## Krypton Help Pages

# Geloc

## April 2003

Research Park Haasrode
Z1 Interleuvenlaan 86
3001 Leuven Belgium
tel.: +32 (0)16 400 410
fax: +32 (0)16 400 386
e-mail: info@krypton.be
http://www.krypton.be

KRYPTON
INDUSTRIAL METROLOGY

# Table of contents

# Table of figures

# Product protection

The software and hardware described in these KRYPTON Help Pages are furnished under a license agreement and may be used only in accordance with the terms of this agreement.

The contents of the KRYPTON Help Pages are strictly confidential and the material described is strict intellectual and industrial property of KRYPTON N.V. Divulgation of any kind of information described in this manual is strictly forbidden.

No warranty is given on damages, including but not limited to loss of profit, and special, incidental, consequential, or other similar claims.

ROCAL, RODYM, GeoLoc and SPACE PROBE are registered trademarks of KRYPTON N.V.
Microsoft and Windows are registered trademarks of Microsoft Corporation.

# 1. Start here

Welcome to the GeoLoc user manual.

GeoLoc is a software package, intended for measurement and localization of geometrical objects.
Its primary use is to supply other Krypton software packages with an easy and consistent tool to localize geometric objects that are in the field-of-view of the Krypton camera, and need to be identified.

This guide is intended for all users working with GeoLoc. This document will guide you through all the features this software has to offer.
This guide will follow a top-down approach, starting off as would a user approaching the system for the first time. Step by step, we'll go through all of the possibilities, starting with the obvious features, and ending with the most powerful and versatile features.

This guide assumes you have a basic knowledge of the Krypton camera system, that all the hardware is installed, and that you have some basic experience with the Windows operating system.

If you are interested in the principles of GeoLoc, please refer to chapter "Introduction" on page 7.

If you want to install the software from scratch, start with the chapter "Installing the GeoLoc software" on page 9.

If you want to make sure that everything is set before the first measurements take place, start with the chapter "Actions before conducting measurements" on page 14.

If you are sure that all preliminary actions are taken, or were not required, start with the chapter "Using GeoLoc" on page 15.

# 2. Introduction

Krypton develops and commercializes camera-based solutions for measuring and evaluating the 6D positions of objects in space. Our systems are based on non-contact measurements using solid-state camera systems. Several software packages are developed by Krypton to support industrial and laboratory position problems, both static and dynamic, but always using the same hardware.

Each camera system consists of 1 or more digital infrared CCD cameras mounted on a fixed beam. Each of the camera's measures an angle of an infrared LED that resides in the camera's field-of-view. By combining the readings from the three cameras, a 3D-position of the LED can be calculated with respect to the camera's reference frame.
If we attach 3 such LED's on an object, we can calculate the 6D-motion of this object by combining the 3D-readings of the three LED's.

GeoLoc is a program that measures and localizes geometrical objects. Software characteristics:

?? Measurement of all geometric objects: planes, lines, points, cylinders, circles, spheres, dynamic frames, …
?? Possible operations: translations, rotations, creation of coordinate systems, …
?? Best-fits on points, point clouds, lines and planes
?? Project-oriented approach in order to allow logical grouping of objects
?? Possibility of "Scripting", teaching macro's to make repetitive measurements easier.

With GeoLoc, the user can measure certain geometrical objects (such as planes, spheres, holes…). By measuring objects with one or more point measurements, the position and orientation of these objects with respect to the camera can be calculated. Simultaneously, certain physical characteristics of the object are calculated (radius, height)

The user can also combine different geometric elements to construct his own coordinate system, e.g. by selecting a measured plane, line and point. GeoLoc will calculate the transformations that allow later measurements to be expressed in the user-specified coordinate system.

In case the user-requested coordinate system cannot be established with easy accessible geometric elements, a "best-fit mapping" might prove useful: a discrete set of points, known in the user coordinate system, will be measured. GeoLoc will try to fit the measured values on the nominal values, yielding a similar transformation from camera coordinate system to user-specified coordinate system.

GeoLoc treats every object as a frame (coordinate system) with additional features or degrees of freedom (e.g. a circle is a frame with 1 undetermined degree of rotation). This means that every object can be used as a reference frame – though circumstances may dictate that this is not a useful thing to do.

"Scripting" allows the user to teach GeoLoc a measurement sequence once, and replaying it many times. This prevents the user from repeated, annoying walks back to the system and keyboard manipulations. Let GeoLoc tell you from a distance what feature to measure, confirm it – and return only to find GeoLoc calculated all the requested elements!

The software runs under Windows 98 or higher versions. A Pentium III 500 MHz processor or better is advised, in order to reduce the calculation time. GeoLoc requires a minimum free hard disk space of 40 Mbyte, and a minimum of 64 Mbyte RAM memory.

The system comes with a CD-ROM containing the installation package of GeoLoc.

This page is only intended for users that are upgrading to a newer version.

## Software changes from version 1.1 to 1.2 :

?? Translations of the software & manuals to French and German

?? An entire alignment or LED file can be imported from an IGES-file. This means that complex alignments can be made using dedicated CMM software, and then be imported in Geoloc. See chapter “Defining geometrical elements” on page 32.

?? Fit point cloud to point: now a point cloud, containing multiple measurements of the same point, can be averaged into 1 point position.

?? Import() specifies reference: when importing elements the frame of reference for this element can be choosen too, and is no longer by default the camera frame. See chapter “Defining geometrical elements” on page 32. Automatic insertion of double quotes when recording a script

?? The cylinder measurement has been modified: the cylinder direction is determined by the direction of the plane, and no longer by measuring the points clockwise or anti-clockwise. See chapter ”Measuring basic geometrical elements” on page 20.

?? Translate() is modified: when translating an element, it will be translated along its own axes and no longer along the axes of the reference frame in which the element is expressed. See chapter “Manipulating & constructing geometrical elements” on 25.

?? Element names are no longer case-sensitive in scripts

?? Scripts return their messages in the tab “messages” and no longer in the separate tab “Script”

?? A number of bugs have been solved.

## Software changes from version 1.0 to version 1.1 :

?? A dynamic frame can be created without a previously defined static frame, or with a non-existent frame. This can be useful when there are very little features to base an alignment on. See chapter “Measuring advanced geometrical objects” on page 21.

?? The script functionality of the function “Change reference” has changed: if an output element is specified, a copy of the original element is created and expressed in another frame of reference. See chapter “Manipulating & constructing geometrical elements” on page 25.

?? Two functions to import and/or export elements and alignments in text files are added to the standard function inventory. See chapter ‘Defining geometrical elements” on page 32.

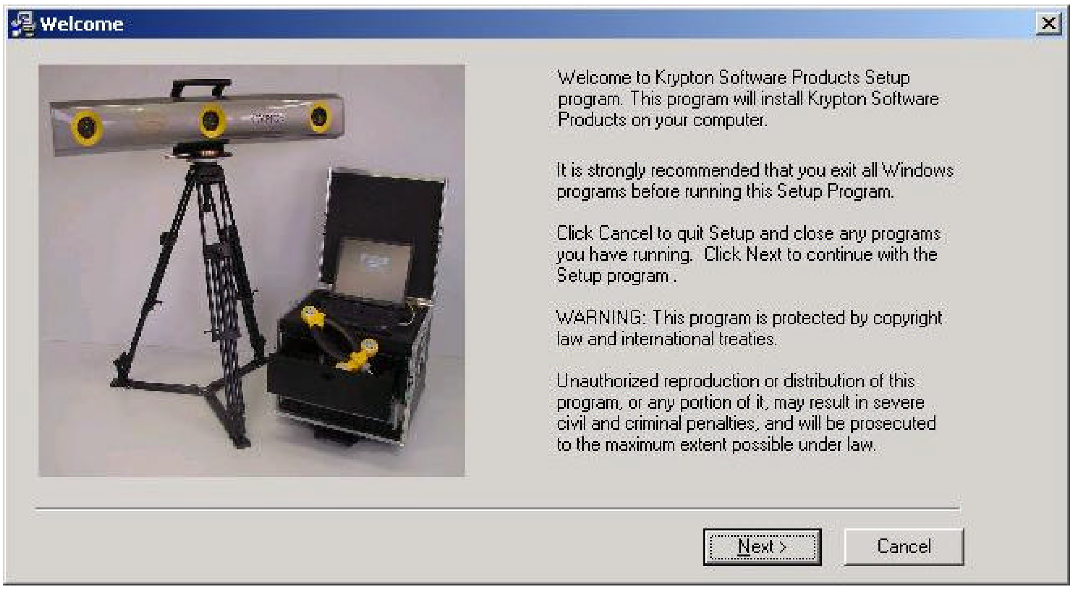
?? When recording a script with the “Record”-button, an existing file will not automatically be overwritten, but GeoLoc will ask whether to overwrite or to append. See chapter “ ‘Macro’ menu ” on page 42.

# 3. Installing the GeoLoc software

This chapter describes the software installation “from scratch” for GeoLoc.

Power up your PC and close all unnecessary applications.

Insert the Krypton installation CD in your CD-ROM drive. The following screen – release date may be more recent – will appear automatically:



Welcome to Krypton Software Products Setup program. This program will install Krypton Software Products on your computer.

It is strongly recommended that you exit all Windows programs before running this Setup Program.

Click Cancel to quit Setup and close any programs you have running. Click Next to continue with the Setup program .

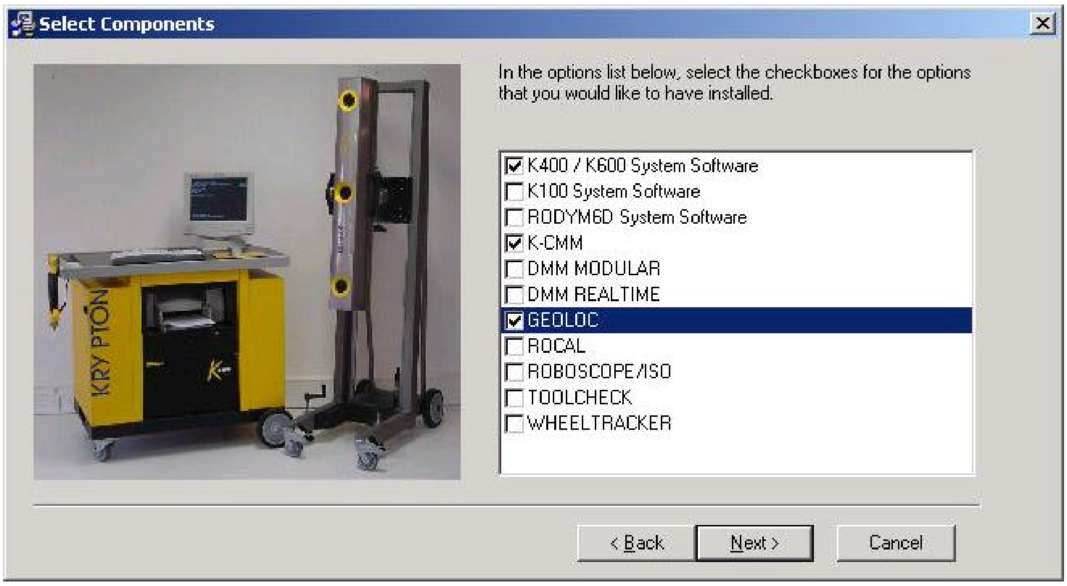
WARNING: This program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.

*Figure 1 : release CD - pop-up form*

If you didn’t get this screen automatically within 30 seconds, use your Windows Explorer to locate the file *SETUP.EXE* on the CD; start this by double-clicking on the file.

Click on the button “*Next*”. A list of software products will appear, sorted per product group:



Select Components

In the options list below, select the checkboxes for the options that you would like to have installed.

☑ K400 / K600 System Software
☐ K100 System Software
☐ RODYM6D System Software
☑ K-CMM
☐ DMM MODULAR
☐ DMM REALTIME
☑ GEOLOC
☐ ROCAL
☐ ROBOSCOPE/ISO
☐ TOOLCHECK
☐ WHEELTRACKER

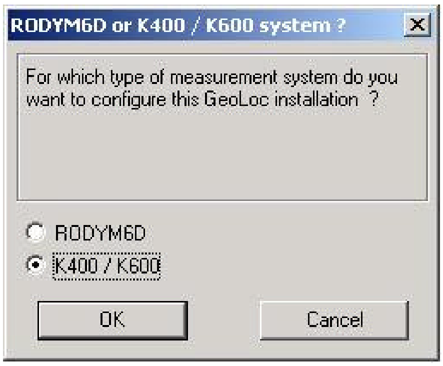
Check the following software packages if they were not already installed on your system; however, we advise you to install these software packages in advance. Please refer to the user manuals of the software packages involved :

?? “K400/K600 system software” (if you have a Kx00 camera system) or “Rodym6D System Software” (if you have a Rodym6D camera system)

?? “K-CMM”

Check “Geoloc” and click “Next”

In the welcome screen, read the instructions as shown on the screen; keep on clicking the “Next” button untill you reach the following screen:



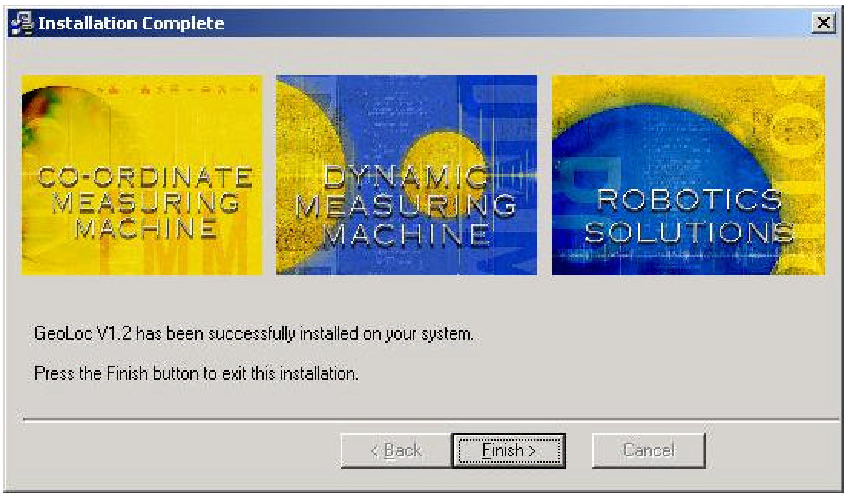
RODYM6D or K400 / K600 system ?

For which type of measurement system do you want to configure this GeoLoc installation ?

○ RODYM6D
● K400 / K600

Select the camera type you currently use ( Kx00 or Rodym6D) and click “OK”.

Follow the instructions on the screen, and keep on clicking “Next”, untill you reach the following screen :



GeoLoc V1.2 has been successfully installed on your system.

Press the Finish button to exit this installation.

Click the “Finish” button to conclude the GeoLoc installation process.
If the main installation program is still active, click the “Finish” button of that program to conclude the entire installation process.

# 4. Actions before conducting measurements

Before measuring, a few steps must be taken. Although they are completely independent from the GeoLoc software, GeoLoc assumes that these steps were carried out before the measurements start. Therefore, please make sure to perform these actions before continuing.

In sequence, these steps are:
1) Configuring the cameras
2) Checking LED visibility and the quality of the measurement environment
3) Calibrating and checking the Space Probe

The following steps must be repeated for each camera PC on the test bench.

## 4.1 Configuring the cameras

If you purchased a Rodym® 6D camera, refer to "*Rodym® 6D system guide*" , chapter "*configuring & operating the Rodym® 6D hardware*"

If you purchased a Kx00® camera, refer to the "K400/K600 Hardware & Software Guide", chapter "Checking hardware & environment"

## 4.2 Checking LED visibility & environment

If you purchased a Rodym® 6D camera, refer to "*Rodym® 6D system guide*" , chapter "*configuring & operating the Rodym® 6D hardware*"

If you purchased a Kx00® camera, refer to the "K400/K600 Hardware & Software Guide", chapter "K-Check: LED visibility & environment"

## 4.3 Checking the Probe

If you purchased a Rodym® 6D camera, refer to "*Rodym6D utilities user guide*" , chapter "*using the Rodym® 6D utilities*"

If you purchased a Kx00® camera, refer to the "K400/K600 Hardware & Software Guide", chapter "ProbeCheck: checking Probe functionality"

## 4.4 Calibrating the Probe

If you purchased a Rodym® 6D camera, refer to "*Rodym6D utilities user guide*" , chapter "*using the Rodym® 6D utilities*"

If you purchased a Kx00® camera, refer to the "K400/K600 Hardware & Software Guide", chapter "ProbeCal: calibrating the Probe"

# 5. Using GeoLoc

This chapter will describe all options and features of GeoLoc.
This chapter is written assuming that all default values for directories, program groups and such where chosen during installation.

First, we'll take a closer look into the GeoLoc user interface, so we know what information can be found or expected in what screen location.
Next, we'll discuss the left-vertical operation bar: all geometrical operations we can perform.
Finally, we'll discuss possibilities of the main menu.

## 5.1 The camera driver

GeoLoc uses a camera driver to obtain his coordinates from. This driver might pop in front of the GeoLoc user interface during the measurement, and might be useful from time to time since he gives feedback on the Probe, stability of the last measurement, …

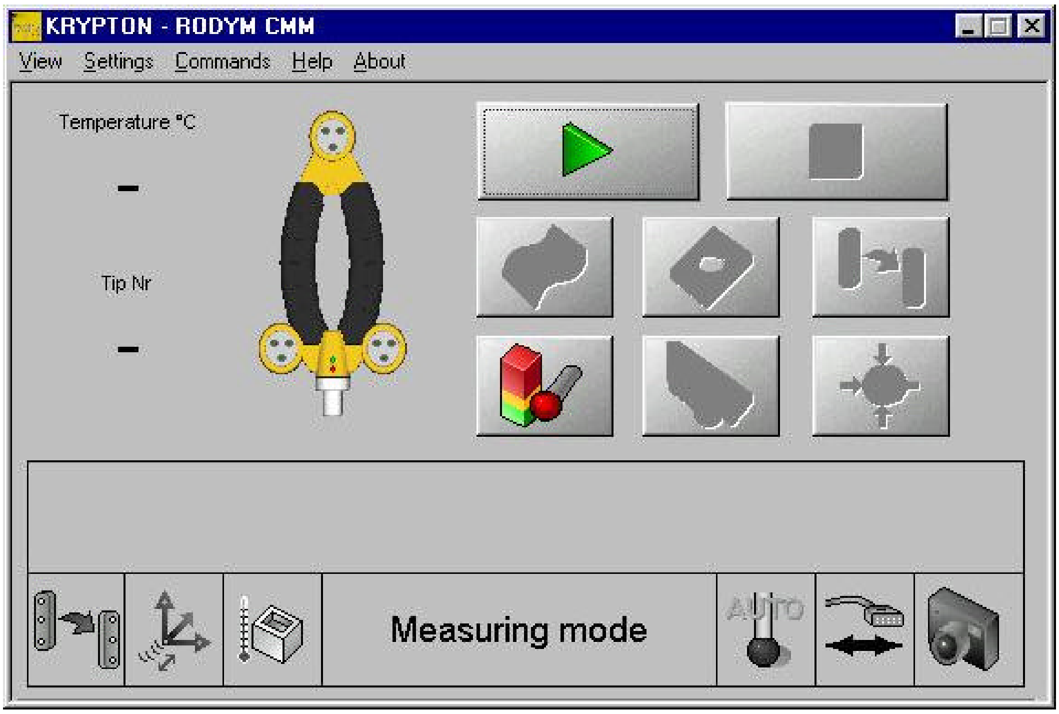


*Figure 2: camera driver - the user interface*

In general, this driver can be neglected.
Although it is possible to start or stop this driver manually, it is not advised to do so: GeoLoc will handle all start- and stop control over this driver. Should the driver pop in front of GeoLoc, you can manually click on the GeoLoc interface somewhere to restore GeoLoc as foreground-running task.

## 5.2 The GeoLoc user interface

In this chapter, we'll look at the GeoLoc user interface, and the division into different zones. The use-and-yield of each zone is independent from the chronology of operations and is therefore discussed as first topic.

Open the program group "*Krypton*". Start "*GeoLoc*". The following screen appears:

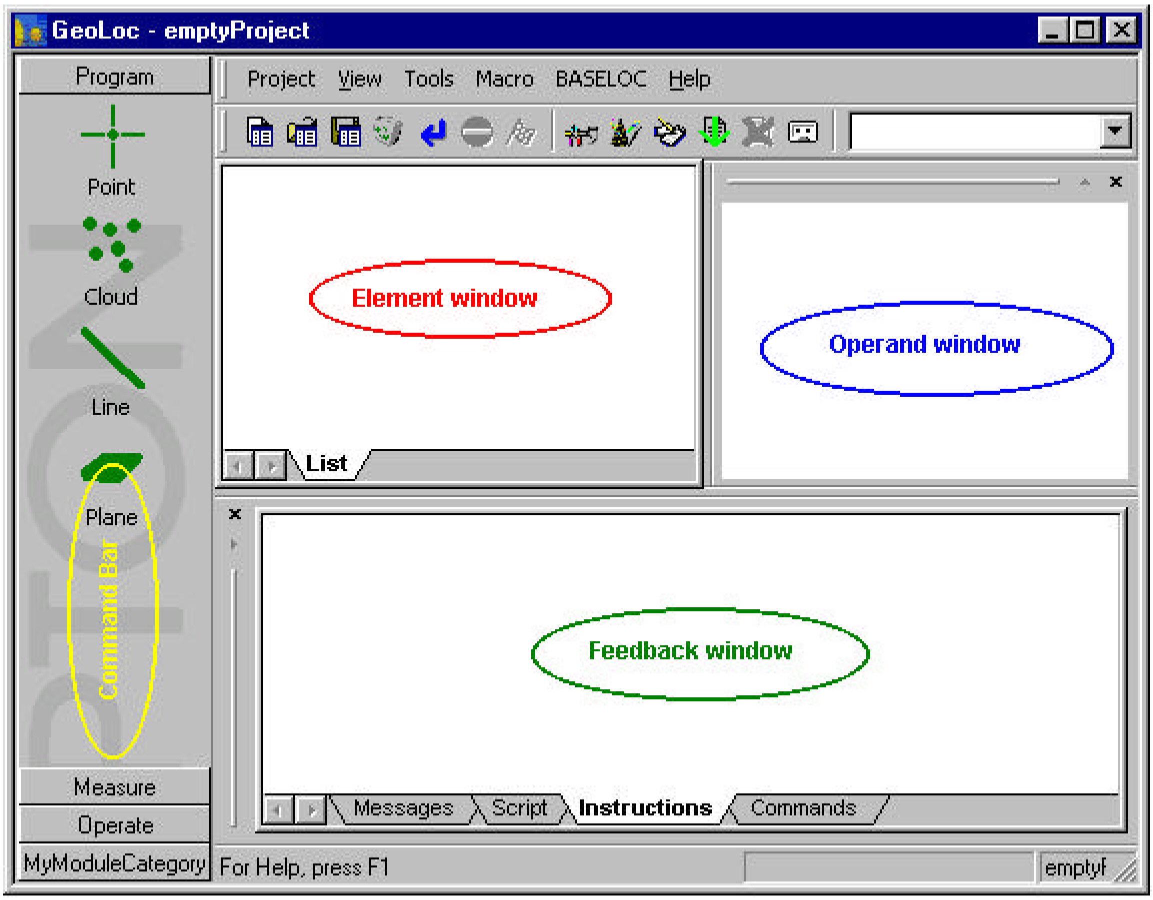


*Figure 3: GeoLoc - the user interface*

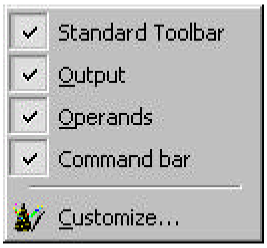
This is the GeoLoc user interface.
The interface is divided into 5 different zones, each with a specific function:

1. **main menu / icon bar** on top. Through this menu, all options, configurations and features are accessible. Each option, configuration or feature corresponds to an icon that acts as a shortcut. Each of these icons can be dragged onto the icon bar or can be removed.
2. "**Command Bar**" (yellow) on the left: this bar holds in fact 3 slider bars, where each slider bar holds a number of elements or operations. By clicking an element or operation with the mouse, the element gets measured or the operation gets executed. **Every manipulation that involves geometrical elements – be it defining, measuring or constructing one – can be found in one of these three slider bars.** All operations are sorted by nature: clicking on the button "*Program*" lists all geometrical elements that can be defined prior to a best-fit alignment ; clicking on the button "*Measure*" lists all geometrical elements that can be measured ; clicking on the button "*Operate*" lists all operations and manipulations on geometrical elements.
3. "**Element Window**" (red) in the middle: this window is the reservoir in which all generated elements are gathered. **Every geometrical element that is defined, measured or constructed is listed in the element window.** Geometric elements, frames, point clouds, … everything that is defined, measured or calculated is displayed in this window with an appropriate icon and a name.
4. "**Operand window**" (blue) on the right: this window holds the items to be used in an operation. Whenever an operation is selected, and some items from the element window were dragged into the operand window, these items will be used as parameters for this operation. The operand window is also the "transit buffer", to be used when items must be teleported to another software package. **Every geometrical item that serves as input for an operation or an export must be dragged into the operand window.** (However, if an operation is performed

*without* parameters in the operand window, GeoLoc will ask for them with a pop-up window with all parameters listed in combo boxes)

5. The "**Output window**" (green) at the bottom: this window holds all feedback from the program to the user. The window has 3 tabs:
   - Messages: these are general messages, warnings or error reports from GeoLoc
   - Script: this tab contains all script-related messages
   - Instructions: this window echoes all directive messages from the scripts.

If you accidentally closed one of the windows, move the mouse over the main menu, and right-click with the mouse : a menu will appear : check the appropriate window to be displayed again



### 5.2.1 The main menu / icon bar

The main menu / icon bar is discussed in chapter "Options in the main menu" on page 36

### 5.2.2 The command bar

The command bar is discussed in chapter "Measuring, constructing & defining geometric elements" on page 19.
The element window will list all measured, constructed and defined geometrical elements that have been created using GeoLoc. Every element will be shown with its own icon and name:

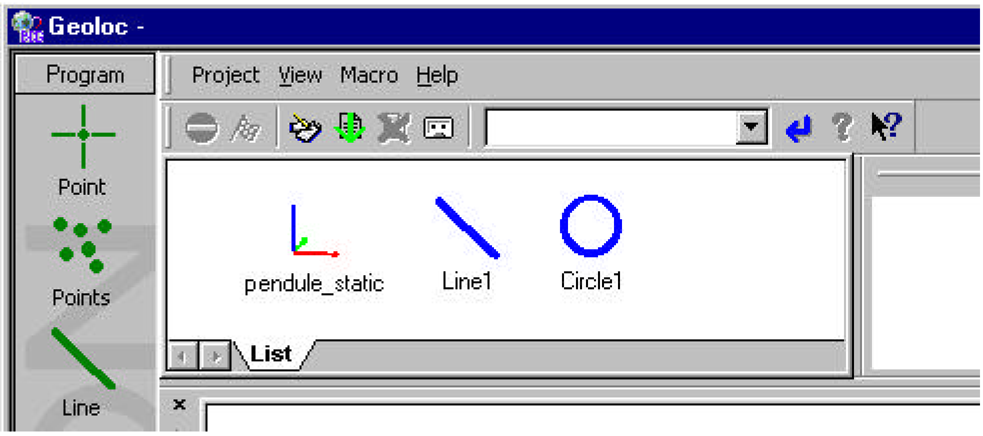
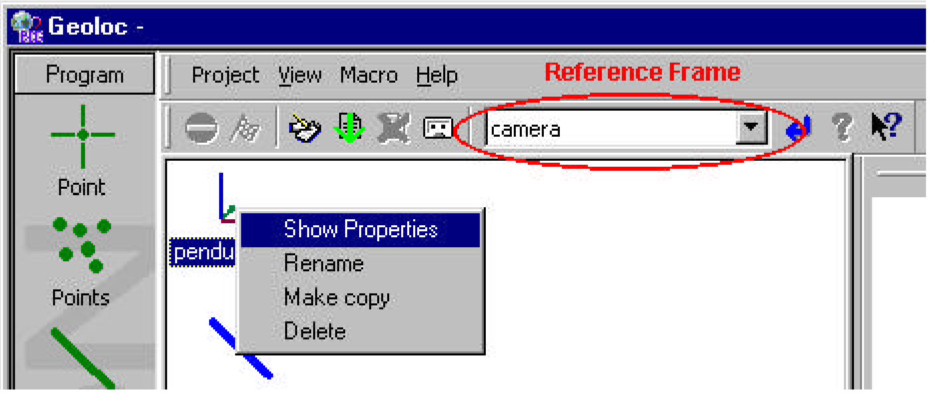
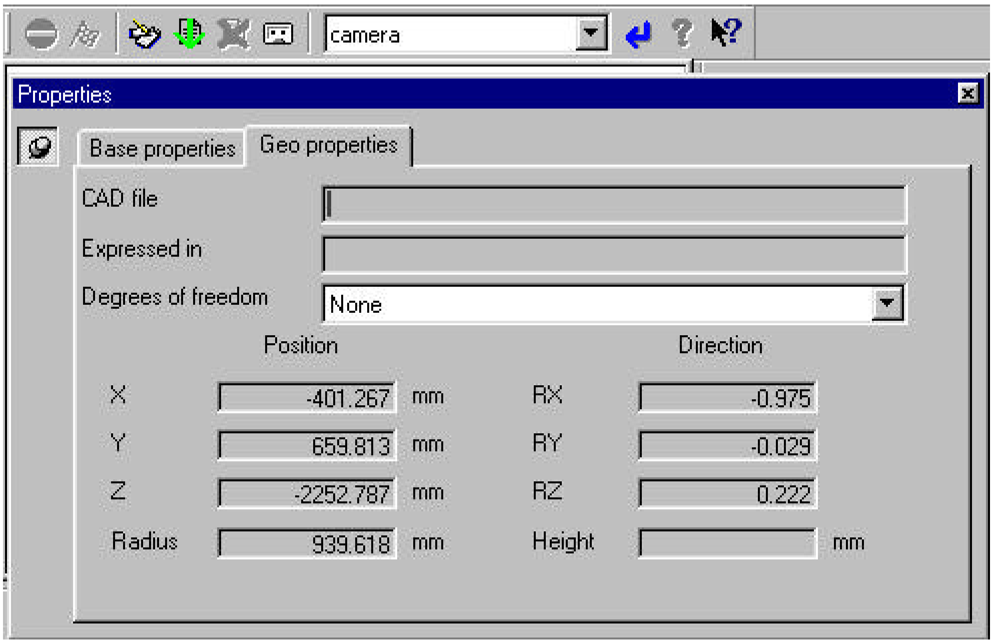


*Figure 4: GeoLoc - the element window*

Measured or constructed items have an icon that is mainly blue; defined items are green. By right-clicking on an item, a number of actions can be performed with this item:



- **Delete**: remove the item from the element list. (The item will not be removed physically, but will be stored in GeoLoc's own 'Deleted items' bin)
- **Make copy**: makes a copy of the original element
- **Rename**: changes the name of the element
- **Show Properties**: this option will show a pop-up window, displaying some general information about the object, and geometrical information about the element with respect to the reference frame. This reference frame can be chosen on-line from the combo box at the top (indicated in red):



CAD file:
Expressed in:
Degrees of freedom: None

| | Position | | | Direction | |
|---|---|---|---|---|---|
| X | -401.267 | mm | RX | -0.975 | |
| Y | 659.813 | mm | RY | -0.029 | |
| Z | -2252.787 | mm | RZ | 0.222 | |
| Radius | 939.618 | mm | Height | | mm |

*Figure 5: GeoLoc - element properties window*

***Element holds position ( -401.267, 659.813, -2252.618 ) with respect to the camera coordinate system***

### 5.2.4 The operand window

The operand window holds the items to be used in an operation. Whenever an operation is selected, and some items from the element window were dragged into the operand window, these items will be used as inputs for this operation.

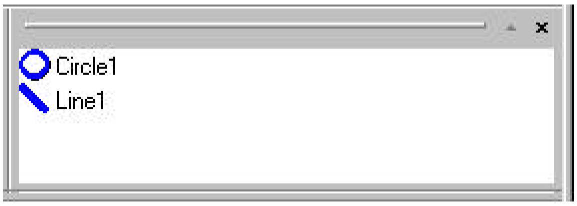


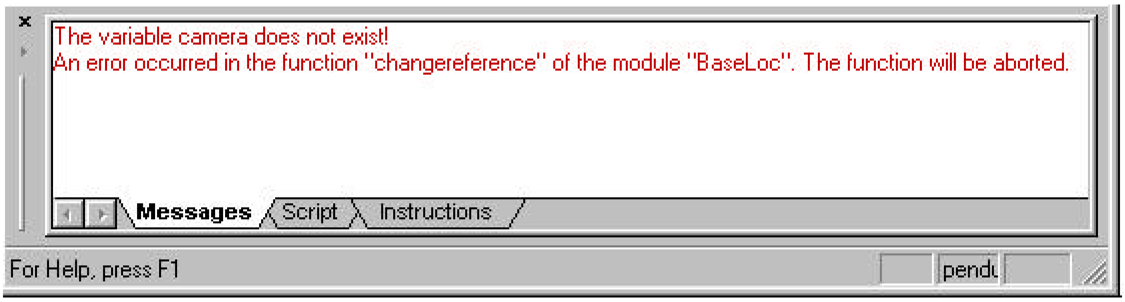
*Figure 6: GeoLoc - the operand window*

***Choosing the operation 'intersect' would calculate the intersection of Circle1 with Line1. The resulting point would be added in the elements window***

The operand also holds the elements that will be teleported to external programs if the user clicks the 'Return'-button (see 'Return'-function in chapter "'Project' menu" on page 37).

### 5.2.5 The output window

The output window will hold all messages from GeoLoc for the user. These messages can be an informational message (e.g. reporting the residual error after a best-fit), a warning message (e.g. inaccurate calculation) or errors (e.g. an error occurred in the module 'BASELOC'):



The variable camera does not exist!
An error occurred in the function "changereference" of the module "BaseLoc". The function will be aborted.

*Figure 7: GeoLoc - the output window*

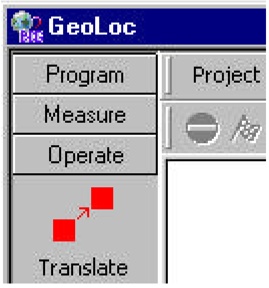
The messages will appear in different tab sheets, depending on the context. GeoLoc itself will activate the proper tab for you when a new message arrives.

## 5.3 Measuring, constructing & defining geometric elements

This chapter focuses on the key actions of GeoLoc: how do I measure geometrical objects, how do I make alignments, how can I perform a best fit.

All operations are grouped in 3 vertical slider bars; they are sorted by nature: clicking on the button "*Program*" lists all geometrical elements that can be defined prior to a best-fit; clicking on the button "*Measure*" lists all geometrical elements that can be measured; clicking on the button "*Operate*" lists all operations and manipulations on geometrical elements.

Every time the user clicks on one of these buttons, the appropriate menu will unfold, covering the other two.



GeoLoc has the ability to auto-detect new calculation modules, and to add the functions from these modules to the menu or the icon bar.

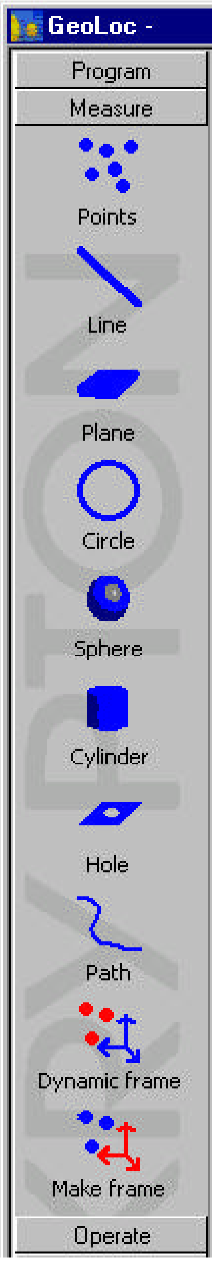
Some of these modules can also add a button and corresponding slider bar. This does not change the functionality of the original slider bars.

### 5.3.1 Measuring basic geometrical elements

With the "measure" command bar, the user can measure a number of points that GeoLoc will try to fit into a 'geometric primitive' shape, such as a plane, a cylinder, a sphere, … The last three options on the "measure" command bar are not as straight-forward as the other elements, so they will be discussed in chapter "Measuring advanced geometrical objects" on page 21

GeoLoc treats every element as a frame (coordinate system with an origin and orientation), which are shown when the properties of an element are reported (see "The element window" on 18).

For each geometric primitive, two settings can be configured to control the fitting:
1) The number of points
2) The tolerance (max. residual error allowed during fitting)
These settings can be configured in the menu "**Tools**" – "**Options**" and selecting the tab "**Measurement options**"
The geometric primitives currently available are:



**Points**: a set of discrete points, used for fitting
Origin: N/A
Direction: N/A
Tolerance: applicable only when fitting a measured point cloud on a defined point cloud
No probe tip radius compensation

**Line**: a straight line through at least two points
Origin: N/A
Direction: positive from the first point towards the last point
Tolerance: applicable when more than 2 points are fitted
No probe tip radius compensation

**Plane**: a plane through at least three points
Origin: N/A
Direction: perpendicular to the plane. Positive sense is equal to positive Z-vector of the measurement probe
Tolerance: applicable when more than 3 points are fitted
Probe tip compensation: opposite to the positive Z-vector

**Circle**: a circle through at least three points
Origin: circle center
Direction: perpendicular to the plane. Positive sense is determined by the point measurement sequence (right-hand rule)
Tolerance: applicable when more than 3 points are fitted
No probe tip radius compensation

**Sphere**: a sphere through at least four points
Origin: sphere center
Direction: N/A
Tolerance: applicable when more than 4 points are fitted
Probe tip compensation: always to the inside of the sphere

**Cylinder**: a cylinder through at least six points. The first *n* points determine the cylinder's base plane; the last three or more are measured on the shaft. The cylinder is assumed to be perpendicular on the plane.
Origin: N/A
Direction: perpendicular to the plane. Positive sense is equal to positive Z-vector of the measurement probe
Tolerance: applicable when more than 3 points are fitted on

either plane or shaft.

Probe tip compensation: opposite to the positive Z-vector

NOTE:

*n* equals the number of points, configured for a normal plane. The number of points, configured for a cylinder, *only* refers to the points taken on the cylinder shaft!

If a plane was dragged into the operand window before measuring the cylinder, GeoLoc will *still* ask to measure the plane, contrary to a hole measurement

**Hole**: center of a bore in a surface. First, a temporary plane is measured on the surface. Then the probe sinks into the bore where a 'centerpoint' measurement is taken. This point is then projected on the measured plane.

?? Origin: bore center

?? Direction: N/A. The temporary plane has a direction: see "Plane"

?? Tolerance: applicable when more than 3 points are fitted into a plane.

?? Probe tip compensation: not necessary. For the temporary plane, the compensation is the same as for a "Plane"

NOTE:

?? To measure the plane, the settings for the geometric primitive 'plane' do NOT apply! A hole has different settings: "small hole -> plane" and "small hole -> centerpoint"

?? If a previously measured plane was dragged into the operand window prior to starting the hole measurement, GeoLoc will skip the plane measurement and project the centerpoint on the plane in the operand window.

During the measurement, GeoLoc will show the number of points to take in the output window, and how many of these points are already taken. E.g.: 3 points of the expected 5 are already measured.

3 / 5

### 5.3.2 Measuring advanced geometrical objects

The last three items on the "measure" command bar are powerful items that do not require the use of the Space Probe, but only of individual LED's. Nevertheless, they yield valuable information that can be used in a number of Krypton products.
Each item also requires more configuration than the number-of-points and the tolerance, as did their basic counterparts.
These items are a **path**, a **dynamic frame** and a **make frame**.

**Path**: a point cloud of subsequent LED positions. If multiple LED's are configured, the subsequent position of each LED is ordered into *channels*.

- Origin: N/A
- Direction: N/A
- Tolerance: N/A

When a path measurement is activated, a pop-up form appears.

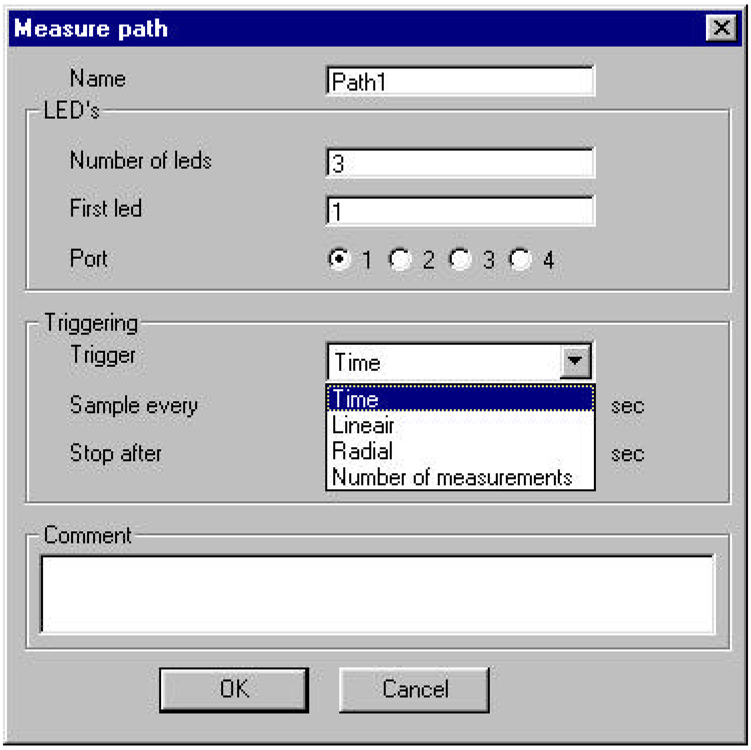
The parameters are:

- **Name**: name of the generated point cloud
- **Port**: strober port onto which the strober of the LED's is connected
- **First LED**: sequence number of the first LED of the cloud
- **Number of LED's** number of LED's in the cloud to be measured
- **Trigger**: select 1 of 4 possible trigger types to make a snapshot of the LED cloud
- **Sample every …** : depending on the type of trigger, this option says when exactly to take the snapshot, e.g. after every 5 mm movement, or every 2 seconds, …

- **Stop after:** stop criterium, based on the type of trigger. The “sample every” parameter and the stop criterium together determine the number of snapshots that are taken
- **Comment**: (optional) description of the element

Clicking “OK” will start the controller, and will command the camera to look for the LED’s. Should any of the LED’s not be visible, the measurement is halted until all LED’s are visible. GeoLoc reports the invisibilities in the feedback window.

Note: when using a path-generated point cloud in typical fittings, the ordering in channels is discarded.



Measure path

Name: Path1

LED's

Number of leds: 3

First led: 1

Port: ● 1 ○ 2 ○ 3 ○ 4

Triggering

Trigger: Time

Sample every ... sec

Stop after ... sec

Comment

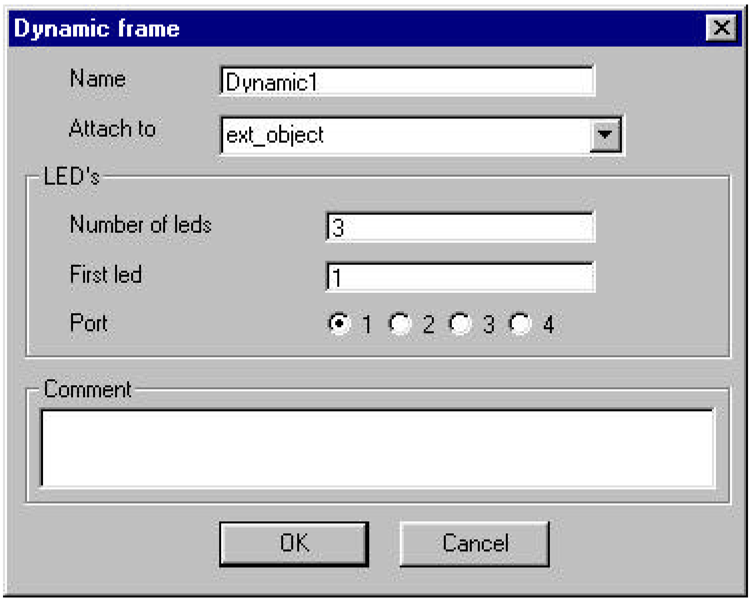
*Figure 8: GeoLoc - path measurement form*

**Dynamic frame**: a dynamic frame is a cloud of **static** LED positions with respect to a chosen external coordinate system. These LED’s are assumed to be rigidly mounted onto the coordinate system. If the camera can see the LED’s in a later stadium, the software can calculate the momentary position of the external coordinate system. This option is used in dynamic measurements where 6D-motions of objects are measured.

When a dynamic frame measurement is activated, a pop-up form appears.

The parameters are:

- **Name**: name of the generated dynamic frame
- **Attach to**: name of the external coordinate system. This frame must be measured before the dynamic frame is measured. If no static frame is specified (box empty), or a non-existent static frame, a frame is created through the first three LED’s (see notes below)
- **Port**: strober port onto which the strober of the LED’s is connected
- **First LED**: sequence number of the first LED of the cloud
- **Number of LED’s** number of LED’s in the cloud to be measured
- **Comment**: (optional) description of the element



Dynamic frame

Name: Dynamic1

Attach to: ext_object

LED's

Number of leds: 3

First led: 1

Port: ● 1 ○ 2 ○ 3 ○ 4

Comment:

*Figure 9: GeoLoc - dynamic frame form*

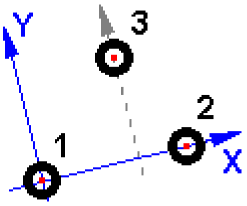
A dynamic frame has two additional parameters in the menu "**Tools**" – "**Options**":

- **Number of points**: number of snapshots over which each LED gets averaged.
- **Tolerance**: if the dynamic frame is used later on to retrieve the position of the coordinate system, the residual mapping error cannot exceed this value. It is a check on the integrity of the frame.

***Notes:***

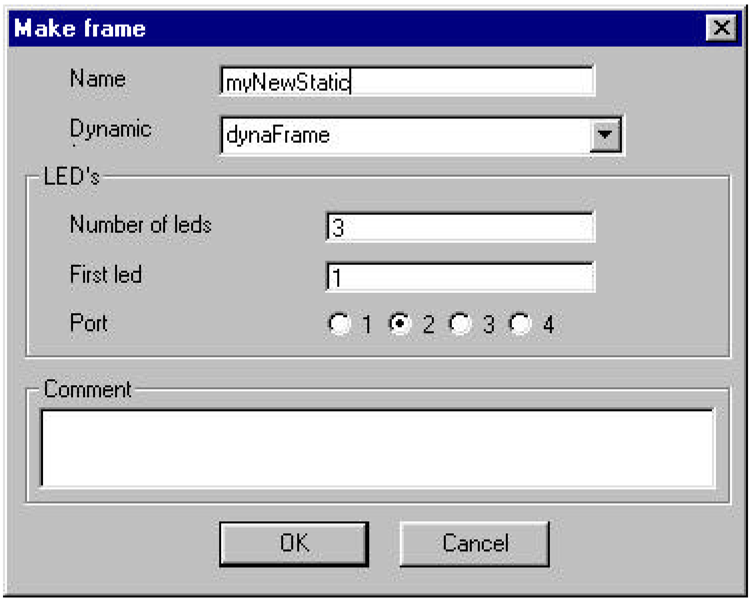
If no static frame was specified, GeoLoc will construct a frame using the first three LED's that are configured. The origin will be LED1. The X-axis leads from LED1 to LED2. The Y-axis is perpendicular to the X-axis, pointing to LED3 through LED1. The Z-axis can be found with the right-hand rule.

If the element, entered in the "Attach to" field does not exist, a static frame with this name is constructed as described above.



**Make frame**: make frame is actually the reverse of 'dynamic frame': it converts the momentary position of a dynamic frame into a static coordinate system. This feature is useful when a dynamic frame was identified in an environment other than the object's initial measurement position.

When a 'make frame' measurement is activated, the following form pops up:


Make frame

Name: myNewStatic

Dynamic: dynaFrame

LED's

Number of leds: 3

First led: 1

Port: ○ 1 ● 2 ○ 3 ○ 4

Comment

*Figure 10: GeoLoc - dynamic frame form*

The parameters are:

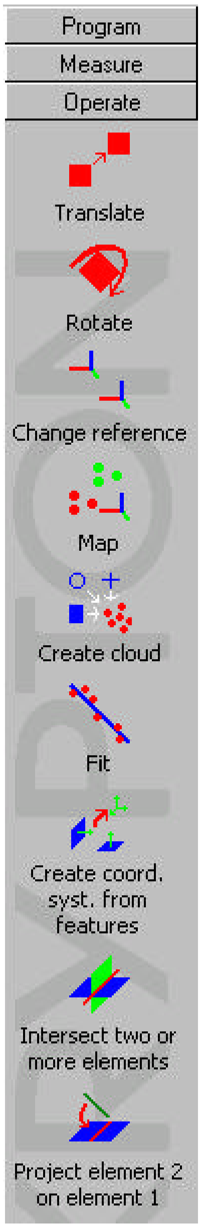
- **Name**: name of the generated dynamic frame
- **Dynamic**: name of the dynamic frame whose momentary position will be copied into <name>. This frame must be measured before the dynamic frame is measured.
- **Port**: strober port onto which the strober of the LED's is connected
- **First LED**: sequence number of the first LED of the cloud
- **Number of LED's**: number of LED's in the cloud to be measured
- **Comment**: (optional) description of the element

A 'make frame' has two additional parameters in the menu "**Tools**" – "**Options**":

- **Number of points**: number of snapshots over which each LED gets averaged.
- **Tolerance**: the fitting of the current LED positions should not yield a residual error exceeding this tolerance (compared with the LED positions measured at the time the dynamic frame was acquired)

### 5.3.3 Manipulating & constructing geometrical elements

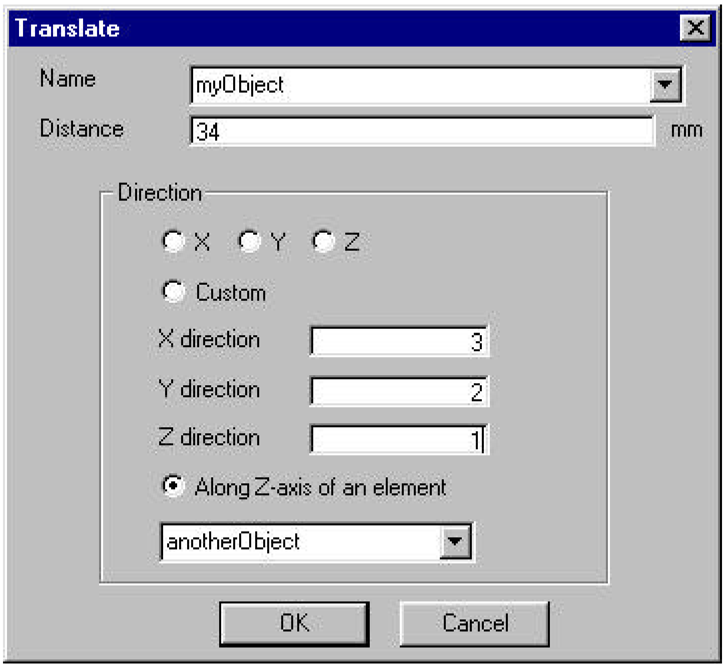
With the “Operate” command bar, the user can manipulate existing objects, or can create new geometrical elements based on existing elements. This way, geometrical features that cannot be measured directly can be identified indirectly.


Program
Measure
Operate
Translate
Rotate
Change reference
Map
Create cloud
Fit
Create coord. syst. from features
Intersect two or more elements
Project element 2 on element 1

Theoretically, every operation needs a number of input parameters and yields 1 output parameter. However, if the operand window contains geometrical elements before the operation is initiated, the operation will use those elements as input parameters and yield an output with an automatically generated name. The first element in the operand window (highest positioned element) will be parameter 1; the second will be parameter 2, etc. In some circumstances, the use of the operand window is not advised in order not to create ambiguity.

This chapter will explain every operation in detail, with its objective, in- and output parameters. Operations will not necessarily be discussed in order of appearance, but in logical sequence. Every operation is shown as if no parameters are passed through the operand window.

**Translate:** this operation translates an element over a certain distance. The element will be translated along its own axes.


Translate

Name: myObject

Distance: 34 mm

Direction

○ X ○ Y ○ Z

○ Custom

X direction: 3

Y direction: 2

Z direction: 1

● Along Z-axis of an element

anotherObject

*Figure 11: GeoLoc - 'Translate' form*

Parameters:

?? **Name**: name of the object to translate

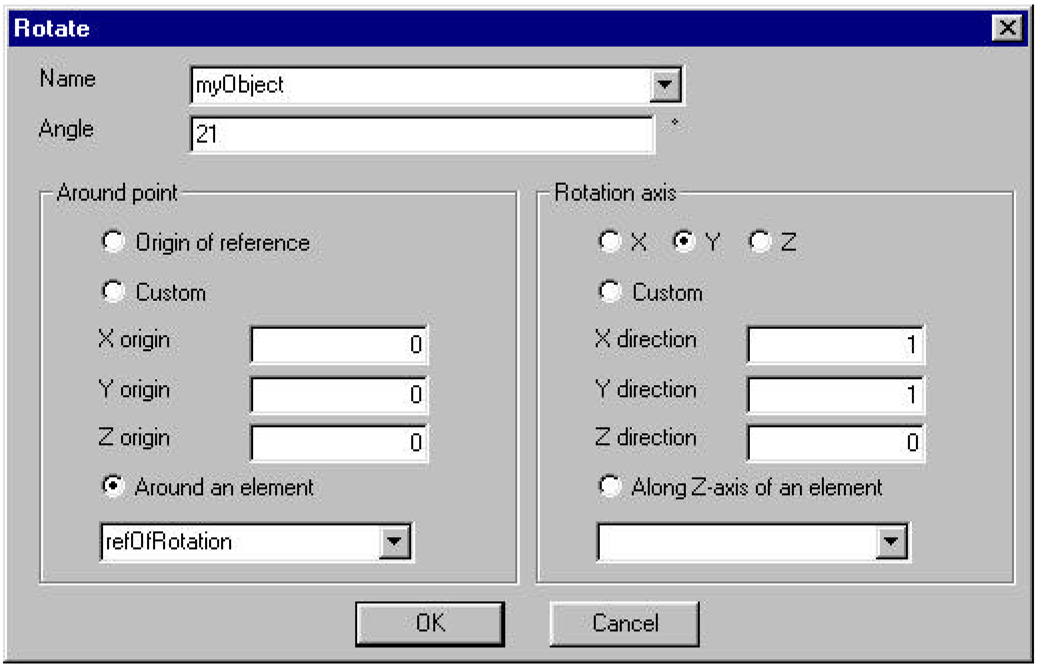
?? **Distance**: distance, positive or negative, over which to translate the object.

?? **Direction**: direction to translate the object along. The user has three possibilities:

> ?? Pure X, Y or Z along the axes of the coordinate system in which the object is expressed.
>
> ?? ‘Custom’: translation according to entered weight factors for each axis of the coordinate system in which the object is expressed.
>
> ?? Along the Z-axis of another selected object. When working with weight factors, GeoLoc will translate in each direction, taking into account the ratio between that direction and the weight of the 3D-motion

For example:
The weight of the 3D-motion is ? ( $3^2 + 2^2 + 1^2$ ) = ? 14 = 3.74
The weight for the Y-motion is 2
Therefore, the object will be moved over 34 mm x ( 2 / 3.74 ) = 18.18 mm in Y
Similar calculations count for X and Z-motion.

**Rotate:** this operation rotates an element over a certain angle, around an axis parallel to a chosen axis. The parallel axis runs through a chosen rotation point.



Rotate

Name: myObject
Angle: 21 °

Around point
○ Origin of reference
○ Custom
X origin: 0
Y origin: 0
Z origin: 0
● Around an element
refOfRotation

Rotation axis
○ X ● Y ○ Z
○ Custom
X direction: 1
Y direction: 1
Z direction: 0
○ Along Z-axis of an element

*Figure 12: GeoLoc - 'Rotate' form*

Parameters:

- **Name**: name of the object to rotate
- **Angle**: angle over which to rotate the element. A positive angle rotates the object counter-clockwise (right-hand rule); a negative angle rotates the object clockwise.
- **Around point**: the point that acts as rotation center. The user has three possibilities:
  - ?? Origin of reference: this is the origin of the coordinate system in which the object was expressed (often the camera coordinate system)
  - ?? 'Custom' coordinates of a point, expressed in the coordinate system in which the object is expressed (often the camera coordinate system)
  - ?? Around an element
- **Rotation axis**: the axis to which a parallel will be constructed through the rotation point, around which the rotation will be executed. The user has three possibilities:
  - ?? Pure X, Y or Z around the axes of the object.
  - ?? 'Custom': rotation according to entered weight factors for each axis of the coordinate system in which the object is expressed.
  - ?? Around the Z-axis of a other, selected, object

When working with the X, Y and Z-axes, the axes of the element itself are taken (NOT the axes of its reference frame).

When working with weight factors, GeoLoc will construct an axis, taking into account the ratio between the weight for that direction and the total weight. The resulting ratios form the components of a unity vector on the requested axis.

For example:
The weight of the all the axes is ? ( $1^2 + 1^2 + 0^2$ ) = ? 2 = 1.41
The weight for the Y-axis is 1
The ratio is 1/1.41 = 0.707
X has the same ratio, Z has ratio 0
The unity vector is the XY-bissectrice in the XY-plane. The object will be rotated over 21° around this bissectrice.

**Note:** when asking for the 6D-properties of an element after the rotation (see chapter "The element window" on 18), the displayed rotation angles must be interpreted as follows:

- Roll-Pitch-Yaw: the consecutive rotations are around the axes of the reference frame
- Steer-Camber-Spin: the consecutive rotations are around axes that got rotated themselves ( camber- and spin axes follow the object in a steer rotation. The spin axis follows the object in a camber rotation)

**Change reference:** this operation recalculates the position and orientation of an element in a given static reference to another static reference.

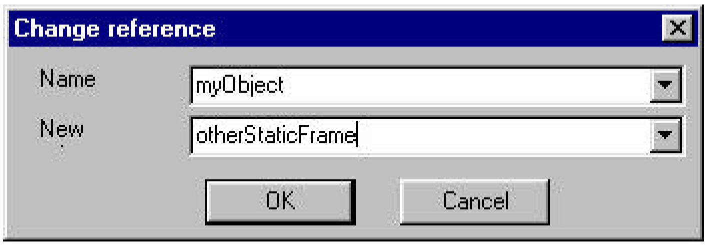

Change reference
Name: myObject
New: otherStaticFrame

*Figure 13: GeoLoc - 'Change reference' form*

Parameters:

- **Name**: name of the object to express into another reference frame
- **New**: new frame of reference (no need to enter the old reference frame – GeoLoc knows in which frame the object was initially expressed).

***Warning!*** The object itself will be affected by the operation! Only if the function "Change reference" is used in a script (see chapter "Scripting – writing and using macro's" on page 43), an output parameter can be specified, e.g.:

*["myCopy"] = ChangeReference("myObject", "otherStaticFrame")*

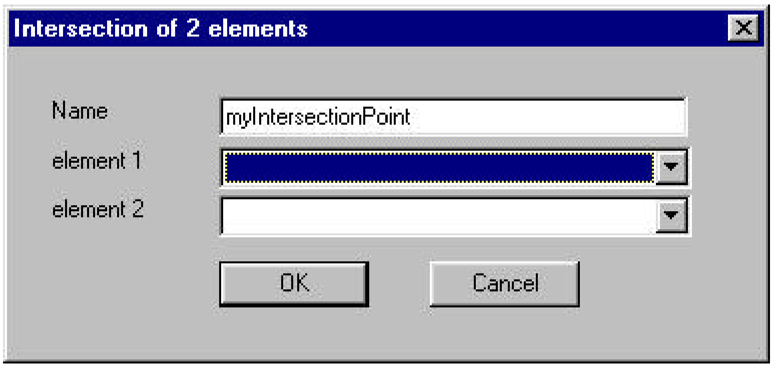
In this case, first an exact copy *myCopy* of element *myObject* is made; next, *myCopy* will be expressed in *otherStaticFrame* instead of *myObject*.

**Intersect two or more elements:** this operation intersects two or more geometric elements and generates the intersection element. The type of the element depends on the intersecting elements:

?? 2 intersecting planes yield a line, but 3 intersecting planes yield a point
?? A plane and an intersecting line yield a point, and two intersecting lines yield a point.

Parameters:

- **Name**: name of the constructed element
- **Element 1**: first element that participates in the intersection.
- **Element 2**: second element that participates in the intersection.



Name: myIntersectionPoint
element 1:
element 2:

*Figure 14: GeoLoc - 'Intersection' form*

**Notes:**

- Intersection: with this pop-up only allows an intersection of two elements at the same time. Intersecting by using elements previously dropped in the operand window does not suffer this inconvenience. Intersections rarely suffer from ambiguities, so intersecting via the operand window in general is recommended.
- In case the yielded element has a direction (such as a line), the direction can be predicted as follows: the positive direction of the yielded element is indicated by the right-hand rule when rotating from element1 to element2. (If you want to enforce this while intersecting with the operand window, make sure element1 is the uppermost element in the operand window)

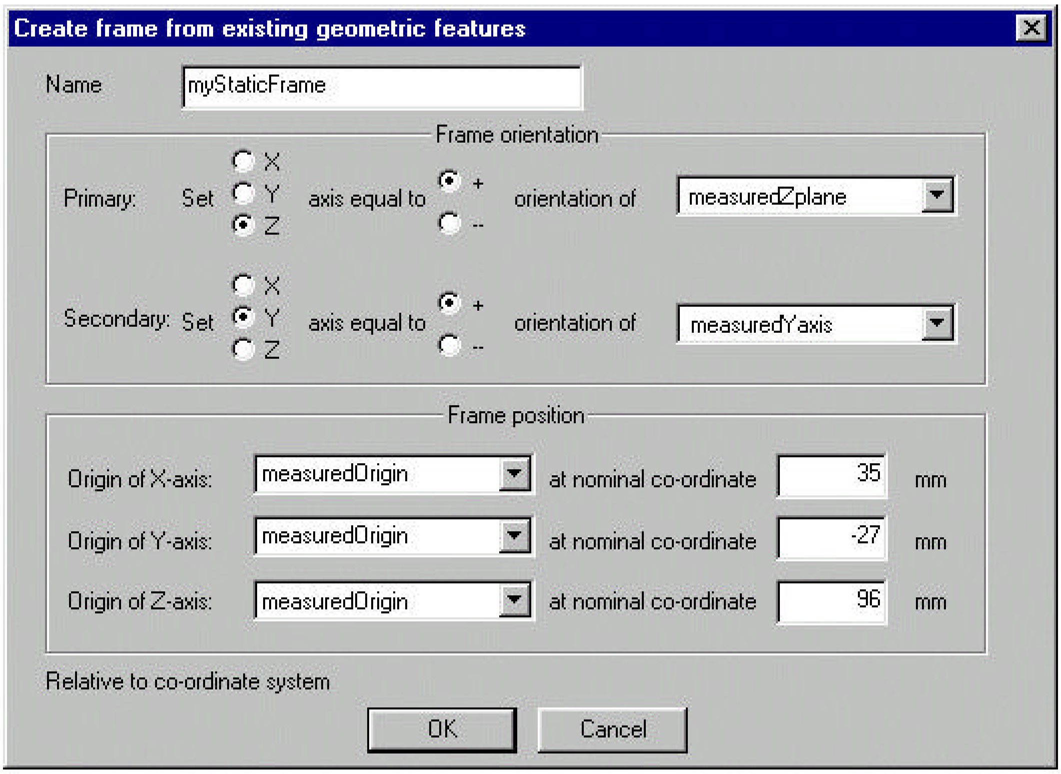
**Create coord. Sys. from features:** this operation allows the user to construct a static coordinate system (or static frame), using previously measured geometrical elements such as planes, lines and points. This feature corresponds to a classical alignment as commonly known in CMM applications. This frame can be used to express other elements in, or to attach a dynamic frame to.

In order to establish an alignment (that is, determine the transformation from the camera coordinate system to the user-chosen coordinate system), we first select a primary orientation; this is a measured axis, which we assume to be an axis of our user-chosen coordinate system.
Next, we select a secondary orientation. This is a second measured axis, which will serve as second axis of our user-chosen coordinate system. If the second axis is not perpendicular to the first axis, she is projected first on the plane perpendicular to the primary axis (keep this in mind when choosing a secondary axis).
Once we know two perpendicular axes, the 3rd axis is known too, since she is perpendicular to the first and second axis.
Finally, we must translate our constructed reference frame to a chosen origin.

For obvious reasons, the measured elements, selected as primary and secondary orientation, must be elements with a direction (planes, lines, circles). The frame position can be an element without a direction (point).

Parameters:

- **Name**: name of the new static reference frame
- **Primary: set <*d*> axis to <*s*> orientation of <*e*>**: configures our new reference primary axis:
- <*d*> determines whether we'll choose the X, Y and Z axis to be parallel to our primary orientation
- <*s*> determines whether this axis has the same direction (+) or the opposite direction (-) as the primary orientation
- <*e*> selects the measured element, which is chosen to be our primary orientation.



Create frame from existing geometric features

Name: myStaticFrame

Frame orientation

Primary: Set ○ X ○ Y ● Z axis equal to ● + ○ - orientation of measuredZplane

Secondary: Set ○ X ● Y ○ Z axis equal to ● + ○ - orientation of measuredYaxis

Frame position

| | | | | |
|---|---|---|---|---|
| Origin of X-axis: | measuredOrigin | at nominal co-ordinate | 35 | mm |
| Origin of Y-axis: | measuredOrigin | at nominal co-ordinate | -27 | mm |
| Origin of Z-axis: | measuredOrigin | at nominal co-ordinate | 96 | mm |

Relative to co-ordinate system

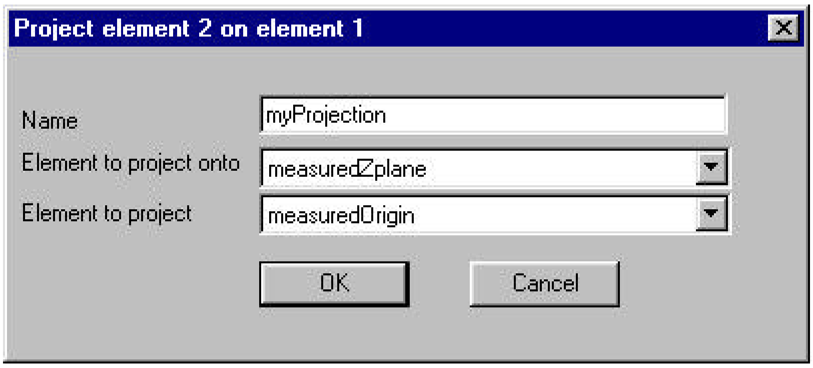
*Figure 15: GeoLoc - 'create coord. sys. from features' form*

- **Secondary: set <*d*> axis to <*s*> orientation of <*e*>**: configures our new reference secondary axis:
- <*d*> determines whether we'll choose the X, Y or Z axis to be parallel to our secondary orientation (projected in plane of the primary orientation).
- <*s*> determines whether this axis has the same direction (+) or the opposite direction (-) as the secondary orientation
- <*e*> selects the measured element, which is chosen to be our secondary orientation.
- **Frame position: origin of axis: <*e*> at nominal coordinate <*c*> relative to co-ordinate system**: determines the origin of our new static reference frame with respect to a measured element. The idea is that the measured element gets expressed in the new static reference system. The software proposes to use the same element from all three axes, but is can be configured independently for each axis:
- <*e*> the measured element of which we dictate the position in the new static reference frame.
- <*c*> the dictated coordinate of the measured element in the new static reference frame.

**Project element 2 on element 1:** this operation projects the second element onto the first:

Parameters:

- **Name**: name of the constructed element
- **Element to project onto:** the element onto which the second element will be projected.
- **Element to project**: subject of the projection.

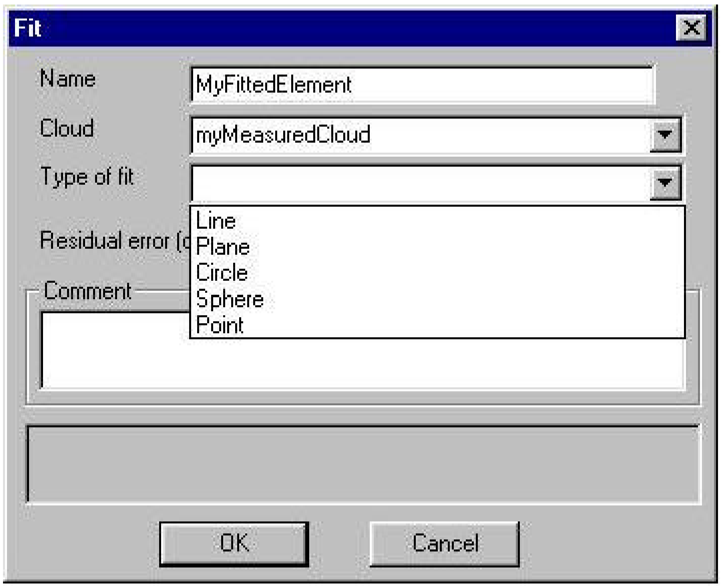


Name myProjection
Element to project onto measuredZplane
Element to project measuredOrigin

Figure 16: GeoLoc - ‘projection’ form

**Create cloud:** this operation creates a cloud of a number of individual geometric elements. This can be useful when a set of individually constructed points must be mapped on a cloud of defined points.
‘Create cloud’ is the only operation that has no pop-up window to configure parameters: ‘Create cloud’ ***must*** use the operand window.

**Fit:** this operation will fit an existing point cloud (measured or constructed) onto one of four geometric primitive elements: a line, a plane, a circle or a sphere:



Name MyFittedElement
Cloud myMeasuredCloud

Figure 17: GeoLoc - ‘fit’ form

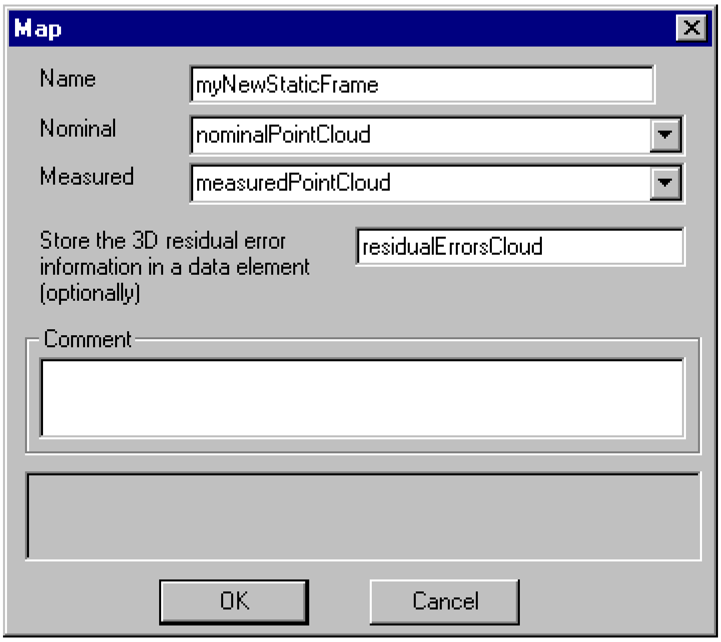
Parameters:

- **Name**: name of the constructed element
- **Cloud**: the point cloud that will be fitted into a geometric primitive.
- **Type of fit**: type of geometric primitive. Fitting a cloud to a point creates a point whos coordinates are the average of all the points in the cloud.
- **Residual error**: (optional) the name of an error point cloud. Each point in the error point cloud is the 3D error between the fit result and the point cloud on which the fit was based.
- **Comment**: (optional) description of the element

**Map:** this operation creates a new static reference frame by mapping two point clouds onto each other: a defined point cloud and a measured point cloud. The defined point cloud contains a set of points, of which the user dictates their coordinates in a coordinate system. The measured point cloud contains the corresponding points, measured in another frame. By mapping the two clouds onto each other, the software can calculate the transformation formulas to convert coordinates from one frame into the other. Defined point clouds have not been treated yet – they will be in the next chapter "Defining geometrical elements" on page 32.

The main difference between 'fitting' and 'mapping' is, that for a fitting, the user does not know what the outcome should be: he constructs a new element in an existing reference frame. For a mapping, the user must specify what the measured points *should* be in his chosen reference frame: he constructs a new reference frame based on existing elements.

It is very important that the measured points are measured in exactly the same sequence as their counterparts in the defined point cloud.



Map

Name: myNewStaticFrame

Nominal: nominalPointCloud

Measured: measuredPointCloud

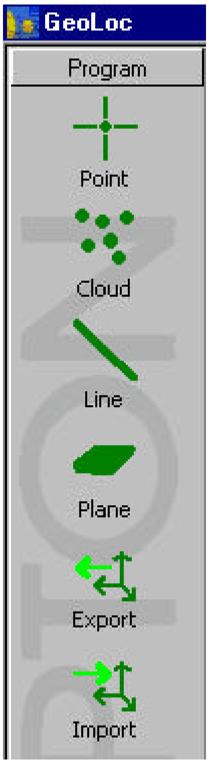
Store the 3D residual error information in a data element (optionally): residualErrorsCloud

*Figure 18: GeoLoc - 'Map' form*

Parameters:

- **Name**: name of the new static reference frame
- **Nominal**: the point cloud with nominal (dictated) coordinates.
- **Measured**: the point cloud with coordinates, measured in camera frame.
- **Residual error**: (optional) the name of an error point cloud. Each point in the error point cloud is the 3D error between the fit result and the point cloud on which the fit was based.
- **Comment**: (optional) description of the element

### 5.3.4 Defining geometrical elements



With the “Program” command bar, the user can define a limited set of geometric elements, expressed in any given reference frame.

Defined geometric elements are requested when an alignment, based on a mapping of two point clouds, is required, or in case we want to construct elements that can or should not be measured.

The easiest way of dividing points into the category of defined points is to call them: “points that were not measured and not constructed”. In other words, these are theoretical points.

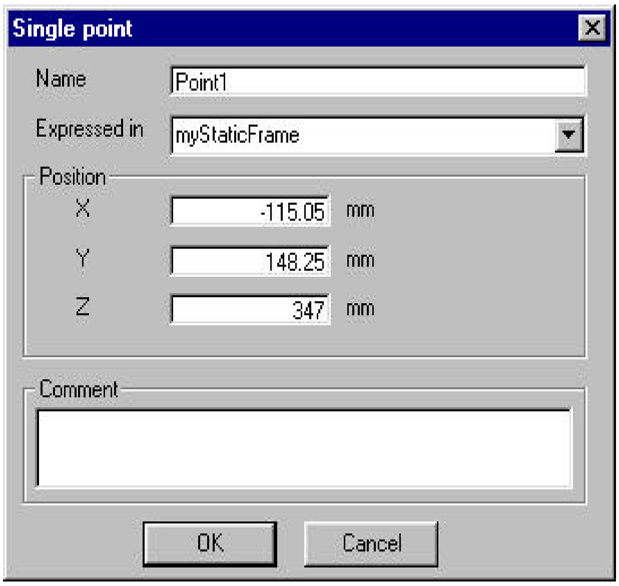
The geometric elements that can be defined are: planes, lines, points and point clouds.

Defined geometric elements do not differ in any way from measured or constructed points: they can be used for constructions, projections, intersections, … Only the circumstances will indicate whether this kind of operation is acceptable or not.

Finally, there is a function to export elements to text files, and to load them again from file. Sometimes, this makes it easier to share data between GeoLoc and other programs.

**Point:** this option defines a single point. Single points can be used e.g. to construct defined lines.

Parameters:
**Name**: name of the defined point
**Expressed in**: name of the static reference frame in which the point is expressed. All available reference frames are listed in the combo box, including the camera frame
**Position**: Cartesian coordinates in the static reference frame.
**Comment**: (optional) description of the element


Single point

Name: Point1

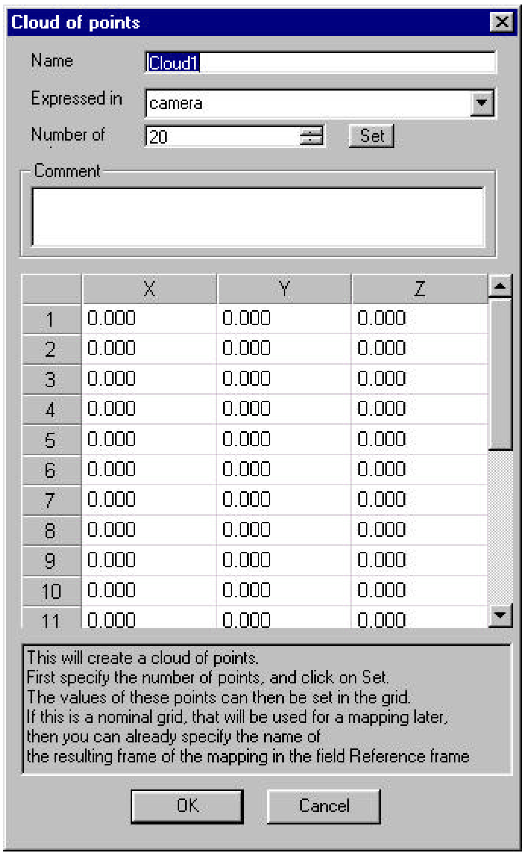
Expressed in: myStaticFrame

Position
X: -115.05 mm
Y: 148.25 mm
Z: 347 mm

Comment:

**Cloud:** this option defines a cloud of single points.  Defined point clouds can be used to create alignments based on mappings (see chapter “*5.3.3Manipulating* The points are entered in a matrix: each row represents a point, each column represents one of three coordinates (X, Y or Z)

Parameters:
**Name**: name of the defined point cloud
**Expressed in**: name of the static reference frame in which the point cloud is expressed. All available reference frames are listed in the combo box, including the camera frame
**Number of**: number of points in the point cloud. It isn’t until the user clicks on the ‘Set’-button that the matrix size is updated. Only then the user can enter the coordinates of each point in the cloud
**Comment**: (optional) description of the element



Cloud of points

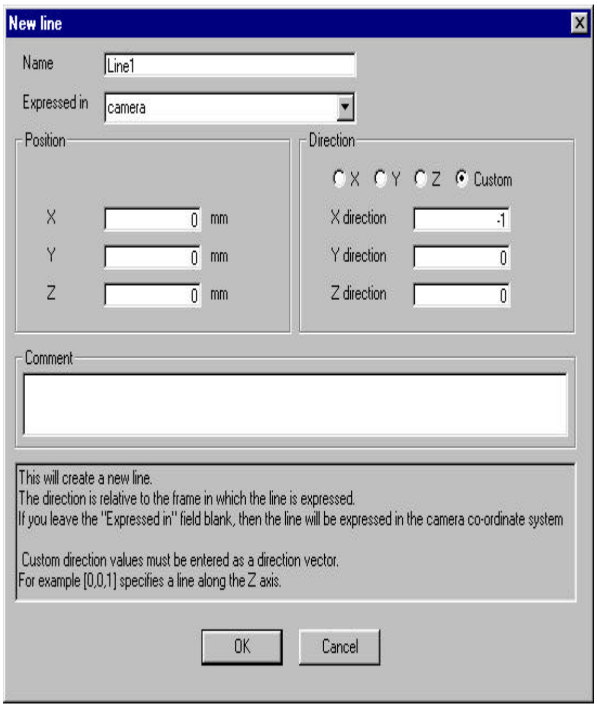
Name: Cloud1
Expressed in: camera
Number of: 20
Comment:

|  | X | Y | Z |
|---|---|---|---|
| 1 | 0.000 | 0.000 | 0.000 |
| 2 | 0.000 | 0.000 | 0.000 |
| 3 | 0.000 | 0.000 | 0.000 |
| 4 | 0.000 | 0.000 | 0.000 |
| 5 | 0.000 | 0.000 | 0.000 |
| 6 | 0.000 | 0.000 | 0.000 |
| 7 | 0.000 | 0.000 | 0.000 |
| 8 | 0.000 | 0.000 | 0.000 |
| 9 | 0.000 | 0.000 | 0.000 |
| 10 | 0.000 | 0.000 | 0.000 |
| 11 | 0.000 | 0.000 | 0.000 |

This will create a cloud of points.
First specify the number of points, and click on Set.
The values of these points can then be set in the grid.
If this is a nominal grid, that will be used for a mapping later,
then you can already specify the name of
the resulting frame of the mapping in the field Reference frame

**Line:** this option defines a line. This type of line is defined by a point on the line and three direction values. Should a defined line only be known by two of its points, it can be constructed using two defined points and a line fit.

Parameters:
**Name**: name of the defined line
**Expressed in**: name of the static reference frame in which the line is expressed. All available reference frames are listed in the combo box, including camera frame
**Posi tion**: Cartesian coordinates of the point on the line, expressed in the static reference frame.
**Direction**: direction of the line. The user has two possibilities:
A line perfectly parallel to, and equal in direction as, one of the axes of the static ref. frame.
‘Custom’: direction according to entered weight factors for each axis of the coordinate system in which the object is expressed.
**Comment**: (optional) description of the element.



New line

Name: Line1
Expressed in: camera

Position
X: 0 mm
Y: 0 mm
Z: 0 mm

Direction
○ X ○ Y ○ Z ● Custom
X direction: -1
Y direction: 0
Z direction: 0

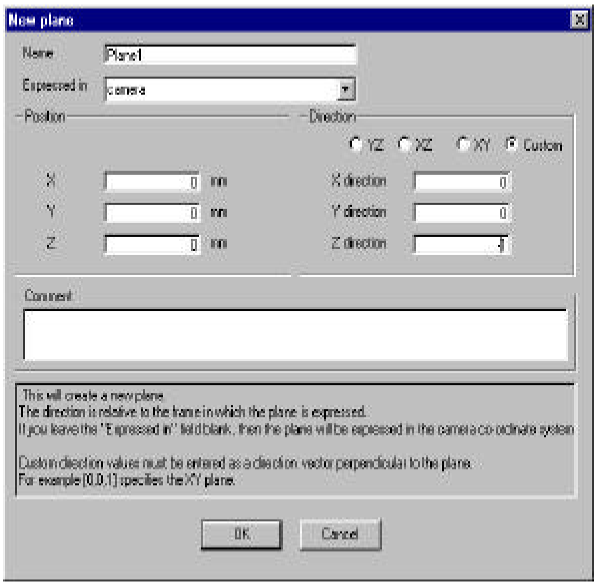
Comment:

This will create a new line.
The direction is relative to the frame in which the line is expressed.
If you leave the "Expressed in" field blank, then the line will be expressed in the camera co-ordinate system

Custom direction values must be entered as a direction vector.
For example [0,0,1] specifies a line along the Z axis.

When working with weight factors, GeoLoc will set the line direction, taking into account the ratio between that direction and the weight of the 3D-motion
For example:
The weight of the 3D-motion is ? ( $(-1)^2 + 0^2 + 0^2$ ) = ? 1 = 1
The weight for the X-motion is -1
The weight for Y and Z is 0
Therefore, the axis will be parallel to X, but in the opposite direction.

**Plane:** this option defines a plane. A plane is defined by a point in the plane and three direction values, determining a vector perpendicular to the plane. Should a plane only be known by three of its points, this plane can be constructed using three defined points and a plane fit.



Name: Plane1
Expressed in: camera
Position
X: 0 mm
Y: 0 mm
Z: 0 mm
Direction
○ YZ ○ XZ ○ XY ● Custom
X direction: 0
Y direction: 0
Z direction: -1
Comment:

This will create a new plane
The direction is relative to the frame in which the plane is expressed.
If you leave the "Expressed in" field blank, then the plane will be expressed in the camera co-ordinate system

Custom direction values must be entered as a direction vector perpendicular to the plane.
For example [0,0,1] specifies the XY plane.

Parameters:
**Name**: name of the defined line
**Expressed in**: name of the static reference frame in which the plane is expressed. All available reference frames are listed in the combo box, including camera frame
**Position**: Cartesian coordinates of the point on the plane, expressed in the static reference frame.
**Direction**: direction of the plane. The user has two possibilities:
A plane perfectly parallel to, and equal in direction as, one of the planes, formed by the axes of the static ref. Frame (XY-plane, XZ-plane or YZ-plane).
'Custom': direction values of the vector perpendicular to the plane, according to entered weight factors for each axis of the coordinate system in which the object is expressed.
**Comment**: (optional) description of the element

When working with weight factors, GeoLoc will set the vector direction, taking into account the ratio between that direction and the weight of the 3D-motion
For example:
The weight of the 3D-motion is ? ( $(-1)^2 + 0^2 + 0^2$ ) = ? 1 = 1
The weight for the Z-motion is -1
The weight for Y and Z is 0
Therefore, the plane will be equal to the XY-plane, but 'inverted' (its positive vector is opposite to the Z-axis).

**Export:** this function exports an element into a text file. (Extension does not necessarily have to be *.TXT)

If the exported element is a point cloud of *n* points, the text file is a matrix with *n* rows. The layout of each row is as follows:

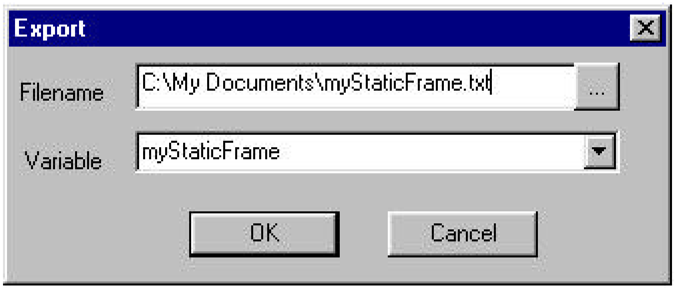
- Column 1: sequence number of the point
- Columns 2, 3 and 4 : X-, Y- and Z-coordinates of the point with respect to the frame of reference in which the element is expressed
- Columns 5, 6 and 7: directional weight factors in X, Y and Z of the point. This is useful if the point was measured with a Space® Probe, since the probe direction is stored in these columns.

If the exported element is a dynamic frame with *n* LED's, the text file is a matrix with *n* columns. The layout of each column is as follows:

- Row 1: X-coordinate of the LED in the frame to which the dynamic frame was attached

- Row 2: Y-coordinate of the LED in the frame to which the dynamic frame was attached
- Row 3: Z-coordinate of the LED in the frame to which the dynamic frame was attached
- Row 4: ‘1’. This is required for mathematical manipulations after this file will be imported.

If the exported element is another type of element (plane, static frame…), the text file will hold the transformation matrix between the frame and the frame of reference.



Filename: C:\My Documents\myStaticFrame.txt

Variable: myStaticFrame

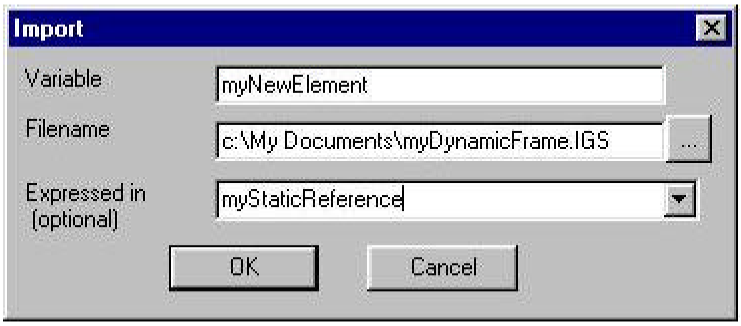
*Figure 19: GeoLoc - 'Export' form*

Parameters:

- **Filename**: name of the text file in which the element data will be stored.
- **Variable**: select the element that you want to export into the text file.

**Import:** this function imports an element from a specified file. The valid extensions are listed when clicking on the browse button to select the file.

GeoLoc will recognize the type of element that was saved, and will write the file data into an element of the same type, be it a new element or an existing one.



Variable: myNewElement

Filename: c:\My Documents\myDynamicFrame.IGS

Expressed in (optional): myStaticReference

*Figure 20: GeoLoc - 'Import' form*

Parameters:

- **Variable**: enter the name of the element in which you want to import the file. If the name does not exist, the element will be created.
- **Filename**: name of the file from which the element data will be read. This can be a text file, a Matlab file or an IGES-file, containing an element, an alignment or a dynamic frame (led file)
- **Expressed in**: reference frame in which the element is to be expressed. If this field remains blank, the camera reference frame is the default.

## 5.4 Options in the main menu

Until now, we've seen all possibilities to measure elements interactively. However, all these possibilities are explained on a 'once only' basis.
The main menu holds a number of features that allow us to organize and configure settings that are not likely to change often.

GeoLoc has the ability to detect new calculation modules. Every time such a module is found, it will be added to the menu. This manual will only describe the standard menu options.

The standard menu options are:

'**Project menu**': this menu allows for organizing measurement sessions into projects. It also gives the user access to the deleted items bin, and the ability to return geometric elements to another Krypton software product.

'**View**': this menu allows the user to customize the user interface: hide/show toolbars, adding/removing icons, status bars, …

'**Tools**': this menu allows the user to configure the measurement of some geometric elements.

'**Macro**': this menu holds all tools to record, play, or edit 'scripts', macro's that can automate repetitive tasks.

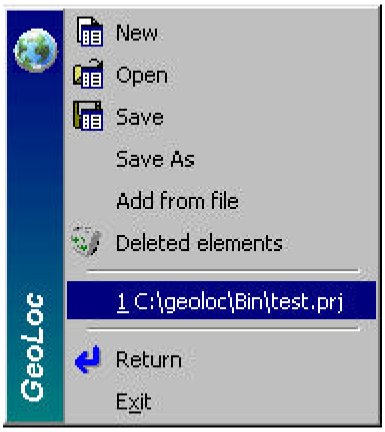
'**Baseloc**': this is the basic localisation module - in fact, every measurement or construction operation we've seen until now in chapter "Measuring, constructing & defining geometric elements" on page 19.

### 5.4.1 ‘Project’ menu

The project menu deals with three features: organizing projects, managing deleted elements and exporting elements to other Krypton products.

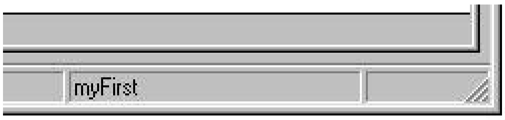
In general, GeoLoc sorts all measurements in projects. These are separate directories on the hard-disk. By storing geometric elements, the element window will not be flooded with elements that are outlived, or have got nothing to do with the current measurement context.

The user must not be aware where the project files are stored. GeoLoc stores this information for him. All the user has to do is open the appropriate project file. GeoLoc will fill the element window with the corresponding elements. From then on, it’s just as if the user never stopped measuring.



Setting the project is something that must not always be done manually by the user: if another Krypton product calls GeoLoc, this calling program will instruct GeoLoc to create or open a project.

The current project is denoted in GeoLoc’s status bar: e.g. the project ‘myFirst’



Should it be necessary to exchange elements between projects, GeoLoc can import element files from other projects.

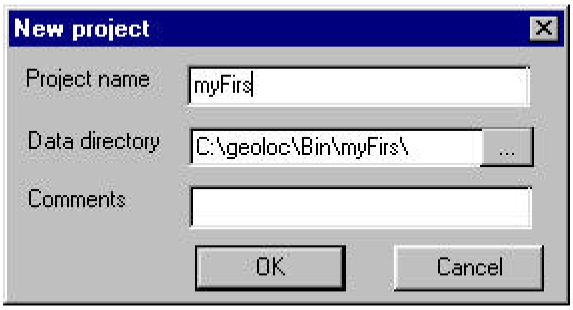
Whenever an element gets deleted, its file is not yet physically removed, but the element’s icon disappears from the element window. The project menu allows to restore elements, or to purge the ‘Deleted items’ bin.

Finally, The menu allows for exporting elements to the calling Krypton software package. This is very often a static reference frame, or a dynamic frame.

We’ll now discuss each option in detail:

**New:** creates a new project. Whenever a new project is created, a new data directory is created, where the geometric element files will be stored, and a project file (extension ***.PRJ**).

The user is prompted to enter a project name. While the user types the project name, the data directory will automatically grow. It is possible to change the data directory *after* the project name was entered.
Should the user change the project name after the data directory was set, the data directory will be reset to the default directory, extended with the project name.


**New project**

Project name: myFirs

Data directory: C:\geoloc\Bin\myFirs\ ...

Comments:

OK    Cancel

**Open:** select the project file (project name with extension *.PRJ). GeoLoc will fill the element window with all elements described in this project.

**Save:** stores the current project file.

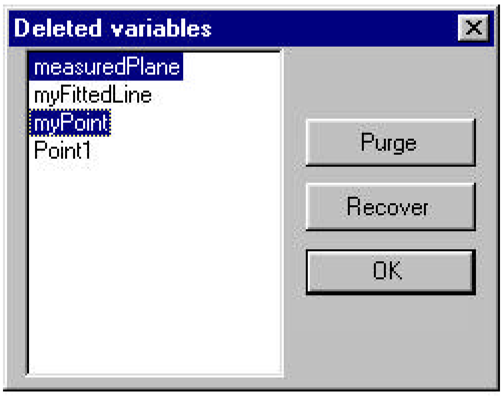
**Save As**: will store the current elements in a new project. The pop-up window is the same as for 'New', but it will not be an empty project.

**Add from file**: allows for manually adding elements from other projects. All elements are stored on disk under Matlab format. Such file is of form 'elementName.MAT'. Select the Matlab file that corresponds to your element with the pop-up form.

**Deleted items**: choosing 'Deleted items' lists the elements that were deleted but not physically removed from the hard disk.

If you select a number of elements and click on 'recover', the elements are restored in the element window.
Clicking 'Purge' will physically remove the selected elements from the hard-disk.

Clicking 'OK' returns you to the main menu.



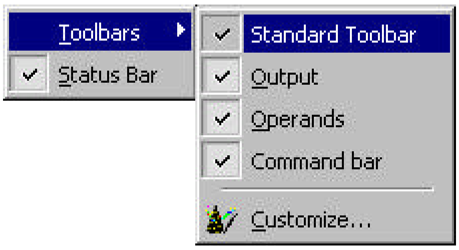
**Return**: when GeoLoc was started by another program, and some elements are dragged into the operand window, clicking 'return' will teleport these elements to the calling program. See also chapter "The operand window" on 18.

**Exit**: closes GeoLoc

### 5.4.2 'View' menu

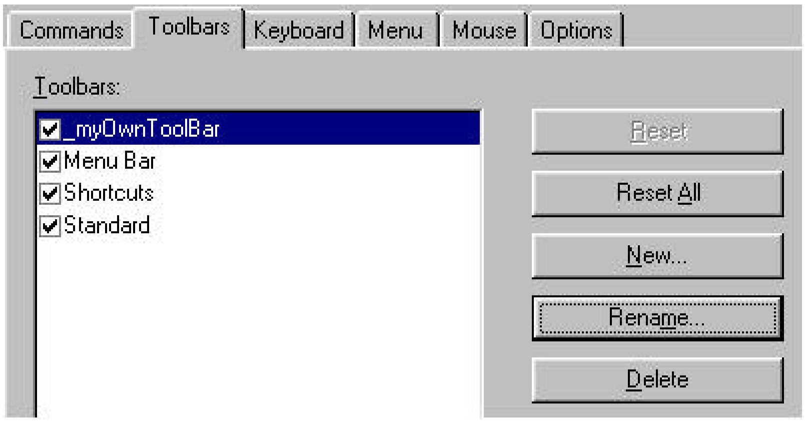
The view menu is a small menu that allows the user to modify the user interface to his/her personal comfort.

The first drop-down level allows the user to hide or show the status bar. Showing the status bar is useful, since it displays a line of description every time the mouse moves over a menu item.



**Toolbars:** Secondly, the menu allows for hiding or showing individual toolbars. 4 Standard toolbars are available.
Unchecking the 'Standard toolbar' will hide the icon bar below the menu
Unchecking the 'Output' will hide the output window
Unchecking the 'Operands' will hide the operands window
Unchecking the 'Command Bar' will hide the command bar

**Customize:** this menu brings up a form that allows you to customize every item on the menu and icon bar. The form has a number of tab sheets:



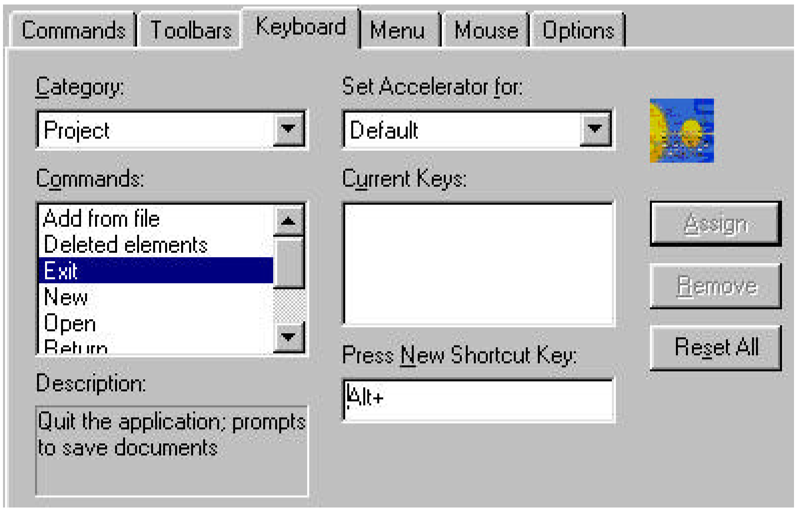
Toolbars:
☑ _myOwnToolBar
☑ Menu Bar
☑ Shortcuts
☑ Standard

*Figure 21: GeoLoc - 'Customize toolbars' form*

**Toolbars**: this tab allows you to create new toolbars, hide existing toolbars, or deleting toolbars. It is suggested to create a personal toolbar to place all user-chosen icons on. When a new toolbar is created, this starts as a very small grey block. Drag it besides the icon bar.
Checking or unchecking a toolbar shows or hides the toolbar.
***Warning!*** If you click 'Reset all', all menu entries of additional modules will be deleted! If you want to reset the toolbars, while retaining the module toolbars, go to the tab 'Menu' and click on 'Reset' for 'Default menu'.

**Commands**: this menu shows, per category, every possible menu item and its corresponding icon. Just select the icon of a function and drag it onto the icon bar. If you want to remove an icon, just click on it on the menu bar, drag it into the form while holding your left mouse-button down.

**Keyboard:** this menu allows configuring keystrokes as shortcuts for certain functions:



Category: Project
Set Accelerator for: Default

Commands:
Add from file
Deleted elements
Exit
New
Open
Return

Current Keys:

Description:
Quit the application; prompts to save documents

Press New Shortcut Key: Alt+

*Figure 22: GeoLoc - 'Customize keyboard' form*

Select a category; in the listbox 'Commands', a list of all functions for that category appear. Select the command you want to create a shortcut for. In the 'Description'-window, you'll get a description for the chosen command. Should a shortcut already be defined for this command, it will appear in the list-box 'Current keys'.
Click with the mouse in the text box labeled 'Press new shortcut key:' Just press the key combination you'd like to use to activate the command: this text box will show the key combination you just pressed.
If this key combination was not yet assigned, a message will appear below the text box.
Click on the 'assign' – button to confirm the assignment. The shortcut will be added to the 'Current keys' listbox.

If a key combination was already assigned, a message will appear indicating which command is already linked to that shortcut. It will be impossible to re-assign this shortcut. If you wish to remove a shortcut, select it in the 'Current keys' listbox and click on 'Remove'.

It is suggested not to change the default settings for the following 3 menus.

**Menu**: this menu allows to configure some menu animations or context menus, or to reset the default menu and all toolbars for additional modules.

**Mouse**: allows to configure the double-click command of the mouse.

**Options**: allows configuring the Windows Screen Tips and the Windows 2000 look.

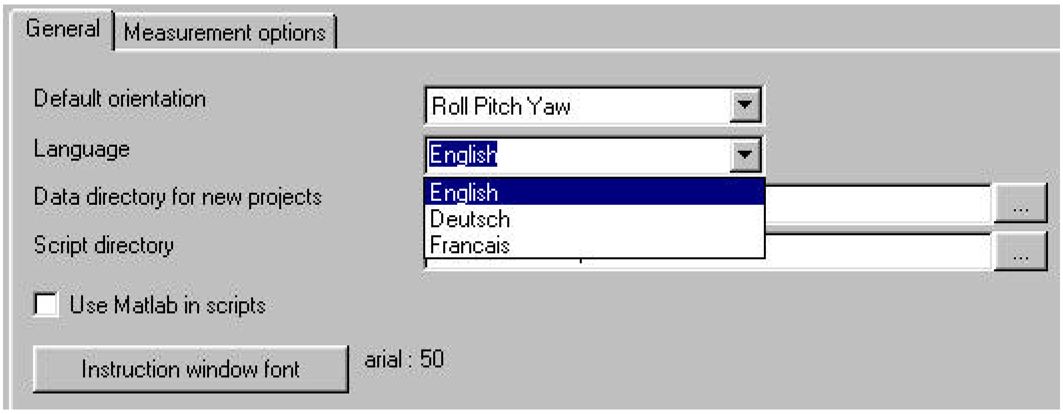
### 5.4.3 'Tools' menu

The 'Tools' menu holds two configuration functions.

**Customize:** function has been discussed in the chapter "'View' menu" on 38.

**Options**: this function brings up a pop-up with a few tab sheets. If a module is added to GeoLoc, a new tab sheet appears here, on which the settings for this new module can be changed. The default tab sheets are discussed in the following subchapters.

#### *5.4.3.1 General:*



General | Measurement options

Default orientation: Roll Pitch Yaw
Language: English
Data directory for new projects:
Script directory:
□ Use Matlab in scripts
Instruction window font: arial : 50

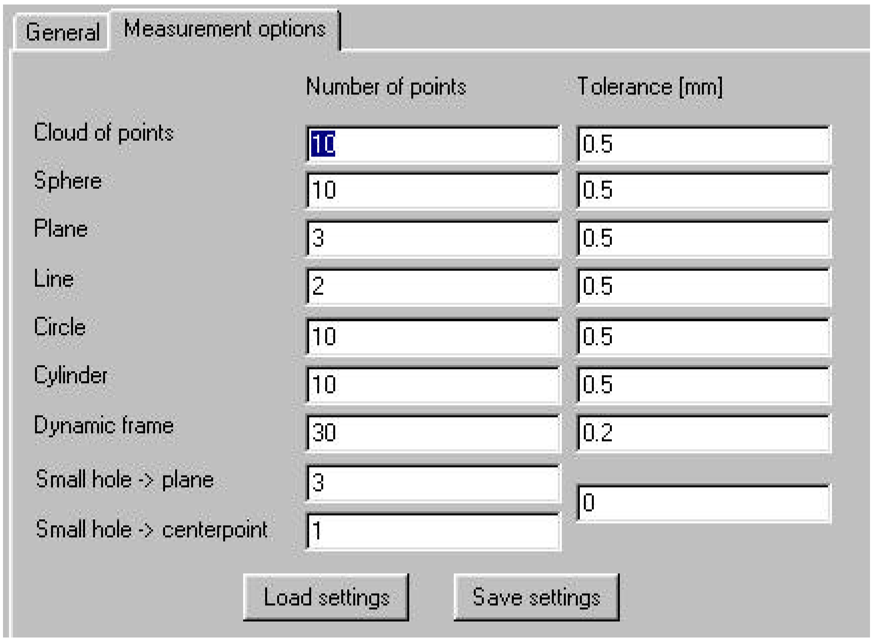
*Figure 23: GeoLoc - general settings tab form*

- **Default orientation**: this is the angular convention in which frames will be expressed. At this time, three conventions are supported:
- Roll-Pitch-Yaw: first around X, then Y, then Z. Rotations are relative to the reference frame axes
- Steer-Camber-Spin: first Z, then X, then Y. Rotations are relative to the objects' own axes (axes have rotated with object)
- Skrew vector: rotation around a vector in a 3D-space. The three coordinates are the lengths of the three vectors $S_x$, $S_y$ and $S_z$ that compose the skrew vector. The angle of rotation is the vector norm (in radians) – this is always the smallest angle to transit.
- **Language**: language in which GeoLoc presents itself to the user.
- **Data directory for new projects**: this is the default directory GeoLoc will create subdirectories in, every time a new project is created.
- **Script directory:** default directory in which Geoloc saves or retrieves script files.

- **Use Matlab in scripts:** if GeoLoc finds an unknown function in a script, he will interpret this as an error, unless this option is checked: if checked, GeoLoc will try to execute the function in Matlab and proceed with the results as yielded by Matlab.
- **Instruction window font**: this is the font in which script messages will be shown to the user. The bigger the font, the easier it is to read from large distances, but the faster a message will clip.

#### *5.4.3.2 Measurement options:*

This tab lists for every geometric element the number of points to take before attempting a fit for that element. The 'Tolerance' column holds the residual error (least-square error) of the fitting that should not be exceeded.



| | Number of points | Tolerance [mm] |
|---|---|---|
| Cloud of points | 10 | 0.5 |
| Sphere | 10 | 0.5 |
| Plane | 3 | 0.5 |
| Line | 2 | 0.5 |
| Circle | 10 | 0.5 |
| Cylinder | 10 | 0.5 |
| Dynamic frame | 30 | 0.2 |
| Small hole -> plane | 3 | 0 |
| Small hole -> centerpoint | 1 | |

*Figure 24: GeoLoc - measurement options tab form*

After a fitting, the residual error and the tolerance are displayed in the output window as follows:

**0.137 / 0.500**

If the actual residual error exceeds the tolerance, this message is **red** instead of black.

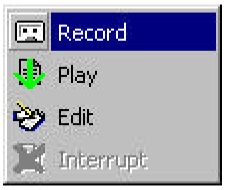
The settings can be saved to file, or can be loaded from a previously stored file, by clicking on the buttons "Load settings" and "save settings".

**Notes:**

- The 'number of points' for a cylinder are only the points measured on the shaft. For measuring the base plane of the cylinder, the number-of-points for a plane apply.
- The 'Number of points' for a dynamic frame is the number of 'snapshots' that will be taken of the LED's on the frame. Every LED position gets averaged over this number of snapshots.
- Measuring a bore consists of measuring a plane, and then measuring the probe as it rests into the bore. For measuring the plane of the bore, GeoLoc does NOT use the number-of-points of a plane, but uses the number-of-points set in 'small hole -> plane' .

### 5.4.4 'Macro' menu

Although the macro's themselves are fully described in chapter "Scripting – writing and using macro's" on page 43 we'll list the menu possibilities here already.



**Record:** when this option is chosen, the macro 'teach' mode is toggled. If the 'teach'-mode is active, the icon will look as if it is pushed in.
From this moment on, every action performed is recorded into a macro and can be replayed later. When activating the 'teach'-mode, GeoLoc asks in which script to dump the actions. This file is an ASCII-file, and can be edited with any editor.
If the file already exists, GeoLoc asks you whether to overwrite the file, or to append the new actions to the existing file.

**Play**: run a previously recorded macro. When clicking the function, it will ask for a macro to run. Macro's have the file extension "*.SCRIPT".

**Edit**: this script opens the macro editor (see chapter "Scripting – writing and using macro's" on page 43)

**Interrupt**: when a script is running, this will interrupt the active script and return program control to the GeoLoc user interface.

### 5.4.5 'Baseloc' menu

This menu is the same as its shortcuts in the command bar, only in text form. Refer to chapter "Measuring, constructing & defining geometric elements" on 19.

### 5.4.6 'Help' menu

This menu has two functions:
**Help**: activates the Windows on-line help function on GeoLoc
**About GeoLoc**: returns a splash screen with GeoLoc's version number and build number. Always communicate these numbers when reporting problems to Krypton.

## 5.5 Miscellaneous functions

The following functions do not belong to any of the previously described menus but they belong to the standard GeoLoc user interface and function inventory.

**Abort**: clicking this button will terminate and reject the measurement currently in progress. If this happens while running a script, the script will be aborted too.

**Finish**: clicking this button will terminate and accept the measurement currently in progress. If this happens while running a script, the script will continue as usual.

# 5.6 Scripting – writing and using macro's

Scripting is a very powerful utility of GeoLoc.
Scripting allows you to record or program a set of measurements. This macro can be replayed an infinite number of times.
In a script, clarifying messages can be sent to the operator: this means that one person can write scripts, distribute them to multiple measurement stations, where an operator just follows screen instructions.

GeoLoc scripts can execute a variety of functions:
- They can execute menu actions
- They can execute other DMM Modular scripts
- They can execute Matlab scripts

GeoLoc scripts are ASCII-files: they can be read, modified or saved by any text editor. GeoLoc's own script editor however has some utilities that make it a very handy tool when dealing with scripts.

We'll start with explaining how to write an easy, top-level script. Next, we'll go into how to use scripts within scripts

## 5.6.1 Writing a single-level script

GeoLoc scripts can be created in two ways:
- They can be written off-line using a dedicated GeoLoc editor
- They can be taught: GeoLoc records all the operations performed by the user

Both methods have their advantages and drawbacks:
- Off-line script writing has the greatest flexibility, but requires the user to be familiar with the GeoLoc script syntax. Inexperienced users can easily make mistakes this way
- Teaching is the easiest: the user is not confronted with GeoLoc's syntax. But this means that a number of functions (such as the message( ) function) are not accessible in teach-mode. Scripts are sometimes longer, due to rename-operations, …

The advised method is a mix of the two: record the first draft of the script using the 'teach'-mode; then, open the script in the GeoLoc editor (or any other editor), filter out the overhead manually, and add message( )-functions at will.

When reading the following chapters, please bear in mind that options from the main menu can also be chosen by clicking the corresponding icon on the icon bar. See chapter "'Macro menu" on 42 for these icons.

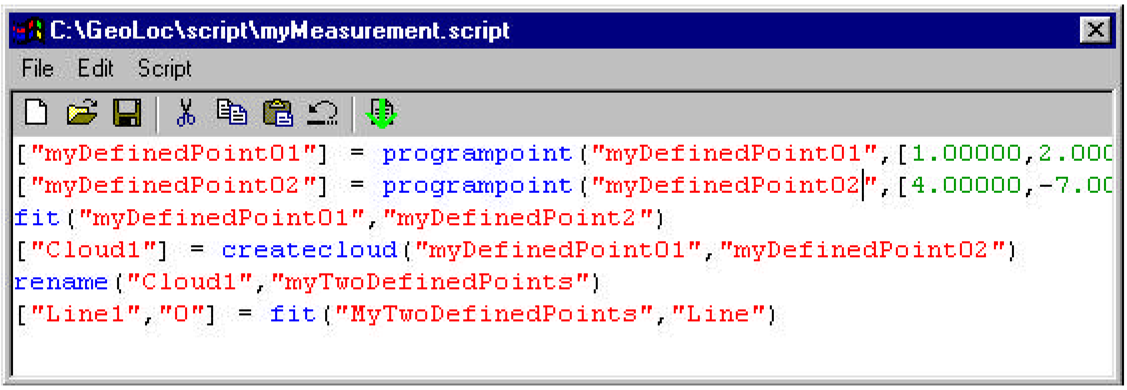
### *5.6.1.1 Teaching a script*

This chapter will clarify how we teach a script.
Go to the main menu and choose 'Macro' – 'Record'
GeoLoc will ask for a script file name in the current project directory. Enter a suitable file name. Make sure the extension is 'script'. If the file already exists, GeoLoc will ask you to overwrite the old file, or to append the new actions to the old file.

Start measuring/operating/defining elements with the probe as if no script was recorded. In the background, every action is recorded. You'll see that the record-icon looks pushed into the user interface.

When you are done, go again to the main menu and choose 'Macro' – 'Record'. The icon will click out of the menu.

Now choose ‘Macro’ – ‘Edit’. Enter the filename under which you recorded your previous program. The GeoLoc editor will open the script:

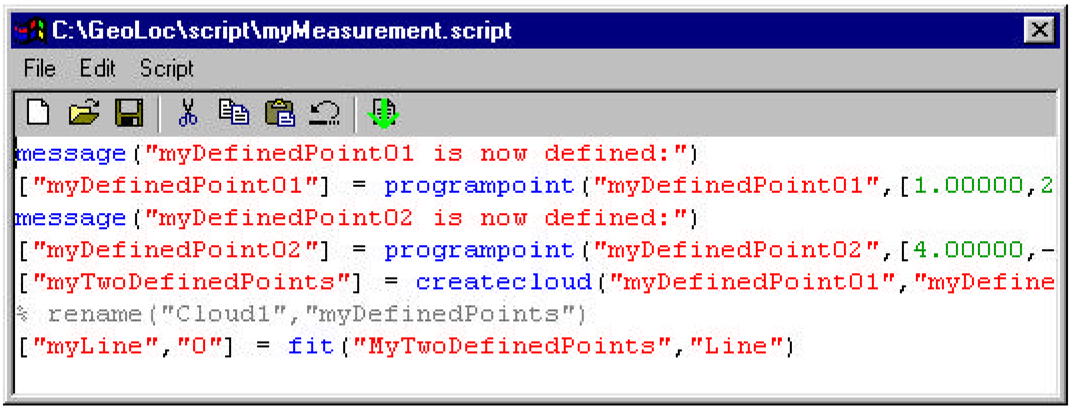


```
["myDefinedPoint01"] = programpoint("myDefinedPoint01",[1.00000,2.000
["myDefinedPoint02"] = programpoint("myDefinedPoint02",[4.00000,-7.00
fit("myDefinedPoint01","myDefinedPoint2")
["Cloud1"] = createcloud("myDefinedPoint01","myDefinedPoint02")
rename("Cloud1","myTwoDefinedPoints")
["Line1","0"] = fit("MyTwoDefinedPoints","Line")
```

As you can see in the example above, no messages are programmed to be sent, and there are two lines of code to measure the point cloud myTwoDefinedPoints: one line to measure the cloud (but gets saved under a default name), and one to rename the cloud with the default name to the requested one. We also made a false calculation, being the fit on line 3.

Therefore we can manually edit the script:
- We’ll remove the faulty fit
- We’ll create the cloud without renaming it
- We’ll comment out the cloud rename operation
- We’ll rename the final fitted line directly (line 7)
- We’ll add some messages

The result is the following:



```
message("myDefinedPoint01 is now defined:")
["myDefinedPoint01"] = programpoint("myDefinedPoint01",[1.00000,2
message("myDefinedPoint02 is now defined:")
["myDefinedPoint02"] = programpoint("myDefinedPoint02",[4.00000,-
["myTwoDefinedPoints"] = createcloud("myDefinedPoint01","myDefine
% rename("Cloud1","myDefinedPoints")
["myLine","0"] = fit("MyTwoDefinedPoints","Line")
```

This script can be saved and can now be run.
The functionality of the GeoLoc script editor will be explained in the next chapter.

### 5.6.1.2 *The GeoLoc script editor*

This chapter will describe the possibilities of the GeoLoc script editor.
The GeoLoc script editor is a small tool that allows for fast and easy script writing.

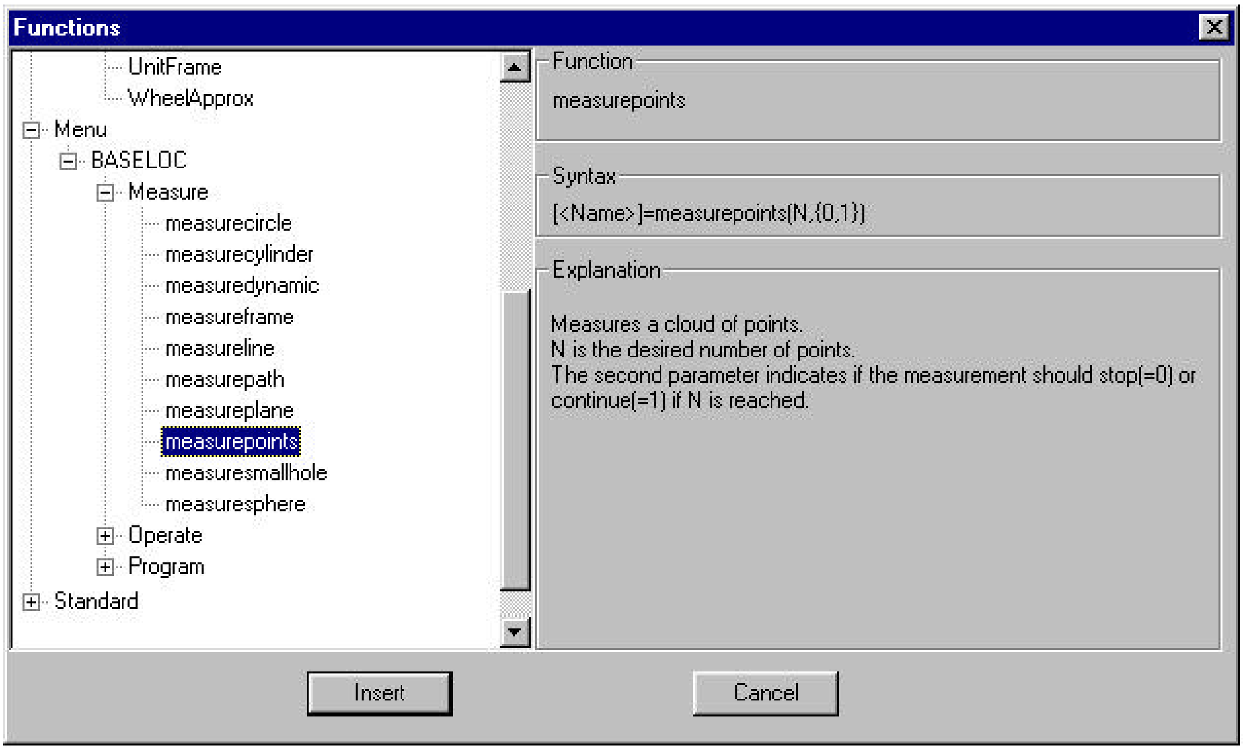
The editor recognizes the GeoLoc script syntax, and will color the words according to their function: functions are blue, names of variables and strings are red, and variables are black.

The editor has 8 functions/icons that are very similar to Windows, also in functionality:

‘File’ menu:
- New: new script
- Save: save a script under its original name
- Save as: save script under a different name
- Open: open an existing script

‘Edit’ menu:
- Cut: move a portion of text to Windows’ clipboard
- Copy: copy a portion of text to Windows’ clipboard
- Paste: insert the clipboard contents
- Undo: neutralize last edit action

Besides these, the editor has two additional functions:
- Run: run the current opened script in GeoLoc
- Show functions: this shows a structured overview of all the available functions for GeoLoc scripts:



Function
measurepoints

Syntax
[<Name>]=measurepoints(N,{0,1})

Explanation
Measures a cloud of points.
N is the desired number of points.
The second parameter indicates if the measurement should stop(=0) or continue(=1) if N is reached.

*Figure 25: Scripting - the function browser*

If you select a function from the tree, on the right will appear the function name, syntax, and an explanation on the function.

By clicking on ‘Insert’, the function will be inserted in the current script wherever the text cursor was positioned at the time of insertion.

Note that it is also possible to include previously defined scripts.

When a function is inserted, it is inserted with dummy parameters. The user must replace these parameters with their ‘real-life’ counterparts.

#### *5.6.1.3 The script syntax*

The GeoLoc scripting syntax is very similar to the Matlab syntax:

- Both variables and function calls are case sensitive
- Most functions are of the form [element] = function (parameter). Return parameters are put between square brackets; parameters are put between normal brackets.
- It is possible to use multiple parameters: parameters are separated by commas
- It is possible to return multiple return values: return values are separated by commas
- Elements can be put between double-quotes ( **“ “** ) or not. Users are advised though to put element names ALWAYS between double quotes. E.g. [myPoint] or [“myPoint”]
- Variables **have** to be put between double-quotes if their name contains spaces, or if they are constants, e.g. [“my Point”] or [“0.5”]
- If in the function selector, a function’s syntax is depicted as <element>, than this means it is optional. Optional elements should **NOT** be written in the script. E.g.:

Right:
*[“mySphere”, “fitError”] = fitSphere(“measuredCloud”)*
*[“mySphere”] = fitSphere(“measuredCloud”)*

Wrong:
*[“mySphere”, <”fitError”>] = fitSphere(“measuredCloud”)*

### 5.6.2 Using scripts within scripts

A script can be regarded in two ways: as a fixed set of operations, or as a function that calculates one specific sort of data.

The single-level script (see chapter “Writing a single-level script” on page 43) usually is a fixed set of operations.
When using nested scripts, the top-level script usually also is just a fixed set, but the branches can be of two sorts.

We’ll clarify the difference with an example. Suppose I have a calculated a data file “measurement1” and “measurement2”. Suppose I have three scripts, TOPLEVEL.script, FOO.script and BAR.script:

```
=== top of file  =====================
% this is script TOPLEVEL.script
% this executes the script FOO.script
FOO
% this executes the script BAR.script
[“output3”]=BAR(“measurement1”)
=== Bottom of file ==================
```

The script FOO is a fixed set of operations. All the parameters it uses and generates are global, i.e. they are created and known within GeoLoc.

```
=== top of file  =====================
% this is script FOO.script
[“output1”]=FitCircle(“measurement1”)
[“output2”]= FitCircle(“measurement2”)
=== Bottom of file ==================
```

The script BAR is a function: The first line states the syntax and return parameters to call this function. The first comment block (starting with percentage signs) will return information in the user interface, should you call this function without specifying parameters. All data files created within a script only have a local scope, i.e. they are only known to the script, and only live as long as the script is executed. The variable *myCloud* will not appear in GeoLoc, since it is locally created.

```
=== top of file  =====================
function [result] = BAR (“local Parameter”)
% BAR calculates the acceleration of your input parameter

[“myCloud”]=createCloud(“Point1”,”Point2”,”local Parameter”)
[“result”]=fitPlane(“myCloud”)
=== Bottom of file ===================
```

In case you would run a function script as top-level script, DMM Modular will show you a pop-up box, asking for the required in- and output parameters. The function BAR( ) would invoke this pop-up box:

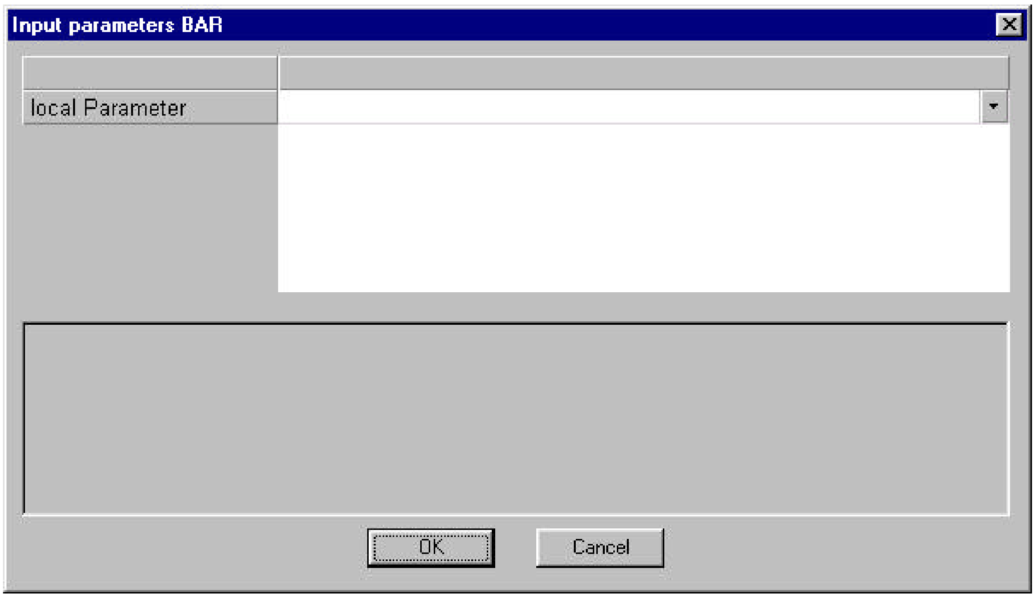


*Figure 26: Scripting - parameters field form*

These parameters can be selected from combo boxes in the pop-up, or can be drag-and-dropped from the data window.

# 6. Troubleshooting

This chapter describes the problems that are most likely to arise, and the way to solve them.
Should any problem not occur in the list, please notify your nearest Krypton representative.

This chapter is ordered parallel to all chapters and subchapters of this manual.

?? **After how many points measured does Geoloc start to calculate my element?**

- The number of expected points is listed in the output window during the measurement. See chapter "Measuring basic geometrical elements" on page 20.

?? **I want to measure an element with less points than expected by GeoLoc**

- Click the 'Finish'-button (see chapter "Miscellaneous functions" on page 42): this will calculate the element anyway, with the points measured until now.

?? **The position and orientation of my element is not what I expected**

- Make sure you are using the correct datum, or reference frame (e.g. when estimating the position of a point with respect to the camera, make sure the camera frame is used as frame of reference). See chapter "The element window" on page 18
- If your element was constructed, please check if the position and orientation of the "parent" elements was OK. If you don't know which elements were the "parent" elements, refer to the "base properties" of your construction. See chapter "The element window" on page 18.

?? **If I want to measure a path or a dynamic frame, a pop-up form appears instead of measuring led's**

- Geoloc can't find the application CamCheck, probably because this file was deleted or moved.
- Enter the full path to the CamCheck application.
- If you entered the wrong application, some registry settings will have to be deleted: contact Krypton for assistance.

# 7. Index