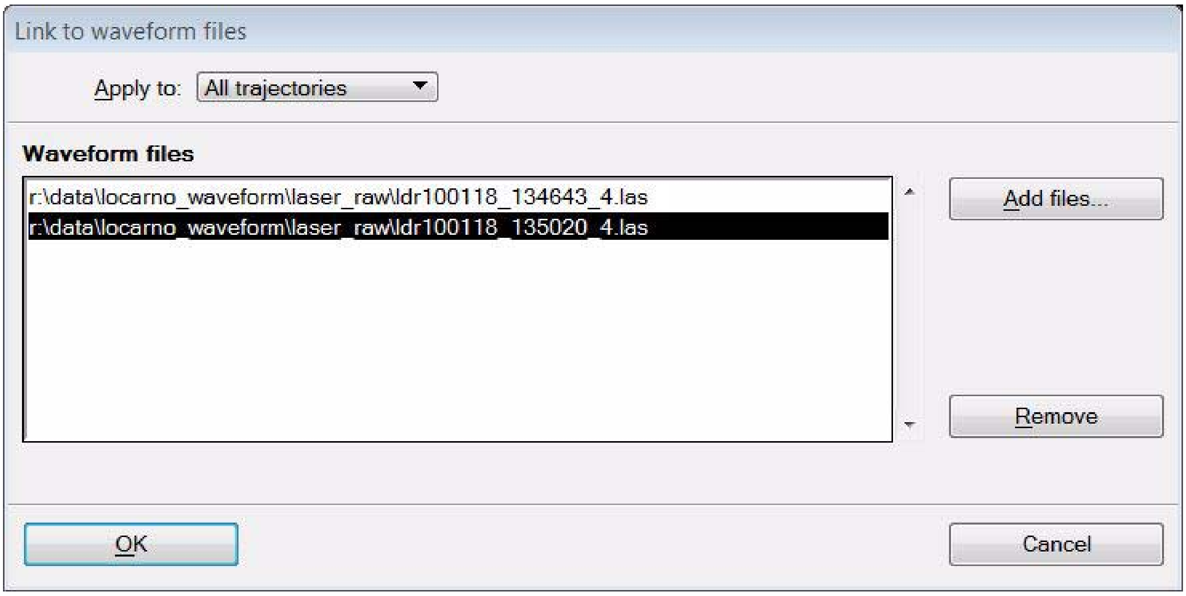
Link to waveform files

Link to waveform files command links trajectories to waveform files. This is required for any processing tasks based on waveform data. See Chapter **Waveform Processing** on page 155 for more information. You can check if a waveform file is linked to a trajectory in the **Trajectory information** dialog that is opened by the **Edit information** command.

➢ **To link trajectories to waveform files:**

1. (Optional) Select the trajectory file(s) you want to link in the **Trajectories** window.
2. Select **Link to waveform files** command from the **Tools** pulldown menu.

   This opens the **Link to waveform files** dialog:



3. Click on the **Add files** button.

   This opens the **Waveform files** dialog, a standard dialog for opening files.

4. Select the waveform file(s) and click **Done**.

   This adds the file(s) to the list of waveform files in the **Link to waveform files** dialog.

5. Click OK.

   This links trajectories to the waveform files and updates the trajectory files in the active trajectory directory. An information dialog shows the amount of trajectories that are effected by the process.

Field:	Description:
Apply to	Trajectories linked to waveform files: **All trajectories** or **Selected only**.
Add files	Opens a dialog for selecting waveform files.
Remove	Removes a selected waveform file from the list.

Renumber trajectories

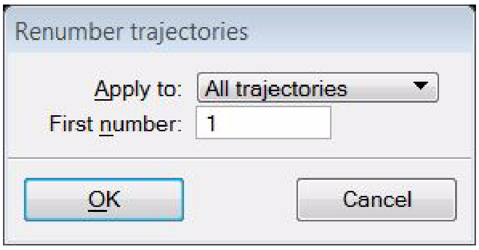
Renumber trajectories command applies a new numbering to trajectories. It assigns increasing numbers to the trajectories according to their order in the **Trajectories** window.

Renumbering can be useful, for example after a new sorting has been applied to the active trajectories using the **Sort** command.

➢ **To renumber trajectories:**

1. (Optional) Select the trajectory file(s) you want to sort in the **Trajectories** window.
2. Select **Renumber trajectories** command from the **Tools** pulldown menu.

   This opens the **Renumber trajectories** dialog:



3. Define settings and click OK.

   This assigns new numbers to the trajectories and updates the trajectory files in the active trajectory directory. An information dialog shows the number of the trajectories effected by the process.

Field:	Description:
Apply to	Trajectories effected by the process: **All trajectories** or **Selected only**.
First number	Number of the first trajectory in the list.

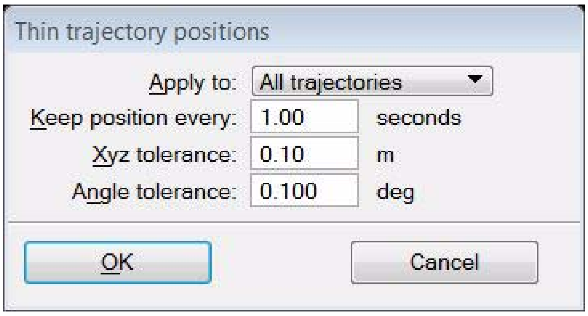
Thin positions

Thin positions command removes positions from trajectories. The process removes positions as long as the trajectory line stays within given accuracy tolerance values.

➢ **To thin positions of trajectories:**

1. (Optional) Select the trajectory file(s) you want to thin in the **Trajectories** window.
2. Select **Thin positions** command from the **Tools** pulldown menu.

   This opens the **Thin trajectory positions** dialog:



3. Define settings and click OK.

   This removes unnecessary trajectory positions and updates the trajectory files in the active trajectory directory. An information dialog shows the number of the trajectories effected by the process.

Field:	Description:
Apply to	Trajectories effected by the process: **All trajectories** or **Selected only**.
Keep position every	Time interval between two trajectory positions to keep.
Xyz tolerance	Maximum allowed locational change of the trajectory caused by thinning.
Angle tolerance	Maximum allowed angular change of the trajectory caused by thinning.

✍ Thinning can also be applied to a trajectory in the import process. See **Import files** command for more information.

Transform

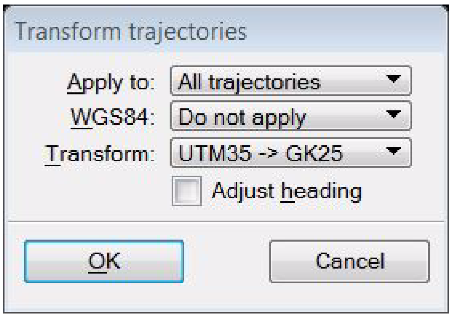
Transform command applies a transformation to the coordinates of a trajectory. The transformation can be, for example, a change of the projections system or any other transformation defined in **Coordinate transformations / Transformations** category of TerraScan **Settings**.

You can find more detailed information about transformations in Chapter **Coordinate Transformations** on page 312.

➢ **To transform a trajectory:**

1. (Optional) Select trajectory file(s) to transform.
2. Select **Transform** command from the **Tools** pulldown menu.

   This opens the **Transform trajectories** dialog:



3. Define settings and click OK.

   The coordinates of the trajectory are changed. The modification is saved to the trajectory binary files in the active trajectory directory. An information dialog shows the number of effected trajectories.

Setting:	Effect:
Apply to	Trajectories to transform: • **All trajectories** - all trajectories in the list. • **Selected only** - selected trajectories only.
WGS84	Target projection system for applying a transformation from WGS84 coordinates to the given projection system. You can choose from any of the builtin or user-defined projections systems which are set as active in **Coordinate transformations / Built-in projection systems**, **Coordinate transformations / US State Planes**, or **Coordinate transformations / User projection systems** categories of TerraScan **Settings**.
Transform	User-defined transformation to apply. The list includes transformations that is defined in **Coordinate transformations / Transformations** of TerraScan **Settings**.
Adjust heading	If on, the software applies a meridian convergence correction to heading values. The correction is based on the projection system set for **WGS84** or the coordinate transformation set for **Transform**.

Add lever arm

Add lever arm command applies a lever arm to trajectories. A lever arm is expressed by the X,Y, and Z components of a vector between the original trajectory position and the lever arm-corrected position.

The direction of the three vector components relative to the trajectory or system movement direction is as follows:

- **X -** positive values to the right, negative to the left.
- **Y -** positive values forward, negative backward.
- **Z -** positive values up, negative down.

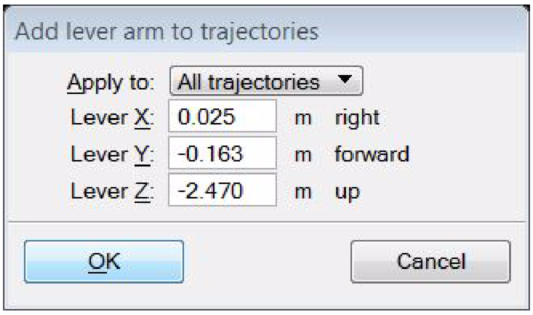
A lever arm should be applied, if the trajectory has been computed for the IMU and the point cloud has been generated without considering the lever arm values. Then, the lever arm vector describes the distance between the IMU and the scanner. However, this is commonly done by post-processing software for one scanner systems. For multiple scanner systems, the **Scanner systems** definition in TerraScan **Settings** defines the lever arms of the different scanners which can be applied in the import process of trajectories. See **Import files** command for more information.

An application example for applying lever arms to active trajectories is to project the trajectory of a MMS survey to the center of a rail track. In this case, the lever arm vector describes the distance between the IMU (trajectory location) and the center between the wheels of the vehicle carrying the system along the tracks.

➢ To add a lever arm to trajectories:

1. (Optional) Select the trajectory file(s) you want to modify in the **Trajectories** window.
2. Select **Add lever arm** command from the **Tools** pulldown menu.

   This opens the **Add lever arm to trajectories** dialog:



3. Define settings and click OK.

   This modifies the trajectory positions and updates the trajectory files in the active trajectory directory. An information dialog shows the number of the trajectories effected by the process.

Field:	Description:
Apply to	Trajectories effected by the process: **All trajectories** or **Selected only**.
Lever X	X component of the lever arm vector.
Lever Y	Y component of the lever arm vector.
Lever Z	Z component of the lever arm vector.

Adjust to geoid

Adjust to geoid command applies an elevation correction to trajectory files. The command is used, for example, to transform the WGS84-based ellipsoidal elevation values of a raw trajectory file to a local height model. The input model for geoid adjustment must be provided in one of the following formats:

- **Points from file -** text file containing space-delimited X Y dZ- points.
- **TerraModeler surface -** triangulated surface model created from X Y dZ - points. The surface model in TerraModeler has the advantage that you can visualize the shape of the adjustment model.
- **Selected linear chain -** linear element of which the vertices represent the X Y dZ - points.

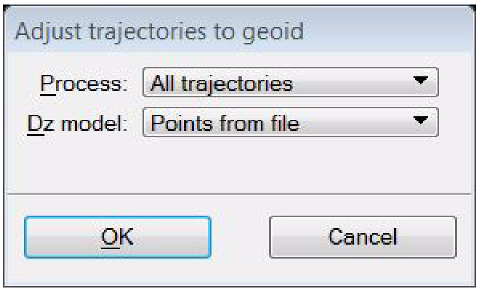
XY are the easting and northing coordinates of the geoid model points, dZ is the elevation difference between ellipsoidal and local heights at the location of each geoid model point. Intermediate adjustment values of the model are derived by aerial (text file or surface model as input) or linear (linear element as input) interpolation between the known geoid model points.

You can find more detailed information about elevation adjustment in Section **Geoid adjustment** on page 313.

➢ **To adjust trajectories to a geoid model:**

1. (Optional) Load a geoid model into TerraModeler.
2. (Optional) Select trajectory file(s) to adjust.
3. Select **Adjust to geoid** command from the **Tools** pulldown menu.

   This opens the **Adjust trajectories to geoid** dialog:



4. Define settings and click OK.

   If **Points from file** is selected as the **Dz model**, the **Geoid dz file** dialog opens, a standard dialog for opening files.
5. Define the text file that contains the geoid point coordinates and elevation differences and click **Open**.

   This applies the elevation adjustment to all or selected trajectories. The modification is saved to the trajectory binary files in the active trajectory directory. An information dialog shows the minimum and maximum values of the adjustment.

Setting:	Effect:
Process	Trajectories to adjust: • **All trajectories -** all trajectories in the list. • **Selected only -** selected trajectories only.
Dz model	Source file that provides the geoid correction model: • **Points from file -** text file. • **Selected linear chain -** linear element selected in the design file. • <*name*> - name of the geoid model surface loaded in TerraModeler.
Extend	Distance of a linear extension. This is only active if **Dz model** is set to **Selected linear chain**.

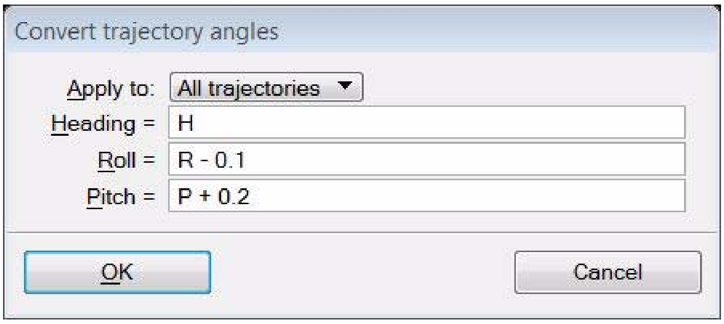
Convert angles

Convert angles command lets you apply a mathematical equation to the attitude angles heading, pitch, and roll of each trajectory position. The current angle value can be accessed by using constants H (heading), R (roll), and P (pitch). Thus, the command can also be used to exchange angle values.

➢ **To convert angles of trajectory positions:**

1. (Optional) Select trajectory file(s) for which to manipulate angles.
2. Select **Convert angles** command from the **Tools** pulldown menu.

   This opens the **Convert trajectory angles** dialog:



3. Define equations and click OK.

   This computes the new values for the orientation angles. The modification is saved to the trajectory binary files in the active trajectory directory. An information dialog shows the number of effected trajectories.

Setting:	Effect:
Apply to	Trajectories for which the computation of new angles is applied: • **All trajectories -** all trajectories in the list. • **Selected only -** selected trajectories only.
Heading	Equation for modifying the heading angle.
Roll	Equation for modifying the roll angle.
Pitch	Equation for modifying the pitch angle.

Convert time stamps

Convert time stamps command can be used to convert the format of time stamps. Supported conversions are GPS seconds-of-week to GPS standard time, Unix time to GPS standard time, GPS standard time to GPS seconds-of-week, and Unix time to GPS seconds-of-week.

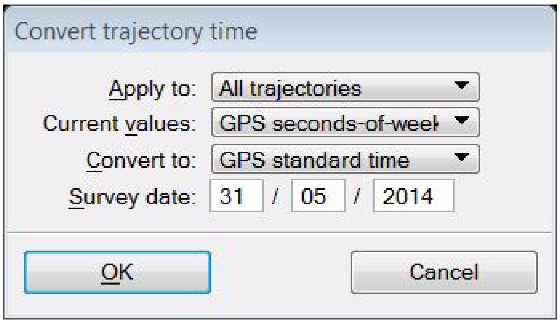
The conversion is necessary, for example, if data collected in several weeks is processed together in one project. Then, GPS seconds-of-week time stamps result in repeated values and GPS standard time must be used in order to provide unique time stamps for each trajectory position. This is a requirement for many processes that rely on trajectory information. Some post-processing software generates data with Unix seconds-of-day time stamps. They must be converted into another GPS time format as well.

It is essential that time stamps of trajectories and laser data are stored in the same GPS time format.

➢ **To convert time stamps:**

1. (Optional) Select trajectory file(s) for which to manipulate angles.
2. Select **Convert time stamps** command from the **Tools** pulldown menu.

   This opens the **Convert trajectory time** dialog:



3. Define settings and click OK.

   This converts the trajectory time stamps to the new format. The modification is saved to the trajectory binary files in the active trajectory directory. An information dialog shows the number of effected trajectories.

Setting:	Effect:
Apply to	Trajectories for which the conversion of time stamps is applied: • **All trajectories -** all trajectories in the list. • **Selected only -** selected trajectories only.
Current values	Original time stamp format of the trajectory positions.
Convert to	Target time stamp format.
Survey date	Date when the trajectory data was captured. The format is day/month/year (dd/mm/yyyy). This is only active for the conversion from **GPS seconds-of-week** to **GPS standard time**.

Create macro / For stops and turns

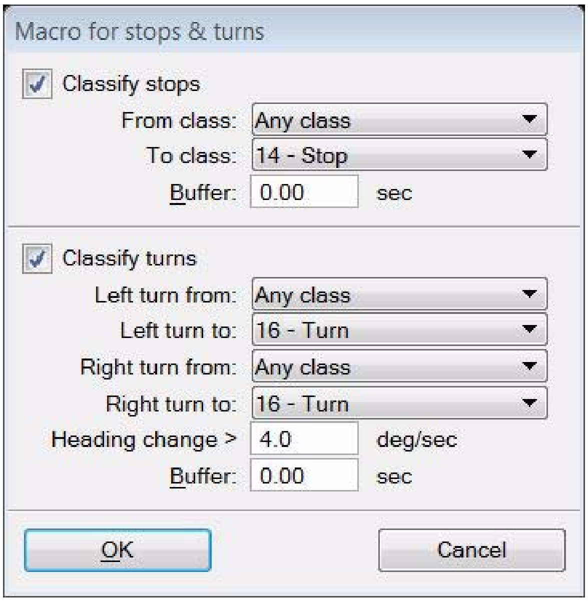
Create macro / For stops and turns command creates a TerraScan macro automatically. The macro is used to classify points of an MLS data set that were collected during stops or in sharp turns.

Since stops and turns cause a slowing-down of the vehicles speed, the scanner collects significantly more data than at normal operating speed. The command can be used to identify locations of stops and turns based on trajectory position attributes. If the search for stop and turns finds such locations, the resulting macro contains steps that classify points based on time intervals.

➢ **To create a macro for stops and turns:**

1. Select **Create macro / For stops & turns** command from the **Tools** pulldown menu.

   This opens the **Macro for stops & turns** dialog:



2. Define settings and click OK.

   The software computes stops and turns from the trajectory positions and creates a TerraScan macro. An information dialog shows the number of added time intervals.

   The macro can be saved and applied to the laser points. See Chapter **Macros** on page 429 for more information about macros in TerraScan.

Setting:	Effect:
Classify stops	If on, the software searches for stops.
From class	Point class(es) from which to classify points. The list contains the active point classes in TerraScan.
To class	Target class for points collected during stops. The list contains the active point classes in TerraScan.
Buffer	Number of seconds that is added to each stop time interval. The seconds are added at the beginning and in the end of a stop.
Classify turns	If on, the software searches for turns.
Left turn from	Point class(es) from which to classify points collected during turns to the left. The list contains the active point classes in TerraScan.
Left turn to	Target class for points collected during a turns to the left. The list contains the active point classes in TerraScan.
Right turn from	Point class(es) from which to classify points collected during turns to the right. The list contains the active point classes in TerraScan.

Setting:	Effect:
Left turn to	Target class for points collected during a turns to the right. The list contains the active point classes in TerraScan.
Heading change	Minimum change in heading angle that defines a turn. As long as the heading angle between consecutive trajectory positions changes more than the given degree value per second, the change is considered a turn and the respective time stamps are added to the macro step.
Buffer	Number of seconds that is added to each turn time interval. The seconds are added at the beginning and in the end of a turn.

Create macro / For poor accuracy

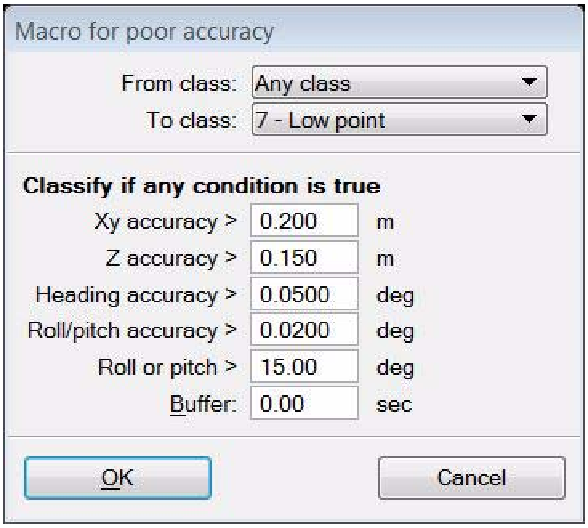
Create macro / For poor accuracy command creates a TerraScan macro automatically. The macro is used to classify points that were collected from locations of bad trajectory accuracy.

Especially in MLS data sets, there might be places of poor trajectory accuracy caused, for example, by the lack of GPS signals. The command can be used to identify such locations based on trajectory position attributes. The process uses accuracy values that are assigned to trajectory positions. See **Import accuracy files** for more information. If the search finds poor accuracy locations, the resulting macro contains steps that classify points based on time intervals.

➢ **To create a macro for poor accuracy:**

1. Select **Create macro / For poor accuracy** command from the **Tools** pulldown menu.

   This opens the **Macro for poor accuracy** dialog:



2. Define settings and click OK.

   The software computes poor accuracy time ranges from the trajectory positions and creates a TerraScan macro. An information dialog shows the number of added time intervals.

   The macro can be saved and applied to the laser points. See Chapter **Macros** on page 429 for more information about macros in TerraScan.

Setting:	**Effect:**
From class	Point class(es) from which to classify points. The list contains the active point classes in TerraScan.
To class	Target class for points collected during poor trajectory accuracy. The list contains the active point classes in TerraScan.
Xy accuracy	If the xy accuracy value is bigger than the given value, the trajectory position is added to the macro.
Z accuracy	If the z accuracy value is bigger than the given value, the trajectory position is added to the macro.
Heading accuracy	If the heading accuracy value is bigger than the given value, the trajectory position is added to the poor accuracy macro.
Roll/pitch accuracy	If the roll/pitch accuracy value is bigger than the given value, the trajectory position is added to the macro.
Roll or pitch	If the value of roll or pitch angle is bigger than the given value, the trajectory position is added to the macro.
Buffer	Number of seconds that is added to each poor accuracy time interval. The seconds are added at the beginning and in the end of a time interval.

Create macro / For repeated passes

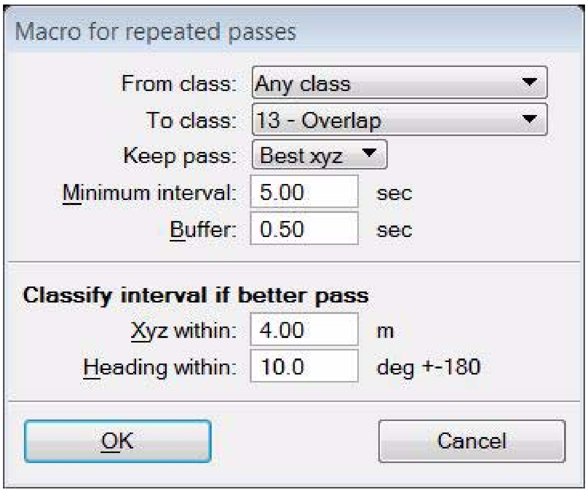
Create macro / For repeated passes command creates a TerraScan macro automatically. The macro is used to classify points collected from the same location in several strips.

Especially in MLS data sets of roads inside urban areas, there might be places where data was collected several times during a survey. The command can be used to identify such locations based on trajectory position attributes. The process can make use of accuracy values that are assigned to trajectory positions. See **Import accuracy files** for more information. If the search finds repeated pass locations, the resulting macro contains steps that classify points based on time intervals.

➢ **To create a macro for repeated passes:**

1. Select **Create macro / For repeated passes** command from the **Tools** pulldown menu.

   This opens the **Macro for repeated passes** dialog:



2. Define settings and click OK.

   The software computes time ranges of repeated passes from the trajectory positions and creates a TerraScan macro. An information dialog shows the number of added time intervals.

   The macro can be saved and applied to the laser points. See Chapter **Macros** on page 429 for more information about macros in TerraScan.

Setting:	Effect:
From class	Point class(es) from which to classify points. The list contains the active point classes in TerraScan.
To class	Target class for points collected in repeated passes. The list contains the active point classes in TerraScan.
Keep pass	Defines which data is kept in the original class: • **First -** data of the first pass. • **Last -** data of the last pass. • **Best xyz -** data of any pass with the best positional accuracy. • **Best hrp** - data of any pass with the best attitude accuracy.
Minimum interval	Minimum time interval within which the software searches for repeated pass points.
Buffer	Number of seconds that is added to each repeated pass time interval. The seconds are added at the beginning and in the end of a time interval.
Xyz within	Defines the maximum distance between repeated passes. If passes are less than the given distance apart from each other, they are considered repeated passes.

Setting:	Effect:
Heading within	Defines the maximum angular difference between repeated passes. If the heading angle between passes is smaller than the given value, they are considered as repeated passes.

Draw into design

Draw into design command draws the trajectories as line elements into the design file. The line elements are drawn on the active level using the active line width and line style settings of MicroStation. The color(s) of the line elements are defined by the command's settings.

The command can use accuracy values that are assigned to trajectory positions. See **Import accuracy files** for more information.

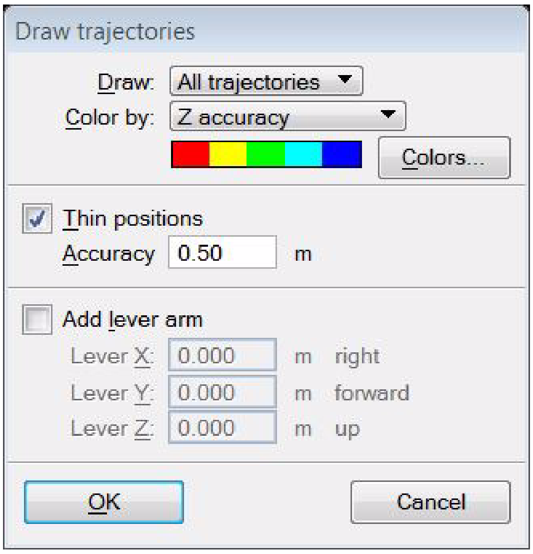
The line elements are drawn by placing a vertex for each trajectory position. The lines can by simplified by removing positions within a given tolerance.

You can apply lever arms to trajectories when drawing them into the design file. This is useful, for example, if a centerline or other line elements along rails are derived from the trajectories of an MLS survey. The lever arm values are only applied to the line elements drawn into the design file but do not effect the original trajectory files.

➢ **To draw trajectory lines into the design file:**

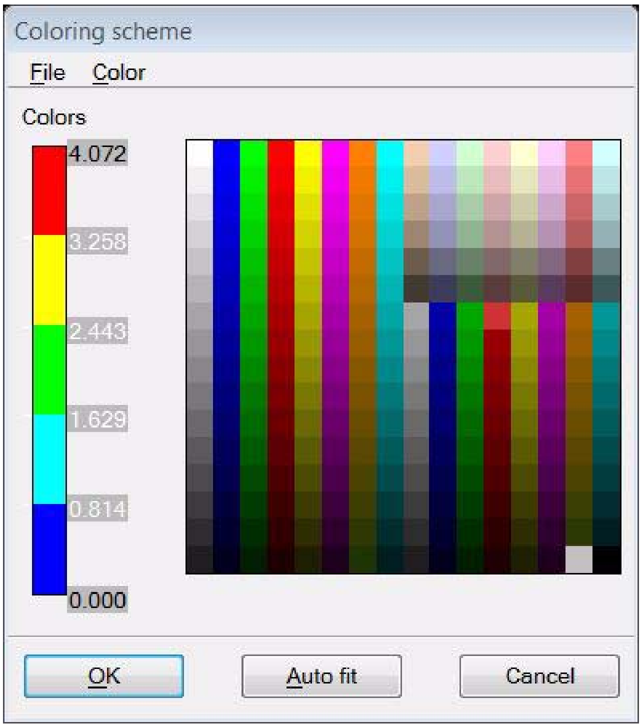
1. (Optional) Select trajectory file(s) to draw.
2. Select **Draw into design** command from the **Tools** pulldown menu.

   This opens the **Draw trajectories** dialog:



3. Define settings.
4. If the trajectory is drawn with an accuracy-based coloring option, click on the **Colors** button.

   This opens the **Coloring scheme** dialog:



5. (Optional) Define your own coloring scheme for drawing trajectories.
6. Click on the **Auto fit** button in order to fit the colors to RMS value ranges.
7. Click OK to the **Coloring scheme** dialog.
8. Click OK to the **Draw trajectories** dialog.

   This draws the line element(s) into the design file.

Setting:	Effect:
Draw	Trajectories that are drawn: • **All trajectories -** all trajectories in the list. • **Selected only -** selected trajectories only.
Color by	Determines how the color is chosen for drawing a trajectory line: • **Active color -** the active color of MicroStation is used. • **Trajectory number -** the color whose number in the active color table of MicroStation corresponds to the trajectory number is used. • **Xy accuracy -** x/y accuracy values are applied to a color scheme. • **Z accuracy -** z accuracy values are applied to a color scheme. • **H accuracy -** heading accuracy values are applied to a color scheme. • **Rp accuracy -** roll/pitch accuracy values are applied to a color scheme.
Colors	Button to open the coloring scheme for accuracy-based coloring methods.
Thin positions	If on, intermediate trajectory positions are skipped when the line is drawn as long as the line accuracy stays within the given positional **Accuracy** tolerance.

✍ You can undo the drawing of trajectories by using the **Undo** command from the **Edit** pulldown menu of MicroStation.

14 Geometry Component Fitting

15 Key-in commands

Some of the tools in TerraScan can be started by entering a key-in command.

Key-in commands are handy because you can assign those to function keys in MicroStation and thus have hot key combinations for specific operations.

This chapter gives information about those key-in commands which you may consider assigning to function keys.

Add Point To Ground

Add Point To Ground key-in command adds a point to the ground class and reiterates the surroundings. You identify the laser point to add after starting the command.

Assign Point Class

Assign Point Class key-in command assigns a point class to a single laser point or to all points belonging to the same group. You identify the laser point with a mouse click after starting the command.

Syntax is:

```
Assign Point Class from=2/to=6/classify=single/select=highest/within=2.0
```

Possible parameters are:

Parameter:	Effect:
from=n	Source class from which to classify. Use from=999 or from=any to use any visible points.
to=n	Target class to classify to.
classify=x	Points to classify: single or group.
select=x	How to select point to classify: closest, highest or lowest.
within=n	Search radius in master units if selecting highest or lowest point.

Classify Above Line

Classify Above Line key-in command classifies points above a given line in a section view. You specify the line with two mouse clicks after starting the command.

Syntax is:

```
Classify Above Line [From] To
```

where **From** is optional point class number to classify from and **To** is point class to classify to. If only one parameter is given, all visible points above the line will be classified.

Classify Below Line

Classify Below Line key-in command classifies points below a given line in a section view. You specify the line with two mouse clicks after starting the command.

Syntax is:

```
Classify Below Line [From] To
```

where **From** is optional point class number to classify from and **To** is point class to classify to. If only one parameter is given, all visible points below the line will be classified.

Classify Close To Line

Classify Close To Line key-in command classifies points above, below or close to given line in a section view. Source classes, destination classes and line tolerances may be given as parameters following the key-in command. Parameters are separated from each other with / character, for example:

```
Classify Close To Line abovefrom=any/aboveto=3/abovetol=0.1
```

Possible parameters are:

Parameter:	Effect:
abovefrom=n	Source class above line.
aboveto=n	Target class above line.
abovetol=x	Tolerance above line.
closeform=n	Source class close to line.
closeto=n	Target class close to line.
belowfrom=n	Source class below line.
belowto=n	Target class to classify to below line.
belowtol=x	Tolerance below line.

Classify Fence

Classify Fence key-in command classifies points inside fence and then starts the *Place Fence* tool so that you can immediately place the next fence.

Syntax is:

```
Classify Fence [From] To
```

where **From** is optional point class number to classify from and **To** is point class to classify to. If only one parameter is given, points in any class are classified to the given class.

Classify Inside Shapes

Classify Inside Shapes key-in command classifies points inside selected shapes. Source class, destination class and expand distance may be given as parameters following the key-in command. Parameters are separated from each other with / character, for example:

```
Classify Inside Shapes from=5/to=6/expand=1.0
```

Possible parameters are:

Parameter:	Effect:
from=n	Source class from which to classify. Use from=999 or from=any for any class.
to=n	Target class to classify to.
expand=n	Distance by which to expand shapes. Positive values expand and negative values shrink the shapes.

Classify View

Classify View key-in command classifies points which are visible inside a view. You select the view with a mouse click after starting the command.

Syntax is:

Classify View **[From] To**

where **From** is optional point class number to classify from and **To** is point class to classify to. If only one parameter is given, points in any class are classified to the given class.

Classify Using Brush

Classify Using Brush key-in command classifies points inside a circular brush. Source class, destination class and brush size are given as parameters following the key-in command. Parameters are separated from each other with / character, for example:

Classify Using Brush from=2/to=6/size=15

Possible parameters are:

Parameter:	Effect:
from=n	Source class from which to classify. Use from=999 or from=any to use any visible points.
to=n	Target class to classify to.
size=n	Brush radius as pixels on screen.

Move Backward

Move Backward key-in command moves a section view backward by full view depth. The key-in command must include a view number (1-8).

Syntax is:

Move Backward **ViewNumber**

Move Forward

Move Forward key-in command moves a section view forward by full view depth. The key-in command must include a view number (1-8).

Syntax is:

Move Forward **ViewNumber**

Scan Display

Scan Display key-in command changes view display mode for laser points. The new display settings are given as parameters following the key-in command. Parameters are separated from each other with / character, for example:

```
Scan Display off=all/on=2/color=class
```

Possible parameters are:

Parameter:	Effect:
view=n	View to apply to where n is view number 1-8. If this parameter is given, it must be the first one and causes an immediate change to the specified view. If not given, the application will wait for the user to select a view.
on=n	Classes to switch on. For example: on=2 or on=1,5-8 or on=all.
off=n	Classes to switch off. For example: off=7 or off=3-11 or off=all.
color=n	Coloring mode where n is class, echo, elevation, line, intensity, distance, color, lineint, echolen, scanner, dimension, group, shading or slope. For example: color=class.
weight=n	Weight mode where n is weight value 0-7 or -1 for class based weight. For example: weight=1 or weight=-1.
lineon=n	Flightlines to switch on. For example: lineon=1-3 or lineon=all.
lineoff=n	Flightlines to switch off. For example: lineoff=1-3 or lineoff=all.
sparse=n	Sparse display mode. For example: sparse=on or sparse=off.
depth=	Depth display mode. For example: depth=on or depth=off.

Scan Fit View

Scan Fit View key-in command fits a view to display the area covered by laser points. You can specify the view to fit as an optional parameter or by selecting a view with a mouse click.

Syntax is:

Scan Fit View **[View]**

where **View** is an optional view number (1-8).

Scan Run Macro

Scan Run Macro key-in command executes a macro.

Syntax is:

Scan Run Macro **Macrofile**

where **Macrofile** is the file name of the macro to execute. If the file name does not include a directory path, the value of TSCAN_MACRODIR environment variable is used as the directory to search from.

Travel Step Backward [Count]

Travel Step Backward key-in command moves *Travel Path* windows by one section backward. The key-in may be followed by an optional number which specifies how many steps to move.

Travel Step Forward [Count]

Travel Step Forward key-in command moves *Travel Path* windows by one section forward. The key-in may be followed by an optional number which specifies how many steps to move.

Fix Elevation

Fix Elevation key-in command changes the elevations of laser points inside given polygons. Each laser point inside one polygon is fixed to the same elevation value.

Parameters are separated from each other with / character, for example:

```
Fix Elevation class=2,8/percentile=5/level=10/color=7
```

Possible parameters are:

Parameter:	Effect:
class=n	Classes to use and modify. For example: class=2,5-8 or class=all.
polygon=n	Type of polygons where n is fence, selected or level.
percentile=n	If given, computes elevation from the points inside the polygon and n specifies the percentile (0=minimum, 50=median, 100=maximum).
elevation=n	If given, uses a fixed elevation n. For example: elevation=0.0.
level=n	Design file level to search for polygons. If given, forces polygon=level.
color=n	If given, filters polygons by color in addition to level.
weight=n	If given, filters polygons by line weight in addition to level.
style=n	If given, filters polygons by line style in addition to level.

Open Block

Open Block key-in command opens a project block for viewing or modification.

Syntax is:

```
Open Block [BlockFile [options]]
```

BlockFile may specify block file name or block number.

Options can be:

Option:	Effect:
neighbours=n	Distance to load points from neighbouring blocks.
fit=n	View(s) to fit where n is between -1 and 8 (-1=all, 0=none, 1-8=view).
lock=n	If n is 0, opens block for viewing only. If n is not zero, opens block for modification.

Scan Delete Inside Fence

Scan Delete Inside Fence key-in command deletes points inside fence. Tool waits for user to accept the operation.

Scan Delete Outside Fence

Scan Delete Outside Fence key-in command deletes points outside fence. Tool waits for user to accept the operation.

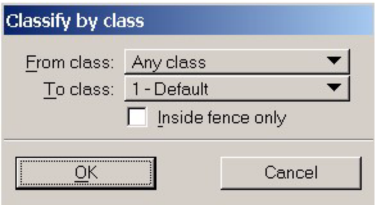
Batch Processing Reference

16 Classification Routines

The **Routine** menu in the TerraScan **Main** window offers a number of classification tools. Additionally, there are some routines which can be used in macros only. This chapter describes the basic logic used in various classification routines.

By class

By class routine simply changes all points in a given class to another class.



Setting:	Effect:
From class	Source class from which to classify points.
To class	Target class to classify points into.
Inside fence only	If on, points inside a fence or selected polygon are classified.

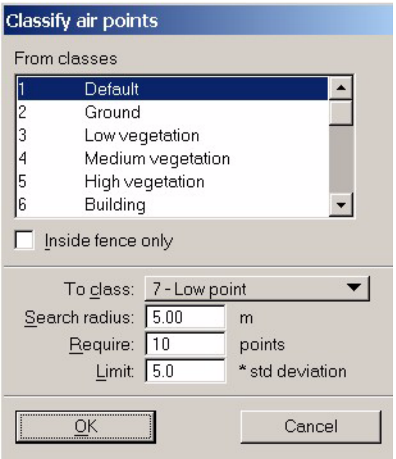
✓ You can use this to undo a classification.

Air points

Not Lite

Air points routine classifies points which are clearly higher than the median elevation of surrounding points. It can be used to classify noise up in the air.

For each point, this routine will find all the neighboring points within a given search radius. It will compute the median elevation of the points and the standard deviation of the elevations. A point will be classified if it is more than **Limit** times standard deviation above the median elevation. Comparison using standard deviation results in the routine being less likely to classify points in places where there is more elevation variation.



Setting:	Effect:
From classes	Source class(es) from which to classify points. Select several classes by pressing the **Ctrl-key** while selecting classes from the list.
Inside fence only	If on, points inside a fence or selected polygon are classified.
To class	Target class to classify points into.
Search radius	2D search radius around a point. For points within this radius the median elevation and standard deviation are computed. Normally 2.0 - 10.0 m.
Require	Minimum amount of points required within the radius. Routine will not classify a point if it does not find enough points within the xy search radius.
Limit	Factor for multiplication with standard deviation of elevation values of points within the search radius.

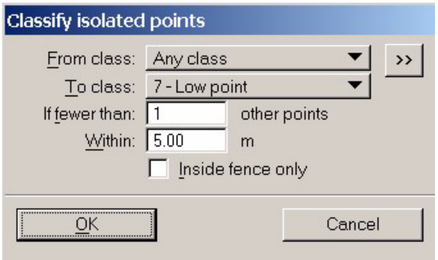
Closeby points

Not Lite

Isolated points

Not Lite

Isolated points routine classifies points which do not have very many neighbor points within a 3D search radius. This routine is useful for finding isolated points up in the air or below the ground.



Setting:	Effect:
From class	Source class(es) from which to classify points. Click the >> button to select multiple source classes.
To class	Target class to classify points into.
If fewer than	A point is classified if there are less than the given number of neighbouring points within a 3D search radius. Normally 1 - 5.
Within	3D search radius. Normally 2.0 - 10.0 m.
Inside fence only	If on, points inside a fence or selected polygon are classified.

Low points

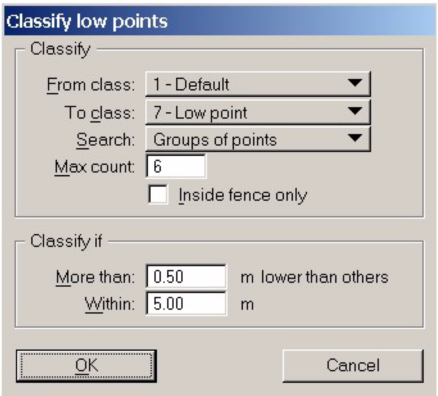
Not Lite

Low points routine classifies points which are lower than other points in the vicinity. It is often used to search for possible error points which are clearly below the ground.

This routine will basically compare the elevation of each point (=center) with every other point within a given xy distance. If the center point is clearly lower than any other point, it will be classified.

Sometimes you may have a higher density of error points. If there are several error points close to each other, those will not be detected if searching for single low points. However, this routine can also search for groups of low points where the whole group is lower than other points in the vicinity.

The routine finds the lowest point or group of points in a data set when it runs once. If there are low points on several elevation levels, this routine should be executed several times with different settings.



Setting:	Effect:
From class	Source class from which to classify points.
To class	Target class to classify points into.
Search	Whether to search for groups or single points.
Max count	Maximum size of a group of low points.
Inside fence only	If on, points inside a fence or selected polygon are classified.
More than	Minimum height difference. This is normally 0.3 - 1.0 m.
Within	Xy search range. Each point is compared with every other point less than **Within** distance away. Normally 2.0 - 8.0 m.

Ground

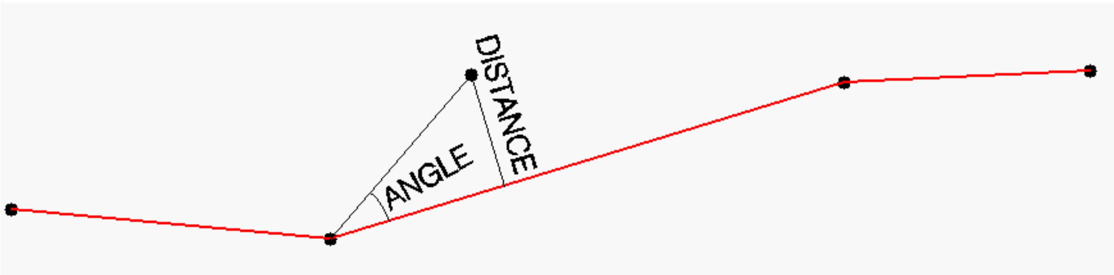
Not Lite

Ground routine classifies ground points by iteratively building a triangulated surface model.

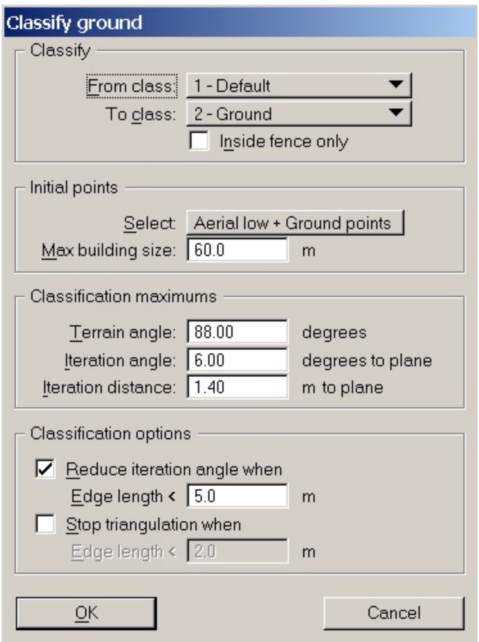
The routine starts by selecting some local low points that are confident hits on the ground. You control initial point selection with the **Max building size** parameter. If maximum building size is 60.0 m, the application assumes that any 60 by 60 m area will have at least one hit on the ground (provided there are points around different parts of the area) and that the lowest point is a ground hit.

The routine builds an initial model from the selected low points. Triangles in this initial model are mostly below the ground with only the vertices touching ground. The routine then starts molding the model upwards by iteratively adding new laser points to it. Each added point makes the model following the ground surface more closely.

Iteration parameters determine how close a point must be to a triangle plane for being accepted as ground point and added to the model. **Iteration angle** is the maximum angle between a point, its projection on triangle plane and the closest triangle vertex. **Iteration distance** parameter makes sure that the iteration does not make big jumps upwards when triangles are large. This helps to keep low buildings out of the model.



The smaller the **Iteration angle**, the less eager the routine is to follow changes in the point cloud (small undulations in terrain or hits on low vegetation). Use a small angle (close to 4.0) in flat terrain and a bigger angle (close to 10.0) in hilly terrain.



Setting:	Effect:
From class	Source class from which to classify points.
To class	Target class to classify points into.
Inside fence only	If on, points inside a fence or selected polygon are classified.
Select	Selection of initial ground points. If starting a new ground classification, use **Aerial low + Ground points**. Use **Current ground points** when you want to continue iteration in a fenced, previously classified area.
Max building size	Edge length of largest buildings.
Terrain angle	Steepest allowed slope in ground terrain.
Iteration angle	Maximum angle between point, its projection on triangle plane and closest triangle vertex. Normally between 4.0 and 10.0 degrees.
Iteration distance	Maximum distance from point to triangle plane during iteration. Normally between 0.5 and 1.5 m.
Reduce iteration angle when	If on, reduce eagerness to add new points to ground inside a triangle when every edge of triangle is shorter than **Edge length**. Helps to avoid adding unnecessary point density to the ground model and reduces memory requirement.
Stop triangulation when	If on, quit processing a triangle when every edge in triangle is shorter than **Edge length**. Helps to avoid adding unnecessary point density to the ground model and reduces memory requirement.

Add point to ground

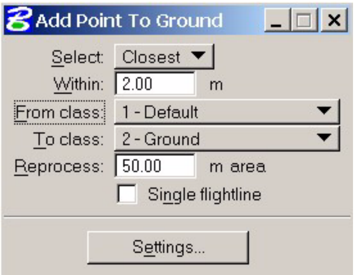
Add point to ground menu command lets you classify points inside a certain area to the ground class. The command classifies points to the ground class based on one initial point and additional settings defined in the command's setting dialog. The source class from which the points are added to ground has to be visible in the MicroStation view.

The command might be useful to correct classification errors effectively in areas where the automatic ground classification does not provide a good result.

➢ **To add points to the ground:**

1. Select **Add point to ground** command from **Classify** pulldown menu.

   This opens the **Add Point to Ground** dialog:

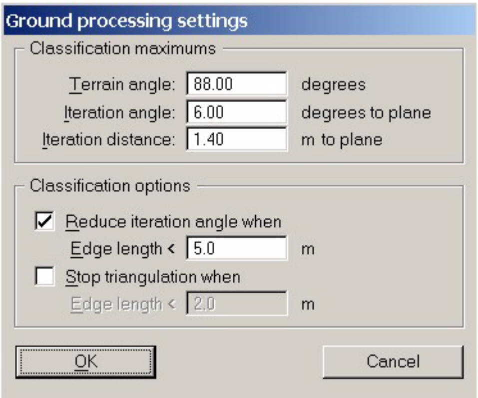


2. Define settings for the search for additional ground points.

Setting:	Effect:
Select	Defines which point is selected as initial point for the ground classification: • Closest - the point closest to the mouse click. • Highest - the highest point within the search radius. • Lowest - the lowest point within the search radius.
Within	Search radius around the mouse click to find the initial point.
From class	Source class from which points are classified into ground.
To class	Target class to add points to.
Reprocess	Area within which points are classified.
Single flightline	If on, only points from one flightline at a time are classified.

3. (Optional) Click **Settings** to change settings for the ground processing parameters.

   This opens the **Ground processing settings** dialog:



   The settings are the same as for the automatic ground classification. See **Ground** classification routine for more information.

4. Click inside a view to define the initial location for adding ground points.

   This classifies visible points from the source class to the ground class according to the given parameters and within the defined reprocessing area.

Hard surface

Not Lite

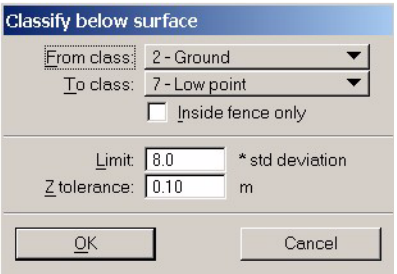
Below surface

Not Lite

Below surface routine classifies points which are lower than neighboring points in the source class. This routine can be run after ground classification to locate points which are below the true ground surface.

The algorithm for this routine can be outlined as follows:

- For each point (=central point) in the source class, find up to 25 closest neighboring source points.
- Fit a plane equation to the neighboring points.
- If the central point is above the plane or less than **Z tolerance** below, it will not be classified.
- Compute standard deviation of the elevation differences from the neighboring points to the fitted plane.
- If the central point is more than **Limit** times standard deviation below the plane, classify it into the target class.



Setting:	Effect:
From class	Source class from which to classify points. Normally ‘Ground’.
To class	Target class to classify points into.
Inside fence only	If on, points inside a fence or selected polygon are classified.
Limit	Factor for multiplication with standard deviation of the elevation differences of neighbouring source points. A point has to be more than **Limit * std.-dev.** below the fitting plane to be classified.
Z tolerance	Minimum elevation difference a point has to be below surface to be classified.

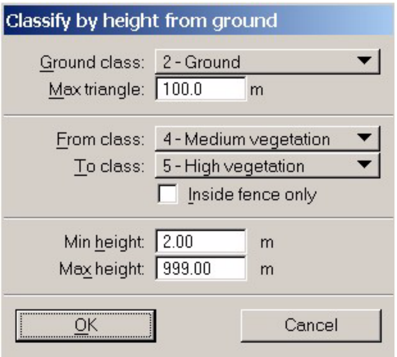
By height from ground

Not Lite

By height from ground routine classifies points which are within a given height range compared to the ground points surface model. The routine requires that you have already classified ground points successfully.

This routine will build a temporary triangulated surface model from ground points and compare other points against the elevation of the triangulated model.

You might use this routine to classify points into different vegetation classes for preparing building classification, powerline processing or tree detection. As a result, the highest vegetation class should include all hits on the target objects of interest (building roofs, wires and towers, or trees).

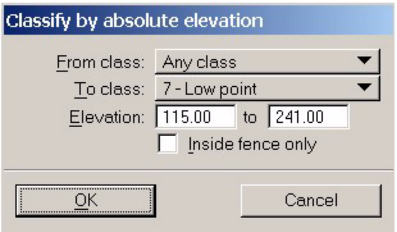


Setting:	Effect:
Ground class	Point class into which ground points have been classified before.
Max triangle	Maximum length of a triangle edge in the temporary calculated surface model.
From class	Source class from which to classify points.
To class	Target class to classify points into.
Inside fence only	If on, points inside a fence or selected polygon are classified.
Min height	Start of height range above ground surface.
Max height	End of height range above ground surface.

By absolute elevation

Not Lite

By absolute elevation routine classifies points which are within a given elevation range. This can be used to classify error points high up in the air or clearly below the ground.

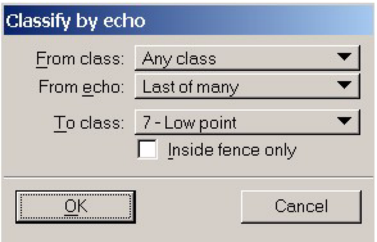


Setting:	Effect:
From class	Source class from which to classify points.
To class	Target class to classify points into.
Elevation	Absolute elevation range within which points are classified.
Inside fence only	If on, points inside a fence or selected polygon are classified.

By echo

Not Lite

By echo routine classifies points based on echo information.

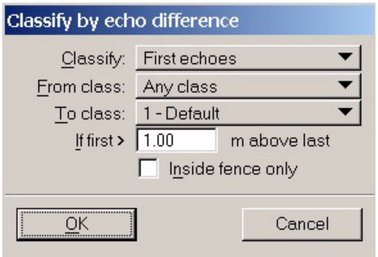


Setting:	Effect:
From class	Source class from which to classify points.
From echo	Echo type from which to classify points: • Only echo - only echo from a pulse. • First of many - first echo from a pulse which produced at least two echoes. • Intermediate - intermediate echo from a pulse which produced at least three echoes. • Last of many - last echo from a pulse which produced at least two echoes. • Any first - combination of Only echo and First of many • Any last - combination of Only echo and Last of many • First ... Seventh - points with a specific echo number. This requires LAS file format.
To class	Target class to classify points into.
Inside fence only	If on, points inside a fence or selected polygon are classified.

By echo difference

Not Lite

By echo difference routine classifies points based on elevation difference between first and last echoes. The classification affects only points with the echo types ‘first of many’ or ‘last of many’.



Setting:	Effect:
Classify	Echo type from which to classify points: **First echoes** or **Last echos**.
From class	Source class from which to classify points.
To class	Target class to classify points into.
If first >	Elevation difference between first and last echos. A point is classified, if the difference is larger than the given value.
Inside fence only	If on, points inside a fence or selected polygon are classified.

By echo length

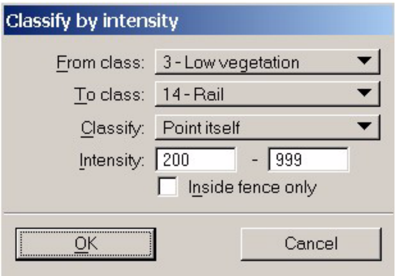
Not Lite

By intensity

Not Lite

By intensity routine classifies points which have an intensity value (strength of return) within a given range. The intensity value is affected by the type of surface material hit by the laser beam.

This can be used to classify points which are possible hits on railroad rails or on road paint markings because the metal surface or the white paintings produce high intensity returns while the surroundings like dark gravel or asphalt often result in low intensity returns.



Setting:	Effect:
From class	Source class from which to classify points.
To class	Target class to classify points into.
Classify	Determines which points are classified: • Point itself - points with an intensity value in the given range are classified. • Later echos - last and intermediate echo points are classified if the intensity value of the first echo point of the same pulse is within the given range. First and only echo points are not affected with this setting.
Intensity	Range of absolute intensity values within which points are classified.
Inside fence only	If on, points inside a fence or selected polygon are classified.

By color

Not Lite

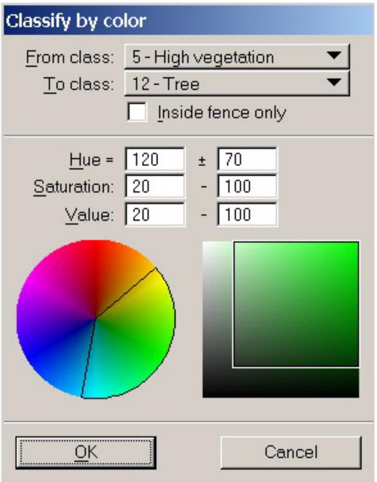
By color routine classifies points based on the color value stored for each point. Classification by color can help in solving classification tasks which would be very difficult relying on xyz positions only.

The routine will classify points in the source class whose color falls into a specified HSV color range. HSV color model consists of three parameters:

- Hue - pure color. This can be red, yellow, green, cyan, blue or magenta.
- Saturation - purity of the color or how much grey is mixed in.
- Value - brightness of the color.

The classification routine can use color values stored with the laser points. These color values have to be derived beforehand from images using **Extract color from images** command. Source images can be raster references attached to TerraPhoto or raw images stored in a TerraPhoto image list. The colors are stored as RGB values for each laser point but corresponding HSV values can be displayed in the TerraScan **Main** window point list.

After the extraction of colors, you may check the distribution of color values for the objects that you want to classify by color. This can be done using the **Identify** button in TerraScan **Main** window. In the **Main** window, the attribute **Point color HSV** should be switched on for display. Select different laser points to check the HSV color values. You can also run classification by color multiple times and test different parameter settings.



Setting:	Effect:
From class	Source class from which to classify points.
To class	Target class to classify points into.
Inside fence only	If on, points inside a fence or selected polygon are classified.
Hue	Hue value and tolerance.
Saturation	Minimum and maximum value for saturation component.
Value	Minimum and maximum value for value component.

✍ You can effectively classify by grey scale if you set **Hue** tolerance to +- 180. All hue values will then fall within the range.

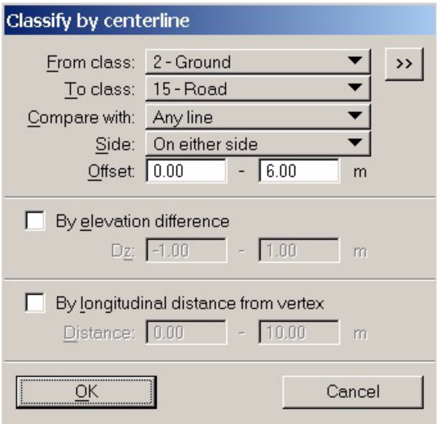
By centerline

Not Lite

By centerline routine classifies points based on how close they are to a linear element. You can use it to classify points which are close to or far away from a linear element.

You must select the centerline element(s) before starting the tool. Centerline elements can be MicroStation lines, line strings, arcs, shapes, or complex elements consisting these element types.

If multiple elements are selected, each laser point will be classified according to the offset distance to either the closest linear element or to any of the selected linear elements.



Setting:	Effect:
From class	Source class(es) from which to classify points. Select multiple classes by clicking the >> button.
To class	Target class to classify points into.
Compare with	Method of assigning points to linear elements: • Any line - points are classified if they are within the offset distance of any linear element. • Closest line - points are classified if they are within the offset distance of the closest linear element.
Side	Side on which to classify points: **On left side**, **On either side** or **On right side**. Direction is relative to the digitization direction of the linear element.
Offset	Minimum and maximum side offset. Points within the offset range are classified.
By elevation difference	If on, classifies only point which are within a given elevation distance range from the linear element.
By longitudinal distance from vertex	If on, classifies only points which are within a given longitudinal distance range from the closest element vertex.

By section template

Not Lite

By section template routine classifies points based on the shape of a tunnel defined by a three dimensional alignment and a tunnel section template.

This classification routine is primarily used with laser data from terrestrial scanners.

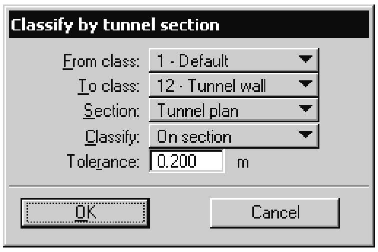
Defining section templates

You can define tunnel sections in the **Section templates** of TerraScan user settings.

Classifying by section template

The tunnel section definition can be used to classify points which are inside the tunnel, outside the tunnel or close to the cross section of the tunnel. You will first need to create a three dimensional vector in the design file which runs along the tunnel. The cross section template of the tunnel will be linked to the three dimensional alignment.

You must select the centerline element before starting the tool.

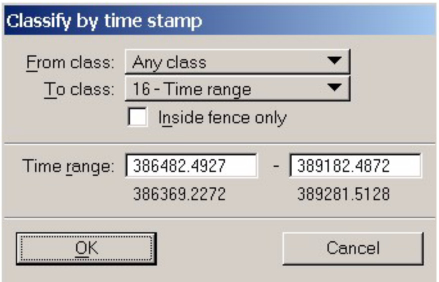


Setting:	Effect:
From class	Source class from which to classify points.
To class	Target class to classify points into.
Section	Tunnel section type defined in user settings.
Classify	Type of points to classify: • Inside section - points inside the section. • Outside section - points outside the section. • On section - points close to the section shape.
Tolerance	Defines how close to the section shape a point has to be in order to be classified. This is only active if **Classify** is set to **On section**.

By time stamp

Not Lite

By time stamp routine classifies points within a specified time range. This requires time stamps stored for each laser point.



Setting:	Effect:
From class	Source class from which to classify points.
To class	Target class to classify points into.
Inside fence only	If on, points inside a fence or selected polygon are classified.
Time range	Range of time stamps within which points are classified. The numbers below show the minimum and maximum time stamp of the data set.

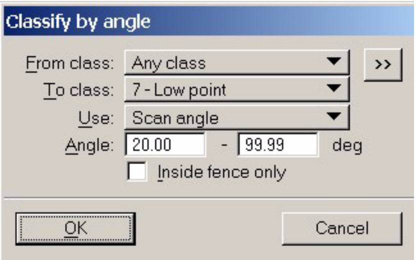
By angle

Not Lite

By angle routine classifies points according to the scan angle or the angle from vertical.

Scan angles range from -128 and +127 degree and are stored in LAS files. If TerraScan binary files are used, a scan angle is computed from the trajectories. Angles from vertical range from 0 to 90 degree.

Classify by angle may be useful to classify points in a small corridor, e.g. between -20 and +20 degree or to remove edges in the data set from further processing.

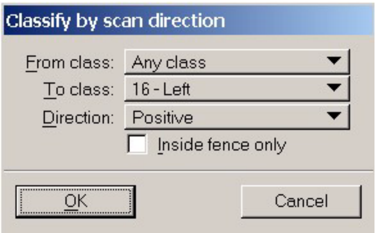


Setting:	**Effect:**
From class	Source class(es) from which to classify points.
To class	Target class to classify points into.
Use	Type of angle used for classification: **Scan angle** or **Angle from vertical**.
Angle	Range of angle values within which points are classified.
Inside fence only	If on, points inside a fence or selected polygon are classified.

By scan direction

Not Lite

By scan direction routine classifies points in negative or positive scan direction, or edge points. This requires laser data in LAS file format. Classification is based on the bit fields present in LAS files.



Setting:	Effect:
From class	Source class from which to classify points.
To class	Target class to classify points into.
Direction	Scan direction from which to classify points: **Negative**, **Positive** or **Edge points**.
Inside fence only	If on, points inside a fence or selected polygon are classified.

By normal vector

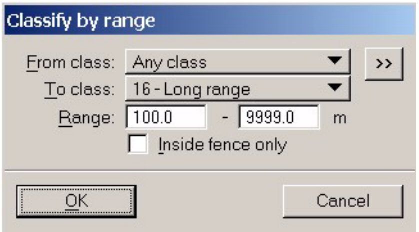
Not Lite

By range

Not Lite

By range routine classifies points based on the measurement distance which is the distance of a point from the scanner. This classification routine is primarily used with laser data from mobile scanners.

Classify by range is useful to exclude points from the data set that are located far away from the scanner and thus not suited for matching tasks. It should be used before processes for drive path matching are started.



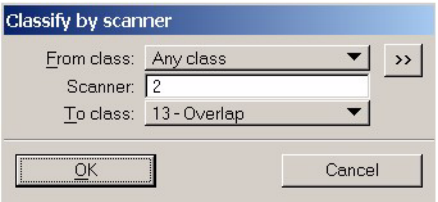
Setting:	Effect:
From class	Source class(es) from which to classify points. Click the >> button to select multiple source classes.
To class	Target class to classify points into.
Range	Measurement distance range within which points are classified.
Inside fence only	If on, points inside a fence or selected polygon are classified.

By scanner

Not Lite

By scanner routine classifies points according to the scanner number of the scanner which recorded a point. This classification routine is primarily used with laser data from mobile scanners and requires laser data in LAS file format.

Classify by scanner is required for the calibration of a mobile scanner system using more than one scanner. The classification separates points recorded by different scanners into different classes which enables the correction of misalignment between the scanners.



Setting:	Effect:
From class	Source class(es) from which to classify points. Click the >> button to select multiple source classes.
Scanner	Scanner number of the scanner from which to classify points.
To class	Target class to classify points into.

Railroad

Not Lite

Railroad routine classifies points which match the elevation pattern of a railroad rail.

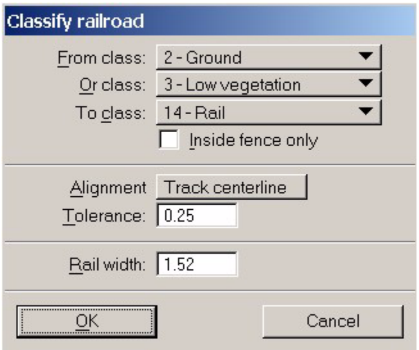
For a point to match elevation pattern, it:

- must have some points in the vicinity which are 0.05 - 0.35 m lower.
- can not have any points in the vicinity which would be 0.15 - 2.50 m higher.
- can not have any points in the vicinity which would be more than 0.35 m lower.

Railroad classification can use an alignment element in three different ways:

- **None** - No alignment element. Find all points which match the elevation pattern and have another matching point at **Rail width** ± **Tolerance** distance in any direction.
- **Track centerline** - Alignment element follows the centerline between two rails. Find all points which match the elevation pattern and are at **(0.5 * Rail width)** ± **Tolerance** offset from centerline.
- **General direction** - Alignment element defines the general direction of the railroad. Find all points which match the elevation pattern and have another matching point at **Rail width** ± **Tolerance** distance. The line between the two points must be close to perpendicular with alignment direction.

Alignment elements can be created using the *Place Railroad String* tool in TerraScan or any other line placement tool from MicroStation. The alignment element(s) must be selected before starting the classification tool.

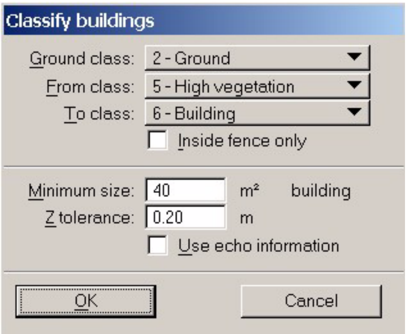


Setting:	Effect:
From class	Source class from which to classify points.
Or class	Second source class from which to classify points.
To class	Target class to classify points into.
Inside fence only	If on, points inside a fence or selected polygon are classified.
Alignment	Centerline usage: **None**, **Track centerline** or **General direction**.
Tolerance	Vertical distance about which the rail width can differ.
Rail width	Distance from rail center to rail center.

Buildings

Not Lite

Buildings routine classifies points on building roofs which form some kind of a planar surface. This routine requires that you have classified ground points before. It is also advisable to classify low vegetation so that only points more than two meters above the ground will be considered as possible building points.



Setting:	Effect:
Ground class	Class into which ground points have been classified before.
From class	Source class from which to search building points.
To class	Target class to classify points into.
Inside fence only	If on, points inside a fence or selected polygon are classified.
Minimum size	Smallest building footprint size in square meters.
Z tolerance	Approximate elevation accuracy of laser points.
Use echo information	If on, echo information of laser points is considered for the process. This can support the classification because points on roofs mostly belong to the echo type ‘only echo’ where as trees have a lot of ‘first of many’ and ‘intermediate’ echoes.

Model keypoints

Not lite

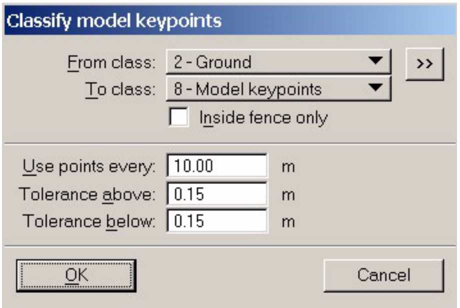
Model keypoints routine classifies laser points which are needed to create a triangulated surface model of a given accuracy. This routine is normally used to create a thinned data set from points classified as ground hits.

You control the accuracy with elevation tolerance settings **Above model** and **Below model**. These settings determine the maximum allowed elevation differences from a ground laser point to a triangulated model. **Above model** determines how high laser points can be above the model. **Below model** determines how low laser points can be below the model.

The application will try to find a relatively small set of points (= keypoints) which would create a triangulated model of the given accuracy. These keypoints will be classified into another class. All remaining ground points are within the given elevation tolerances from a model that the keypoints would produce when triangulated. Some of the ground points are above the model, some ground points are below.

This classification is an iterative process similar to ground classification. The process starts by searching for initial points inside rectangular regions of a given size. The lowest and the highest source point inside each rectangle is classified as keypoint and those are used to create an initial triangulated model. During each iteration loop the routine searches for source points which are more than the given tolerance above or below the current model. If such points are found, the highest or lowest points are classified and added to the model.

The **Use points every** setting provides a method for ensuring a minimum point density in the final model even in flat places. For example, if you want to have at least a point every 10 meters, you should set the **Use points every** setting to 10.0.



Setting:	Effect:
From class	Source class(es) from which to search keypoints. Click the >> button to select multiple source classes.
To class	Target class to classify keypoints into.
Inside fence only	If on, points inside a fence or selected polygon are classified.
Use points every	Rectangle size for searching initial points. The highest and the lowest point inside each rectangle will be classified.
Tolerance above	Maximum allowed elevation difference from the triangulated model to a source point above the model.
Tolerance below	Maximum allowed elevation difference from the triangulated model to a source point below the model.

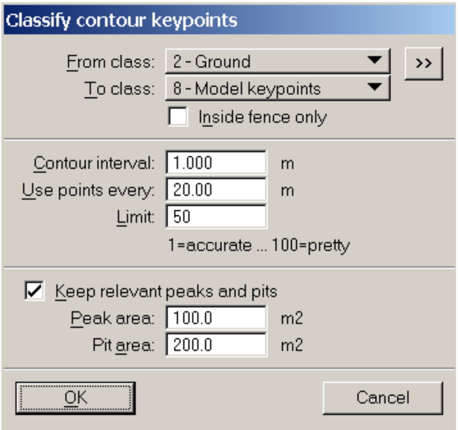
Contour keypoints

Contour keypoints routine classifies model keypoints tailored for the production of contours. It works similar to the **Model keypoints** routine but produces a ground model which is suited to create smooth, cartographic contours for map production.

When contours are derived from original laser data, they are very accurate, detailed and mathematically correct but less suited for being used for representation purposes. Since the small details in contour lines are caused by laser points close to the contour elevation, those points are removed from the model formed by contour keypoints.

The user controls the volumetric difference to the true ground model by setting a **Limit** value for contour keypoints classification. The value determines how much the contours derived from the model will be smoothed. So higher the limit value, the smoother, nicer but less accurate the contours will be compared with the true ground model.

In addition to contours, the classification routine also considers peaks and pits in the elevation model. The values for **Peak area** and **Pit area** are used to define the minimum area that is enclosed by a contour line on top of hills or in depressions.



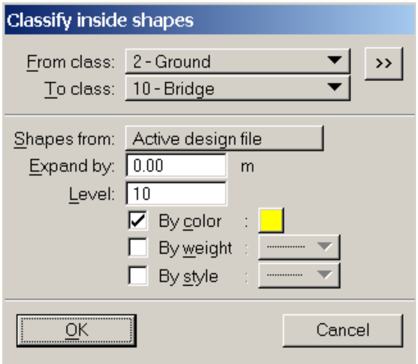
Setting:	Effect:
From class	Source class(es) from which to search keypoints. Click the >> button to select multiple source classes.
To class	Target class to classify keypoints into.
Inside fence only	If on, points inside a fence or selected polygon are classified.
Contour interval	Interval of contours.
Use points every	Rectangle size for searching initial keypoints. The highest and the lowest point inside each rectangle will be classified
Limit	Determines the smoothness of the contours.
Keep relevant peaks and pits	If on, areas on top of hills and in depressions are considered in the model.
Peak area	Minimum size of a peak area enclosed by a contour line.
Pit area	Minimum size of a pit area enclosed by a contour line.

Inside shapes

Not lite

Inside shapes routine classifies points that are located inside a shape on a specified level in the active design file or in a reference design file. Optionally, the shape can be further defined by color, weight or style settings. The shapes have to be drawn before the routine can be used.

The routine works similar to **Inside fence** command in TerraScan **Main** window but it is only available as a macro step. Therefore the command is not in the **Routine** list of the **Classify** pull-down menu. See **Macros** on page 429 for more information about macro creation and macro actions.



Setting:	**Effect:**
From class	Source class(es) from which to search building points. Click the >> button to select multiple source classes.
To class	Target class to classify points into.
Shapes from	Source for shapes: • Active or reference files - shapes are defined in the active design file or in an attached reference file. • Active design file - shapes are defined in the active design file. • Reference file - shapes are defined in an attached reference file.
Expand by	Distance for expanding the shapes.
Level	Design file level on which the shapes are placed.
By color	If on, only shapes with the given color are used.
By weight	If on, only shapes with the given line weight are used.
By style	If on, only shapes with the given line style are used.

Wire danger points

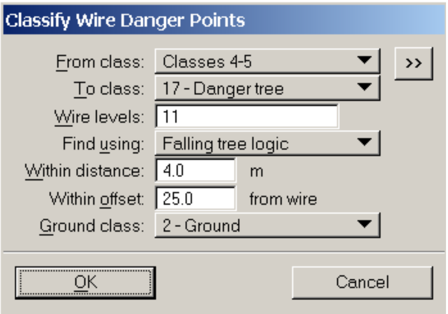
Not lite

Wire danger points routine classifies laser points which are within a given three dimensional distance limit from vectorized catenaries. This routine is related to powerline processing which is explained in detail in chapter **Coordinate Transformations** on page 312. It requires the vectorization of wires before the routine can be used.

There are three different methods how danger points can be defined:

- **Vertical distance to wire -** danger points are searched within a 3D radius around each wire and within a vertical distance from the wire.
- **3D distance to wire -** danger points are searched within a 3D radius around each wire.
- **Falling tree logic -** each vegetation point is considered as a falling tree and classified if the point is travels too close to the wire when falling down as a tree. The falling tree computation treats each vegetation point as a tip of a tree with the trunk at the xy location of that vegetation point.

The routine works similar as classification part of the *Find Danger Objects* tool in TerraScan **View Powerline** tools. The routine is only available as a macro step, therefore the command is not in the **Routine** list of the **Classify** pulldown menu. See **Macros** on page 429 for more information about macro creation and macro actions.

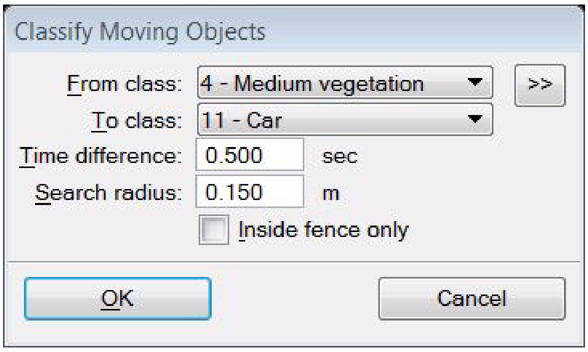


Setting:	Effect:
From class	Source class(es) from which to search keypoints. Click the >> button to select multiple source classes.
To class	Target class to classify keypoints into.
Wire levels	Design file level(s) on which the vectorized wires are placed. Separate different levels by comma.
Find using	Method of wire danger point definition: **Vertical distance to wire**, **3D distance to wire** or **Falling tree logic**.
Within distance	3D radius around a wire used as search distance for danger points.
Within offset	Vertical distance from a wire used in **Vertical distance to wire** and **Falling tree logic** methods to search for danger points.
Ground class	Class into which laser points on ground are classified. This is only active for **Falling tree logic** method.

Moving objects

Not lite

Moving objects routine classifies groups of points.



Setting:	Effect:
From class	Source class(es) from which to classify points.
	Opens the **Select classes** dialog which contains the list of active classes in TerraScan. You can select multiple source classes from the list that are then used in the **From class** field.
To class	Target class of classified points.
Time difference	
Search radius	
Inside fence only	If on, points inside a fence or selected polygon are classified. This requires a fence element or a selected polygon in the design file.

17 Macros

Macros provide a method to automate processing steps. The best level of automation is reached when using macros together with a project definition. See chapter **Coordinate Transformations** on page 312 for information about projects in TerraScan.

Macros consist of a number of processing steps which are executed one after the other. Processing steps can classify points, modify points, delete points, transform points, output points, update views, execute commands or call functions from other MDL applications.

Macros are stored as text files with the default extension '*.mac'.

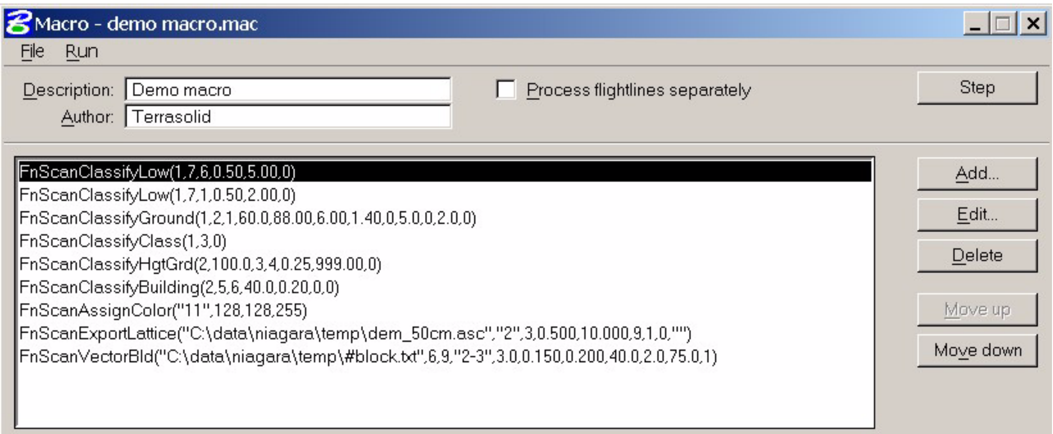
Creating a macro

A macro can be created in TerraScan using the **Macro** command from **Tools** menu in the TerraScan **Main** window. The **Macro** dialog lets you add, modify, delete and arrange processing steps, and offers the possibilities to save and run macros. Each line in the macro dialog represents one processing step that is performed on the laser points when running a macro.

➢ **To create a macro:**

1. Choose **Macro** command from **Tools** pulldown menu in the **Main** window.

   This opens the **Macro** dialog:



2. Enter description of the macro in the **Description** field.
3. Enter your name in the **Author** field.
4. Click **Add** to add a new processing step to the macro.

   This opens a dialog for defining the processing step and related settings. See more information in **Macro actions** on page 431.

5. Define settings for the processing step and click OK.

   You can continue to step 4 until all the steps are complete.

6. Choose **Save as** from **File** pulldown menu to save the macro to a file.

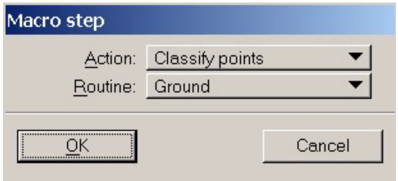
Setting:	**Effect:**
Description	Free description of the macro.
Author	Name of the author.
Process flightlines separately	If on, the macro steps are performed for each flightline separately.
Step	By clicking this button, the macro is executed step-by-step on loaded points.

Setting:	**Effect:**
Add	Add a new processing step to the macro.
Edit	Modify settings for a selected processing step.
Delete	Delete a selected processing step.
Move up	Moves a selected processing step one line up in the macro.
Move down	Moves a selected processing step one line down in the macro.

To:	**Choose menu command:**
Start to create a new macro	**File / New**
Open an existing macro file	**File / Open**
Save changes to an existing macro file	**File / Save**
Save a macro into a new file	**File / Save As**
Run a macro on loaded points	**Run / On loaded points**
Run a macro on selected files	**Run / On selected files**

Macro actions

When the **Add** button in the Macro dialog is pressed, the **Macro step** dialog opens:



The **Action** field includes a list of all possible macro actions. Depending to the chosen action, another dialog opens to define the settings for a specific processing step.

Classify points action requires the selection of a classification routine in the **Routine** field. The routines are basically the same as in the **Routine** sub-menu in TerraScan **Main** window which can be performed on loaded points. However, a few classification routines can be executed in macros only.

Actions that are executable in a macro are listed in the following table. For most of the actions, there is a corresponding tool or menu command for processing loaded points in TerraScan.

To:	Choose action:	Description:
Apply a classification routine	**Classify points**	**Classification Routines**
Call an Addon routine	**Addon command**	**Addon**
Adjust misalignment angles	**Adjust angles**	**Adjust laser angles**
Apply TerraMatch corrections to laser data	**Apply corrections**	**Apply corrections**
Assign a color to laser points	**Assign color**	**Assign color to points**
Assign groups to laser points	**Assign groups**	**Assign groups**
Create an elevation model in TerraModeler	**Create model**	**Create Editable Model**
Add a comment to a macro	**Comment**	**Comment**
Compute normal vectors for laser points	**Compute normal vectors**	**Compute normal vectors**
Compute parameters of road cross sections	**Compute section parameters**	**Compute section parameters**
Compute slopes on road surfaces	**Compute slope arrows**	**Compute slope arrows**
Convert time stamps of laser data	**Convert time stamps**	**Convert time stamps**
Cut points from overlapping flightlines	**Cut overlap**	**Cut overlap**
Assign line number from trajectories to laser points	**Deduce line numbers**	**Deduce using time**
Delete points of a class	**Delete by class**	**Delete by point class**
Delete points of a flightline	**Delete by flightline**	**Delete by flightline**
Call a function from another MDL application	**Evaluate expression**	**Evaluate expression**
Call a TerraScan function	**Execute command**	**Execute command**
Export a lattice model	**Export lattice**	**Export lattice model**
Detect paint markings from laser data	**Find paint lines**	**Find paint lines**
Fix the elevation of laser points	**Fix elevation**	**Fix elevation**
Run a keyin command in MicroStation	**Keyin command**	**Keyin command**

To:	Choose action:	Description:
Modify flightline numbers	**Modify line numbering**	**Modify numbering**
Save points in new files	**Output points**	**Output points**
Save points by flightline	**Output by flightline**	**Output by flightline**
Save points in original file	**Save points**	**Save points**
Manipulate intensity values	**Scale intensity**	**Scale intensity**
Smooth the elevation of points	**Smoothen points**	**Smoothen points**
Sort points	**Sort points**	**Sort**
Thin points to a lower density	**Thin points**	**Thin points**
Apply a transformation to the laser points	**Transform points**	**Transform loaded points**
Update a MicroStation view	**Update views**	**Fit view**
Vectorize buildings automatically	**Vectorize buildings**	**Vectorize buildings**
Create a section points data set	**Write section points**	**Write section points**
Write points into design file	**Write to design**	**Write to design file**

Apply corrections

Comment

Compute normal vectors

Compute section parameters

Compute slope arrows

Convert time stamps

Evaluate expression

Execute command

Find paint lines

Keyin command

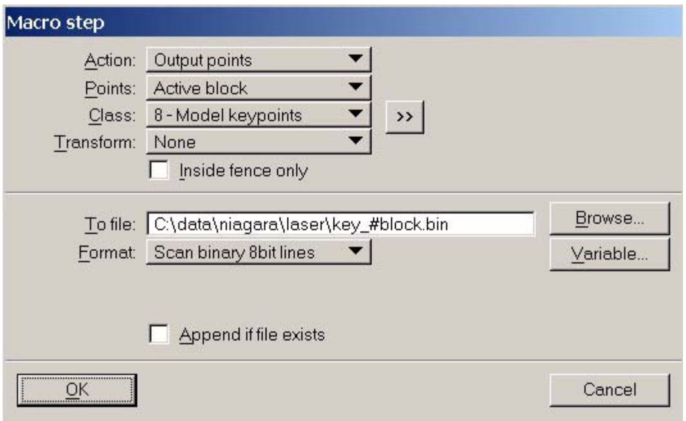
Output points

Output points macro step can be used to output selected points to a new file. A typical example is a macro which:

- classifies model keypoints
- output model keypoints only

Such a macro can be run to extract model keypoints from all project blocks to a single file or to individual files for each block.

The **Output points** macro step would have the following settings:

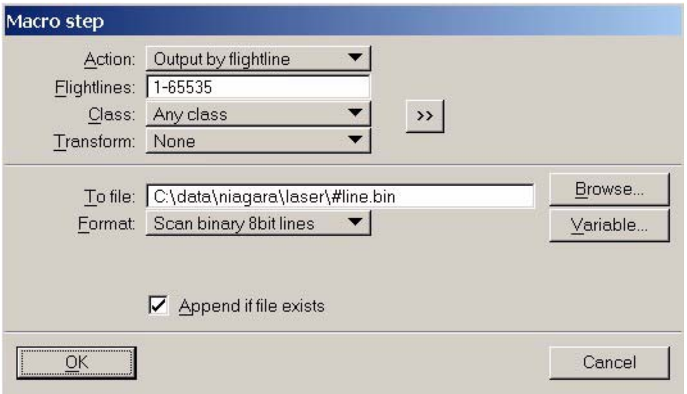


Setting:	Effect:
Action	Output points action.
Points	Source points for output: **All points** which would include neighbour points or **Active block**.
Class	Point class to output.
Transform	Coordinate transformation to apply when computing output coordinates for a point.
Inside fence only	If on, output points inside fence only.
To file	Name of output file. If it does not include a directory, project data directory is used.
Variable	Add a variable to the output file name: • #bnumber - block number. • #block - block file name. • #bpath - block file path. • #bemin - block minimum easting. • #bnmin - block minimum northing. • #bemax - block maximum easting. • #bnmax - block maximum northing. • #pdir - project data directory. • #name - active file name. Use this to replace the file name component when running a macro on selected files.
Format	Format of the output file.
Delimiter	Delimiter character when outputting to a text file format.
Xyz decimals	Number of decimal places when outputting to a text file format.
Append if file exists	Set this on, if you want to output all input files to a single file. Set this off, if every input file should generate an individual output file and you have used a variable as part of the file name specification.

Output by flightline

Output by flightline macro step can be used to divide points into output files by flightline number.

You would typically use this to generate final output for a customer who wants have each flightline in a separate file.



Setting:	Effect:
Action	Output by flightline action.
Flightline	Number of flightline to output. A value of -1 outputs all flightlines and generates one output file for each flightline.
Class	Point class to output.
Transform	Coordinate transformation to apply when computing output coordinates for a coordinate.
To file	Name of output file. If **Flightline** is set to output all flightlines, the file name should include the **Variable** ‘#line’ which will be replaced by the number of each flightline. See other variables in **Output points** section.
Format	Format of the output file.
Delimiter	Delimiter character when outputting to a text file format.
Xyz decimals	Number of decimal places when outputting to a text file format.
Append if file exists	Set this on, if you want to output all blocks to a single file. Set this off, if every block should generate an individual file.

Scale intensity

Vectorize buildings

Building vectorization can be also performed on project level by using a macro that includes the **Vectorize Buildings** command as macro step. The main difference in the process is that the building models are not drawn into the design file directly. Instead, a text file is created for each processed project block including the corner coordinates for all building model shapes. These text files can be loaded into TerraScan after the vectorization process is finished.

➢ **To vectorize buildings based on a TerraScan project:**

1. Create a macro using **Macro** command in TerraScan **Main** window.
2. Select **Vectorize Buildings** as **Action** to be added to the macro.

   This opens the Macro Step dialog which is basically the same as the **Vectorize Buildings** dialog described above.

3. Define settings for the vectorization task.
4. Define a location and naming scheme for storing the text files in the **Output files** field using the **Browse** and **Variable** buttons.
5. Save the macro.
6. Run the macro on the TerraScan project using the **Run macro** command from the **Project** window.

   This creates the text files for each block in the defined output directory.

7. Load the building models into TerraScan using the **Read / Building models** command from TerraScan **Main** window.

This reads the text files and draws the building models into the design file.

Write section points

Macro action for writing points suitable for viewing corridor object such as road.

Artificial coordinate system:

- X = scaled station
- Y = offset
- Z = scaled elevation

Advantages:

- Faster viewing as road is shortened
- Exaggerated elevation changes
- Easier panning – result runs along x axis
- Better sun shading – direction does not change

Run macros

Run a macro on loaded points

You can execute the active macro on the currently loaded points simply by choosing **On loaded points** command from **Run** pulldown menu in the **Macro** dialog.

You can also execute a macro by entering a key-in command such as:

```
scan run macro modelkey.mac
```

which would read 'modelkey.mac' macro from the directory specified by TSCAN_MACRODIR environment variable in **Configuration Variables** and execute that.

A macro can be executed step-by-step on loaded points using the **Step** button in the **Macro** dialog. This is useful to test the effect of the separate processing steps when creating a macro.

After a macro has been executed on loaded points, a report appears that lists all macro steps and a return value. The return value can be the amount of points that has been affected by the processing step or another value specifying the result of the processing step. If a processing step could not be executed, the line in the report appears in red color. A definition of the return values for each processing function is given in **Function Prototypes** on page 460.

The macro report can be saved as text file using the **Save as** command from **File** menu or printed using the **Print** command from **File** menu in the **Macro execution** dialog. The size of the dialog can be changed using commands from the **View** menu.

Run a macro on selected files

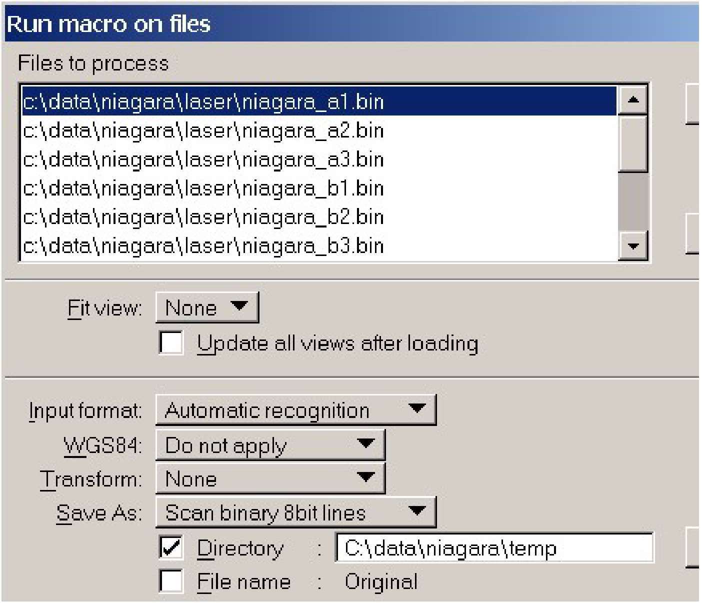
You can execute a macro on selected files in batch mode. You select a number of input files which the application will load one at a time, execute macro steps on the points extracted from the input and optionally writes the results to an output file. Additionally, different transformations can be applied to the files.

The input files can be in any format that TerraScan supports.

➢ **To run a macro on selected files:**

1. Choose **On selected files** command from **Run** pulldown menu in the **Macro** dialog.

   This opens the **Run macro on files** dialog:



   The dialog displays a list of files to process and related settings.

2. Click **Add** to add files into the list of files to process.

   This opens a TerraScan dialog for selecting multiple input files.

3. Add files to the list and click **Done**.

   You may continue with step 2 to add additional files.

4. Enter additional settings in the **Run macro on files** dialog and click OK to start the batch processing.

   The application starts processing input files one at a time. Your settings determine how the macro results are written out.

Setting:	Effect:
Add	Add files for processing.
Remove	Remove files from the list of files for processing.
Fit view	Views to fit after loading each file.
Update all views after loading	If on, updates all views after loading a file.
Input format	Definition of the input file format. Binary formats that are supported by TerraScan are usually recognized automatically.
WGS84	Definition of a projection system into which the points are transformed from WGS84 projections system. The projections system has to be enabled in TerraScan settings,

Setting:	Effect:
Transform	Definition of a transformation that is applied to the files. The transformation has to be defined in TerraScan settings.
Save As	Format of the output file into which all of the points will be written. Select **Do not save all points** option if you have included a more selective output action as one step in the macro.
Directory	If on, you can specify the output directory into which each file will be saved.
File name	If on, you can specify the output file name under which each input file will be saved. You may include the Variable'#name' as part of the file name. This will be replaced by the file name component of the active file. See more information about Variables is file names in **Output points** section.
Extension	If on, you can specify the output file extension.

After a macro has been executed on selected files, a report appears that lists all files and for each file the amount of loaded points, the executed macro steps and a return value as well as the amount of saved points. The return value can be the amount of points that has been affected by the processing step or another value specifying the result of the processing step. If a processing step could not be executed, the line in the report appears in red color. A definition of the return values for each processing function is given in **Function Prototypes** on page 460.

The macro report can be saved as text file using the **Save as** command from **File** menu or printed using the **Print** command from **File** menu in the **Macro execution** dialog. The size of the dialog can be changed using commands from the **View** menu.

Run a macro on a project

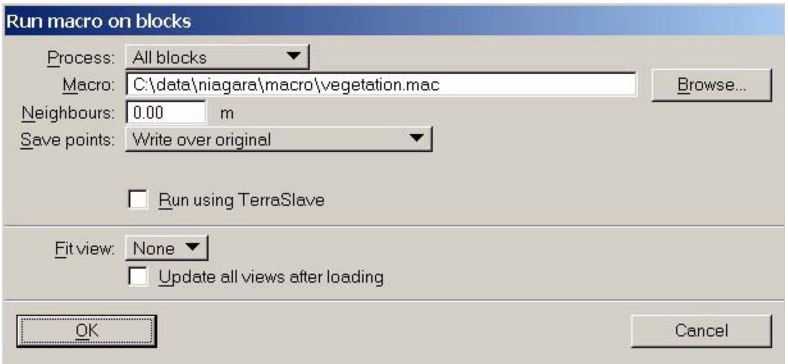
If a project is defined, a macro can be executed on selected block files or all block files of a project definition using the **Run macro** command from **Tool** menu in the **Project** window. See **Coordinate Transformations** on page 312 for detailed information about projects in TerraScan.

A macro can be performed on project level either by using TerraScan or by using TerraSlave. Using TerraSlave has the advantage that TerraScan and MicroStation are not blocked when a macro is processed. See **Run macros in TerraSlave** on page 453 for more information.

➢ **To execute a macro on blocks:**

1. (Optional) Select the desired block files in the **Project** window's file list if you want to execute the macro on selected blocks only.
2. Choose **Run macro** command from **Tools** pulldown menu in TerraScan **Project** window.

   This opens the **Run macro on blocks** dialog:



3. Define values for settings and click OK.

   This performs the macro on all or on selected blocks. The application will loop through the blocks, load the corresponding binary file, perform macro steps and optionally save the modified points back to the binary file.

Setting:	Effect:
Process	Blocks to process: **All blocks** or **Selected blocks**.
Macro	Macro file to execute.
Neighbours	Width of overlap region around the active block boundaries for which the application will load points from neighbouring blocks. See more information in **Neighbour points** section.
Save points	Method of saving the block files after processing: • Do not save - block files are not saved. This should be used if the macro includes an output step which writes the results to a new output file. • Write over original - original block files are overwritten directly by the processed files. • Temporary copy & replace original - a copy is created for each processed block file. TerraScan will move the temporary copies to replace the original block files at the end of the process. • Create new copy - a new file is created for each processed block file. Additionally, the project file is copied to the same directory where the new files are stored.

Setting:	Effect:
Temporary	Directory for storing the temporary copies of the block files if **Save points** is set to **Temporary copy & replace original**.
Write to	Directory for storing the new block files if **Save points** is set to **Create new copy**.
Run using TerraSlave	If on, the macro is performed by TerraSlave.
Fit view	Views to fit after loading each block.
Update all views after loading	If on, all views are updated after loading a block.

After a macro has been executed on a project, a report appears that lists all files and for each file the amount of loaded points, the executed macro steps and a return value as well as the amount of saved points. The return value can be the amount of points that has been affected by the processing step or another value specifying the result of the processing step. If a processing step could not be executed, the line in the report appears in red color. A definition of the return values for each processing function is given in **Function Prototypes** on page 460.

The macro report can be saved as text file using the **Save as** command from **File** menu or printed using the **Print** command from **File** menu in the **Macro execution** dialog. The size of the dialog can be changed using commands from the **View** menu.

Neighbour points

Neighbour points can be loaded additionally to the points of the active block to support classification steps. This should be used when performing processing steps that require neighboring points to ensure smooth transitions on block boundaries. Those steps include:

- ground classification
- model keypoint and contour keypoint classification
- building classification
- export lattice
- fix elevation
- smoothen points
- thin points
- vectorize buildings

For classification steps it is recommended to set the width of the neighbour points region to the same value as the maximum building size. Laser points from neighbour blocks do not belong to the active block and are not saved with the active block points.

Run macros in TerraSlave

TerraSlave is a program for executing batch processing tasks. It is included in a full TerraScan version but it can also run with an own license on a computer without TerraScan and MicroStation installed. At the moment, TerraSlave is able to execute macros on project level and it supports distributed computing. It can be launched from TerraScan **Run macro on blocks** dialog.

The main benefits of using TerraSlave are:

- MicroStation and TerraScan are immediately free for interactive work while a batch process is running.
- A separate executable can use more memory than a MicroStation based application.
- The software can queue several tasks for being executed one after the other without further interaction.
- It can utilize several computers in the network.
- Powerful servers can be used for batch processes.

Besides executing the processing steps, TerraSlave also includes a simple task management system. This includes the display of queued tasks, the ability to delete tasks from the queue, the display of active tasks (blocks, start time, processing machine), the ability to restart processing for a block, stop an ongoing process or end a failed process, as well as showing reports. See **TerraSlave quick guide** on page 455 for a short introduction.

When TerraSlave is started from TerraScan, it performs the following actions:

- TerraScan writes a task file (*.tsk) to \terra\tslave\queue and launches TerraSlave if not running already.
- TerraSlave checks \queue every 2 seconds for a new task and moves the .tsk file to \terra\tslave\task
- When TerraSlave works on a block, it writes a report into \terra\tslave\progress.
- When TerraSlave completes a block, it moves the report into \terra\tslave\report.
- When TerraSlave has completed the task, it moves the .tsk file from \task to \report.

The task file is a text file with the ending *.tsk consisting of all information that is required for TerraSlave to perform the task. The task file is named automatically: “<date>_<time>.tsk”. The report includes information about the processing steps performed on the blocks and a status remark. Reports are named in the same way as task files with the addition of a block number: “<date>_<time>_<blocknumber>.txt”.

There are some macro actions and processing options that do not work with TerraSlave. This includes:

- Start one task on computer 1 and start another task on computer 2 modifying the same data set.
- Run on another computer if local paths are used.
- Run on multiple computers if all blocks output to the same file (**Output points** or **Output by flightline** macro steps).
- Use **Inside fence only** setting in a macro step.
- Use **Key-in command** step.
- Use **Create model** step.

When a macro is created, the options for TerraSlave processing are stored in the macro file header:

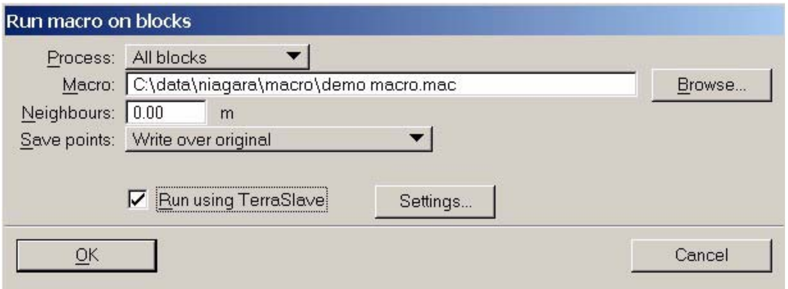
- **SlaveCanRun=1 -** Macro can be executed by TerraSlave.
- **AnotherComputerCanRun=1 -** Macro can be executed by TerraSlave on another computer.
- **CanBeDistributed=1 -** Macro can be executed by TerraSlave on multiple computers.

If one of these options is set to **0**, the option is not available due to the above listed restrictions for running macros in TerraSlave.

➢ **To run a macro in TerraSlave:**

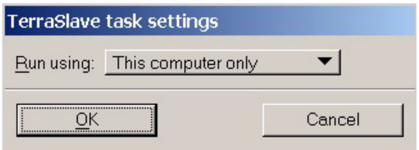
1. (Optional) Select the desired block files in the **Project** window's file list if you want to execute the macro on selected blocks only.
2. Choose **Run macro** command from **Tools** pulldown menu in TerraScan **Project** window.

   This opens the **Run macro on blocks** dialog:



3. Define settings as described in **Run a macro on a project**.
4. Switch **Run using TerraSlave** on.
5. Click Settings.

   This opens the **TerraSlave task settings** dialog:



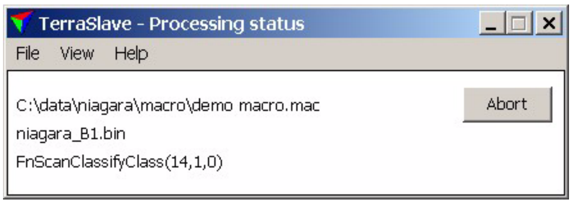
6. Select, on which computer(s) TerraSlave executes the task.
7. Click OK.
8. Click OK to the **Run macro on blocks** dialog.

   This launches TerraSlave if it is not already open and starts processing the task. If another process is already running in TerraSlave, the task file is added to the queue and processed as soon as the other tasks are finished.

Setting:	**Effect:**
Run using	Defines the computer that is performing the task: • This computer only - TerraSlave on the same computer will execute the macro. • Any available computer - TerraSlave on any available computer from the network will execute the macro. • Select computers from list - TerraSlave on all selected computers from the list will execute the macro. • Enter computers names - TerraSlave on all computers whose names are given will execute the macro.

TerraSlave quick guide

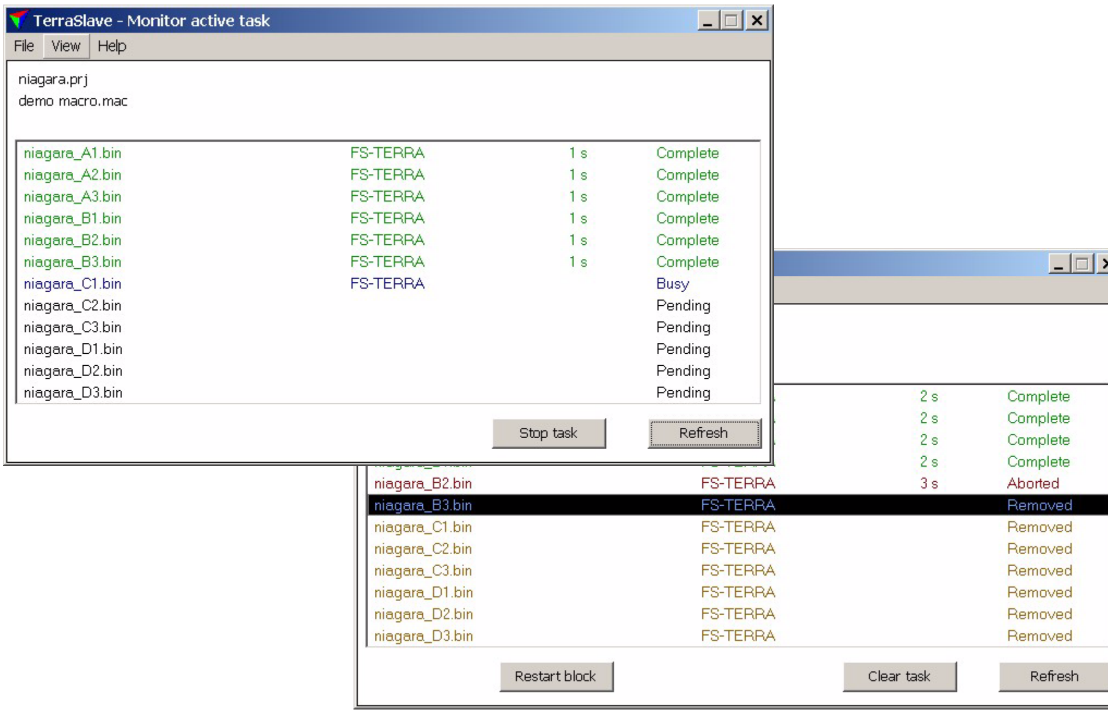
When TerraSlave is started, the TerraSlave **Processing window** opens:



It is empty, if no task is running. If a macro is executed, it shows the status of the process including the currently running macro, processed block and processing step.

A task can be stopped by clicking on the **Abort** button. This stops the processing of the current block. An new button **Activate** appears in the window. Click this to reactivate the task. This restarts the macro execution with the next block.

The processing of a task can be monitored using the **Monitor active task** command from **View** pulldown menu. This opens the **Monitor active task** window:



It shows a detailed list of the blocks to be processed, the executing computer's name and the current processing status.

Stop task stops the task and sets the status of all remaining blocks to **Removed**. It also activates the **Clear task** button which can be used to delete a task file completely.

If one or more blocks with status **Removed** are selected in the list, the **Restart block** button appears. This sets the status of the selected blocks to **Pending**. To restart a task for remaining blocks, TerraSlave has to be restarted. After the restart, the task is processed for all blocks with the status **Pending**.

Pending blocks can be removed from a running task by selecting the blocks and clicking the **Remove blocks** button. The task is processed on all other blocks.

✍ If a task is stopped, the task file remains in the task folder in the TerraSlave installation directory. Make sure to delete the task file from the directory to avoid the unwanted execution of old task files when TerraSlave is restarted.

Programming Interface

18 DLL Interface

TerraScan installation includes an example dynamic link library together with source code illustrating how tools can be added into TerraScan.

These addon tools may perform such tasks as classifying points, writing points to output files or displaying statistical information. The names of the tools will appear in the **Addon** menu in the TerraScan’s **Main window**. The tools are responsible for providing modal or mode-less dialogs for the input of required user settings.

The example link library \terra\ma\tscanadd.dll has been built using Visual C++ 6.0 from source codes included in \terra\addon directory. It provides one tool: the display of an intensity histogram.

You may modify the source code and build a dynamic link library to replace the one supplied during installation. The addon library must provide four predefined functions which TerraScan will call at different stages of application execution. These functions are:

- AddonFillCommands() is called at start-up so that the DLL will inform TerraScan about available tools.
- AddonLinkVariables() is called after start-up to provide the DLL with information about some of TerraScan’s internal tables and variables.
- AddonRunCommand() is called whenever user starts an addon command.
- AddonEnd() function is called before the application is unloaded so that the DLL may release any resources it has allocated.

See \terra\addon directory for example source code files.

The example source code uses C syntax and accesses WIN32 API to create the user interface items.

19 MDL Public Functions

TerraScan has a number of public functions which can be called by other MDL applications or used in macros. When creating a macro, you can normally define the parameters by selecting values in a dialog and you do not need to refer to the parameter specifications in this chapter.

Public functions make it possible for another MDL application to interact with TerraScan. These same routines are used internally by Terrasolid for interaction between TerraScan and TerraPhoto.

Laser point memory structure

When laser points are loaded into RAM, TerraScan keeps the different attributes of laser points in separate tables. For example, all the point flightline numbers of laser points will be kept in one table which is stored in one continuous memory area.

The possible attributes which may be present are:

Attribute	Always	C Data Type	Size
Point coordinates	Yes	Point3d	12 bytes
Time stamps	-	double	8 bytes
Group identifiers	-	unsigned int	4 bytes
Normal vectors	-	unsigned int	4 bytes
Distances	-	int	4 bytes
RGB colors	-	RgbClr	3 bytes
Intensity values	-	unsigned short	2 bytes
Line numbers	-	unsigned short	2 bytes
Echo lengths	-	short	2 bytes
Parameter values	-	unsigned short	2 bytes
Point classes	Yes	unsigned char	1 byte
Mark values -- run time only	Yes	unsigned char	1 byte
Flag values -- run time only	Yes	unsigned char	1 byte
Scanner angles	-	char	1 byte
Echo bits	-	unsigned char	1 byte
Scanner numbers	-	unsigned char	1 byte
Echo normality values	-	unsigned char	1 byte

Calling application must not assume that all of the attributes are present in the memory. Only attributes marked **Always** are always allocated when reading points into memory.

All attribute tables hold the same number of items.

Another application can retrieve pointer to laser point attributes by calling FnScanGetPnt(), FnScanGetCls() or other FnScanGetXxxx() routines. You should call these every time user starts a new operation in your application as the pointer may become invalid if user has modified laser points using TerraScan tools (deleted points, loaded new points in or closed TerraScan's **Main window**).

Calling Method

The functions can be called with mdlCExpression_ getValue(). The code example below illustrates the method:

```
void    Example( void)
{
    int     Ret ;

    if (TsCall( &Ret, "FnScanGetTable(0)") > 0)
        mdlOutput_printf( MSG_PROMPT, “%d points loaded into TerraScan”, Ret) ;
}

/*-------------------------------------------------------------------
    Call a function in TerraScan.
    Set *Ret to be the return value.

    Return 1 if successful.
    Return 0 if could not load TSCAN.
    Return -1 if failed.
*/

int     TsCall( int *Ret, const char *Expr)
{
    CExprValue    Val ;
    CExprResult   Res ;
    int           Ok ;

    if (!LoadApp( “TSCAN”))
        return (0) ;
    Ok = mdlCExpression_getValue( &Val, &Res, Expr, VISIBILITY_CALCULATOR);
    if (Ok != SUCCESS)
        return (-1) ;
    if (Ret)
        *Ret = (int) Val.val.valLong ;
    return (1) ;
}

/*-------------------------------------------------------------------
    Load MDL application with Name (such as “TSCAN”).

    Return 1 if successful.
    Return 0 if application not found.
*/

int    LoadApp( char *Name)
{
    void    *Ptr ;
    int     Ok ;

    // Is application already loaded?

    Ptr = mdlSystem_findMdlDesc( Name) ;
    if (Ptr)                                return (1) ;

    // Not loaded, attempt loading

    Ok = mdlSystem_loadMdlProgram( Name, NULL, "") ;
    if (Ok == SUCCESS)                      return (1) ;

    return (0) ;
}
```

Function Prototypes

```
/*-----------------------------------------------------------------
      Classify points from one class to another.

        FromClass    source class (999 for any)
        ToClass      destination class
        Fence        0=all points, 1=inside fence

      Return number of points classified (0...n).
      Return -1 if no fence defined (and Fence was nonzero).
      Return -3 if invalid parameters.
*/

int   FnScanClassifyClass( int FromClass, int ToClass, int Fence) ;

/*-----------------------------------------------------------------
      Classify local minimum points which are more than Dz lower than
      any other source point within XyDst distance.

        FromClass    source class (999 for any)
        ToClass      destination class
        GrpCnt       maximum number of low points in a group (1...99)
        Dz           minimum elevation difference to other points (m)
        XyDst        search radius (m)
        Fence        0=all points, 1=inside fence

      Return number of points classified (0...n).
      Return -1 if no fence defined (and Fence was nonzero).
      Return -2 if was aborted.
      Return -3 if invalid parameters.
*/

int   FnScanClassifyLow( int FromClass, int ToClass, int GrpCnt, double Dz,
                         double XyDst, int Fence) ;

/*-----------------------------------------------------------------
      Classify points up in the air. Compare each point against all
      points within a search rectangle with radius (=half of rectangle
      edge length) XyDst. If there are at least ReqCnt points and
      point in question is higher than:

        MedianElevation + (Lim * StandardDeviationOfElevations)

      classify it into ToClass class.

        ClsLst       list of source classes, example "1,7-9"
        ToClass      destination class
        XyDst        search radius (m)
        ReqCnt       minimum point count within search rectangle
        Lim          minimum factor
        Fence        0=all points, 1=inside fence

      Return number of points classified (0...n).
      Return -1 if no fence defined (and Fence was nonzero).
      Return -2 if was aborted.
      Return -3 if invalid parameters.
*/

int   FnScanClassifyAir( char *ClsLst, int ToClass, double XyDst, int ReqCnt,
                         double Lim, int Fence) ;
```

```
/*-----------------------------------------------------------------
      Classify isolated points which have fewer than LimCnt other
      points within 3D search radius Dst.

        ClsLst        list of source classes, example "1,7-9"
        ToClass       destination class
        LimCnt        classify if fewer than Lim other points
        Dst           search radius (m)
        Fence         0=all points, 1=inside fence

      Return number of points classified (0...n).
      Return -1 if no fence defined (and Fence was nonzero).
      Return -2 if was aborted.
      Return -3 if invalid parameters.
*/

int   FnScanClassifyIsolated( char *ClsLst, int ToClass, int Lim, double Dst, int Fence);

/*-----------------------------------------------------------------
      Classify ground points by iteratively molding a triangulated
      ground model upwards.

        FromClass     source class (999 for any)
        ToClass       destination class
        InitLow       initial points: 0=current ground points, 1=low
        BldSz         - maximum building size (m) if InitLow=1
        MaxAng        maximum terrain angle (degrees)
        IterAng       iteration angle (degrees)
        IterDst       iteration distance (m)
        Reduce        reduce iteration length?
        RedLen        - when edge length < RedLen
        Stop          stop iteration?
        StopLen       - when edge length < StopLen
        Fence         0=all points, 1=inside fence

      Return number of points classified (0...n).
      Return -1 if no fence defined (and Fence was nonzero).
      Return -2 if was aborted.
      Return -3 if invalid parameters.
*/

int   FnScanClassifyGround( int FromClass, int ToClass, int InitLow, double BldSz,
                            double MaxAng, double IterAng, double IterDst, int Reduce,
                            double RedLen, int Stop, double StopLen, int Fence) ;
```

```
/*------------------------------------------------------------------
    Classify points below surface. Compare each point against
    a plane equation fitted to closest neighbours. Classify point
    if it is clearly below the plane.

    Routine computes the distance from each point to the plane and
    computes the standard deviation of these distances. Point is
    classified if it is more than Tol below the plane and more than
    Lim * StandardDeviation below the plane.

      FromClass    source class -- normally ground
      ToClass      destination class -- normally low point
      Lim          limit
      Tol          elevation tolerance
      Fence        0=all points, 1=inside fence

      Return number of points classified (0...n).
      Return -1 if no fence defined (and Fence was nonzero).
      Return -2 if was aborted.
      Return -3 if invalid parameters.
*/

int   FnScanClassifyBelow( int FromClass, int ToClass, double Lim, double TolZ, int Fence);

/*------------------------------------------------------------------
    Classify points points by height from ground.

      Grd          ground class or 1000000+TerraModeler surface id
      MaxLen       maximum ground triangle length
      FromClass    source class (999 for any)
      ToClass      destination class
      MinH         minimum height above ground
      MaxH         maximum height above ground
      Fence        0=all points, 1=inside fence

    Return number of points classified (0...n).
    Return -1 if no fence defined (and Fence was nonzero).
    Return -2 if was aborted.
    Return -3 if invalid parameters.
*/

int   FnScanClassifyHgtGrd( int Grd, double MaxLen, int FromClass, int ToClass,
                            double MinH, double MaxH, int Fence) ;

/*------------------------------------------------------------------
      Classify points by height from multiple ground classes.

        GrdLst       list of ground classes, example "2,8"
        MaxLen       maximum ground triangle length
        FromClass    source class (999 for any)
        ToClass      destination class
        MinH         minimum height above ground
        MaxH         maximum height above ground
        Fence        0=all points, 1=inside fence

      Return number of points classified (0...n).
      Return -1 if no fence defined (and Fence was nonzero).
      Return -2 if was aborted.
      Return -3 if invalid parameters.
*/

int   FnScanClassifyHgtLst( char *GrdLst, double MaxLen, int FromClass, int ToClass,
                            double MinH, double MaxH, int Fence) ;
```

```
/*----------------------------------------------------------------------
      Classify points within given elevation range. This classifies
      points which satisfy MinZ <= Z < MaxZ.

        FromClass    source class (999 for any)
        ToClass      destination class
        MinZ         minimum elevation
        MaxZ         maximum elevation
        Fence        0=all points, 1=inside fence

      Return number of points classified (0...n).
      Return -1 if no fence defined (and Fence was nonzero).
      Return -2 if was aborted.
      Return -3 if invalid parameters.
*/

int   FnScanClassifyAbsElev( int FromClass, int ToClass, double MinZ, double MaxZ,
                             int Fence) ;

/*----------------------------------------------------------------------
      Classify points within given intensity range. This classifies
      points which satisfy MinV <= Intensity < MaxV.

        FromClass    source class (999 for any)
        ToClass      destination class
        MinV         minimum intensity
        MaxV         maximum intensity
        Fence        0=all points, 1=inside fence

      Return number of points classified (0...n).
      Return -1 if no fence defined (and Fence was nonzero).
      Return -2 if was aborted.
      Return -3 if invalid parameters.
*/

int   FnScanClassifyIntensity( int FromClass, int ToClass, double MinV, double MaxV,
                               int Fence) ;

/*----------------------------------------------------------------------
      Classify points within given color range which is given using
      HSV color model.

        FromClass    source class (999 for any)
        ToClass      destination class
        Source       color source: 0=laser, 1=ortho, 2=raw images
        Foot         laser footprint diameter (m) in ortho/raw images
        Hue          hue value
        Tol          +- tolerance for hue
        SatMin       minimum saturation
        SatMax       maximum saturation
        ValMin       minimum value
        ValMax       maximum value
        Fence        0=all points, 1=inside fence

    Return number of points classified (0...n).
      Return -1 if no fence defined (and Fence was nonzero).
      Return -2 if was aborted.
      Return -3 if invalid parameters.
*/

int   FnScanClassifyColor( int FromClass, int ToClass, int Source, double Foot, int Hue,
                           int Tol, int SatMin, int SatMax, int ValMin, int ValMax,
                           int Fence) ;
```

```
/*-------------------------------------------------------------------
      Classify points within given time range. This classifies
      points which satisfy Beg <= Time <= End.

        FromClass     source class (999 for any)
        ToClass       destination class
        Beg           start time (seconds)
        End           end time (seconds)
        Fence         0=all points, 1=inside fence

      Return number of points classified (0...n).
      Return -1 if no fence defined (and Fence was nonzero).
      Return -2 if was aborted.
      Return -3 if invalid parameters.
*/

int   FnScanClassifyTime( int FromClass, int ToClass, double Beg, double End, int Fence) ;

/*-------------------------------------------------------------------
      Classify model keypoints which produce a triangulated surface
      model of given accuracy.

        ClsLst        list of source classes, example "2,14"
        ToClass       destination class
        InitDst       initial sampling distance (m)
        Above         distance points may be above model (m)
        Below         distance points may be below model (m)
        Fence         0=all points, 1=inside fence

      Return number of points classified (0...n).
      Return -1 if no fence defined (and Fence was nonzero).
      Return -2 if was aborted.
      Return -3 if invalid parameters.
*/

int   FnScanClassifyModelKey( char *ClsLst, int ToClass, double InitDst,
                              double Above, double Below, int Fence) ;

/*-------------------------------------------------------------------
      Classify contour keypoints which produce a triangulated surface
      suitable for generating contours.

        ClsLst        source classes, for example "2,14"
        ToClass       destination class
        Interval      contour interval (m)
        Init          initial sampling distance (m)
        Lim           limit for classification
                      (1=accure/many points, 100=pretty/few points)
        Peaks         0=no, 1=keep peaks and pits
        PeakArea      minimum area (m2) for peak to keep
        PitArea       minimum area (m2) for pit to keep
        Fence         0=all points, 1=inside fence

      Return number of points classified (0...n).
      Return -1 if no fence defined (and Fence was nonzero).
      Return -2 if was aborted.
      Return -3 if invalid parameters.
*/

int   FnScanClassifyContourKey( char *ClsLst, int ToClass, double Interval, double Init,
                         int Lim, int Peaks, double PeakArea, double PitArea, int Fence)
```

```
/*-------------------------------------------------------------------
      Classify points based on selected centerline element. Alignment
      element must be selected with selection tool before executing
      this function.

        ClsLst       list of source classes, example "1,7-9"
        ToClass      destination class
        Cmp          compare with 0=any line, 1=closest line
        Side         -1=left, 0=either, 1=right side
        OffMin       minimum xy offset from element (m)
        OffMax       maximum xy offset from element (m)
        UseDz        if true, use dz from alignment
        DzMin           minimum dz value
        DzMax           maximum dz value
        UseDst       if true, use distance from closest vertex
        DstMin       minimum longitudinal distance to vertex
        DstMax       minimum longitudinal distance to vertex

      Return number of points classified (0...n).
      Return -1 if invalid or no selection.
      Return -2 if was aborted.
      Return -3 if invalid parameters.
*/

int   FnScanClassifyCtrline( char *ClstLst, int ToClass, int Cmp,
                             int Side, double OffMin, double OffMax,
                             int UseDz, double DzMin, double DzMax,
                             int UseDst, double DstMin, double DstMax) ;

/*-------------------------------------------------------------------
      Classify points based on echo information.

        FromClass    source class (999 for any)
        ToClass      destination class
        Echo         from echo
        Fence        0=all points, 1=inside fence

      Valid values for Echo parameter are:
        0  only echo
        1  first of many
        2  intermediate
        3  last of many
       11  any first (only echo or first of many)
       13  any last (only echo or last of many)

      Return number of points classified (0...n).
      Return -1 if no fence defined (and Fence was nonzero).
      Return -3 if invalid parameters.
*/

int   FnScanClassifyEcho( int FromClass, int ToClass, int Echo, int Fence) ;
```

```
/*-------------------------------------------------------------------
    Classify points based on scan direction or edge flag.

      FromClass    source class (999 for any)
      ToClass      destination class
      Direction    0=negative direction, 1=positive direction, 2=edge
      Fence        0=all points, 1=inside fence

    Return number of points classified (0...n).
    Return -1 if no fence defined (and Fence was nonzero).
    Return -2 if no direction flag available.
    Return -3 if invalid parameters.
*/

int   FnScanClassifyDirection( int FromClass, int ToClass, int Direction, int Fence) ;

/*-------------------------------------------------------------------
      Classify points which are hits on building roofs.

        GrdClass     ground class
        FromClass    source class (high vegetation)
        ToClass      destination class
        MinArea      minimum area of building (m2)
        Tol          plane tolerance (m)
        UseEcho      use echo information?
        Fence        0=all points, 1=inside fence

      Return number of points classified (0...n).
      Return -1 if no fence defined (and Fence was nonzero).
      Return -3 if invalid parameters.
*/

int   FnScanClassifyBuilding( int GrdClass, int FromClass, int ToClass, double MinArea,
                              double Tol, int UseEcho, int Fence) ;

/*-------------------------------------------------------------------
      Classify points inside shape elements on a given level in the
      active design file or any reference file with Locate lock on.

        ClsLst       list of source classes, example "4-5,11"
        ToClass      destination class
        Lvl          level
        ByClr        filter by color?
        Clr          - color
        ByWgt        filter by weight?
        Wgt          - weight
        BySty        filter by style?
        Sty          - style
        Dgn          0=active design and references,
                     1=active design file,
         2=reference files

      Return number of points classified (0...n).
      Return -2 if out of memory.
*/

int   FnScanClassifyShapes( char *ClsLst, int ToClass, int Lvl, int ByClr, int Clr,
                       int ByWgt, int Wgt, int BySty, int Sty, double Expand, int Dgn)
```

```
/*-------------------------------------------------------------------
      Classify every Mark=1 point from class FromClass to class
      ToClass. Redraw the points during the process.

      Return number of classified points.
*/

int   FnScanClassifyMarked( int FromClass, int ToClass) ;

/*-------------------------------------------------------------------
      Delete points in a given class.

        Class         source class to delete
        Fence         0=all points, -1=outside fence, 1=inside fence

      Return number of points deleted (0...n).
      Return -1 if no fence defined (and Fence was nonzero).
*/

int   FnScanDeleteClass( int Class, int Fence) ;

/*-------------------------------------------------------------------
      Delete points from a given flightline or measurement.

        Line          flightline or measurement to delete
        Fence         0=all points, -1=outside fence, 1=inside fence

      Return number of points deleted (0...n).
      Return -1 if no fence defined (and Fence was nonzero).
*/

int   FnScanDeleteLine( int Line, int Fence) ;

/*-------------------------------------------------------------------
      Transform coordinates of points using a transformation defined
      in TerraScan settings.

        Class         source class to transform (999 for any)
        Name          name of transformation in TerraScan settings
        Fence         0=all points, 1=inside fence

      Return number of points deleted (0...n).
      Return -1 if no fence defined (and Fence was nonzero).
      Return -2 if was aborted.
      Return -3 if invalid parameters.
      Return -4 if transformation was not found.
*/

int   FnScanTransform( int Class, char *Name, int Fence) ;

/*-------------------------------------------------------------------
      Redraw contents of all views which may be displaying points.

      Return 1 always.
*/

int   FnScanUpdateViews( void) ;

/*-------------------------------------------------------------------
    Sort points to By order:
      0   increasing easting
      1   decreasing northing
      2   increasing easting
      3   decreasing northing
      4   increasing time stamp
```

```
    Return  1 if points were sorted.
    Return  0 if nothing was done.
*/

int   FnScanSort( int By) ;

/*-------------------------------------------------------------------
      Output points to a file.

        File      Output file name. If it contains string "######",
                  that string will be replaced by active block number.
                  Active project directory will be used if File does
                  not include a directory specification.
                  If File contains string "#name", that string will
                  replaced by active file name. In that case, File
                  should always include directory and extension such
                  as "c:\output\#name.grd".
        ClsLst    list of classes to output, example "1,7-9"
        Fmt       Format of output file.
        Delim     Delimiter (0=space, 1=tabulator, 2=comma).
        Trans     Name of transformation in TerraScan settings.
                  Pass zero or "" for none.
        Append    If true, add to end of a possibly existing file.
        Fence     0=all points, 1=inside fence

      Format can be:
          1        X Y Z
          2        Class X Y Z
          3        X Y Z Intensity
          5        Class X Y Z Intensity
          7        TerraScan binary 8 bit line numbers
          8        TerraScan binary 16 bit line numbers
         13        EarthData binary
         23        Grass Sites format
        200+       User defined file format

    Return number of points written.
    Return -1 if no fence defined (and Fence was nonzero).
    Return -2 if failed to write to file.
    Return -3 if invalid parameters.
    Return -4 if transformation was not found.
*/

int   FnScanOutput( char *File, char *ClsLst, int Fmt, int Delim, char *Trans,
                    int Append, int Fence) ;
```

```
/*-------------------------------------------------------------------
      Output points to a file based on flightline number. This
      will generate multiple output files if Line == -1 in which
      case one output file will be generated for every flightline.

        File      Output file name.
                  If File contains string "#line", that string will
                  be replaced by flight line number.
                  Active project directory will be used if File does
                  not include a directory specification.
        Line      Line to output. Pass -1 for all lines and use "#line"
                  as part of file name.
        Class     Class of points to output. Pass 999 for all points.
        Fmt       Format of output file, see FnScanOutput().
        Delim     Delimiter (0=space, 1=tabulator, 2=comma).
        Trans     Name of transformation in TerraScan settings.
                  Pass zero or "" for none.
        Append    If true, add to end of a possibly existing file.

      Return number of points written.
      Return -3 if invalid parameters.
      Return -4 if transformation was not found.
*/

int   FnScanOutputLine( char *File, int Line, int Class, int Fmt, int Delim,
                        char *Trans, int Append)

/*-------------------------------------------------------------------
      Return information about the active project. Set:

        Type   project type: 0=airborne, 1=ground based
        Fmt    storage format: 0=TerraScan binary, 1=EarthData
        Dir    data file directory

      Return number of blocks (0 - n).
      Return -1 if no active project.
*/

int   FnScanProjectInfo( int *Type, int *Fmt, char *Dir) ;

/*-------------------------------------------------------------------
      Return information about project block index Ind. Set:

        Vrt[]   boundary vertices (up to 41)
        File    binary file name including full path

      Return number of boundary vertices (0 - 41).
      Return -1 if invalid index or no active project.
*/

int   FnScanProjectBlock( Dpoint3d *Vrt, char *File, int Ind) ;
```

```
/*----------------------------------------------------------------------
      Create lattice model from laser points in class Class and
      write that to file File.

        File      Output file name.
                  Active project directory will be used if File does
                  not include a directory specification.
                  If it contains string "######" or "#block", that
                  string will be replaced by active block number.
                  If File contains string "#name", that string will
                  be replaced by active file name.
        Class     Class of points to output. Pass 999 for all points.
        Elev      0=lowest, 1=average, 2=highest hit, 3=triangle model
        Step      Grid step size (m).
        Len       Number of gap pixels to fill if Elev != 3.
                  Maximum triangle length if Elev == 3.
        Fmt       Format of output file.
        UnitZ     Unit of Z values: 0=uor, 1=mm, 2=cm, 3=dm, 4=m
                  Used only with Intergraph and Raw binary formats.
        CrdBlk    0=no coordinate block, nonzero=output.
                  Used only with Intergraph format.
        Out       Value to output in undefined area. Pass NULL to skip.
                  Used only with ArcInfo and Xyz text formats.

      Format can be:
        0         Intergraph GRD
        5         Raw binary 16 bit
        6         Raw binary 32 bit
        7         Raw binary 64 bit double
        8         ArcInfo GRIDASCII
        9         Xyz text

      Return  1 on success.
      Return  0 if no source points.
      Return -2 if failed to write to file.
*/

int   FnScanExportLattice( char *File, int Class, int Elev, double Step, double Len,
                           int Fmt, int UnitZ, int CrdBlk, char *Out)

/*----------------------------------------------------------------------
      Read points from file File into RAM.

        File      Full path of input file(s)
                  Multiple file names should be separated by '\n'
        First     Class for non-last echos (if not in input)
        Last      Class for last echos (if not in input)
        Line      Flightline number (if not in input)
        Inc       When flightline number is incremented
                  0=never, 2=new file, 3=new file name, 4=new dir
        Trans     Name of transformation in TerraScan settings
                  Pass zero or "" for none
        Fence     0=all points, 1=inside fence
        Every     0=every, 1=every, n=every n:th

      Return number of points found.
      Return -2 if failed to open or recognize format of first file.
      Return -3 if invalid parameters.
      Return -4 if transformation was not found.
      Return -5 if out of memory.
*/

int   FnScanRead( char *File, int First, int Last, int Line, int Inc,
                  char *Trans, int Fence, int Every)
```

```
/*-------------------------------------------------------------------
      Smoothen surface by adjusting point elevations.
      moving points points by adjusting elevations by a maximum of

        ModLst       classes to smoothen, example "1,7-9"
        UpCm         maximum movement up (cm)
        DnCm         maximum movement down (cm)
      FixLst       immovable classes to include in surface, "5-6"
      Fence        0=all points, -1=outside fence, 1=inside fence

      Return  1 if points were adjusted.
      Return  0 if nothing was done.
      Return -2 if out of memory.
*/

int   FnScanSmoothen( char *ModLst, int UpCm, int DnCm, char *FixLst, int Fence) ;

/*-------------------------------------------------------------------
      Thin data set by removing points close to another point.
      Point can be removed if it is within Dxy horizontal distance
      and Dz elevation difference from another point.

        Class        source class to thin (999 for any)
        Keep         point to keep in a group:
                        0=highest,1=lowest,2=central,3=create average
        Dxy          xy distance limit
        Dz           dz limit
        Fence        0=all points, -1=outside fence, 1=inside fence

      Return number of points removed (0...n).
      Return -1 if no fence defined (and Fence was nonzero).
      Return -3 if invalid parameters.
*/

int   FnScanThinPoints( int Class, int Keep, double Dxy, double Dz, int Fence) ;

/*-------------------------------------------------------------------
      Cut points from overlapping flightlines by either flightline
      quality or by offset from trajectory.

        ClsLst       List of coverage classes, example "0-255"
        Action       Action: 0=add constant, 1=classify, 2=delete
        Val          If Action == 0, value to add to class
                     If Action == 1, class to classify to
         If Action == 2, ignored
        Quality      0=no, 1=cut by trajectory quality
        Gap           - radius of empty area to consider a gap
        Offset       0=no, 1=cut by offset
        Keep          - minimum corridor to keep (degrees)

      Return number of points affected (0...n).
      Return -3 if invalid parameters.
      Return -4 if no trajectories.
*/

int   FnScanCutOverlap( char *ClsLst, int Action, int Val, int Quality, double Gap,
                        int Offset, int Keep) ;
```

```
/*-------------------------------------------------------------------
      Adjust laser points using an angular change to the vector
      from scanner position to the point.

        Line      Line to adjust. Pass -1 for all lines.
        Fix       Coordinate axis to fix: 1=z, 2=xy, 3=xyz
        Input     Angle unit used: 0=degrees, 1=radians, 2=ratio
        Head      Heading correction angle
        Roll      Roll correction angle
        Pitch     Pitch correction angle
        Div       Angle divider corresponding to full circle.
                  Used if Input == 2.

      Return number of points adjusted.
      Return -1 if no trajectories loaded.
*/

int   FnScanAdjustAngles( int Line, int Fix, int Input, double Head, double Roll,
                          double Pitch, double Div) ;


/*-------------------------------------------------------------------
      Return address of point coordinates and number of points.

      Note that this address may change any time when points have
      been added or deleted from the table.

      Return number of points.
      Return zero if failed.
*/

int   FnScanGetPnt( Point3d **Tbl) ;

/*-------------------------------------------------------------------
      Return address to laser point classes and number of points.

      Return number of points.
      Return zero if no point classes.
*/

int   FnScanGetCls( BYTE **Tbl) ;

/*-------------------------------------------------------------------
      Return address to laser point mark values and number of points.

      Return number of points.
      Return zero if no mark values.
*/

int   FnScanGetMrk( BYTE **Tbl) ;

/*-------------------------------------------------------------------
      Return address to laser point flag values and number of points.

      Return number of points.
      Return zero if no flag values.
*/

int   FnScanGetFlg( BYTE **Tbl) ;
```

```
/*----------------------------------------------------------------------
      Return address to laser point intensities and number of points.

      Return number of points.
      Return zero if no intensities.
*/

int   FnScanGetInt( USHORT **Tbl) ;

/*----------------------------------------------------------------------
      Return address to laser point line numbers and number of points.

      Return number of points.
      Return zero if no line values.
*/

int   FnScanGetLin( USHORT **Tbl) ;

/*----------------------------------------------------------------------
      Return address to time stamp table and number of points.

      Return number of points.
      Return zero if no time stamps.
*/

int   FnScanGetDbl( double **Tbl) ;

/*----------------------------------------------------------------------
      Return address to laser point echo bits and number of points.

      Return number of points.
      Return zero if no echo bits.
*/

int   FnScanGetEch( BYTE **Tbl) ;

/*----------------------------------------------------------------------
      Return address to laser point mirror angles and number of points.

      Return number of points.
      Return zero if no angles.
*/

int   FnScanGetAng( char **Tbl) ;

/*----------------------------------------------------------------------
      Return address to color table and number of points.

      Return number of points.
      Return zero if no color values.
*/

int   FnScanGetClr( RgbClr **Tbl) ;

/*----------------------------------------------------------------------
      Return address to scanner numbers and number of points.

      Return number of points.
      Return zero if no scanner numbers.
*/

int   FnScanGetScr( BYTE **Tbl) ;
```

```
//      ================================================================
//     Get current coordinate setup. Routine stores coordinate system
//     origin in *Org.
//
//     To compute master unit coordinates *D from laser point integer
//     coordinates *P, you would use:
//
//         doubleMul ;
//
//     Mul = 1.0 / UorPerMast ;
//     D->x = Org->x + (Mul * P->x) ;
//     D->y = Org->y + (Mul * P->y) ;
//     D->z = Org->z + (Mul * P->z) ;
//
//     where UorPerMast is the return value from this routine.
//
//     Return number of integer units per master unit.
//      ================================================================

int    FnScanGetCoordSetup( Dp3d *Org) ;

/*-------------------------------------------------------------------
      Mark every point to have mark value MarkVal.
*/

int    FnScanMarkAll( int MarkVal) ;

/*-------------------------------------------------------------------
      Mark every point inside fence to have Mark=1 and every point
      outside fence to have Mark=0.

      Set *Fst and *Lst as the indexes of first and last point inside
      fence.

      Return 1 on success.
      Return 0 if failed.
*/

int    FnScanMarkFence( int *Fst, int *Lst) ;

/*-------------------------------------------------------------------
      Fill option button with class names.

      Return 0 always.
*/

int    FnScanClassOption( RawItemHdr *Raw, int Nbr, int Fst) ;

/*-------------------------------------------------------------------
      Fill string list with:
       - class numbers and names if Nbr is true
       - class names only if Nbr is zero

      Return number of classes.
*/

int    FnScanClassList( StringList *Lst, int Nbr) ;
```

```
/*-----------------------------------------------------------------
      Ask user to select point classes to use. Set character array
      to indicate which classes were selected.

      Sel[] should be allocated for 256 items.
      Sel[0] indicates if code 0 is selected.
      Sel[1] indicates if code 1 is selected.

      Return number of classes selected.
      Return -1 if canceled.
*/

int   FnScanSelectClasses( char *Sel) ;

/*-----------------------------------------------------------------
      Save laser points to a temporary file Path in binary format.

      Return 1 on success.
      Return 0 if failed to write.
*/

int   FnScanSave( char *Path) ;

/*-----------------------------------------------------------------
      Reload laser points from a temporary file Path.

      Return  1 on success.
      Return  0 if failed to read file.
      Return -1 if we already have points in memory.
      Return -2 if out of memory.
*/

int   FnScanReload( char *Path) ;

/*-----------------------------------------------------------------
      Outside application plans to call FnScanFreeTable().
      Ask user if he wants to save modified points.

      Return  1 if were saved or user decided not to.
      Return  0 if no points or not modified.
*/

int   FnScanPreFree( void) ;

/*-----------------------------------------------------------------
      Free memory allocated for laser points.

      Return 1 always.
*/

int   FnScanFreeTable( void) ;
```

```
/*---------------------------------------------------------------
      Find laser point closest to point Cp which is within distance
      R of point Cp when drawn in view Vw.

        Fp         Point where the coordinates will be stored.
        Class      Class of points to output. Pass 999 for all points.
        Cp         Point at which to search.
        R          Search radius (m).
        Vw         View index for rotation of search (pass -1 for top
                   view search logic).

      Return index of point if found (0,1,2,...).
      Return -1 if no point found.
*/

int   FnScanFindClosest( Dp3d *Fp, int Class, Dp3d *Cp, double R, int Vw) ;
```

20 File formats

TerraScan supports a large number of input file formats which have been hard coded into the program logic. Some of these built-in file formats are binary but most are text file formats. Whenever you read files into the application, it will try to recognize the file format automatically.

Point cloud file formats

TerraScan binary files

Laser points are normally stored in TerraScan binary format which provides a compact way of storing laser points and all the information the application can associate with the points.

The information below refers to file formats which were last revised on 12.07.2001 and on 15.07.2002. These version dates are stored in the file header. Future versions of TerraScan may store laser points into another format but will always recognize and read in the old files.

Current version of TerraScan reads and writes two versions of the binary file format:

- **Scan binary 8 bit line -** a more compact version which can accommodate flightline numbers 0-255. Files with this format have HdrVersion field set to 20010712.
- **Scan binary 16 bit line -** a slightly bigger version which can accommodate flightline numbers 0-65535. Files with this format have HdrVersion field set to 20020715.

See \terra\addon\routines.c for example source code for reading in TerraScan binary files.

File organization

TerraScan binary file consists of a file header of 48 bytes and a number of point records. The size of the point record is 16 bytes for file version 20010712 and 20 bytes for file version 20020715. Each point record may be followed by an optional four byte unsigned integer time stamp and an optional four byte RGB color value.

For example, a file containing four laser points and their time stamps using format 20020715 would consist of:

- 48 byte header (ScanHdr)
- 20 byte record for first point (ScanPnt)
- 4 byte time stamp for first point
- 20 byte record for second point (ScanPnt)
- 4 byte time stamp for second point
- 20 byte record for first three (ScanPnt)
- 4 byte time stamp for three point

Structure definitions

The structure of the file header is:

```
typedef struct {
int      HdrSize ;     // sizeof(ScanHdr)
int      HdrVersion ;  // Version 20020715, 20010712, 20010129 or 970404
int      RecogVal ;    // Always 970401
char     RecogStr[4];  // CXYZ
long     PntCnt ;      // Number of points stored
int      Units ;       // Units per meter = subpermast * uorpersub
double   OrgX ;        // Coordinate system origin
double   OrgY ;
double   OrgZ ;
int      Time ;       // 32 bit integer time stamps appended to points
```

```
int     Color ;      // Color values appended to points
} ScanHdr ;
```

The structure of a point record for file version 20010712 is:

```
typedef struct {
BYTE    Code ;        // Classification code 0-255
BYTE    Line ;        // Flightline number 0-255
USHORT  EchoInt ;     // Intensity bits 0-13, echo bits 14-15
long    X ;           // Easting
long    Y ;           // Northing
long    Z ;           // Elevation
} ScanRow ;
```

The structure of a point record for file version 20020715 is:

```
typedef struct {
Point3d Pnt ;         // Coordinates
BYTE    Code ;        // Classification code
BYTE    Echo ;        // Echo information
BYTE    Flag ;        // Runtime flag (view visibility)
BYTE  Mark ;          // Runtime flag
USHORT  Line ;        // Flightline number
USHORT  Intensity ;   // Intensity value
} ScanPnt ;
```

Coordinate system

Laser point coordinates are stored as integer values which are relative to an origin point stored in the header. To compute user coordinate values X, Y and Z (normally meters), use:

```
X = (Pnt.X - Hdr.OrgX) / (double) Hdr.Units ;
Y = (Pnt.Y - Hdr.OrgY) / (double) Hdr.Units ;
Z = (Pnt.Z - Hdr.OrgZ) / (double) Hdr.Units ;
```

Time stamps

Time stamps are assumed to be GPS week seconds. The storage format is a 32 bit unsigned integer where each integer step is 0.0002 seconds.

Echo information

TerraScan uses two bits for storing echo information. The possible values are:

```
0Only echo
1First of many echo
2Intermediate echo
3Last of many echo
```

Supported file formats

TerraScan has built-in support for the following laser point file formats:

Name:	**Type:**	**Content:**
XYZ	Text	Easting Northing Elevation
CXYZ	Text	Class Easting Northing Elevation
XYZI	Text	Easting Northing Elevation Intensity
PXYZI	Text	Pulse_number Easting Northing Elevation Intensity
CXYZI	Text	Class Easting Northing Elevation Intensity
XYZXYZII	Text	East1 North1 Elev1 East2 North2 Elev2 Intensity1 Intensity2
TXYZXYZII	Text	Time East1 North1 Elev1 East2 North2 Elev2 Intensity1 Intensity2
TXYZI	Text	Time Easting Northing Elevation Intensity

Name:	Type:	Content:
XYZIXYZI	Text	East1 North1 Elev1 Intensity1 East2 North2 Elev2 Intensity2
TXYZIXYZI	Text	Time East1 North1 Elev1 Intensity1 East2 North2 Elev2 Intensity2
SCAN8	Binary	TerraScan binary with 8 bit flightlines (BIN).
SCAN16	Binary	TerraScan binary with 16 bit flightlines (BIN).
TOPEYE	Binary	TopEye binary (DTE) with geocentric coordinates.
EARTHDATA	Binary	EarthData binary (EBN).
LEICA	Binary	Leica-Helava binary (LDI).
LAS	Binary	Open binary format (LAS).

Trajectory file formats

TerraScan trajectory binary files

Imported trajectories are stored as binary files with TRJ extension. These files contain a header followed by a number of trajectory position records.

The structure of the file header is:

```
typedef struct {
char     Recog[8] ;       // TSCANTRJ
int      Version ;        // File version 20010715
int      HdrSize ;        // sizeof(TrajHdr)
int      PosCnt ;         // Number of position records
int      PosSize ;        // Size of position records

char     Desc[78] ;       // Description
BYTE     SysIdv ;         // System identifier (for lever arms)
BYTE     Quality ;        // Quality for whole trajectory (1-5)
double   BegTime ;        // First time stamp
double   EndTime ;        // Last time stamp
int      OrigNbr ;        // Original number (before any splitting)
int      Number ;         // Flightline number (in laser points)
char     VrtVideo[400] ;  // Vertical facing video
double   VrtBeg ;         // Start time of VrtVideo[]
double   VrtEnd ;         // End time of VrtVideo[]
char     FwdVideo[400] ;  // Forward facing video
double   FwdBeg ;         // Start time of FwdVideo[]
double   FwdEnd ;         // End time of FwdVideo[]
char     WaveFile[400] ;  // Waveform data file
char     Group[16] ;      // Group (session description)
} TrajHdr ;
```

The structure of the trajectory position records is:

```
typedef struct {
double   Time ;           // Time stamp (seconds in some system)
Dp3d     Xyz ;            // Position
double   Head ;           // Heading (degrees)
double   Roll ;           // Roll (degrees)
double   Pitch ;          // Pitch (degrees)
BYTE     QtyXy ;          // Quality for xy, 0=not set
BYTE     QtyZ ;           // Quality for z, 0=not set
BYTE     QtyH ;           // Quality for headingy, 0=not set
BYTE     QtyRp ;          // Quality for roll/pitch, 0=not set
short    Mark ;           // Run time flag
short    Flag ;           // Run time flag
} TrajPos ;
```

where Dp3d structure is:

```
typedef struct {
double   x ;
double   y ;
double   z ;
} Dp3d ;
```

and where xy and z quality values translate to meters as:

```
QualityInMeters = pow(QtyXy,1.5) * 0.001 m
(1=0.0010m, 2=0.0028m, 3=0.0052m, 4=0.0080m, ... 255=4.072m)
```

and where heading, and roll/pitch quality values translate to degrees as:

```
QualityInDegrees = pow(QtyH,1.5) * 0.0001 deg
(1=0.00010, 2=0.00028, 3=0.00052, 4=0.00080, ... 255=0.4072 de
```

Supported file formats

TerraScan has built-in support for the following trajectory file formats:

Name:	Type:	Content:
TYXZRPH	Text	Time Northing Easting Elevation Roll Pitch Heading
TYXZSSSQSS	Text	Time Northing Easting Elevation Skip Skip Skip Quality Skip Skip
TYXZQSSSS	Text	Time Northing Easting Elevation Quality Skip Skip Skip Skip
TTXYZHPR	Text	Time Time Easting Northing Elevation Heading Roll Pitch
TXYZ	Text	Time Easting Northing Elevation
SBET.OUT	Binary	Proprietary file format of Applanix
POF 1.1	Binary	Proprietary file format of Riegl.

Additional Information

21 Installation Directories

TerraScan shares the same directory structure with all Terra Applications. It is recommended that you install all Terra Applications in the same directory.

The list below shows a typical directory structure when TerraScan has been installed in path C:\TERRA.

c:\terra	directory where TerraScan was installed
addon	example addon source files
config	for configuration files
tscan.cfg	defines environment variables
docs	for documentation
tscan.pdf	documentation in Acrobat Reader format
license	for user license files
tscan.lic	user license
ma	for application files
tscan.ma	application
tscan.dll	library
tscanadd.dll	addon library
tscan	for point class lists and settings
tscan.ptc	example list of point classes

22 Configuration Variables

MicroStation is able to locate TerraScan with the help of configuration variables. When you install TerraScan, the installation program will create a configuration file TERRA.CFG which defines the required environment variables. This file is placed in MicroStation's CONFIG\APPL sub-directory.

For example, C:\USTATION\CONFIG\APPL\TERRA.CFG may contain:

```
#-------------------------------------------------------
#
#  TERRA.CFG - Configuration for Terra Applications
#
#-------------------------------------------------------

TERRADIR=c:/terra/
TERRACFG=$(TERRADIR)config/
TERRADOCS=$(TERRADIR)docs/

MS_MDLAPPS < $(TERRADIR)ma/

%if exists ($(TERRACFG)*.cfg)
%   include $(TERRACFG)*.cfg
%endif
```

This configuration file will include all the configuration files in C:\TERRA\CONFIG directory. TerraScan's configuration file TSCAN.CFG contains:

```
#--------------------------------------------------------------
#
#  TSCAN.CFG - TerraScan Configuration File
#
#--------------------------------------------------------------

TSCAN_DATA=$(TERRADIR)data/
TSCAN_LICENSE=$(TERRADIR)license/
TSCAN_MACRODIR=$(TERRADIR)macro/

#Directory for user preferences (user has write access)

TSCAN_PREF=$(TERRADIR)tscan/

#Directory for settings (may point to read-only directory)

TSCAN_SET=$(TERRADIR)tscan/

#Files for settings (may be shared by organization)

TSCAN_ALIGNREP  = $(TSCAN_SET)alrepfmt.inf
TSCAN_OUTFMT    = $(TSCAN_SET)outfmt.inf
TSCAN_TRANSFORM = $(TSCAN_SET)trans.inf
TSCAN_TARGETS   = $(TSCAN_SET)targets.inf
TSCAN_CODES     = $(TSCAN_SET)codes.inf
```

In a default configuration, MicroStation will automatically include these settings as configuration variables. You can use MicroStation's **Configuration** command from **Workspace** menu to check the values for these variables. In case these variables have not been defined correctly, you should define them manually.

- MS_MDLAPPS should include the directory where TSCAN.MA is located.
- TSCAN_DATA defines a default directory for incoming laser points.
- TSCAN_LICENSE should point to the directory where user license TSCAN.LIC is located.
- TSCAN_MACRODIR defines a directory where macros are searched from.
- TSCAN_SET should point to a directory where settings can be stored. This directory may be shared by an organization and the user may lack write access to it.
- TSCAN_PREF should point to a directory where user preferences can be stored. The user must have write access to this directory.
- TSCAN_ALIGNREP defines the file in which alignment report formats are stored.
- TSCAN_OUTFMT defines the file in which output file formats are stored.
- TSCAN_TRANSFORM defines the file in which coordinate transformations are stored.
- TSCAN_TARGETS defines the file in which target object types are stored.
- TSCAN_CODES defines the file in which EarthData code translation table is stored.