Technology by QVI

VMS

REFERENCE GUIDE

This document was produced by the Marketing Communications Group of Quality Vision International, Inc.

Warranty

Quality Vision International, Inc. (QVI® company) warrants that the software will operate according to the specifications described in this manual and be free from known defects in materials or workmanship for a period of one year from the date of shipment. During this warranty period, QVI will, at its option, repair, replace or provide a workaround solution to any items that prove to be defective. In order to qualify for such warranty service, a complete description of the problem, with appropriate documentation (such as results, program listing, sample part and program) should be forwarded to QVI for our inspection. Such items will be returned to the customer if requested.

QVI does not warrant that the operation of the software will be uninterrupted or error-free. The information in this manual is subject to change without notice.

This warranty does not apply to defects resulting from customer-supplied or -configured computer equipment, operating systems or software, unauthorized alteration or misuse, or operation outside the environmental specifications for the product.

THESE WARRANTIES AND QVI'S LIABILITY HEREUNDER, ARE THE SOLE AND EXCLUSIVE WARRANTIES AND ARE EXPRESSLY IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL QVI BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR OTHER DAMAGES INCLUDING, AND NOT BY WAY OF LIMITATION, LOSS OF PROFITS, AND WITHOUT REGARD TO THE FORM OF THE ACTION OR THE NATURE OF THE CLAIM WHICH IS MADE.

The information contained herein is based on the experience and knowledge relating to the subject matter gained by QVI prior to publication. No patent license is granted by this information. QVI RESERVES THE RIGHT TO CHANGE THIS INFORMATION WITHOUT NOTICE AND MAKES NO WARRANTY, EXPRESS OR IMPLIED, WITH RESPECT TO THIS INFORMATION. QVI shall not be liable for any loss or damage, including consequential or special damages, resulting from the use of this information, even if loss or damage is caused by negligence or other fault on the part of QVI.

Due to the nature of this material, a number of hardware and software products may be mentioned by name. In most, if not all, cases these product names are claimed as trademarks by the companies that manufacture the products. It is not our intent to claim these names or trademarks as our own.

© 2005 Quality Vision International, Inc. All rights reserved. Printed in USA.

No part of this document may be reproduced or disclosed in any form or for any purpose, other than personal use, without the written permission of Quality Vision International, Inc.

Part No. 790411 *First Printing 05-2005*

Table of Contents

About This Guide

Section 1: Introduction

Section 2: Looking at VMS Software

Section 3: Setup Operations

Section 4: Measurement Programs

Section 5: Alignment and Coordinate Systems

Section 6: Measurements

Section 7: Finders

Section 8: Step and Repeat Programming

Appendix A: Advanced Programming

Appendix B: CAD Import Option

Appendix C: Offline Programming Option

Appendix D: Error Messages

Index

This page was left blank intentionally.

About This Guide

Welcome!

Quality Vision International (QVI) has produced this *Reference Guide* as part of its ongoing effort to provide users with useful, comprehensive documentation. This guide has been developed using new documentation standards and a document design that enhances readability and makes it easier to find information.

We believe this manual will assist you in the use of the VMS software. If you have any questions that are beyond the scope of this manual, please do not hesitate to contact your authorized Sales Representative.

Organization

The *VMS Reference Guide* contains the following sections:

About This Guide (this section), describes the content and organization of this manual, explains the documentation conventions used in this manual, and includes customer support information.

Section 1, Introduction, provides a general overview of VMS.

Section 2, Looking at VMS Software, describes all the components comprising the user interface.

Section 3, Setup Operations, describes lens calibration and explains configuration options.

Section 4, Measurement Programs, describes how to create and run measurement programs.

Section 5, Alignment and Coordinate Systems, describes the alignment and coordinate system used in the VMS software

Section 6, Measurements, describes the various feature measurements available in the VMS software.

Section 7, Finders, describes the different Finders available in the VMS software.

Section 8, Step and Repeat Programming, describes how to use the Step and Repeat functions available in the VMS software.

Appendix A, Advanced Programming, explains how to build programming routines into VMS measurement programs.

Appendix B, CAD Import Option, describes the optional CAD Import component available with the VMS software.

Appendix C, Offline Programming Option, describes the optional Offline Programming component available with the VMS software.

Appendix D, Error Messages, lists the different error messages you may encounter while working with the VMS software.

Documentation Conventions

To help you locate, interpret, enter, or select information easily, this manual uses consistent visual cues and standard text formats. For example, capital letters (or upper-case letters) are used to call attention to certain words and to help eliminate ambiguity. These documentation conventions are explained in the table below.

Type Style; Symbol	Used for	Examples and Explanations
slash: /	➢ Selections from a main menu and submenu(s)	➢ Select **System / Exit**
italic or **bold**	➢ Emphasized words	➢ Select the *highest* magnification ➢ **Do not** repeat this step
bold, sans-serif typeface	➢ Commands to be typed ➢ Keys to be pressed ➢ Buttons to be pressed or selected ➢ Menu items to be selected	➢ Type **b:install**; type **Exit** ➢ Press the **Enter** key ➢ Select the **Done** button or press the **Start / Stop** button ➢ Select **System / Calibration**
ALL CAPS	➢ Acronyms ➢ File names ➢ Path statements	➢ ASCII; QVI ➢ Edit INPUT.INI ➢ View C:\\W3_1
Initial Caps	➢ Proper nouns ➢ Product names ➢ Sections; figures	➢ System menu ➢ VMS ➢ See Section 3

Technical Support and Customer Service

QVI values all its customers. Please contact your local sales representative for:

- Customer service
- Questions and comments
- Information about additional VMS training or support for your application

Section 1

Introduction

Overview of VMS Operation

If you are new to using VMS, there are a few ideas you need to understand before you can experiment with it on your own. This section explains these most fundamental concepts. The ideas here will become second nature to you as you gain experience.

- You teach VMS a series of steps to perform in order to inspect a part. This series of steps is called a Measurement Program. An outline of the Measurement Program is shown in the *Measurement Steps* window, but there is more detail to each step than you see in this window.
- You save the Measurement Program on disk, as a file with any name you like (it should end in .VOY). Later, you can recall the program in VMS with a few mouse clicks. To save a Measurement Program, click the **Save** button in the File Group of the VMS toolbar.
- Once the Measurement Program is created or loaded, you just click on the Run button (the single green arrow in the toolbar) to have VMS perform all the steps of the inspection, automatically.
- There are many kinds of steps in measurement programs, but the most important ones are Feature measurement steps. A Feature is a point, line, circle, arc, slot, or plane that is part of the Part being inspected. For example, a straight edge on the Part is a line; a hole is a circle. A Measurement Program usually contains many Feature measurement steps.
- Each Feature measurement step uses the camera to measure a Feature by taking pictures and analyzing the images. It then judges several properties of the Feature (position, size, form, etc.) against tolerances you supply.
- You begin creating a Feature measurement step by clicking on the desired **Feature** button in the *Measurement Group* of the VMS toolbar. You get a dialog box to see and control all information about that Feature.
- While setting up a Feature measurement step, you set up one or more Finders as part of that step. A Finder uses the camera to look at one particular area of the part and get raw XYZ coordinates of point(s) along the edge.

- To create a Finder, you select the Finder type from the *Video* window toolbar. You then set the position of the camera, the light levels, the area to scan, the direction to scan, and perhaps some other details about how to analyze the image.
- You test-run the finder by double-clicking an open area in the *Video* window. VMS will display the found points in light blue, overlaid on the picture from the camera.
- You add the Finder to the Feature measurement step by clicking the **Finder Accept** (green check mark) in the *Video* window toolbar. The *Feature Measurement* dialog box shows pictures of the Finders you have in this step.
- If the Feature does not fit in the field of view, you move the stage with the Joystick to other places on the Feature, create Finders at those locations, and add them to the Feature measurement step.
- Back in the dialog box for the Feature measurement, you type in **Nominal** values for position, size, etc. You select which properties to inspect by turning on the check marks next to the **Property Inspection** (at the far left side of the box). You also enter **Tolerances** for each inspected property, so VMS can make a pass/fail determination on each property.
- After adding one or more Finders and specifying Nominals and Tolerances, and selecting Properties, you click the **Run** button in the *Feature Measurement* dialog box to have VMS re-run all Finders, determine the **Actual** measured values and make pass/fail determination. The results of the measurement are displayed in the *Results* window.
- When you are done, you click on **OK** to add the Feature measurement step to the Measurement Program.
- The *Features* window displays a picture of all the Features in the Measurement Program. While you are editing a Feature measurement step, all the Finders for the step are also displayed in the *Features* window.

Computer Components

This manual assumes that you know how to use a personal computer and Microsoft Windows. Listed below is additional information you should know about the user components provided as part of the system:

- Mouse
- Trackball
- Keyboard
- Joystick

Mouse

Use the left mouse button to complete most actions. The right mouse button is for special actions, such as opening context menus in certain windows. This manual refers to the right mouse button specifically by name when it is required.

Trackball

Some systems have a trackball instead of a mouse. The trackball has an extra button that is the equivalent to holding down the left button while you spin the ball. This makes it easier to drag items.

Keyboard

The [F1] key is reserved for Help. Press this key to display the Help menu on the monitor. The [Tab] key moves you from one field to another within a dialog box.

Joystick

The joystick is used to move the stage in the X and Y directions, and the optics and lens assembly in the Z (up-down) direction.

To Move	You Must
The stage along the X axis	Move the joystick right (positive X) and left (negative X).
The stage along the Y axis	Move the joystick forward (positive Y) and backward (negative Y).
The optic/lens assembly along the Z axis	Rotate the knob of the joystick to the left and right. This will move the lens closer to (clockwise) and further from (counterclockwise) the stage.

The speed of the joystick can be adjusted using the *Stage and Light Control* dialog box in VMS. You can set the speed of the joystick to your preference -- fast or slow. You can also use the joystick in discrete mode by holding the [Ctrl] key. While in either Fast or Slow continuous mode, pressing either of the buttons on the joystick will make it go 10 times slower.

While the fast and slow settings are self-explanatory, discrete is not. In fast and slow mode, the stage moves continuously when you move the joystick. In discrete mode, the stage moves the specified distance and stops. You must let go of the joystick and move it again to make it move further. If you set the joystick discrete distance to 1 millimeter and hold the [Ctrl] key, the stage will move 1 millimeter every time you move the joystick. Discrete mode is useful for precise, specific movement of the stage. When you know the distance that you want to move the stage -- for example, when you are working with a CAD file or Features repeated at a known distance -- discrete mode is useful.

Continuous moves in XY may also be performed by pressing the mouse button in the *Video* window while holding the [Alt] key.

Discrete moves in XY may also be performed using the keypad arrows while holding the [Ctrl] key.

Section 2

Looking at VMS Software

VMS Components

VMS comes in a variety of different configurations. All VMS configurations come with the basic software. In addition, the software may include either of the following optional components:

- **CAD Import** – allows you to import Computer-Aided Design (CAD) drawings into VMS. Once imported, you can use the physical specifications in the CAD drawings to help you build measurement programs.
- **Offline Programming** – allows you to create measurement programs offline – that is, when you are not physically connected to the system.

You can configure your system with any or all of the optional components. If you have purchased either or both of these components, please see the applicable appendix in this manual.

VMS Shortcut Keys

Shortcut Key	Description
[Ctrl] + N	Starts a new measurement program.
[Ctrl] + O	Opens an existing measurement program file.
[Ctrl] + S	Saves the current measurement program.
[Ctrl] + [Shift] + D	Activates the *DRO* window
[Ctrl] + [Shift] + F	Activates the *Features* window

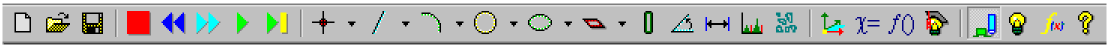
Shortcut Key	Description
[Ctrl] + [Shift] + L	Activates the *Stage & Lights* dialog box
[Ctrl] + [Shift] + M	Activates the *Measurement Steps* window
[Ctrl] + [Shift] + O	Activates the *Object Names* dialog box
[Ctrl] + [Shift] + R	Activates the *Results* window
[Ctrl] + [Shift] + S	Activates the current step-creation dialog box (if one exists)
[Ctrl] + [Shift] + V	Activates the *Video* window
[Pause] or [Ctrl] + [Break]	Stops a running measurement program. After clicking Stop, VMS finishes the step it is currently running, and then stops. The step that will be performed next, if you click Run or Single Step, is highlighted in green in the *Measurement Steps* window.
[Ctrl] + R	Resets the measurement program to the beginning. The program does not actually start until you click Run or Single Step.
[Ctrl] + G	Starts the program from any given step in the program. While the program is paused, select a step in the *Measurement Steps* window and click on this button. The selected step will turn green, indicating that this step will be the next one to run if the Run or Single Step buttons are clicked.
[F1]	Displays VMS Help.

Shortcut Key	Description
[F5]	Starts the inspection you have programmed. Normally it runs from the beginning, but sometimes you can pause in the middle of the program. The step that is about to be run is highlighted in green in the *Measurement Steps* window. After clicking the Run button, VMS will continue to the end of the measurement program, unless there is some kind of error or you halt it with the Stop button. After VMS has reached the end of the measurement program, it will reset to start the program from the beginning the next time the Run button is clicked.
[F8]	Runs just one step, the next one, which is highlighted in green in the *Measurement Steps* window. It will stop when the step is finished.
[Ctrl] + keypad #1, 2, 3, 4, 6, 7, 8, 9	Performs a discrete stage move in the indicated up, down, left right or diagonal direction. **Note:** The numeric keypad number/arrow keys are used. The [Num Lock] key must be on to use this capability.

VMS Toolbar

Toolbar Commands

The toolbar below the main VMS menu provides a means for you to perform commonly used operations with buttons.

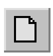
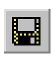


The VMS toolbar is divided into the following groups:

- File Group
- Program Control Group
- Measurement Group
- Programming Group
- General Assistance Group

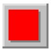
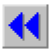
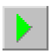
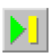
File Group

The buttons in this group are used to save and load VMS measurement programs to and from files. All of these buttons have keyboard shortcuts. The keyboard shortcuts can be activated while any VMS window is active (see *VMS Shortcut Keys*).

Icon	Name	Description
	New	Unloads the current measurement program and begins a new measurement program.
	Open	Opens an existing measurement program.
	Save	Saves the current measurement program.

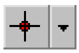
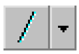
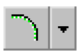
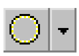
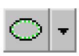
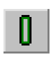
Program Control Group

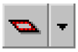
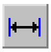
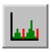
Options available in the *Program Control* group of the toolbar are listed in the table below. These buttons control how VMS automatically runs the sequence of inspections of the programs you have built. All of these buttons have keyboard shortcuts. The keyboard shortcuts can be activated while any VMS window is active (see *VMS Shortcut Keys*).

Icon	Name	Description
	Stop	Stops a running measurement program. After clicking **Stop**, VMS finishes the step it is currently running, and then stops. The step that is to be run next, if you click **Run** or **Single Step**, is highlighted in green in the *Measurement Steps* window.
	Restart	Resets the measurement program to the beginning. The program does not actually start running until you click **Run** or **Single Step**.
	Jump to Step	Starts the program from any given step in the program. While the program is paused, select a step in the *Measurement Steps* window and click on this button. The selected step will turn green, indicating that this step will be the next one to run if you click on the **Run** or **Single Step** buttons.
	Run	Starts the inspection you have programmed. Normally it runs from the beginning, but sometimes you can pause in the middle of the program. The step that is about to be run is highlighted in green in the *Measurement Steps* window. After clicking the **Run** button, VMS will continue to the end of the measurement program, unless there is some kind of error, a program breakpoint is reached, or you halt it with the **Stop** button. After VMS has reached the end of the measurement program, it will be set to start the program from the beginning the next time the **Run** button is clicked.
	Single Step	Runs just one step, the next one, which is highlighted in green in the *Measurement Steps* window. It will stop when the step is finished.

Measurement Group

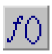
Select the appropriate measurement button to start describing a new Feature in the measurement process. Selecting one of these buttons is the same as selecting an option from the Measure menu.

Icon	Name	Description	Application
	Define Point	Measures or Constructs a Point Feature	There are seldom real Features on a part that are points. Measured points are often used to construct other Features such as Planes. Points may be constructed to define theoretical Features such as a corner where two edges intersect
	Define Line	Measures or Constructs a Line Feature	Most commonly, these represent straight edges of a part, although imaginary lines between two holes, etc. are also common.
	Define Arc	Measures or Constructs an Arc Feature	Rounded corners, cutouts, etc.
	Define Circle	Measures or Constructs a Circle Feature	Usually represents holes or cylindrical pins.
	Define Ellipse	Measures or Constructs an Ellipse Feature	Usually represents elliptical holes, pins, or radii.
	Construct Slot/Tab	Constructs a slot or a tab and determines if it is vertical or horizontal.	Square notches or tabs. Also, many rectangular Features do not require the combined size/position tolerancing that Slot/Tab provides, but the ease of constructing a midline, center and width makes this very useful.

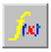
Icon	Name	Description	Application
	Define Plane	Measures or Constructs a Plane (Surface) Feature	Usually measured with Autofocus or Laser Finders, this often is used as a primary Datum for establishing coordinate systems. Flatness of a surface is another common inspection.
	Measure Angle	Measures the angle between two Features, such as lines.	
	Measure Distance	Measures the distance between two Features, such as points.	
	Define Histogram	Creates a histogram by tallying the number or pixels of each brightness value from 0 to 255.	Often used to determine whether or not a particular Feature exists in the field of view before performing other measurements.
	Blob Analysis	Measures a set (array) of Blob Features.	See *About Blob Analysis*

Programming Group

The buttons in this group are used to create some common programming steps. Note that many more types of programming steps are available through the VMS menus.

Icon	Name	Description	Application
	Define Align Block	Creates an Align Block containing a Plane-Line-Point Part Alignment Step.	Simplified Alignment block creation.
	Assignment Button	Creates a variable Assignment step.	Advanced programming.
	Procedure Button	Creates a Procedure Call step.	Advanced programming.
	Set Light Values Button	Creates a Set Light Values step.	Obtains consistent lighting on different machines or parts.

General Assistance Group

Icon	Name	Description
	Confirm Stage Motion	Asks before moving the stage when editing steps. This allows the user to check for possible obstacles before motion occurs.
	Stage & Lights	Displays the *Stage and Lights* dialog box.
	Function/Structure Help	Brings up a dialog box containing descriptions of VMS functions and structure components.
	General Help	Brings up the Help menu and associated files.

VMS Windows

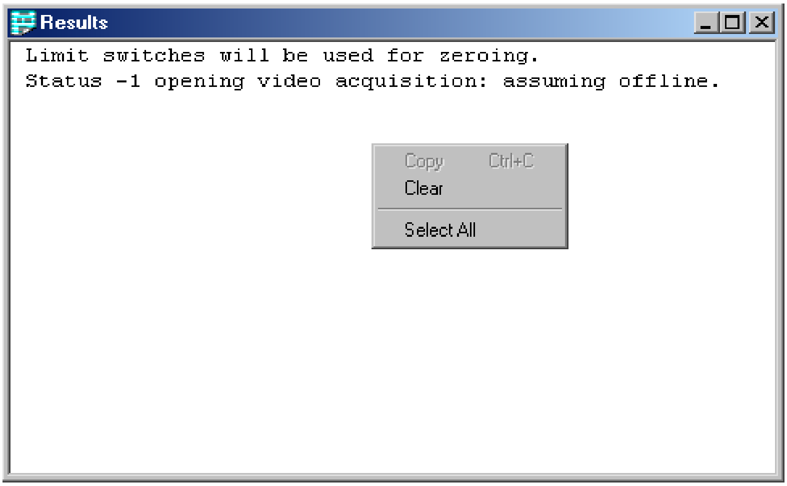
About VMS Windows

The VMS windows are used to create, edit, and run measurement programs. They display the video image, measurement steps, measurement results, measured Features on the part, and other information.

To change the window arrangement of the VMS application main window, move, close, resize, minimize and maximize individual windows in the display. After the window arrangement has been changed, you can return to the original window arrangement by selecting **Window/Arrange All** from the menu or you can save the customized window arrangement by selecting **Window/Save View** from the menu. Once you have saved the customized arrangement, it becomes the default setting that will be restored by **Arrange All**.

Results Window

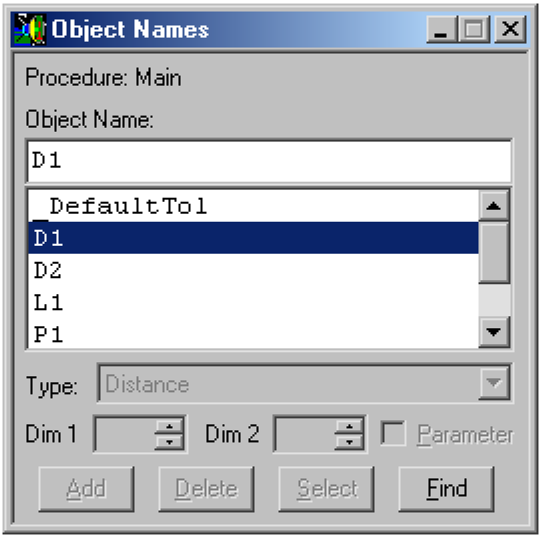
The *Results* window displays measurement results and other related messages, such as information from general output steps built into your measurement program. Use the scroll bars for the window if you cannot see the entire message in the window. You can also send measurement results to the printer or to a data file. Text in the *Results* window may be selected and copied to the clipboard using the context menu invoked by the right mouse button.



The format for measurement results is specified using the *Result Output Options* dialog box.

Object Names Dialog Box

The following figure shows the *Object Names* dialog box:



It may be displayed by:

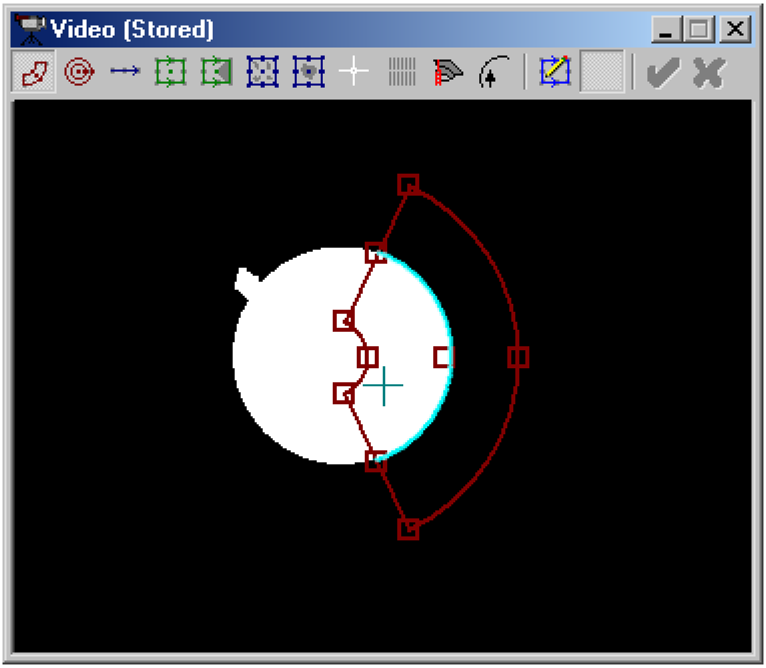
- Selecting Window/View/Object Names
- Selecting Program/Define/Declare Variable
- Pressing [Ctrl]+[Shift]+O simultaneously

The *Object Names* dialog box allows you to add Feature objects into a step. To add a Feature into a construction measurement step, the *Measurement Step* dialog box for the step must be active. In the *Object Names* dialog box, select or type the name of the Feature, or type an expression for a Feature object (e.g., expression for a component of an array of Features) in the **Name** field, then click Select. The Feature or expression will be added to the measurement step.

The *Object Names* dialog box also allows you to declare variables used in programming steps.

Video Window

The *Video* window displays a gray-scale camera image of a part on the stage. The image can be adjusted by changing the lighting, adjusting the lens or moving the stage. Use the *Video* window to direct the machine where and how to inspect a part, and to watch an inspection in progress.

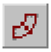
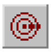
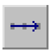
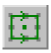
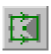
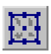
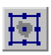
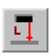
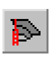


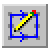
The *Video* window is used to create and edit the Finders that are used in measurement steps to create Features.

The *Video* window may also be used to move the stage. To do this, hold down the [Alt] key and press the left mouse button in the *Video* window. You can vary the direction and speed of motion by adjusting the direction and distance of the mouse cursor from the center of the *Video* window. When the mouse is near the center of the *Video* window, motion is slow; as the mouse is moved farther from the center, motion is faster. A light blue cross is displayed in the center of the *Video* window (as shown above) when the [Alt] key is pressed to aid this capability.

Video Window Toolbar

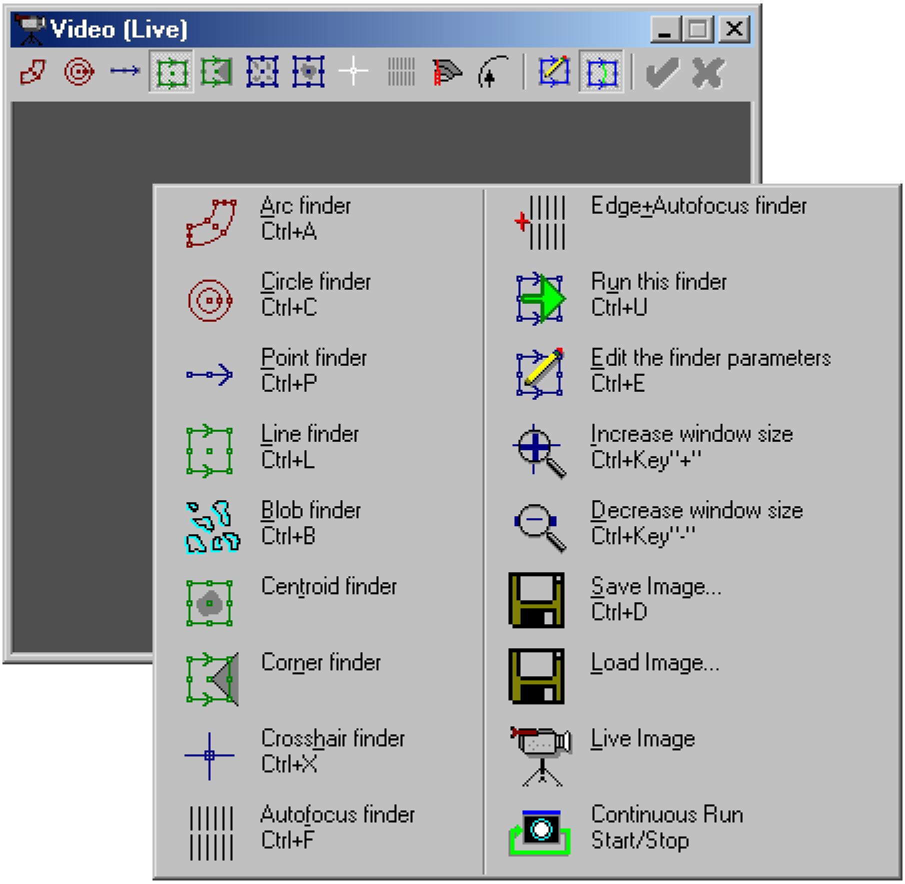
The *Video* window toolbar provides convenient access to common video operations. Many of these operations are also available from the *Video* window context menu.

Icon	Name	Description
	Arc Finder	Displays an Arc Finder
	Circle Finder	Displays a Circle Finder
	Point Finder	Displays a Point Finder
	Line Finder	Displays a Line Finder
	Corner Finder	Displays a Corner Finder
	Blob Finder	Displays a Blob Finder
	Centroid Finder	Displays a Centroid Finder
	Crosshair Finder	Displays a Crosshair Finder
	Autofocus Finder	Displays an Autofocus Finder
	Laser Finder	Displays a Point Laser Scan Finder
	Smart Finder	Runs the Smart Finder to locate an edge that the user clicks on and generate a Finder for that edge

Icon	Name	Description
	Edge Trace Finder	Displays an Edge Trace Finder
	Edit Finder Parameters	Displays a *Finder Parameters* dialog box for the current finder
	Show Finders	Enables or disables display of finders when a program is running (see *Rapid Inspection Mode*)
	Accept Finder	Accepts the current Finder and saves it in the step being created or edited
	Reject Finder	Cancels Finder editing and discards any changes to the Finder

Video Window Context Menu (Finder Menu)

Click the right mouse button anywhere inside the *Video* window to display the context menu with a list of Finders and other options. Finders are VMS on-screen tools used to define certain attributes of parts. Select Finders that are similar in shape to the Feature you want to define. For example, use the Line Finder to define straight edges or the Arc Finder to define curved ones. A Finder returns (finds) one or more points of a Feature that you are teaching VMS about. These returned points help you measure parts in the measurement programs you create. You can rotate, resize and move Finders to help you define part edges. You can select Finders from the *Video* window context menu or the *Video* window toolbar.



The most frequently used Finder options have keyboard shortcuts. To activate the Finder via the shortcut, make sure you are in the *Video* window, hold down the [Ctrl] key and type the letter specified (see *Video Window Shortcut Keys*). Finders can be rotated in the *Video* window by using the [Ctrl] and arrow keys. Finder options available and their typical applications are listed below:

Finder	Description and Placement	Typical Applications
Arc Finder	Defines all points on a radius. Place this Finder on the curved end of a Feature. The scan direction of the Arc Finder is from the inside curve to the outside curve.	Used to measure pins or holes that do not fit completely within the Field of View (FOV) in the *Video* window. Used to measure curved edges.
Circle Finder	Defines all points on a circle. The entire circle must be in the FOV. Place the inner circle of the Finder well inside of the Feature being measured, and then encircle the Feature being measured with the outer circle of the Finder.	Used to measure pins or holes on parts.
Point Finder	Defines a single point on an edge. Place the center point of the Finder over the edge of the Feature being measured. The arrow should point from the "clean" surface to the "dirty" or "noisy" surface.	Defines a single edge point.
Line Finder	Defines all points on a line or on a curve. Position the center point of the Finder on the light transition edge of the Feature being measured. The arrow should point from the "clean" surface to the "dirty" or "noisy" surface.	Finds multiple points along a straight edge of a part.

Finder	Description and Placement	Typical Applications
Centroid Finder	Defines the area or centroid of any Feature that falls within the FOV.	Quickly finds the center of a hole if you don't care about the size, shape or position of something (that fits in the FOV) that isn't a circle, slot, etc.
Corner Finder	Defines a minimum or maximum point on an edge	Locates a corner or tip, but doesn't measure the lines that make up the sides of it. Using this is faster than measuring the sides and intersecting them.
Crosshair Finder	Defines a single point on an edge. Usually positioned manually by the operator. Place the crosshairs on the Feature being measured.	Helps the operator identify locations that the automatic Finders can't, e.g., changes of texture rather than brightness, or items that because of loose manufacturing or fixturing tolerances, may not always fall within the FOV.
Edge+-Autofocus Finder	Defines a Z-axis point. Adds a focus step to an Edge Finder.	Brings edges into focus before the associated Edge Finder runs, in cases where manufacturing or fixturing tolerances make the Z height of the edge unpredictable.
Autofocus Finder	Defines a Z-axis point. Used without an Edge Finder.	Finds single points on a surface (close to parallel with the XY plane).

Finder	Description and Placement	Typical Applications
Run	Runs the Finder (same as double-clicking in the *Video* window).	
Edit	Allows you to fine-tune image processing controls, by brining up the *Finder* dialog box (the same as double-clicking on a Finder handle) so you can edit the Finder parameters.	
Increase and Decrease Window Size	Makes the *Video* window larger and smaller respectively.	
Save Image	Creates a .bmp file. The image will have a 256-color palette with the entire range of gray scale provided by the imaging system. If the image displayed is a “loaded” image, this function will not be allowed, since the image is already stored on disk.	Saves the current video image in a disk file.
Load Image	Attempts to load an image (.bmp) that has been stored by VMS. Most of the Finders can be used with this image provided there is no stage motion. The .bmp file is checked to ensure that it has a 256 color palette entry, fewer than 8 bits per pixel will not be displayed, nor will images with greater than 256 color palette entries. The banner in the image window will indicate “Stored”.	Loads a video image from a stored file.

Finder	Description and Placement	Typical Applications
Live Image	Takes the system back to displaying and using the live image from the camera. The banner in the image window will indicate “Live”.	Restores the use of live video after viewing a loaded image.

Video Window Shortcut Keys

Shortcut Key	Description
[Enter]	Accepts (stores) the current Finder into the measurement step being created or edited.
[Esc]	Rejects the Finder being edited.
[Ctrl] + right arrow	Rotates the Finder to 0 degrees
[Ctrl] + up arrow	Rotates the Finder to 90 degrees
[Ctrl] + left arrow	Rotates the Finder to 180 degrees
[Ctrl] + down arrow	Rotates the Finder to 270 degrees
[Ctrl] + A	Displays an Arc Finder
[Ctrl] + C	Displays a Circle Finder
[Ctrl] + D	Saves image to disk
[Ctrl] + E	Edits the current Finder
[Ctrl] + F	Displays an Autofocus Finder
[Ctrl] + L	Displays a Line (2D) Finder
[Ctrl] + P	Displays a Point (1D) Finder
[Ctrl] + U	Runs the current Finder

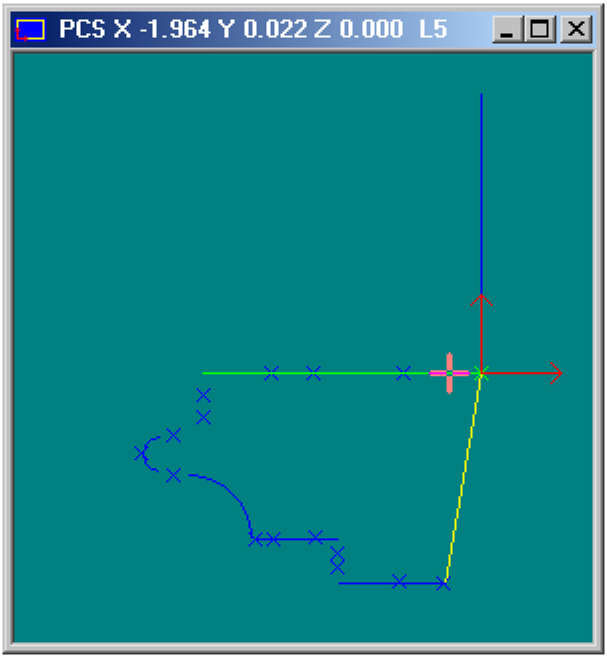
Shortcut Key	Description
[Ctrl] + X	Displays a Crosshair Finder
[Alt] + D	Enables the Finder Diagnostics mode

Features Window

The *Features* window displays a drawing of the stage and all of the known Features of the part being measured. Use the *Features* window to:

- Display a stage or part view.
- Position the stage with the mouse – Position the cursor at the desired stage position in the *Features* window and hold down [Ctrl] while clicking the left mouse button.
- Select, construct, or edit Features – select a Feature by clicking the mouse button on or near the Feature. Display or edit a dialog box containing information about the Feature by double-clicking the mouse button on or near the Feature.
- Set a mouse Z point for CAD import or offline programming.
- Delete Features and Finders.

A picture of the *Features* window is shown below.



The color of a Feature in the *Features* window indicates its status:

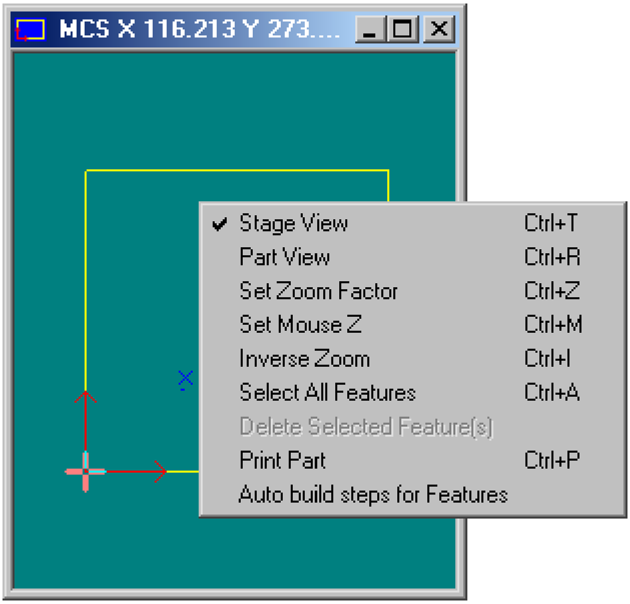
Color	Status
Green	All measured attributes of the Feature passed inspection
Red	One or more measured attributes of the Feature failed inspection
Yellow	You have selected the Feature
Blue	The Feature has not been selected or inspected yet

Reconfigure the color of Features in the *Features* window by selecting **Setup/Options/Colors** on the menu.

Features and Finders that appear in the *Features* window are usually drawn to scale, based on the current Zoom Factor. However, when the item is too small to see it is drawn at a minimum size so it can be seen in the window.

Features Window Context Menu

Click the right mouse button anywhere inside the *Features* window to display the context menu.



Options available in the *Features* window context menu are:

Option	Description
Stage View	Allows you to see where a part is located on the stage. The cross shows where the current camera position is. The angle of view is aligned with the stage's axes. For example, if the part is not straight on the stage, it will not appear straight in the *Features* window. As you move the mouse through the window, you see its coordinates, in relation to the Machine Coordinate System (MCS), displayed in the *Features* window title bar. You can Zoom In and Out.
Part View	Displays only the part being measured. The angle of view is aligned to the part, so the part appears straight in the *Features* window. In this view, you can only Zoom Out far enough to see all the known Features of the part. As you move the cursor through the window, you see its coordinates, in relation to the Part Coordinate System (PCS), displayed in the *Features* window title bar. You can Zoom In and Out.
Set Zoom Factor	Press the **1 to 1** button to zoom to an enlargement factor of 1. A factor of 1 means that the part as drawn on the screen is the same size as the actual part. As an alternative, enter a number in the text box and press [Enter] to zoom to the designated factor. For example, enter 1.5 to zoom to one and a half times the actual part size or .5 to zoom to half the actual part size. As an alternative to using the Zoom Factor command when **Zooming In**, click and hold down the mouse button. Drag the mouse to include the area you want to enlarge and release the mouse. Use the scroll bars in the window to move around the view of the stage. To **Zoom Out**, click the right mouse button to display the context menu and select **Stage View**.
Set Mouse Z	Allows you to set the Z coordinate to be used when the stage is moved using the mouse in the *Features* window. This is primarily used in offline mode.
Inverse Zoom	Changes the way the zoom box appears. The default is a rectangle around the perimeter of the area you are selecting. Select **Zoom Display** to toggle this command. This will change the rectangular area to a reverse video image.

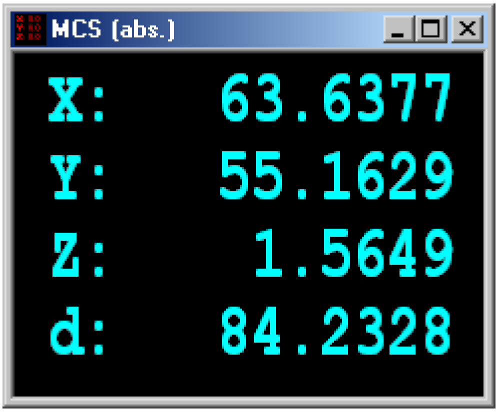
Option	Description
Select All Features	Selects all Features (when not editing a measurement step) or all Finders (when editing a measurement step) for use with a **Delete** command or an **Auto-build steps for Features** command.
Delete Selected Features	When not editing a measurement step, click on one or more Features in the *Features* window (or use **Select All Features**) and select this option from the menu to delete Features. The Features will be marked as unmeasured and deleted. If they are not used in any program step, they will be deleted from the program. When editing a measurement step, click on one or more finders in the *Feature* window and select **Delete Finder**(s) from the menu to delete the Finder from the step.
Print Part	Prints a drawing of the *Features* window in Part View.
Auto-build steps for Features	Inserts a measurement step for each selected Feature into the measurement program; see *Automatic Measurement Step Generation*.

Features Window Shortcut Keys

Shortcut Key	Description
[Ctrl] + A	Select all features
[Ctrl] + I	Turn inverse color zoom box on or off
[Ctrl] + M	Set mouse Z
[Ctrl] + P	Print a drawing of part features
[Ctrl] + R	Select part view
[Ctrl] + T	Select stage view
[Ctrl] +Z	Set zoom factor

Digital Readout (DRO) Window

The *DRO* window displays the coordinates of the center of the field of view (FOV).



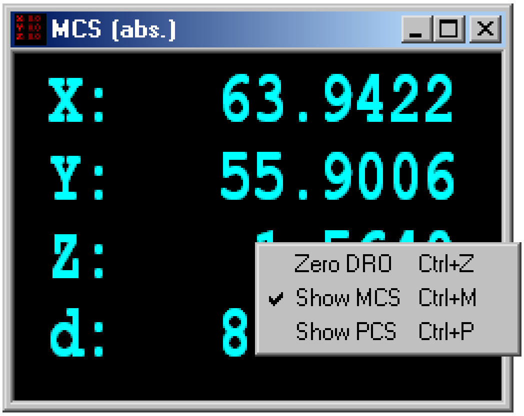
Coordinate	Description
X	Horizontal axis
Y	Vertical axis
Z	Height
D	Direct distance from zero (the square root of the sum of the squares of X, Y, Z)

The coordinates can be displayed in relation to the stage using the Machine Coordinate System (MCS) or the part using the Part Coordinate System (PCS). The *DRO* window also displays the coordinates in relation to an absolute coordinate system (abs.) or a coordinate system relative to a chosen point on the stage or part (rel.). Thus, the coordinate mode is displayed in the title bar of the *DRO* window as follows:

- MCS (abs.)
- MCS (rel.)
- PCS (abs.)
- PCS (rel.)

Digital Readout Context Menu

Click the right mouse button anywhere inside the *DRO* window to display the context menu.



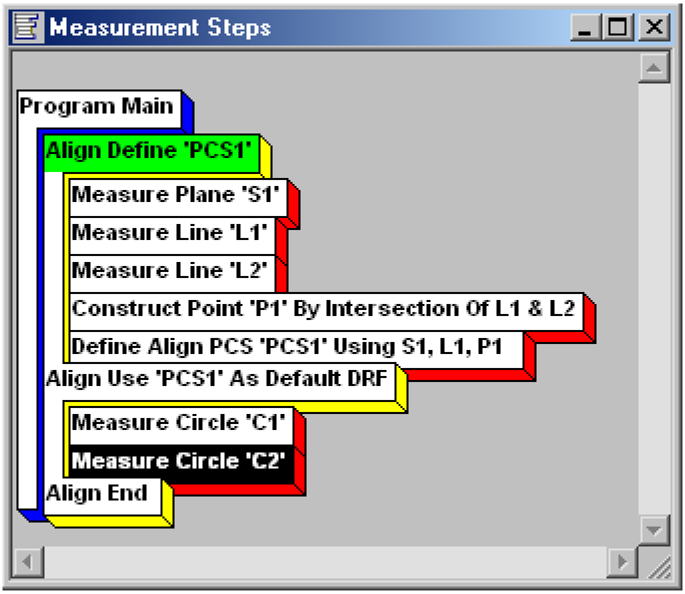
Option	Description
Zero DRO	Establishes the coordinate system relative to the center of the FOV or Manual Crosshair when the DRO is zeroed. (In PCS mode, the PCS angle is taken into account.) Zeroing the DRO is useful in conjunction with the Manual Crosshair Finder. When this Finder is selected, the DRO shows the coordinates of the center of the Crosshairs, not the center of the *Video* window. You can move the Manual Crosshair and the stage to line up with one edge on the part and then zero the DRO. When you move the Manual Crosshair Finder to another edge of a part, the *DRO* window will show the distance between the two points. To “zero the DRO” to a chosen point, use the joystick and Manual Crosshair finder to place the desired point in the center of the Manual Crosshair. Select either **Show MCS** or **PCS** and then select **Zero DRO.** To reset the coordinates to the absolute coordinate system, re-select **Show MCS** or **PCS**.
Show MCS	Sets the coordinate system in relation to the stage.
Show PCS	Sets the coordinate system in relation to the part being measured.

DRO Window Shortcut Keys

Shortcut Key	Description
[Ctrl] + M	Displays Machine Coordinate System (MCS)
[Ctrl] + P	Displays Parts Coordinate System (PCS)
[Ctrl] +Z	Zero DRO

Measurement Steps Window

The following figure shows an example of the *Measurement Steps* window:



The window displays the steps created in building a program to measure the part and the order in which the system will execute the steps to inspect the part. Steps displayed in the window have several distinguishing characteristics:

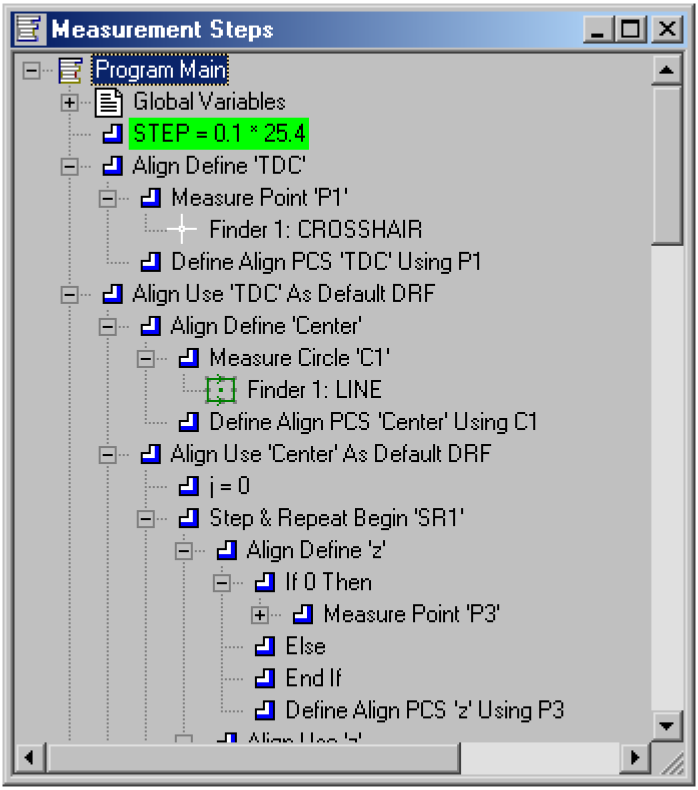
- Steps are 3-dimensional blocks that appear to be stacked.
- Various colors mark the sides of the steps.
- The side colors and stacking appearance show different levels in the program, indicating which steps of the program are contained in or controlled by other steps of the program.
- The first step displayed in the window is always the Main Program step. The blue sides of the step and the blue bar that extends down the left side of the program indicate that all the other steps are part of the Main Program.
- The top face of a step gives a brief description of the step. It is usually written in black on a white background. The colors on the face change to indicate special conditions

Color	Condition Indicated
Black with white printing	A selected step.
Bright green	The next step to run when the program has paused.
Dark green	The next step to run and it is also selected
Red with black text	The step is selected and a step is being edited or inserted.

The window shows steps in the order in which they will be executed.

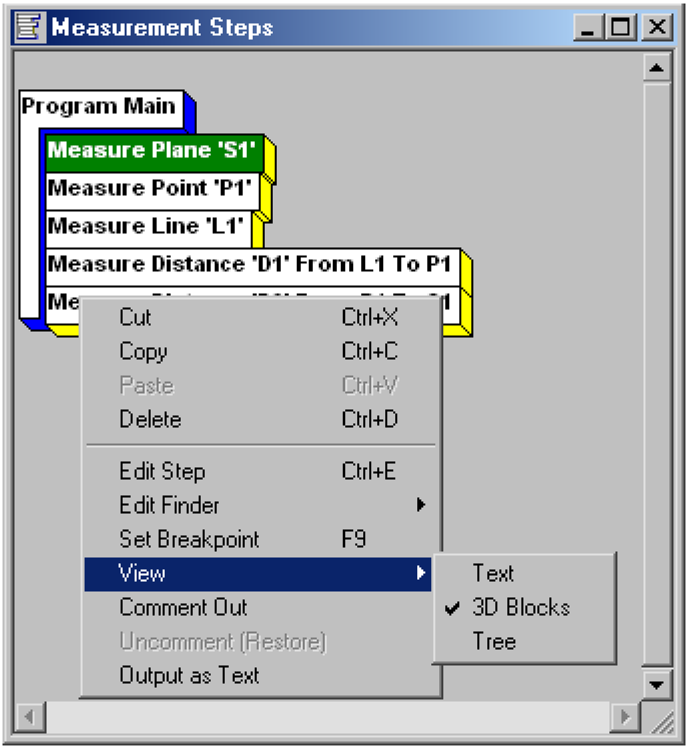
Measurement Steps Tree-View

The *Measurement Steps* window is also capable of displaying a tree-view of the program steps as shown below. This view allows variable and finder types to be viewed without opening additional dialog boxes. It also allows program structures, including the main program, procedures, align blocks, etc. to be collapsed to view a particular section of the program. To see a tree-view of the program, select **View/Tree** from the *Measurement Steps* context menu.



Measurement Steps Context Menu

Click the right mouse button anywhere inside the *Measurement Steps* window to display the following context menu:



The context menu provides operations on a selected step or marked block of steps.

Inserting, Moving and Deleting Steps

To insert, move, or delete steps in the *Measurement Steps* window:

- Highlight the desired step by clicking on the left mouse button, or a range of steps by holding down the [shift] key and clicking the mouse button on the first step of the range, then on the last step of the range. You may not select of range of steps that includes the opening, but not the closing of a Step and Repeat loop or other nesting block structure.
- To insert a step, highlight the step just above where you want to insert a step, then create the step you want to insert.
- To view or change information in a step, double-click the left mouse button on the desired step to bring up a dialog box to edit.
- To change the order of steps, highlight the step or range of steps you want to move, right click on the mouse button to bring up the Measurement Steps Context Menu, and select **Cut**. Highlight the step above where you would like to insert the step(s) and select **Paste**.
- To delete a step or steps, highlight the desired step(s), click the right mouse button and select **Delete**.
- To comment out a step or steps, highlight the desired step(s), click the right mouse button and select **Comment Out** from the context menu. The step(s) may be restored to the program by selecting them individually or as a group and selecting **Uncomment (Restore)** from the context menu.

Setting and Using Breakpoints

A breakpoint is used to stop program execution at a particular place in the program. When the program is run, VMS will stop the program just before executing a step that has a breakpoint set. After the program stops at the breakpoint, clicking the **Run** button will cause VMS to resume execution starting with that step. Breakpoints may be set on more than one step, if desired. To set or clear a breakpoint on a program step, highlight the step where the breakpoint is to be set. Then press **F9** or:

1. Click the right mouse button to activate the *Measurement Steps* context menu
2. Select **Set Breakpoint** or **Clear Breakpoint**.

Outputting a Program Listing

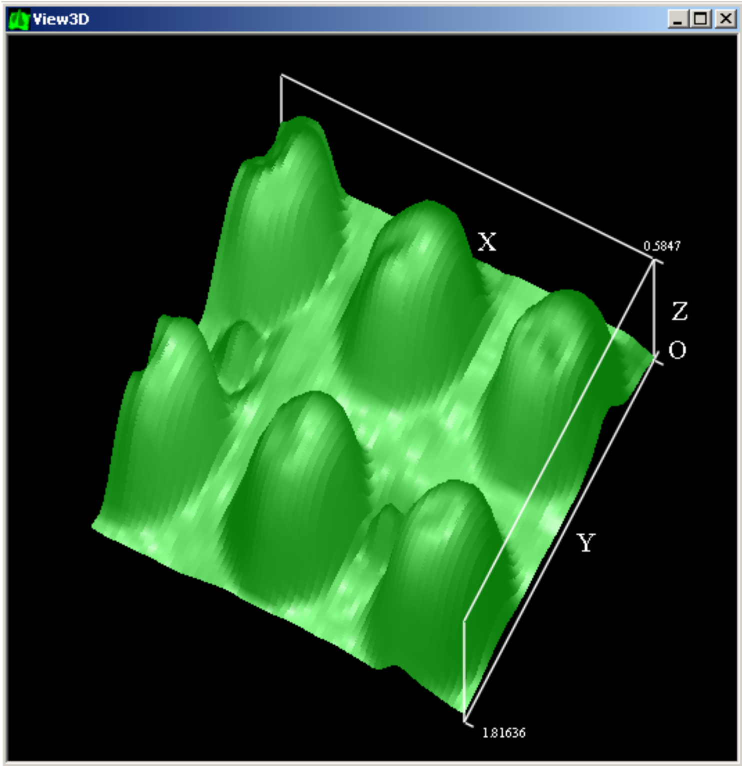
To create a program listing by outputting the contents of the *Measurement Steps* window to a text file, right-click in the *Measurement Steps* window and select **Output as Text** from the *Measurement Steps* context menu. The text will be placed in the same folder as the current program, with the same file name as the current program, but with a ".vpl" extension.

Measurement Steps Shortcut Keys

Shortcut Key	Description
[Ctrl] + C	Copy the selected step or steps to the clipboard.
[Ctrl] + D	Delete the selected step or steps.
[Ctrl] + E	Edit the currently selected step.
[Ctrl] + V	Paste a step or steps from the clipboard after the currently selected step.
[Ctrl] + X	Cut the selected step or steps to the clipboard.
F9	Set or remove a breakpoint at the currently selected step.

View3D Window

The *View3D* window is a new feature in VMS 7.0. View3D displays 2D or 3D data using OpenGL. View3D can be used to visualize data from Area Multi-Focus (AMF) and Point Laser Scan Finders. To display the *View3D* window, select **Window/View/View3D** from the menu.

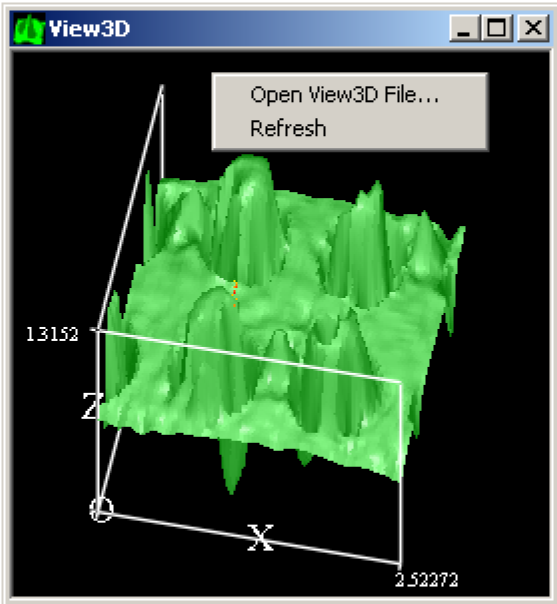


Displaying Data in the View3D Window

To display View3D data, the *View3D* window must first be opened. The following are three ways that data can be displayed in the *View3D* window.

1. From a data file:

   Click the right mouse button anywhere inside the *View3D* window to display the context menu:

   Selecting **Open View3D File** allows the user to load a View3D data file so that a set of 2D or 3D data can be displayed and visualized. **Refresh** can be used to update the window if the loaded file is subsequently changed.

2. Using the VMS setAMF() function:

   The VMS function setAMF() is used to set or clear the Area Multi-Focus (AMF) mode for Autofocus. AMF mode is only available on systems with the software auto focus capability. In AMF mode, View3D can display AMF data directly. To set the AMF mode for View3d display, call the VMS function

   set_AMF(INTEGER mode, INTEGER resultmode,

   INTEGER xsize, INTEGER ysize,

   INTEGER xspacing, INTEGER yspacing,

   INTEGER medianfiltsize, INTEGER smoothfiltsize)

   with mode=1 and resultmode=0. Then run software auto focus. The 3D data acquired from AMF will be automatically sent to and displayed in the *View3D* window. See the VMS Function help for details of the set_AMF() function.

3. Using the VMS view3dio() function:

   The VMS function view3dio(STRING filepath, INTEGER mode) can be used to load a View3D file. Simply specify the View3D data file path and specify mode=0 (mode=0 indicates loading a View3D data file and mode=1 indicates saving a View3D data file.) See the VMS function help for details of the view3dio() function.

Once the data is displayed, the View3D display can be rotated in X, Y, and Z by clicking and dragging the left mouse button. This allows the display to be viewed from any angle.

View3D Data File Formats

A View3D data file can be in either of the following two data formats:

- 2D data format:

  The first line of the View3D data 2D format is the file header. The file header contains items <number of x data points>, <number of y data points>, <x spacing in real units>, <y spacing in real units>. The items <number of x data points> and <number of y data points> are mandatory. The items <x spacing in real units>, <y spacing in real units> are optional, and are assumed to be ones if they are not specified.

  Following the file header are <number of y data points> rows and <number of x data points> columns of 2D format data (each item indicates only z values). The second row and last row of data points should have the minimum and maximum Y coordinates, respectively. The left most and right most columns should have the minimum and maximum X coordinates.

- 3D data format:

  The first line of the View3D data 2D format is the file header. The file header contains items <number of x data points>, <number of y data points>, <x spacing in real units>, <y spacing in real units>. The items <number of x data points> and <number of y data points> are mandatory. The items <x spacing in real units>, <y spacing in real units> are optional, and are assumed to be ones if they are not specified.

  Following the file header are <number of y data points> × <number of x data points> rows and 3 columns of 3D format data (each row contains x, y, z values for a data point). The 3D data format is mostly used to store 3D data from a laser scan. The 3D data points should be sequentially acquired from a set of grid points defined in laser scan finders and in the scan parameters.

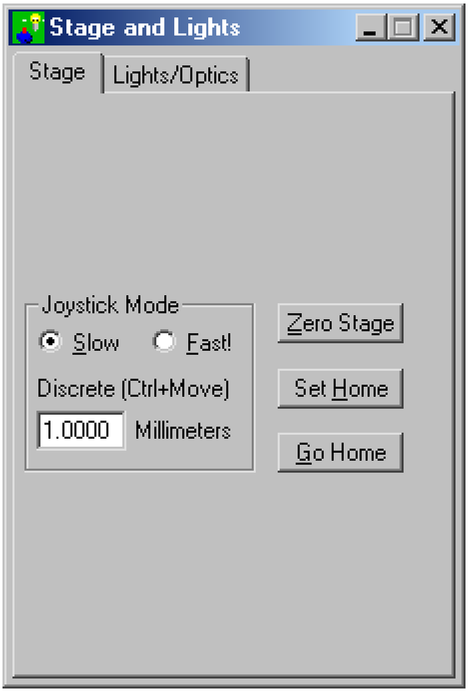
An example of the 3D file format is shown below.

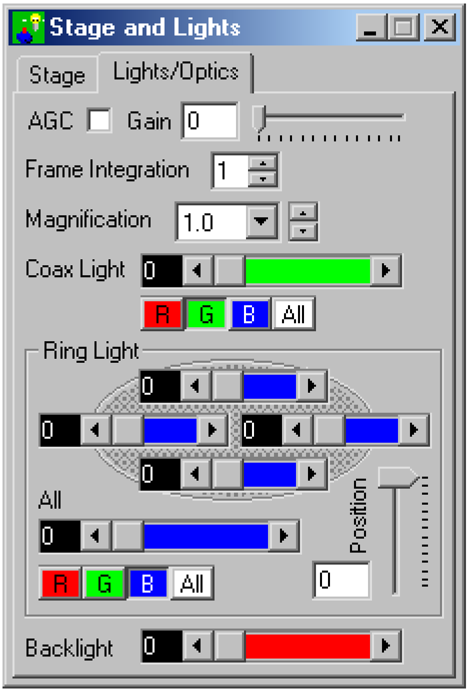
```
50 50

0 0 0

0 -0.3 44.7

0 -0.6 -80.6

0 -0.9 -44.9

0 -1.2 -42.2

0 -1.5 -43

0 -1.8 -40.8

0 -2.1 -41.5

0 -2.4 -36.9

0 -2.7 -35.2

0 -3 -31.5

0 -3.3 -28.8

0 -3.6 -28.6

0 -3.9 -27.6

0 -4.2 -33.7

0 -4.5 -38.1

0 -4.8 -40.5

0 -5.1 -47.6

0 -5.4 -58.8

0 -5.7 -71.8

0 -6 -76.2

0 -6.3 -78.4

0 -6.6 -85
```

```
0 -6.9 -90.3

0 -7.2 -92.5

0 -7.5 -96.7

0 -7.8 -104.7

0 -8.1 -106.9

0 -8.4 -112.5

0 -8.7 -119.1

0 -9 -125.2

0 -9.3 -135

0 -9.6 -133.3

0 -9.9 -137

0 -10.2 -141.4

0 -10.5 -145

...
```

Stage and Lights Dialog Box

The *Stage and Lights* dialog box is part of the standard display; you can also access it by clicking on the light bulb icon in the toolbar. Depending on the options installed, it may have separate **Stage** and **Lights/Optics** tabs as shown below.





The *Stage and Lights* dialog box allows you to:

- Zero the Stage
- Select the Joystick Mode
- Set Home and Go Home
- Adjust Light Values
- Set Frame Integration and Programmable Gain

Stage Control

Zeroing the Stage

When you click **Zero Stage** in the *Stage and Lights* dialog box, the stage seeks out its limits of travel along its X, Y, and Z-axes. The stage then comes to rest at its preprogrammed "home." Zero Stage must be performed once each time VMS is started. As visual reminders, if the stage is not zeroed:

- The status bar at the bottom of the window says "Not Zeroed" in red
- The *Digital Readout* (DRO) window does not display any numbers
- You cannot select any Finders in the *Video* window context menu

Selecting the Joystick Mode

You can move the stage by moving the joystick. The stage moves in the same direction in which you push the joystick. The speed of the joystick can be adjusted using the *Stage and Lights* dialog box in VMS. The joystick has three modes:

Mode	Description
Slow	The stage moves as long as you push the joystick, or until the stage reaches its limit of travel. This mode is for moving the stage when you are close to where you want to be.
Fast	The stage moves as long as you push the joystick, or until the stage reaches its limit of travel. This mode is for moving long distances quickly and is especially useful when writing programs.
Discrete	The stage moves a fixed distance in the X, Y, or Z coordinate each time you move the joystick. You must release the joystick and then move it again to perform successive movements. Set the fixed distance of movement by entering a number in the **Discrete** box. Discrete mode is then selected by holding down the control (Ctrl) key while moving the joystick in the desired direction. For example, if you set the discrete distance to 1 millimeter and hold the [Ctrl] key, the stage will move 1 millimeter every time you move the joystick.

	Discrete mode is useful for precise, specific movement of the stage. When you know the distance that you want to move the stage -- for example, when you are working with a CAD file or Features repeated at a known distance -- discrete mode is useful.

Continuous moves in XY may also be performed by pressing the mouse button in the *Video* window while holding the [Alt] key.

Discrete moves in XY may also be performed using the keypad arrows while holding the [Ctrl] key.

Set Home and Go Home

Click **Set Home** in the *Stage and Lights* dialog box to establish the current XYZ position as the Home position. Once you have set "home," to return the stage to the established position at any time, click **Go Home**. This is not the same thing as returning to the stage's zero position. This is simply an easy way to return the stage to a location you specify. Home is typically defined as the primary datum (a reference point) of a sample while creating a measurement program.

Lights and Optics Control

Adjusting Light Values

In the *Stage and Lights* dialog box, the brightness of the lights is displayed in numbers ranging from 0 (off) to 255 (full on). You can control the brightness of the lights by:

- Clicking and dragging the scroll box ("thumb") in the horizontal scroll bar
- Clicking in an open area left or right of the scroll box to change the settings in increments of 10
- Clicking on the end arrows of the scroll bars to change the settings in increments of 1
- Entering a specific number in the box to the left of the scroll bar

Some lights have more than one color available. You can change the color using the button bar below the scroll bar.

Frame Integration and Programmable Gain

AGC On/Off, Frame Integration and Programmable Gain are standard on current systems. These capabilities are controlled through the *Stage and Lights* dialog Box.

Automatic Gain Control (AGC) is a function inside the video camera that adjusts the brightness and contrast of the picture to a preset level. It automatically compensates for any brightness level, within a limited range. You can see it in action by turning down the light level a notch: first the picture gets darker because you turned the light down; but in about 1/2 second it brightens up as the AGC compensates. If you turn the light back up, it gets brighter momentarily but then returns to the preset level again.

This is not always desirable. For example, if there is a dark feature surrounded by a bright area, the AGC will turn down the brightness and the dark feature will be *too* dark. By turning the AGC OFF and adjusting the light levels and gain (see Section *Programmable Gain*), you can bring out the dark feature to measure it. The bright areas will be saturated, but that is not a problem if you are not measuring there.

Use the checkbox in the *Stage and Lights* dialog box to turn AGC On or Off. When you program a Finder and add it to a step, the status of AGC On/Off will be saved with that Finder, just as the light levels are. When the Finder is run during the course of inspection, the AGC will be turned On or Off as programmed. Finders that were created on systems that did not have AGC On/Off are assumed to have AGC On.

QVI video measuring systems often have to wait for the AGC to settle, so turning it Off can help improve throughput in applications where AGC is not necessary.

If you find that you need both high light levels and high gain for an image, you might obtain better results using Frame Integration.

Programmable Gain

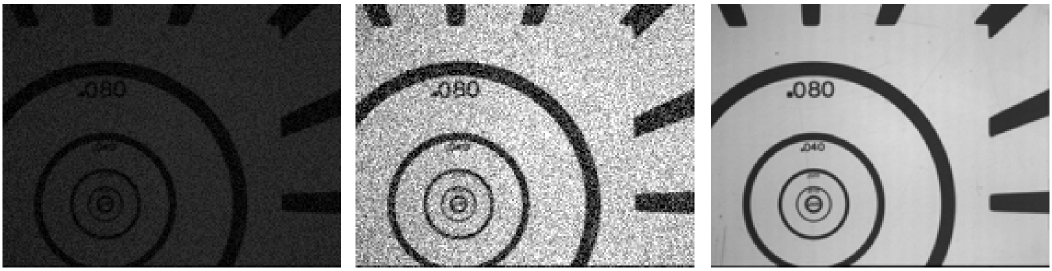
When AGC is Off in the *Stage and Lights* dialog box, you have the ability to set the gain to a fixed value. For the best image, adjust the horizontal slider. This value, like the AGC On/Off state, is saved with each Finder.

You will find that it is possible to get the same image brightness by using low lights and high gain, or by using high lights and low gain. The latter is usually more desirable because it will result in less "noise" in the image. But if you find that you need both high light levels and high gain, you might have better results using Frame Integration.

Frame Integration

The camera normally sends out 25 video frames per second. Each frame is created by collecting light for 1/25 second, analogous to a camera shutter exposing film.

If the scene is dark, the camera's amplifier (via AGC or a high fixed gain) can brighten it to a point. If the original exposed image is very dark, it will also contain a lot of "noise", that is, graininess and lack of clarity. Amplifying this very dark image also amplifies the noise; the resulting image, while it may have good overall brightness, isn’t very clear and gives unacceptable errors in measurements.



Original AGC / Contrast enh. Frame Integration

Frame Integration causes the camera to collect light for a longer period of time, which is analogous to increasing the exposure time on film. The resulting image is brighter and has less noise. So it is a true improvement in the image, not an amplification of a poor image. Parts that are otherwise too dark to see may be seen clearly and measured accurately.

Upgrading a system that does not have Frame Integration will require new equipment installation as well as VMS 6.0 or higher.

In the *Stage and Lights* dialog box, the Frame Integration number controls how many times the exposure time is multiplied. It is in multiples of the normal time. So for a setting of 2, each frame is exposed for twice as long as normal; a setting of three exposes for 3 times normal. The maximum allowed is 8. The setting is saved with the Finder, just like the light settings, and will be used when the program is run. Finders created on systems without Frame Integration will assume the normal setting of 1.

Longer exposure time means that "live" video will be at a slower frame rate, and taking each image for Finders will take longer too. The time multiplication factor is the frame integration number plus one. (The +1 is due to the nature of the timing circuits involved.) For example, with frame integration set to 3, it will actually take 4 times as long for each frame. Thus, each picture for a Finder would take 4/25 second instead of 1/25 second, and "live" video would be at 6.25 frames per second instead of 25 frames per second.

The increased time to take a picture for a Finder will be only a small percent of the inspection times in most cases. Still, it would help to take advantage (when possible) of the picture-taking optimization VMS already has: if several Finders in a row can be programmed without moving the stage or changing the lights, it will not take a separate picture for each Finder. Minimize Frame Integration by using the brightest light settings possible.

The increased time due to Frame Integration is, however, significant when used with Autofocus. For example, instead of taking the typical 2 seconds, the auto focus process could take 8 seconds for a Frame Integration setting of 3 (4 times normal). Also note that the Z-axis accuracy and repeatability specifications apply only to non-Frame-Integration Autofocus. When Frame Integration is used, the performance is slightly degraded, but still good. The alternative would be no Autofocus in these poorly lit situations.

Section 3

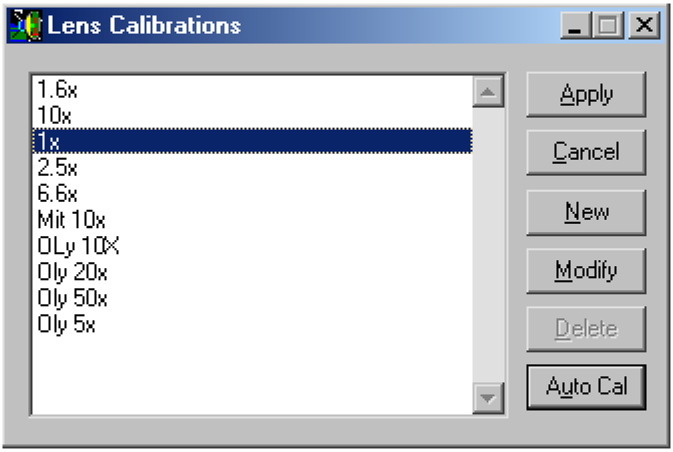
Setup Operations

Lens Calibration

Before measurement, the lens or lenses to be used must be calibrated. Each lens calibration can be given a separate name, if desired, to distinguish between two lenses of the same type or between calibrations done on different days.

When measurement steps are saved in a measurement program, the lens type used is also saved. When the measurement program is recalled, VMS checks the lens calibration that was in effect when the program was saved. If the lens calibration for that lens type is not in effect, VMS will warn you. If you copy a measurement program from one machine to another, the same lens type must be available on both machines or the program cannot be run.

Select **Setup/Calibration** from the menu. The *Lens Calibrations* dialog box is displayed.



To calibrate a lens, follow the procedures in the subsequent sections. The Auto-Calibration Method is recommended; use the older Circle Calibration Method for single-magnification optics only if you do not have an artifact for the newer method.

With dual-magnification optics, each lens must be calibrated for both paths. Auto-Calibration is normally used because it computes the rotational offsets of the two paths as well as the pixel sizes.

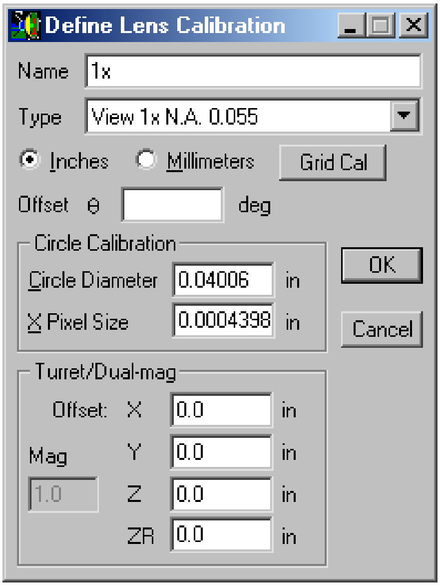
After you have calibrated the lenses that your measurement programs use, you can use the *Lens Calibrations* dialog box to select a calibration when you change lenses. To do this, select the name of the calibration you want to use from the names listed and click **Apply**.

Auto-Calibration Method (FOV Calibration)

VMS 7.0 provides a new and better way to calibrate the lens automatically. In addition to computing the pixel size(s), this method computes a correction for misalignment of the camera(s) with the stage axes.

1. If you are creating a *new* lens calibration, you need to first create a nominal calibration and give it a name:

   a. Select **New** in the *Lens Calibrations* dialog box. The *Define Lens Calibration* dialog box is displayed.

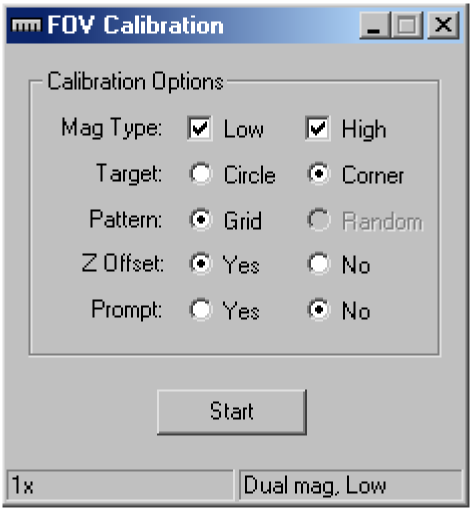


b. Select the type of lens being calibrated. Enter a descriptive name for the lens you are calibrating in the *Lens Name* text box (optional). Typically, a descriptive name for lens calibration is based on the magnification level of the lens, for example, 10x Magnification.

c. Click **OK** to create the new lens calibration.

d. If the system is equipped with dual-magnification optics, select the Hi-Mag path in the *Stage and Lights* dialog box and repeat steps 1-3 to create a default calibration for the Hi-Mag path.

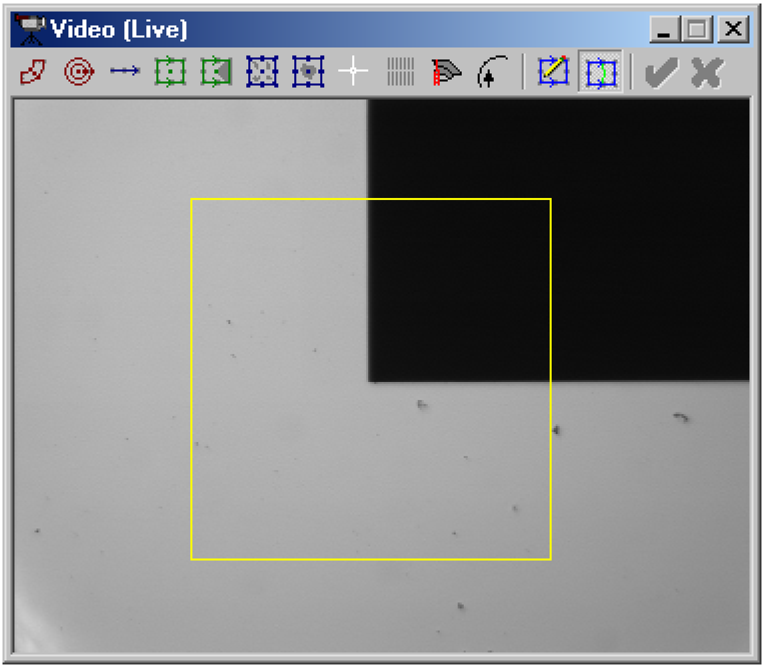
2. After the lens calibration has been created, select **Auto-Cal** in the *Lens Calibrations* dialog box. The *FOV Calibration* window is displayed.



3. Any small, well-defined (backlit) corner or field-of-view circle may be used as an artifact. Select the type of artifact you are using and any other options described below that you want to use for the calibration.

Mag Type	The checkboxes of Mag Type indicate whether **Low** magnification, **High** magnification or both are to be calibrated. For a dual magnification system, both **Low** and **High** are checked by default. For a single magnification system, only **Low** should be checked.
Target	The radio buttons of Target indicate which type of target is to be used. If **Corner** is selected, a well-defined, high contrast corner target (either white on black, or black on white) is used. The two edges of the corner target should be perpendicular to each other. The edges of the corner target should be aligned parallel to the *Video* window. For a corner target, gradient edge detection is used to find the edges of the corner target. If **Circle** is selected, a well-defined, high contrast circle (either white on black or black on white) is used. For a circle target, blob detection is used to find the circle edge. This is not quite as accurate as the gradient edge detection used for a corner target.
Pattern	Currently, the scan pattern of FOV calibration is set to **Grid** only.
Z Offset	The **Yes/No** radio buttons of Z Offset indicate whether the Z offset between the low magnification path and the high magnification path is to be calibrated. This option is only available for dual magnification systems.
Prompt	The radio buttons indicate whether a message box will pop up during each step of FOV calibration.

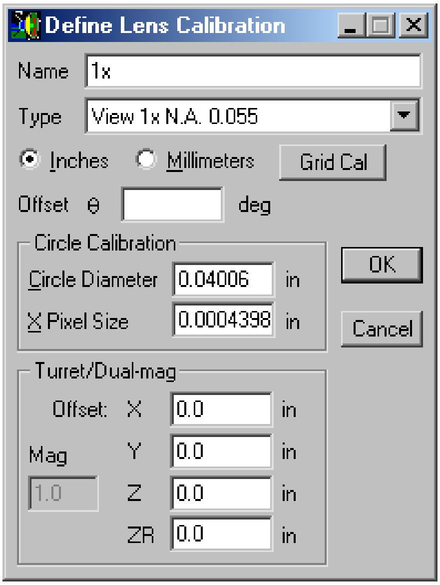
4. Position the corner or circle near the center of the *Video* window. If a corner is used, it should look similar to the following example



5. Press the **Start** button. The stage will be moved automatically to position the corner (or circle) in a grid pattern of locations within the field-of-view (FOV.) When it is finished, close the *FOV Calibration* window and the *Lens Calibrations* dialog box.

Circle Calibration Method

1. Select **New** (or **Modify** to recalibrate an existing lens) in the *Lens Calibrations* dialog box. The *Define Lens Calibration* dialog box is displayed.



2. Select the type of lens being calibrated. Enter a descriptive name for the lens you are calibrating in the Lens Name text box (optional). Typically, a descriptive name for lens calibration is based on the magnification level of the lens, for example, 10x Magnification. Note the default pixel size for the lens type selected.
3. Select the appropriate unit of measurement in the *Calibration* dialog box, based on the unit of measurement of the calibration standard. QVI calibration standards are based on inches.

   If you plan to measure parts that are based on metric measurements, you still need to calibrate the lens based on the unit of measurement of the calibration standard. Then, when you measure in metric units, VMS will convert the lens calibration data to metric.
4. Clean the calibration standard with a chamois cloth or other soft cloth and place the standard on the stage.
5. Turn the backlight on no higher than required for best contrast.

6. Move the joystick so that the “target” on the calibration standard is centered in the *Video* window. You can use the Autofocus Finder in edge mode to focus the lens on the target.
7. Place the mouse pointer over the handle on the outer circle of the red-colored Circle Finder. Click and hold down the left mouse button and drag the handle to enlarge the outer circle. The calibration standard’s target includes several concentric circles. Place the outer circle of the Circle Finder around the target’s largest circle that is *entirely visible* in the *Video* window.

   The inner circle should be made large enough so that it doesn’t include smaller calibration circles. If you need more information on manipulating Finders, see *Circle Finder*.
8. Double-click the left mouse button on one of the Circle Finder control handles. Th*e Circular Edge Finder* dialog box is displayed.

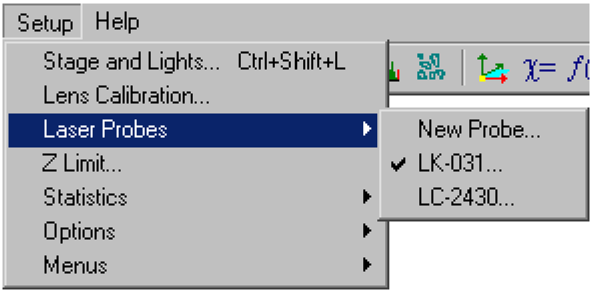
   The Finder needs to find the *outer* edge. As the Finder scans outward, the outer edge will be a “falling” edge (light to dark), since the calibration circle is white on a black background. Select **Falling** under **Polarity** and select **Strongest** under **Edge**. Some calibration standards may be a black circle on a white background; for these you would choose **Rising**.
9. Click the **Close** button in the *Circular Edge Finder* dialog box.
10. In the *Define Lens Calibration* dialog box, enter the nominal value of the circle you’re focusing on in the Circle Diameter text field. A number, such as .080, is visible next to the circle in the *Video* window. Find this circle’s number on the specification sheet that accompanies your calibration standard, and the nominal value (e.g., .08001) will be next to it.

11. Double-click on an open area in the *Video* window to run the Finder. Make sure it found the edges around the outside of the circle, cleanly. Re-adjust the light level and/or Finder attributes if necessary. The image should look similar to this example:

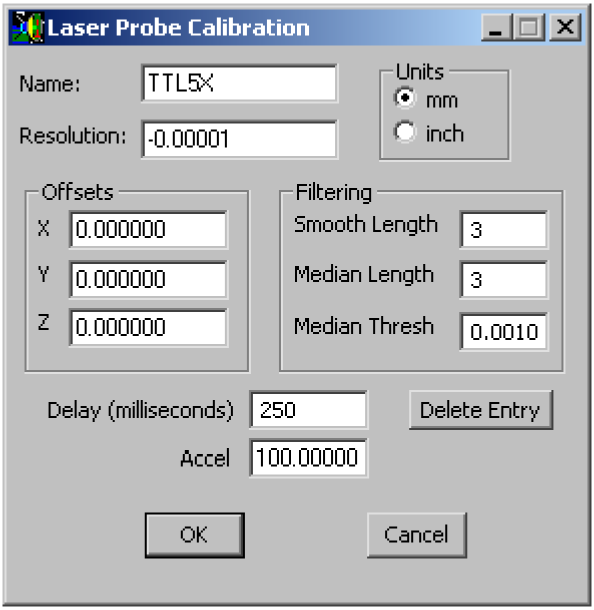
12. Click on the green **Check Mark** in the *Video* window toolbar. VMS calculates X pixel size and inserts this value in the *Define Lens Calibration* dialog box.
13. Check the value in the X Pixel Size text box against the default noted in Step 2 to make sure it seems reasonable.
14. Click **OK** to save and apply the calibration.

Laser Calibration

There is currently no built-in automated calibration procedure for Laser probes; however, the process can be made faster by running the LaserCalibration.voy measurement program to do the calibration. This program uses the VMS calibration functions to set the calibrated values. Before running the LaserCalibration.voy program, some nominal data must be entered by the user. This is done with a special dialog box called up through the *Setup* menu



More than one laser calibration can be defined, similar to multiple lens calibrations. You must set up a separate calibration for each laser/lens combination. Selecting **New Probe** or an existing Laser Probe displays the following dialog box:



You need to enter the X and Y offsets, if any, between the center of the FOV and the zero point in the laser's range, and also the resolution of the laser readings themselves (what distance one count from the A/D represents).

1. Enter the approximate A/D **Resolution**, based on past experience with this type of laser. The LaserCalibration.voy program will obtain a refined value. The value may need to be negative. If a positive value makes the LaserCalibration.voy program run without stopping, stop it and change the value to negative. The value -0.00001 (note the minus sign) shown in the above picture is recommended as a starting value for the TTL-G laser.

2. This step is unnecessary for TTL laser probes because the X and Y offsets are nearly zero. Determining the approximate X and Y **Offsets** can be done by looking at some point-like Feature with the camera, zeroing the DRO, and then moving the laser onto the point. The readings in the DRO are then the offsets that should be entered in the *Laser Calibration* dialog box. If you need to correlate Features measured with the laser to Features measured with Video, then a more accurate offset can be determined as follows.

3. To obtain accurate X and Y **Offsets**, program Video and Laser Finders that go across steps (Z edges.) Each step is aligned with the axis to be calibrated (X or Y.) Use the **Find Edge** checkbox in the Laser Finders to locate the X or Y coordinate of the step with the Laser. The difference between the Laser point X coordinate and the Video point X coordinate for the X-axis step gives the Laser X Offset. The Y offset is similarly computed from the measured Y coordinates of the Y-axis step.

4. **Accel** should be set as high as practical to minimize the number of extra points acquired during acceleration, particularly if high scan rates will be used. This helps to prevent Laser Finders from exceeding limits on the total number of points that can be acquired (see *Laser Parameter Interactions*). The number of extra points acquired during acceleration is given by:

   accel points = Scan Rate x Velocity / **Accel**

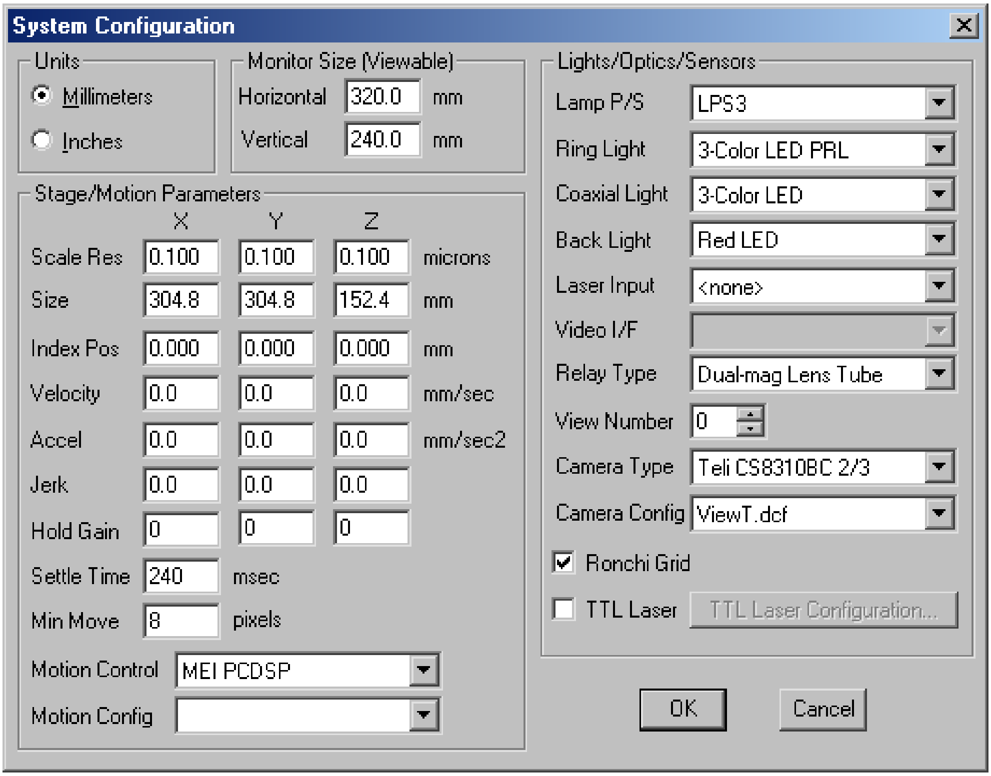
   For example, if a Scan Rate of 1000 pts/sec and a Velocity of 25 mm/sec are used in a Finder, an **Accel** value of 100 will produce 250 extra points. A value of 100 should be good for scan rates up to 1000 pts/sec.

5. Click **OK** to save the nominal calibration data.

   Run the LaserCalibration.voy program. The laser resolution and Z offset will be determined and saved in the currently selected laser calibration. If the LaserCalibration.voy program runs more than five minutes without stopping, stop it manually and edit the nominal calibration data. Change the sign of the laser resolution, click **OK**, and rerun LaserCalibration.voy.

System Configuration Dialog Box

Several lighting options are available. Select **Setup/Options/System** to access the *System Configuration* dialog box.

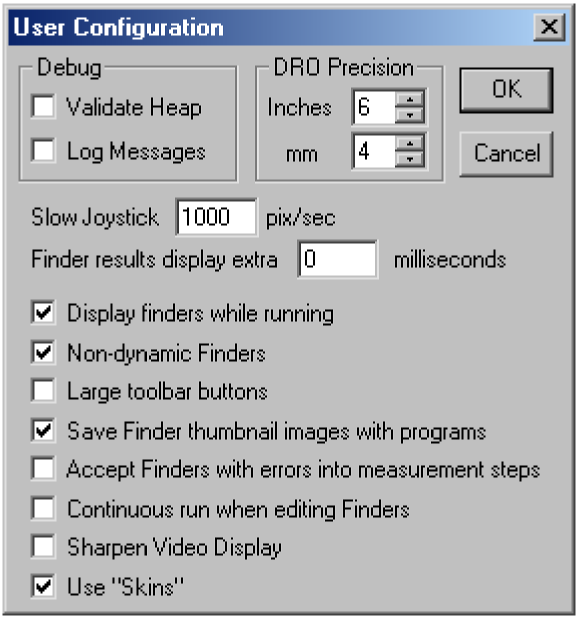


In the *System Configuration* dialog box, verify the lighting configuration in the following controls:

- **Lamp P/S** – this list selection determines what channels control what lights and, therefore, what items appear in the Ring, Coaxial, and Back Light lists. "6x12 LPS1" should be selected if the system is a V612. This selection is set at the factory and is normally disabled so it cannot be changed.
- **Ring Light, Coaxial Light, Back Light** – these list selections determine the lights that can be controlled using the *Stage and Lights* dialog box. They are set at the factory and are normally disabled so they cannot be changed. The *Stage and Lights* dialog box will have the Ring Light controls hidden when Ring Light is set to <none>.

User Configuration Dialog Box

The *User Configuration* dialog box saves individual user preferences. If the operating system is configured for multiple users, each user may save different preferences.



- **Finder results display extra** allows the user to specify a length of time for VMS to delay after running each finder and displaying the finder results. This can be used as a debugging tool to see finder problems that are not apparent when running finders at full speed.
- **Display finders while running** controls the nominal state of Display Finders/No Display Mode (Rapid Inspection Mode.)
- **Non-dynamic Finders** creates Finder parameters with more predictable edge finding characteristics. It should always be used except where there is a need to duplicate the characteristics of the dynamic finder parameters used by older VMS programs.
- **Large toolbar buttons** allows the user to choose between two button sizes for the main toolbar.

- **Save Finder thumbnail images with programs** controls whether or not to save Finder thumbnail images with measurement programs.

  VMS 7.0 adds a more informative, more intuitive way to see and access the Finders in a step. At the bottom of the *Measurement* dialog box, there is a list of pictures representing the Finders in the step, and each one is shown as a “thumbnail” image taken from the camera, with a small color graphic drawn over it, thus resembling what you would see in the *Video* window. So you can see what types of Finders they are, the order they are in, and what each one is looking at. You can right-click on these images to edit or delete the associated finder.

  The images are very handy, but they also take up space in a saved measurement program on disk (.voy files). When the thumbnail images are saved with a Measurement Program, they will increase the size of the .voy file by quite a bit, and this could present new problems if you want to put programs on floppy disks. **Save Finder thumbnail images with programs** controls whether or not to save Finder thumbnail images with programs. If it is off, the images will not be saved in the .voy file. When such a file is later recalled and a Measurement step is edited, you will still get icons with color graphics but the images from the camera will be gone, with just a solid gray image instead. This does not affect how the program runs and measures. Once again: the option affects how programs are **saved to disk**. It does not affect how new Finders are created and displayed.

- **Accept Finders with errors into measurement steps** - Normally, the green check button in the *Video* window is only enabled after successfully test-running a Finder. **Accept Finders with errors into measurement steps** will allow you to add the current Finder to a step even if the test-run produced an error. This can be handy in some circumstances.

- **Continuous Run when Editing Finders** - Before now, you would test-run a Finder only when you specifically commanded it (by double-clicking in the *Video* window or clicking the Run button in a *Finder Editing* dialog box). When **Continuous Run when Editing Finders** is on, the Finder in the *Video* window is run repeatedly. This way, changes to lighting, position, or Finder Parameters are immediately reflected in new results drawn in the window. This makes it easier to experiment with Finder settings. Obviously, the continuous re-running is not used with auto focus or laser Finders because of the motion and time involved.

- **Sharpen Video Display** makes edges appear sharper in the Video image. This is only a “cosmetic” enhancement but it can make it easier to see what you are working with. It does not affect how the system analyzes images and finds edges. (The advanced Finder parameters already perform such processing, and more.) This option is only available on systems that use the MuTech frame grabber.

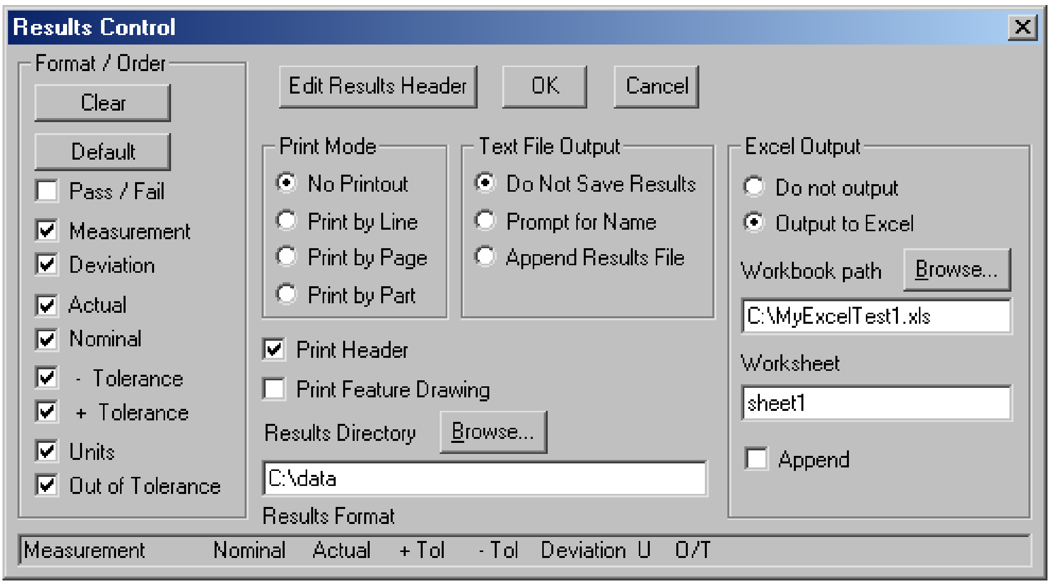
- **Use "Skins" -** What modern program would be complete without cool looking "skins"? **Use "Skins"** will enable a custom look for all measurement- and finder-related dialog boxes. However, a set of BMP files must be installed in the same directory as VMS, otherwise the dialog boxes will be the ordinary solid color. One set is installed by default when VMS is installed, but there are several other sets available on the install CD under \Program\Skins. The pictures in the Measurements and Finders topics show examples of dialog boxes customized using "skins." Note that you might want to customize your Windows scheme colors to match the skin set you choose.

> In Windows® XP, if "Windows and Buttons" is set to "Windows XP Style", you can select between three preset XP styles (maybe more can be found on the Web) but you cannot change the colors of individual elements. To do that you must set "Windows Classic Style", but then you don't get the "jolly, candy-like" look of XP.

Result Output Options

To select options for displaying and outputting results:

1. Select **Setup/Options/Results Control** from the menu. The *Results Control* dialog box is displayed



2. Select a Print Mode.

Option	Description
No Printout (default option)	VMS will not send results to the printer. Results will be displayed on screen in the *Results* window and possibly in a file, depending on your selection in File Output (see next step).
Print by Line	Each time VMS generates one line of results, it sends the results to the printer. This option only works with dot-matrix printers.
Print by Page	Each time VMS generates one page of results, it sends the results to the printer.
Print by Part	Each time VMS generates the results for a part, it sends the results to the printer.

3. Select a File Output. Each time you run a measurement program, VMS stores the measurement results in a file named VRESULTS.TXT. The following options give you greater control over the results:

Option	Description
Do Not Save Result (default option)	VMS will not save a copy of the VRESULTS.TXT file in a user specified file on your computer.
Prompt for Name	When you run a measurement program, the *Results Header* dialog box appears and prompts you to name the results file.
Append Results File	VMS will append the results to the existing data in the specified output file. If you select this option, the *Results Header* dialog box opens so you can enter a header that will separate appended data from data that already exists in the specified output file.

4. In the Format/Order options area, check the options you want VMS to include in the results. The order in which you select results options is the order in which they will print. For example, if you check Pass/Fail followed by Nominal followed by Actual, the results will be printed in this order as well. You can view the order in the **Results Format** field.

5. Select the **Clear** button to uncheck all Format options and clear the **Results Format** field. Select the **Default** button to check all Format options except for **Out of Tolerance**.

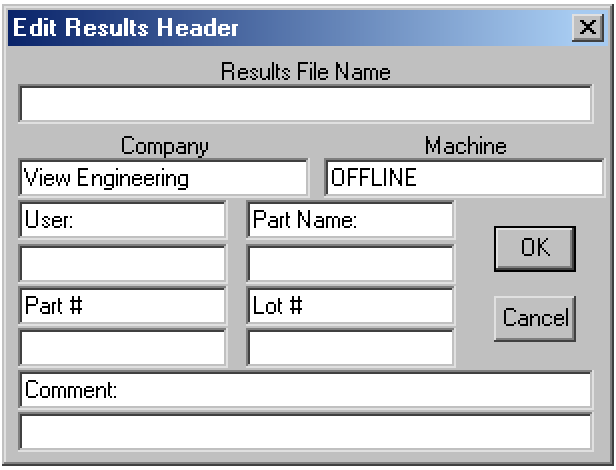
The headings listed in the **Results Format** field at the bottom of the *Results Control* dialog box mirror your selections in the Format/Order options.

6. Check the **Print Feature Drawing** box if you want VMS to print a drawing of the part (default is off). Make sure that **Print Header** is checked if the header is to be displayed when the program is run. If this box is not checked, the header is not displayed.

   To specify a directory in which to save VMS results files, enter a full pathname in the Results Directory box, e.g., C:\Data\PN12345. The directory you specify must already exist on your hard drive. For example, if the director "C:\Data" already existed, but "C:\Data\PN12345" did not, you must first create the PN12345 directory from Windows Explorer (see your Windows documentation for more information on creating directories).

7. To activate automatic data export to Microsoft Excel®, select the radio button **Output to Excel**. It is also necessary to specify a **Workbook path** and **Worksheet** as shown by the example Excel setup in the figure on the previous page. See Section *Result Output to Excel*.

8. Click on the **Edit Results Header** button if you want to customize the *Edit Results Header* dialog box or information printed in the header of the results files.



9. After you have made all your selections, click **OK** to return to the *Results Control* dialog box.

10. Click **OK** in the *Results Control* dialog box to activate the selections you made.

Saving the program after making the desired selections can save Results Control settings in the current measurement program. This allows different programs to use different settings.

Set Z Travel Limit

QVI systems include a safety feature that allows you to set a Z travel limit so that the lens never travels below a certain point above the stage or fixture. This ensures that the lens will not make contact with parts you are measuring. To manually set a lower Z travel limit:

1. Select **Setup/Z Limit** from the VMS menu to display the *Set Lower Z Travel Limit* dialog box.
2. Rotate the joystick so that the lens is at the desired lower travel limit. Make sure that the lens is sufficiently above the top of your part -- at least 1/8 inch. (For microscope optics, you will need to be closer than 1/8 inch.)
3. Click **OK** in the dialog box to reset lower limit.

Section 4

Measurement Programs

Creating a Measurement Program

When you write a measurement (part) program, you are teaching VMS a series of steps necessary to inspect a part. There are many kinds of steps, but the most important ones are Feature measurement steps. Features are part attributes, such as straight and curved edges, pins and holes, or light and dark spots on the part surface. To measure Features, VMS uses tools called Finders. When the system uses the camera to look at one particular Feature of a part, Finders provide raw XYZ coordinates for that Feature. You can have one or several Finders in a Feature measurement step. In dialog boxes for Feature measurements, you can input nominal values for position, size, etc. You can select which properties to inspect and enter tolerances for each inspected property so VMS can make Pass/Fail determinations.

Follow the steps below to create a basic measurement program.

1. Place the part on the stage and turn on the system if it isn't already on.
2. Zero the stage.
3. Place the appropriate lens on the system and calibrate it if needed.
4. Set an appropriate place on the part as the Home position (see *Set Home and Go Home*).
5. Select **File/New** from the menu to start a new program.
6. Set the Z Travel Limit (see *Set Z Travel Limit*).
7. Select the System Alignment and/or define the Part Alignment.
8. Measure the desired Features of the part using Finders. Input nominal values and tolerances for the Features. Test and run diagnostics on Finders and troubleshoot measurement steps as needed.

9. After adding and testing the Finders, click the **Run** button in the *Measurement* dialog box to have VMS re-run the Finders, determine the actual measured values and make pass/fail determinations. To save the step as part of the program, click **OK**.

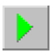
10. Once the measurement program is complete, you can save it on disk as a file with any name you choose (it should end in .VOY). Later, you can load (open) the program in VMS. When the program is loaded, click on the **Run** button to have VMS perform all of the inspection steps automatically, then output the results.

Running a Program

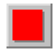
Running a Measurement Program

If no program is currently open, you must either Create a Program or open an existing program (see *File Group*). When you open a program created on another system, it may have Lighting Compatibility Issues that need to be resolved.

To start a measurement program from the beginning:

1. Select  (**Restart**) from the *Program Control Group* of the VMS toolbar.
2. Select  (**Run**) from the VMS toolbar.
3. Select the other Program Control buttons as your needs dictate.

Stopping a Measurement Program

To stop a running measurement program, click the **Stop** button  on the toolbar.

If you stop a program to edit, your editing may change alignment settings, which could cause inaccurate measurement results. Once you are finished editing:

1. Click the **Restart** button on the toolbar. When you select **Restart**, you ensure that the program resets alignments. If you have paused momentarily without making changes in any VMS window, or if the program has stopped as part of its sequence of steps to allow you to take a specified action, you do not need to select Restart.
2. Select **Run**.
3. Select the other Program Control buttons as your needs dictate.

Running Programs with Manually Positioned Finders

If your program includes any manually positioned finders (e.g., Crosshair Finder), the program will stop at the appropriate point to allow you to manually position the Finder. When the program stops, the title bar of the *Video* window displays:

Video: Manually Placed Finder n:n

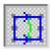
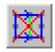
where the first n represents the step number and the second n represents the Finder number within the step. To proceed:

1. Reposition the stage and/or Finder manually.
2. Double-click in the *Video* window to run the Finder.
3. If you aren't satisfied with the edge VMS finds, repeat Steps 1 and 2 before proceeding to the next step.
4. Click on the green **Check Mark** on the *Video* window toolbar to continue.

Rapid Inspection Mode

VMS has a user selectable mode, called Rapid Inspection Mode, that runs measurement programs faster in certain circumstances than it used to in VMS software versions prior to 3.0. It is useful in a production situation when you are running a program that has been tested and is working well. It is not very desirable when you are first writing a program or working out the kinks. It can be used with previously written programs.

Rapid Inspection Mode is activated when the **Display Finders** button in the *Video* window toolbar is in the No Display mode. The default mode (Display Finders or No Display) may be selected in the *User Options* dialog box.

Display Finders Mode	No Display Mode (Rapid Inspection Mode)
	

With Rapid Inspection Mode active, an optimization is applied whenever there are multiple Finders within a step: VMS moves the stage to the next position while it analyzes the previous image. It does not do this between the last Finder of one step and the first Finder of another step, so steps with one or no Finders will not go faster.

Autofocus also takes advantage of Rapid Inspection Mode by not moving the Z-axis to the computed focus position. This is useful only if point data is being gathered to measure a point, plane, etc. If the Autofocus is being done prior to running an edge finder on the focused image, the Z-axis will have to move to that position to run the edge finder anyway.

If a Finder gets an error in Rapid Inspection Mode, the remaining Finders in that step are run, and the error message is shown afterward (and the program will stop, as usual). If you want to determine which Finder got the error and why, turn Rapid Inspection Mode off (set the Display Finder button back to Display Finders mode) and re-run the step. Then you can watch the individual Finder results and the step will abort immediately after the offending Finder is run.

Lighting Compatibility Issues

If you have only one system or if you have several that all have the same lighting configuration, the issues discussed here will not be of any concern. Otherwise, you should read this section carefully if you intend to write programs on a system with one lighting configuration then run them on a system with a different lighting configuration. In general, it can be done without any difficulty.

To handle the growing number lighting configurations, VMS keeps a record in each saved measurement program, of the light configuration that was in place when that measurement program was created. (This is known as the Lighting Function Table and Lamp Type Table, or **LFT/LTT**.) When a measurement program is loaded, the configuration it lists is compared to the machine's current configuration for compatibility. This is intended to accomplish two goals:

- To ensure that VMS will not blindly attempt to use a programmed light type if that light type is not present;
- To allow a program to execute correctly if a needed light is on a physical channel other than the one it was on when programmed.

There are a few important points you need to be aware of:

- You must **always be careful** to change the configuration when you switch between the Red/Green LED ring light and the Blue/White LED ring light.
- There is only one lighting configuration saved with a measurement program, so you cannot have part of a program using one lighting configuration and part of the same program using a different lighting configuration.
- To enforce this, the lighting configuration of a machine cannot be changed unless there is an empty measurement program. So if you go to change the ring light type and it is grayed out, you need to select **File/New** first.

If you load a measurement program and its LFT/LTT calls for types of lighting functions that are not present on that machine, you will see the following warning: **"This program was created using lights that are not present on this system. It may execute incorrectly."** This does not necessarily mean there will be a problem. For example, if a program was written on a machine that had a PRL, but the program did not actually use the PRL, and you loaded it on a machine that did not have a PRL, then you would get the warning but it would run properly. But if it did use the PRL, you would get a different error message (*Illegal Light Value*, error #85) when that particular Finder ran and tried to use the non-present lighting function.

- Note the implied advice there: If you write programs on a machine with a PRL but intend to run them on a machine without one, you **must** leave both its position and light levels set to 0.

- Any lighting functions referred to in a loaded measurement program, that are not actually present on the system, will be removed from the measurement program's LFT/LTT when it is loaded. If the program is then saved, the saved LFT/LTT will no longer include those lighting functions, only those functions the two machines had in common.
- If a program is loaded on a machine that has extra lighting functions (not listed in the loaded program's LFT/LTT), they will **not** be added to the LFT/LTT and they will **not** be available in the *Stage and Lights* dialog box while that program is loaded. This is done in anticipation that the program is likely to be taken back to run on the original system that did not have those functions.
- If a loaded program refers to a particular **lamp type** that is not present on the machine, you will get the warning message listed in note 4 above. But the lamp types will not be removed from the LFT/LTT like missing functions are (note 6); instead, they will be changed to indicate the lamp types that are present on the machine. This allows you to load and run a program that was accidentally written with the wrong LED colors configured.

Measurement programs written before version 5.2 do not have LFT/LTT information. If such a program is loaded, the LFT/LTT is created for it assuming that the program refers to all the lights previously available: Coaxial light, Back light, Single Ring light, and PRL. Since no system can have all of those, some of those functions must be missing on the actual system, but in the case of an older program the warning is not given. But those functions will be removed from the program's newly created LFT/LTT, so if it is now saved it will conform to other new programs. If the program tries to use the missing lighting functions in a Finder, then an error message will be given.

Result Output

Displaying and Printing Results

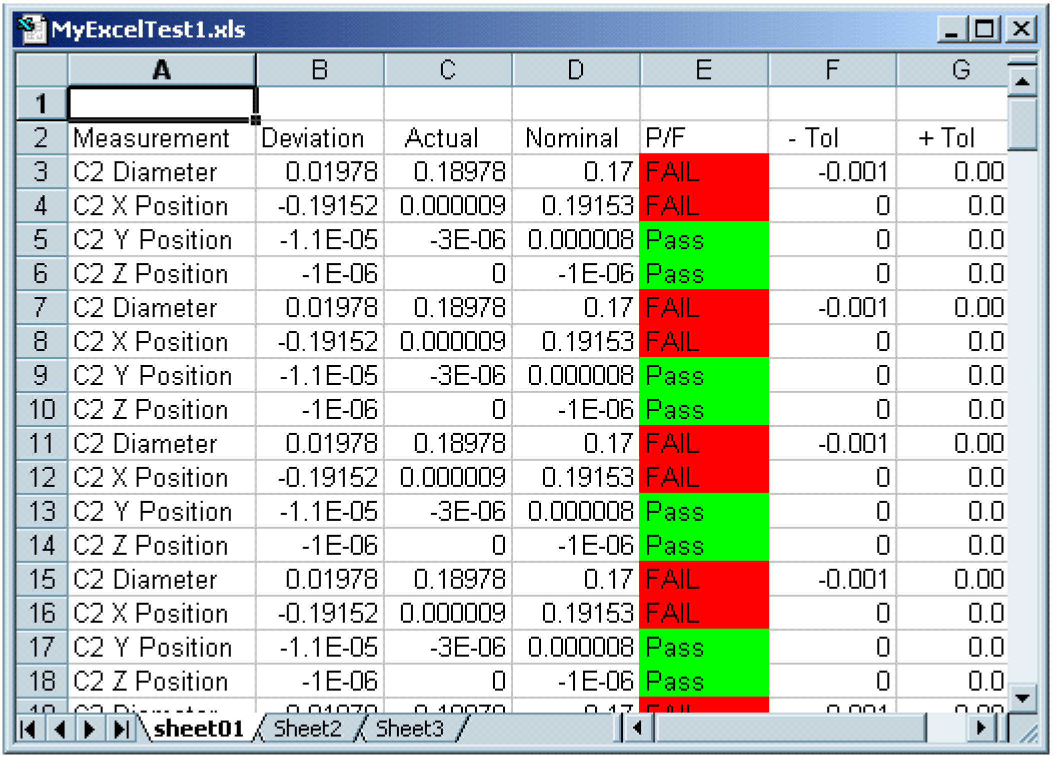
VMS gives you control over how and where your measurement results are output. You can output results:

- on screen
- to a printer
- to a file
- to a Microsoft Excel® spreadsheet

You make your choices about outputting results prior to programming the steps of a measurement program. You record your choices in the *Results Control* dialog box. These choices are saved with the measurement program.

Result Output to Excel

To activate automatic data export to Microsoft Excel®, select the **Output to Excel** radio button in the *Result Output Options*. It is also necessary to specify a **Workbook path** and **Worksheet**; see the example Excel® setup in *Result Output Options*. The user may select the **Append** checkbox for appending results from multiple runs into the same worksheet. Otherwise, it will start at the first row of the worksheet every time you run the program. Only those results checked in "Format/Order" are exported. A sample Excel® worksheet output is shown below.



	A	B	C	D	E	F	G
1							
2	Measurement	Deviation	Actual	Nominal	P/F	- Tol	+ Tol
3	C2 Diameter	0.01978	0.18978	0.17	FAIL	-0.001	0.00
4	C2 X Position	-0.19152	0.000009	0.19153	FAIL	0	0.0
5	C2 Y Position	-1.1E-05	-3E-06	0.000008	Pass	0	0.0
6	C2 Z Position	-1E-06	0	-1E-06	Pass	0	0.0
7	C2 Diameter	0.01978	0.18978	0.17	FAIL	-0.001	0.00
8	C2 X Position	-0.19152	0.000009	0.19153	FAIL	0	0.0
9	C2 Y Position	-1.1E-05	-3E-06	0.000008	Pass	0	0.0
10	C2 Z Position	-1E-06	0	-1E-06	Pass	0	0.0
11	C2 Diameter	0.01978	0.18978	0.17	FAIL	-0.001	0.00
12	C2 X Position	-0.19152	0.000009	0.19153	FAIL	0	0.0
13	C2 Y Position	-1.1E-05	-3E-06	0.000008	Pass	0	0.0
14	C2 Z Position	-1E-06	0	-1E-06	Pass	0	0.0
15	C2 Diameter	0.01978	0.18978	0.17	FAIL	-0.001	0.00
16	C2 X Position	-0.19152	0.000009	0.19153	FAIL	0	0.0
17	C2 Y Position	-1.1E-05	-3E-06	0.000008	Pass	0	0.0
18	C2 Z Position	-1E-06	0	-1E-06	Pass	0	0.0

As you can see from the example spreadsheet above, the columns of the worksheet correspond to the columns in the Results output, e.g., a column for Measurement, a column for Actual, etc. If “Pass/Fail” is checked, the corresponding output cell in Excel sheet will show green for “Pass” and red for “Fail”.

The Excel file will be closed when the last line of the VMS program is executed, or when the rewind button is pressed.

Program Security

About Program Security

This feature only works under Windows® 2000 or XP. It utilizes Windows security mechanisms for very strong security. There should be:

- At least one Windows account that is in the **Administrators Group** for users who are authorized to make changes in measurement programs, and
- At least one Windows account that is not in the Administrators group for users (let’s call them *Operators*) whose abilities are to be restricted. Operator accounts are not automatically restricted - they can be set to restricted or not - but only an Administrator can change the restrictions on an Operator account.

Even Operators need to be Power Users in order to use VMS, because it changes the Registry. Also, be aware that only Administrators will be able to properly run versions of VMS prior to 7.0.

Protecting Measurement Programs

When a measurement program is saved by a member of the Administrators group, the *Save* dialog box includes an extra check box: **Protected**. If this box is checked before the program is saved, then the program will contain a flag indicating that it is protected. When such a program is loaded under an Operator account, the menu level is forced to Restricted. This allows some Operator accounts to be set to Standard or Advanced menu level, so they can modify non-Protected programs but still cannot modify Protected programs.

So far we have discussed how programs are protected from being modified by VMS. But there are other ways that a Program or Library file (.voy and .vml files) could be compromised, outside of VMS. For example, a “doctored” version of a Measurement Program could be created on an unsecured machine, and then this version could be put on a floppy disk and copied over an existing program on a supposedly “secure” machine, simply by using Windows Explorer.

To prevent this, program files can be protected through standard Windows security methods. For example, a program file’s access rights could be set so that only members of the Administrators Group have write access, and all other accounts (meaning Operator accounts) have read-only access. Thus, an Operator could not replace or delete the file in Windows Explorer. In other words, standard Windows security can and should be used to protect program files outside of VMS.

When an **Administrator Override** is in effect, the VMS program (but no other programs) is operating under the Administrator’s credentials, and has write access to such files even though the main Windows logon is under the Operator’s account.

Restricted Menu Level

There is now a third VMS “menu level” called Restricted. When operating at this level, the functionality of VMS is restricted as follows:

- New steps cannot be created.
- Existing steps may not be edited or deleted.
- Programs may not be saved.

But existing programs may be recalled and run.

When VMS is run under an Operator account, the menu level cannot be changed. Each Operator is forced to use a given menu level. So, if you set the menu level of an Operator account to Restricted, they will not be able to accidentally (or purposely) change tolerances or anything else while in VMS.

Note that you can set an Operator (someone not in the Administrator group) to have a menu level of Standard or Advanced. Such an Operator can change measurement programs, *except* programs that are “protected” (see below).

When VMS is run under an Administrator account, menu level may be changed at will.

Changing an Operator's Menu Level

In order to change the menu level of an Operator:

1. The Operator logs in under his own Windows account and starts VMS.
2. The Administrator selects the desired new menu level.
3. A "logon" dialog appears; the Administrator supplies a user name and password.
4. The Administrator exits VMS. **DO NOT SKIP THIS STEP.**

The Operator's new menu level is set until another such change.

Administrator Override

If an Operator who has been set at Restricted level needs to change a measurement program, he or she does not need to log out or even exit VMS, but they do need to call upon an Administrator. Simply follow steps 2 and 3 above: attempt to change the menu level then supply an Administrator account name and password. At this point, VMS is temporarily operating under the Administrator's credentials. The program can be modified, and the modified program can be saved if desired.

After the program has been changed, the Administrator should set the menu level back to Restricted, which also "logs off " (within VMS) the Administrator. **The Administrator should not leave VMS "logged in" to the Administrator's account; that would compromise the system's security.** The modified measurement program remains loaded. The Operator remains logged on to Windows. VMS is again running under the Operator's credentials.

Section 5

Alignment and Coordinate Systems

Introduction to Alignment and Coordinate Systems

VMS expresses measurements based on three coordinates — X, Y, and Z. The coordinates are only useful, however, if you know the coordinate system the measurements are based on. VMS allows you to create several different coordinate systems for various purposes in your measurement programs. They can have different origins and can be at different angular orientations. The proper use of these coordinate systems allows your programs to be more flexible and powerful. Improper use can limit your capabilities or cause you to do unnecessary editing. Understanding coordinate systems requires some explanation and thought.

Alignment is the process of discovering where the coordinate system of a part is located in relation to the stage or in relation to the part. Alignment is one of the most important and one of the most complicated concepts you'll encounter when using the system. Alignment theory and procedures are outlined in this chapter.

Many problems are related to a misunderstanding or misuse of alignment and coordinate systems. Please read this chapter thoroughly and make sure you understand each concept before proceeding.

Before discussing Alignment, it is important that you understand the definitions of Machine Coordinate System (MCS), Datum Reference Frame (DRF), Finder Coordinate System (FCS) and Part Coordinate System (PCS).

MCS - Machine Coordinate System

The Machine Coordinate System (MCS) is built into the system. The MCS consists of the stage coordinates:

Coordinate	Relation to Origin
X	To the right from the origin in positive values
Y	To the rear from the origin in positive values
Z	Up from the origin on the stage surface in positive values

Once you establish the values of the MCS, you can control the movement of the stage and measure the positions of edges or other Features of a part, in MCS, at any time. To establish the values of the MCS, you must "zero the stage" each time VMS is started. This allows the stage to seek out its limits of travel along its X, Y, and Z-axes, and then come to rest at its pre-programmed "home" position:

1. Select **Setup/Stage and Lights** from the menu if the *Stage and Lights* dialog box is not already open on the screen.
2. Click the **Zero Stage** button.

DRF - Datum Reference Frame

A Datum Reference Frame is a 3-dimensional coordinate system whose origin and angular orientation are based on the actual part mounted on the stage. Measurements of part geometry are generally relative to a DRF. DRFs are defined on a part drawing as being based on certain key Features of the part called Datums. VMS finds out the origin and angular orientation of a DRF by inspecting these Features. It then computes, evaluates, and reports measurements in XYZ coordinates relative to the DRF. Often, the same PCS is used both as a DRF and an FCS.

FCS - Finder Coordinate System

A Finder Coordinate System (FCS) is a 3-dimensional coordinate system whose origin and angular orientation are based on the actual part mounted on the stage. It is not used for computing, evaluating, or reporting measurements; rather, it defines the coordinate system used to run Finders. When you show the system where to run a Finder, it remembers the Finder's position relative to the FCS. When the system later runs the Finder automatically, it positions the Finder based on the FCS. Because the FCS is based on where the part really is, it allows the system to position the Finder in the right place on the part. If the part moves, the FCS will move with it, and the Finder will continue to run at the right place. Often, the same PCS is used both as a DRF and an FCS.

PCS - Part Coordinate System

Part Coordinate System (PCS) refers to either a DRF or FCS based on the part. It is simply a generic name. DRFs and FCSs are both types of PCSs. Internally, VMS defines the position and orientation of either type of coordinate system in a special variable called a PCS variable; these are, by default, named PCS1, PCS2, etc.

System Alignment

A System Alignment can be either MCS (also called "Stage") or a Fixture Alignment. Select what to use as System Alignment from the **Align/System Alignment** menu. It gives you a list to choose from. The first item in the list is "Stage", followed by any Fixture Alignments you have created. When you load a measurement program, it automatically selects the System Alignment that was chosen when it was saved.

Fixture Alignment

A *Fixture Alignment* is simply an initial coordinate system to use instead of MCS as the global coordinate system. This coordinate system is based on a particular fixture mounted on a particular machine. It is useful if you have several parts and programs that use a single fixture. If you move the fixture or take the parts and programs to a different machine, you need only recreate the fixture alignment and the programs that use it will run as before.

Usage of Coordinate Systems

VMS uses coordinate systems to express the positions of Features, the XYZ and angle relationships between Features, and to place Finders. Whenever a Feature is measured, VMS saves its coordinates with respect to a defined DRF. You can test these coordinates and use them directly. Sometimes you measure the XYZ distance or angle between two Features. Again, the measurement (distance or angle) is expressed with respect to the DRF. However, the Features involved may be with respect to another DRF. In this case, VMS automatically translates coordinates from one DRF to another for you, so you don’t have to worry about it.

Defining a PCS

A PCS, whether it is used as a DRF or as an FCS, is defined based on Features on the part. According to standard practices, there are three attributes derived from three Datum features: tilt, orientation, and origin. The Features that define these are technically called the Primary, Secondary, and Tertiary datums, respectively.

First, you establish the tilt of the XY plane. This does not determine where the XY plane is, only the angle at which it is tilted. Another way to think of this is the direction the Z-axis points. It does not determine where the Z-axis is, or where the XY plane intersects it. Imagine holding a toy gyroscope. You can move it all around and rotate the cage about its axis, but its spin axis keeps pointing the same way.

Second, you establish the angular orientation (rotation) of the X and Y-axes. This determines which way the X or Y-axis points. Since tilt already determined the direction of the Z-axis, establishing either X or Y will also establish the other one. This does not establish where the origin is, only direction. Using the gyroscope analogy, this is like rotating the cage about its axis to some orientation and then keeping it there. You can still move it up, down, left, right, forward, or backward, since the origin is not set.

Third, you establish the origin. The origin is a single point that serves as the anchor of the coordinate system.

DRF versus FCS

VMS places no limits on what Features may be used to define the DRF (within reason) or on which DRF is used to compute, evaluate, or report measurements. You can define a DRF with a separate step anywhere in the program, and you can choose any DRF in the dialog box used for measuring any Feature. However, if you use DRFs inappropriately, your measurements may not come out as expected.

On the other hand, an FCS is necessary to the process of actually inspecting the part with the camera or other sensors (e.g., laser). FCSs control where the stage will move. So, if they are used inappropriately, the camera may look in the wrong place. Even worse, the stage could move in such a way that the lens hits the part or fixture. Because of this, VMS uses a system of "Alignment Blocks" (see *Align Blocks*) that ensures such mix-ups are not likely to happen.

You MUST define and use DRFs according to the part's drawing or specifications. On the other hand, an FCS can be based on any Features that will help the system find the other things to be measured. Therefore, it is very common to simply use the DRF as the FCS. In other words, you define a coordinate system to be used as the DRF the way the design requires, and then you happen to also use it as the FCS. (See also *Default DRF*).

Default DRF

You can create a step anywhere that uses any Features, to define a PCS. (Select **Measure/Define PCS** from the menu.) This PCS may then be selected, in any measurement step, as the DRF to use for that step. There is a drop-down list in the *Options* dialog box for any Measurement step to select which PCS to use as the DRF in that step.

Quite often you would use the same coordinate system as both the DRF and the FCS. The decision of which Datums to base the DRF on is usually dictated by the part's drawing or other design criteria. So you define the FCS (that is, create an Align Block and Align Define Section steps) based on those Datums, and simply select the resulting PCS as the DRF for each step.

To make that easier, VMS has a concept called the Default DRF. When you create an Align Block, the associated PCS (which is also the FCS) becomes the Default DRF within the **Align Use** section, and every Measurement step you create starts out with "Default" as the DRF setting.

- An Align Block does not *have* to make its PCS the Default DRF. In the Align Define step, you can turn off the Use as Default DRF check box.

- Within a Part Step & Repeat loop, the Step & Repeat variable (e.g., SR1) is the default DRF.

Alignment in Practice

Part Alignment

Sometimes System Alignment is not accurate enough for measurement purposes (for example, the part may be at an angle on the stage, or parts may not fit the same way into a fixture each time). In those cases, use Part Alignment, which you build into your measurement program. Normally, you build Part Alignment as the first steps in your measurement program. To do this, you must:

1. Create an Align Block.
2. Add measurements and constructions for the alignment Features to the Align Define section of the Align Block. If the variation in part position can exceed the size of the field-of-view, it may be useful to define an origin Point Feature using a Crosshair Finder and locate the remaining alignment features relative to it.

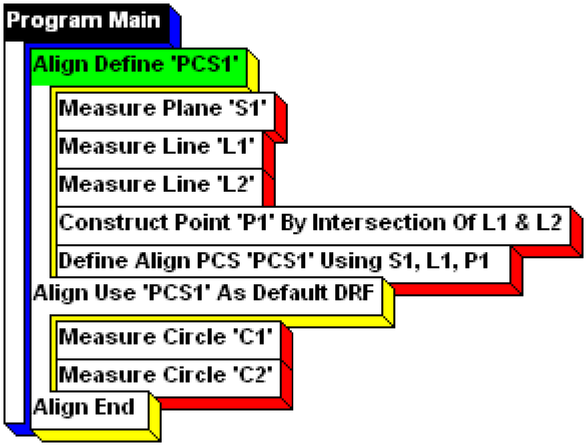
> You must include everything required for defining the PCS in the Align Define section of the block. Steps inside the block should not make reference to Features, relationships, or variables that are defined outside the Align Define section.

3. Add a Define Align PCS (Part Align) step at the end of the Align define section.

Alignment Structure

Align Blocks

In order to program a Finder, the FCS must be set up correctly first. If you were allowed to define an FCS in any section of the program, and then use it anywhere else, you could accidentally program a Finder when the FCS had not been properly set up. So VMS has a system for keeping track of which steps define an FCS and which steps use an FCS, so that it can prevent these problems. This system is called Part Alignment using Align Blocks. An Align Block uses three connected steps to define two groups of other program steps. The first group is called the **Align Define Section**; this is where steps that define a specific FCS are placed. The second group is the **Align Use Section**. All the steps in the **Align Use** section use the specific FCS for their Finders. Here's an example:



First look at the three steps with yellow sides: **Align Define 'PCS1'**, **Align Use 'PCS1' As Default DRF**, and **Align End**. Not only are they the same color but they are connected at the left side to form a big "E" shape. These constitute the **Align Block**.

Now look at the five steps nested between the **Align Define** step and the **Align Use** step. These are the **Align Define Section**.

Finally look at the two steps between the **Align Use** step and the **Align End** step. These are the **Align Use Section**.

The steps in the **Align Define** section measure S1, L1, and L2, construct P1, then construct a PCS from those Features. That PCS is named PCS1. In the steps that measure C1 and C2, the Finders are positioned with respect to PCS1. So PCS1 is being used as a Finder Coordinate System there.

🗎 The alignment in effect before and after the outermost Align Block (and within its **Align Define** section) is whatever was selected for System Alignment (either a Fixture Alignment or MCS).

Get Current Alignment

The Align Block structure is used to prevent us from programming a Finder when its FCS has not been properly set up. How could that happen, and how do Align Blocks prevent it?

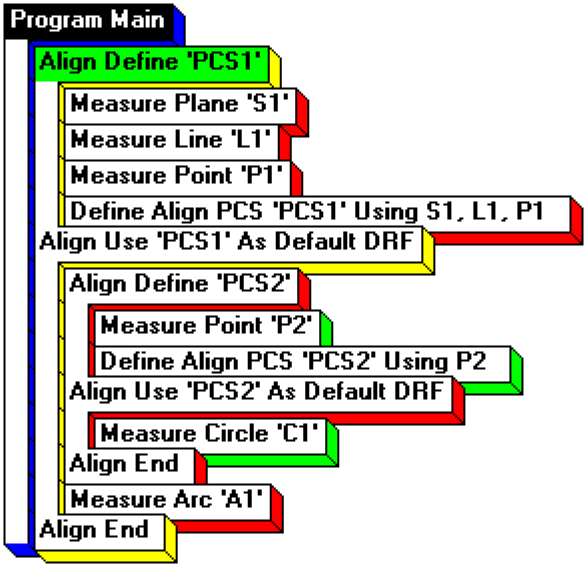
Suppose yesterday we put a part on the system and we wrote a program for it, the first example program above. Then we removed the part. Today, we put the part back on, recall the program we have so far, and we want to add another Feature measurement, say another Arc named A2, after the step for A1. But we don't notice the part has been put on a little crooked this time! All the FCSs match the part's original position, but they don't match its current position. The FCSs need to be re-established before we program any Finders that use it. Without Align Blocks, we could just go in and start programming Finders with the existing, non-matching FCS.

With Align Blocks, when we tell VMS we want to insert a step after the step for A1, VMS looks at the Align Blocks and says to itself, "Hmm. This new step's Finders will be with respect to PCS1. And PCS1 is defined by S1, L1, and P1." (VMS knows this because of the Align Blocks!) VMS then tells us that it wants to re-run just those steps before it lets us create our new step. We say, "OK". (We also get an option to skip that, if we know the part has not been moved.) We call this process of identifying and running the steps that establish the FCS "Getting Current Alignment."

The alternative to this would be to always run the *whole program* from the beginning up to the point where we want to edit. That approach, which was used in the past, often took such a long time that users would add extra IF statements, just for editing usage, to skip over all the steps that did not define the FCS. This way is smarter but it does introduce the Align Block concept.

Nested Align Blocks

Maybe C1 is manufactured in a way that its position may vary with respect to PCS1. We want to measure some other, easier-to-find, Feature as a guide to locating C1. So, we construct another FCS to find it. This requires its own Align Block. It is inside the **Align Use** section of PCS1:



The new FCS is named PCS2, P2 is used as the origin for it. (The tilt and orientation are inherited from PCS1). Note that the Finders used to measure P2 are positioned according to PCS1; we say that PCS1 is the "enclosing" alignment. But the Finders for C1 are positioned according to PCS2, because the step that measures C1 is in the PCS2 **Align Use** section. Also note that the step that measures A1 is *not* inside the PCS2 **Align Use** section, but it is inside the PCS1 **Align Use** section. When VMS crosses the **Align End** step of the PCS2 block (the red **Align End** step), it stops using PCS2 as the FCS and goes back to using PCS1.

Another example of Nested Alignments follows:

Programming Details:

- X, Y and Z boxes checked for Origin alignment to P1 (Step 3)
- Z1 is a point defined by using autofocus finder.
- Z box checked for Origin alignment to Z1.
- Xaxis button checked and X_AXIS line used for rotation alignment (Step 12)
- S1 is specified for tilt alignment (Step 18)
- FOC1 and FOC2 are autofocus points (Steps 20 and 21)

Program Listing:

```
 0 Program Main
 1   Align Define 'XY_ORIGIN'
 2     Measure Point 'P1'
 3     Define Align PCS 'XY_ORIGIN' Using P1
 4   Align Use 'XY_ORIGIN' As Default DRF
 5     Align Define 'Z_ORIGIN'
 6       Measure Point 'Z1'
 7       Define Align PCS 'Z_ORIGIN' Using Z1
 8     Align Use 'Z_ORIGIN' As Default DRF
 9       Align Define 'ROTATION'
10         Measure Point 'P2'
11         Construct Best Fit Line 'X_AXIS' Using P1, P2
12         Define Align PCS 'ROTATION' Using X_AXIS
13       Align Use 'ROTATION' As Default DRF
14         Align Define 'TILT'
15           Measure Point 'Z2'
16           Measure Point 'Z3'
```

17 Construct Best Fit Plane 'S1' Using Z1, Z2, Z3

18 Define Align PCS 'TILT' Using S1

19 Align Use 'TILT' As Default DRF

20 Measure Point 'FOC1'

21 Measure Point 'FOC2'

22 Measure Distance 'STEP_HT' From FOC1 To FOC2

23 Align End

24 Align End

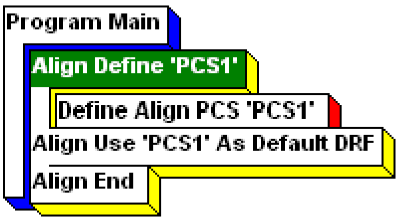
25 Align End

26 Align End

Creating a Part Alignment

Creating an Align Block

To add an Align Block for Part Alignment, click the **Create Align Block Steps** button in the Programming Group of the VMS Toolbar. A block of four program steps -- Align Define, Define Align PCS, Align Use, and Align End -- appears in the *Measurement Steps* window:



Alternately, you can create the Align Block from the **Align** menu. This method is slower, but offers more options.

1. Select **Align/Define Align Block** from the menu. The *Define Align Block* dialog box is displayed:
2. If you wish, change the name suggested in the dialog box to something more descriptive.
3. Leave the **Use As Default DRF** checkbox checked (see *Default DRF*). Click **OK**.

The Align Define step is automatically selected, indicated by the dark green face with white printing on the step. If the Align Define step becomes deselected because you clicked on another block in the *Measurement Steps* window, click on the Align Define step to re-select it. Additional steps are inserted after the selected step. The block structure establishes which PCS is in effect at any given part of the program.

> You must include everything required for defining the PCS in the Align Define section of the block. Steps inside the block should not make reference to Features, relationships, or variables that are defined outside the Align Define section.

Determining the Part Alignment Method

Select **Align/Part Align** from the menu to display the Part Align submenu. There are four options. Choose the option that best corresponds to how the PCS is defined by your CAD file or drawing:

Part Align Options	Description
Use System Align	Sets the System Alignment as the Part Alignment. If this option is selected, the Use System Align method will be the only step in the Align Define section of the measurement program.

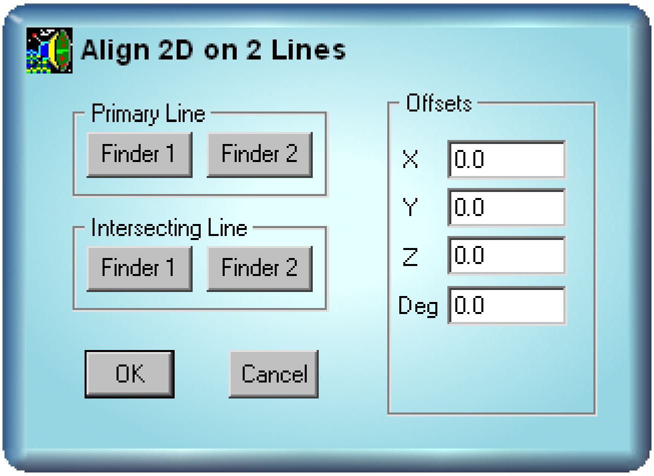
Part Align Options	Description
Two Line (2D)	Generally used to set alignment based on two straight sides of a rectangular part. Two Finders are used to define a "Primary Line," (the X axis of the PCS), typically the bottom or top edge of the part. Two other Finders are used to define an "Intersecting Line," typically a side edge of the part. The intersection of the Intersecting Line with the Primary Line is the PCS origin. If this option is selected, the Two Line (2D) method will be the only step in the Align Define section of the measurement program.
Two Circle (2D)	Sets the alignment using three Finders to define a "Primary Circle" (the PCS origin). Three other Finders are used to define an "Orient Circle." The line between the circles' centers is the X-axis. If this option is selected, the Two Circle (2D) method will be the only step in the Align Define section of the measurement program.
Plane-Line-Point (3D)	Sets the alignment using a full 3D orientation derived from any constructions of plane, line, and point Features. The plane determines the X/Y plane of the PCS; the line determines the X-axis; and the point (or a Feature that reduces to a point) determines the origin. The Plane-Line-Point (3D) method uses Features that are measured in previous steps in the Align Define section, so if this option is selected, the Align Define section will contain several Feature-measuring steps followed by one Part Alignment step. This option is preferred in most cases, however, before selecting this option you must define within the Align Define section, any plane, line(s) or a point(s) to be used.

Features to be used in a Part Align step may be selected from the *Features* window. However, a plane cannot be selected there because a plane, being infinite, cannot be drawn in the *Features* window.

Two Line (2D) Part Alignment

Although Plane-Line-Point is the preferred alignment method in most circumstances, Two Line alignment is a bit simpler and may be appropriate for your application.

1. Position the part on the stage.
2. Create an Align Block.
3. If you created the Align Block with the toolbar button, delete the “Define Align PCS” step from the block and reposition the selected step to the “Align Define” step.
4. Select **Align/Part Align/Two Line (2D)** from the menu. The *Align 2D on 2 Lines* dialog box that is displayed allows you to determine which Finder you are going to work on first. The default Finder is Primary Line Finder 1(Point 1).



5. Click the **Primary Line Finder 1** button. The title of the *Video* window should be Video: Primary Line pt 1.
6. Use the joystick to move to the bottom edge of the part, near the right corner.
7. Select a Finder (Line Finder would work well here) from the *Video* window toolbar.
8. Place the Finder on the bottom edge of the part, near the right corner. Make sure you rotate the Finder so that its arrows are oriented from dark to light. Double-click on the Finder to perform a test run. A line of points is displayed. The **Check Mark** on the *Video* window toolbar should turn green after the test run.

9. Click on the green **Check Mark** to accept the Finder and store the information in the *Align 2D on 2 Lines* dialog box. The **Primary Line Finder 2** button will automatically be selected in the *Align 2D on 2 Lines* dialog box.
10. Using the joystick, move to the bottom edge of the part, near the left corner.
11. Repeat steps 7 through 8. The **Intersecting Line Finder 1** button is automatically selected in the dialog box. The tile of the *Video* window changes to Intersecting Line pt 1.

Using the joystick, move to the left edge of the part, near the bottom.

Repeat steps 6 through 8. The **Intersecting Line Finder 2** button is automatically selected in the *Align 2D on 2 Lines* dialog box.

Using the joystick, move to the upper left edge of the part.

Repeat steps 7 through 8. Click **OK** in the *Align 2D on 2 Lines* dialog box to complete the Two Lines (2D) alignment.

Two Circle (2D) Part Alignment

Two Circle (2D) Part Alignment is a simpler alignment technique than 3D Part Alignment.

1. Position the part on the stage.
2. Create an Align Block.
3. If you created the Align Block with the toolbar button, delete the "Define Align PCS" step from the block and reposition the selected step to the "Align Define" step.
4. Select **Align/Part Align/Two Circle (2D)** from the menu. The *Align 2D on 2 Circles* dialog box that is displayed allows you to determine which Finder you are going to work on first. The default Finder is Primary Circle Finder 1.
5. Select the Primary Circle Finder 1 from the dialog box. The title of the *Video* window should be Video: Primary Circle pt 1.
6. Use the joystick to move to the hole that is the origin Datum. (We are assuming it is a hole, although any other well-defined circular Feature would be fine, such as a pin or an etched target shape.)
7. Click on the *Video* window tool bar to select a Finder (Line or Arc Finder would work well here.) We will use separate Finders, not one field-of-view Circle Finder, in case it does not fit in the FOV.

8. Place the Finder on the edge of the hole. Make sure you rotate the Finder so that its arrows are oriented from dark to light. Double-click on the Finder to perform a test run. An arc of points is displayed. The Check Mark on the *Video* window toolbar should turn green after the test run.
9. Click on the green **Check Mark** to accept the Finder and store the information in the *Align 2D on 2 Circles* dialog box. The Primary Circle Finder 2 button will automatically be selected in the *Align 2D on 2 Circles* dialog box.
10. Using the joystick, move to another edge of the hole. You want to space the three Finders as equally as possible around the hole.
11. Repeat steps 7 and 8, two more times. Then the **Orient Circle Finder** 1 button is automatically selected in the dialog box. The title of the *Video* window changes to Orient Circle pt 1.
12. Using the joystick, move to the other hole that, with the first hole, determines the angular orientation of the part.
13. Repeat steps 7 and 8 three more times for this hole.
14. Click **OK** in the *Align 2D on 2 Circles* dialog box to complete the Two Circles (2D) alignment.

Plane-Line-Point (3D) Part Alignment

Plane-Line-Point is the preferred alignment method in most circumstances, because it gives you alignment in three dimensions for greater accuracy than a two-dimensional alignment. In this alignment method, Point determines the origin of the alignment. Line determines the angular orientation (rotation) of the alignment. Plane determines the tilt of the alignment. You need to have an object with a flat surface, somewhat perpendicular to the lens to do a Plane-Line-Point alignment, or VMS will not be able to adequately focus on the plane. For Plane-Line-Point alignment, do not select Plane-Line-Point (3D) from the Part Align submenu until after you have defined a plane, a line, or a point. It is not required that you use all three types of Features. You can for example, define a new origin point but retain the existing tilt and rotation. Also, the origin Point can be a Circle, Arc, Ellipse or Blob (the center point is used).

Place a 3-dimensional part (such as a block or a washer) on the stage. The part appears through the lens as a rectangular or circular two-dimensional object. In the case of the rectangle, the dimensions are determinable along the X and Y-axes. The circular object has a center point origin, but there is no way to determine the X or Y-axis directions because you can turn it and it looks the same. As the camera moves up and down along the Z-axis to focus on a part, it is determining the Z dimensions of the part.

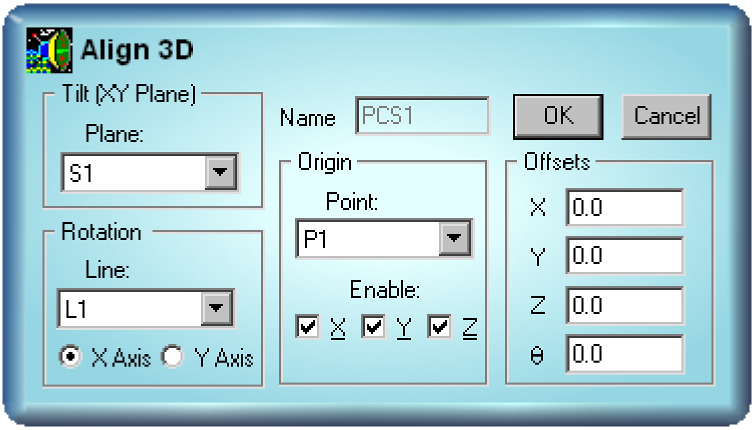
1. Create an Align Block using the VMS toolbar button.
2. (Optional) Define a plane to establish the tilt of the PCS. For the rest of this example, assume the plane is S1.
3. Measure/construct a line to be used as the X or Y-axis of the PCS. For the rest of this example, assume the line is L1.

For a circular object, because there is no X or Y direction, you cannot define a Line. Instead, select **Current** from the Line Feature drop-down list. When current is selected (in either Plane, Line, or Point dialog boxes), it means that that particular aspect of alignment (tilt, rotation, or origin) is to be kept from whatever alignment was in effect before the align block. In this case, the direction of the X and Y axes will remain whatever it was.

4. (Optional) Measure/construct a second line that intersects the first line at the desired part origin.

5. Define/construct a point. In the example below, a point was constructed at the intersection of the two lines. Alternatively, a circle (hole or pin) may be used instead. For the rest of this example, assume the point is P1.

6. Double-click on the "Define Align PCS" step to edit it. The *Align 3D* dialog box is displayed.

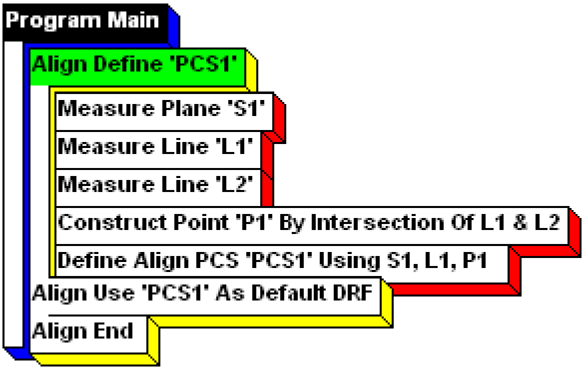


7. Click the arrow in the **Plane** area to activate the drop down list, and select **S1**.

8. Click the arrow in the Line to activate the drop-down list and select L1. Make sure that the **X Axis** is selected if the line you constructed (L1) is parallel to the X-axis. Or **Y Axis** is selected if you use a line that parallels the Y Axis.

9. Click the arrow in to the **Point** area, to activate the drop down list, and select **P1**. Make sure the **Enable X**, **Y**, and **Z** checkboxes are all checked, so that the measurements on all three axes will be used for this point (the origin).

10. Click **OK.** Define Align PCS (PCS1) Using S1, L1, P1 appears in the *Measurement Steps* window, and the Plane-Line-Point (3D) alignment is complete. The completed alignment definition steps are shown below.



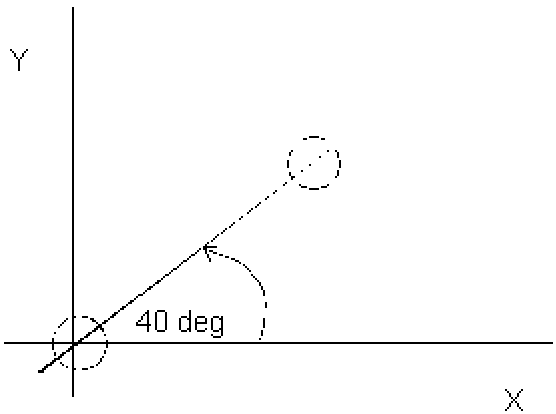
Establishing an origin with a manually placed Crosshair Finder will give you a correct, though approximate, origin in Z as well as X and Y. Establishing an origin by the intersection of two Lines is accurate in X and Y. Lines are usually projected into the pre-existing XY plane; if the previous coordinate system was MCS, the intersections Z will be at the machine Z zero, not up on the part. To generate the Z at the desired height, see *Defining a PCS*.

Establishing an Offset

You use offsets to describe where the PCS should be in relation to a feature or a part. You can establish offsets with all three alignment methods: Two Line (2D), Two Circle (2D), and Plane-Line-Point (3D). Enter offset values in the dialog box of the alignment method that you are using. Using these values, you establish an offset to find the PCS and correct the relationship of the part to the coordinate system in place.

For example, if you want to align to two holes on a part but have no feature or construction that corresponds directly with the X-axis, enter 0 for the X, Y, and Z-offset. Then, in the case of the example shown below, enter –40 for the angular offset (labeled as Deg in the dialog box) in order to create the PCS at the known offset. If you specify both an angular offset and X, Y, and/or Z offsets, VMS performs the rotation (angular offset) first, then performs the X, Y, and Z offsets.

Example:

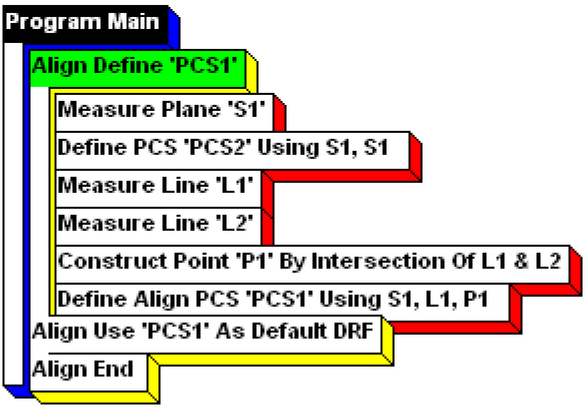


Defining an Alignment PCS on a Surface

This is very similar to the example in Plane-Line-Point Alignment, but in this example, we use the Plane as a DRF for measurement of the remaining alignment Features. This ensures that the Features are created in that plane instead of the MCS plane.

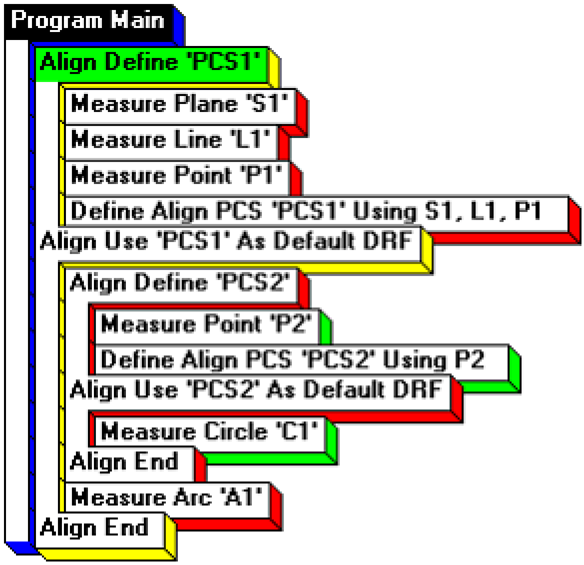
1. Create an Align Block for the part alignment FCS (PCS1).
2. Measure the Plane S1 as in the previous section and select **Datum** in its *Feature Options* dialog box.
3. Define a DRF (PCS2) with **Measure/Define PCS** using plane S1 for the PCS plane and for the PCS origin point. This establishes a tilt and a Z height for the Features to be measured next.
4. Measure Line L1 specifying PCS2 as the DRF in its *Feature Options* dialog box.
5. Measure Line L2 specifying PCS2 as the DRF in its *Feature Options* dialog box.
6. Construct Point P1 as the intersection of L1 and L2, specifying PCS2 as the DRF in its *Feature Options* dialog box.
7. Define the part alignment PCS (PCS1) using plane S1, Line L1, and Point P1.

The completed alignment is shown below.



About System Alignment

A System Alignment can be either MCS (also called "Stage") or a Fixture Alignment. Select what to use as System Alignment from the **Align/System Alignment** menu. It gives you a list to choose from. The first item in the list is "Stage", followed by any Fixture Alignments you have created. When you load a measurement program, it automatically selects the System Alignment that was chosen when it was saved.



In the example above we know that A1's Finders use PCS1 as their FCS because that step is in PCS1's **Align Use** Section. But what FCS controls the Finders for S1, L1, and P1? Those steps are not inside *any* **Align Use** Section.

Steps not inside any **Align Use** section are run according to the System Alignment. "System Alignment" is a term for the coordinate system that is in effect outside any other **Align Use** section. That includes steps in the first Align Define section, like S1, L1, and P1, and steps not inside any Align Block at all.

If a measurement program is written with a System Alignment of "Stage", that means that its first Finders, which are typically in the first Align Define section, are positioned at absolute coordinates relative to the stage. If the fixture is changed, or the measurement program is copied to a different machine, then these absolute coordinates may not be correct, and you would need to edit those first few Finders. If you had 50 such programs you would want a better way, hence, Fixture Alignments.

Specifying System Alignment

To specify System Alignment:

1. Select **Align/System Alignment** from the menu. The *System Alignments* dialog box is displayed.
2. In the *System Alignments* dialog box, select either **Stage** (if no appropriate fixture alignment exists) or a preset fixture alignment.
3. Click the **Apply** button. The *System Alignments* dialog box closes and your selection (stage or preset fixture) now appears as the active alignment in the status bar at the bottom of the window. The selection you make remains the same until you change the setting or open a new file from the **File** menu.

Creating a Fixture Alignment

In order to create a Fixture Alignment that can be called up from the *System Alignment* dialog box, write a program that uses Stage as the System Alignment and establishes a PCS by measuring Features of the fixture (or a part in the fixture.) Then add a special step that takes that alignment and adds it to the list of Fixture Alignments. This special fixture alignment program only needs to be run once (or whenever the fixture is significantly moved on the stage).

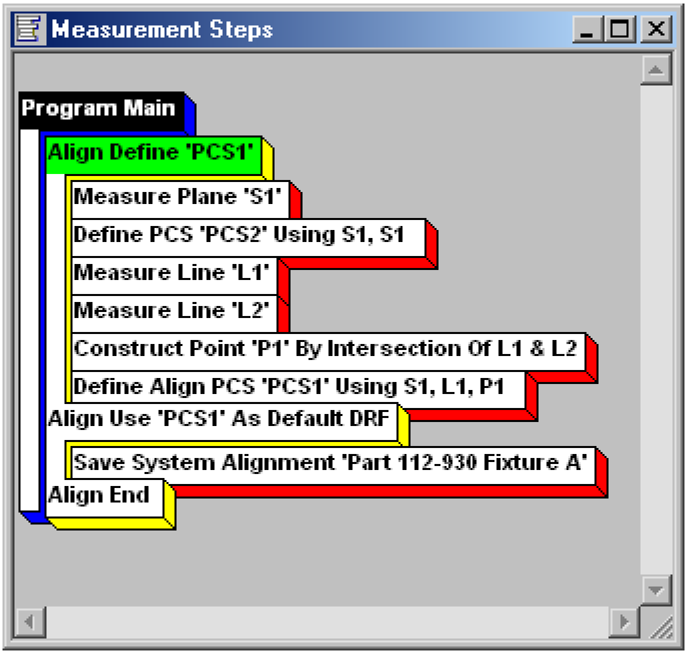
To place a Save Fixture Alignment step in the **Align Use** section of the alignment block:

1. Select the Align Use step of the alignment block in the *Measurement Steps* window.
2. Select **Align/System Alignment** from the menu. The *System Alignments* dialog box is displayed.
3. Click the **New** button. The *Save System Alignment* dialog box is displayed. Enter a descriptive name for your fixture and click **OK**. The *Save System Alignment* dialog box closes and the new fixture name appears in the *System Alignments* dialog box.

4. Close the *System Alignments* dialog box. The new fixture appears as the active alignment in the status bar at the bottom of the VMS window. A step is added under the Align Use step of the alignment block in the *Measurement Steps* window, which specifies "Save System Alignment FixtureName."

5. Save the program. Fixture alignment programs are normally saved with a .ALN extension to avoid confusing them with measurement programs.

An example of a Fixture Alignment program is shown below.



This page was left blank intentionally.

Section 6

Measurements

When you inspect a part, you are inspecting the various Features of the part. Using Finders, VMS collects and stores information about the Features that you inspect. A drop-down list of basic modes for measuring and/or constructing Features is attached to each of the Feature toolbar buttons. Click on the **Arrow** to choose from a list of Feature modes. Click on the **Feature** button to automatically get the last mode chosen for that Feature type. The first time you click on the **Feature** button the drop-down list of basic modes is presented. The mode you choose may set up certain options in the Feature's dialog box that results in, and often will set up a Finder that is appropriate to that choice.

Measure is used to generate a new Feature definition using Finders that acquire data points. **Construct** is used to generate a new Feature definition from existing Features. Examples would be: construction of a "best-fit" line between the centers of two or more previously measured circles, construction of a line perpendicular to a measured line, construction of a point at the intersection of two lines, etc. Construct is also used to create a "best-fit" Feature definition from a General Curve Feature containing point data from Finders. This technique may be used when two or more Features need to be defined using the same point data (e.g., an inscribed and circumscribed circle) or when custom processing of the point data is desired before it is used to define a Feature. Because General Curve Features are not displayed in the *Features* window, they must be selected from the *Object Names* dialog box.

The following are Features that can be inspected by clicking the corresponding Feature button on the VMS toolbar.

- Points
- Lines
- Arcs
- Circles
- Ellipses
- Planes
- Slots/Tabs
- General Curve (Add/Remove Point Data)

Points

Feature	Description	Options	Notes
Point	A point may be defined as: ➢ The result of a measurement tool, ➢ An average of data points ➢ A previously defined reference point ➢ The center of an arc or circular Feature ➢ A point that is related to other points ➢ A point that is related to a set of one or more lines.	Measure FOV	Activates *Measure Point* dialog box and presents the Point Finder.
		Measure Z	In addition to the *Measure Point* dialog box, this option automatically presents the Autofocus Finder.
		Construct	There are many ways to construct points. They are listed in the section Constructing a Point.

Measured Properties of Points

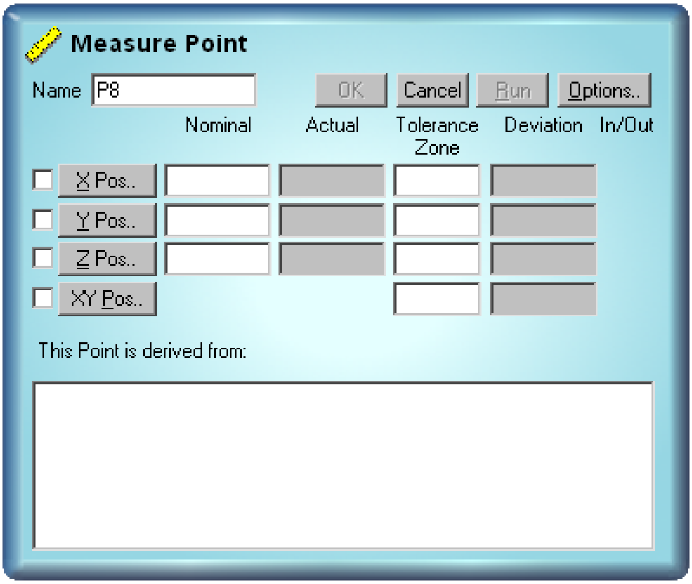
Property	Description
X Pos	Evaluates the X position independently
Y Pos	Evaluates the Y position independently
Z Pos	Evaluates the Z position independently
XY Pos	Evaluates if the actual XY location lies in a circular tolerance zone

Measuring a Point

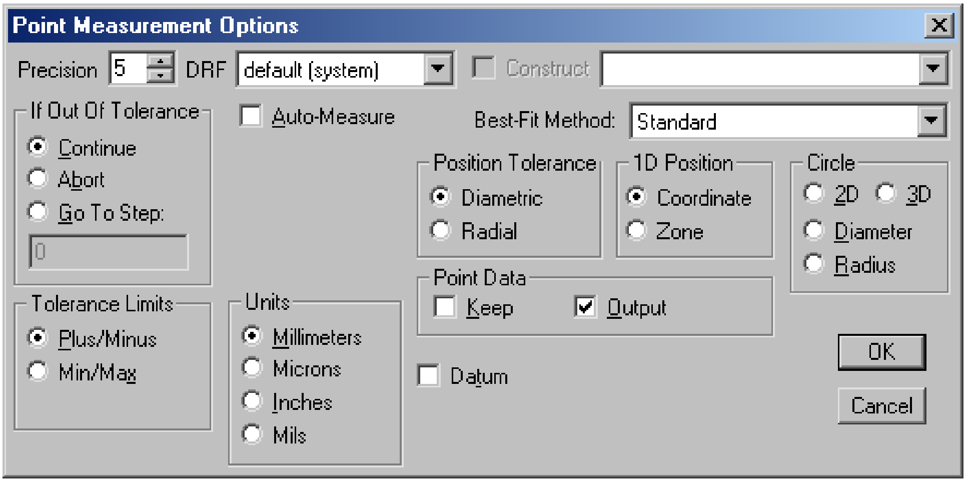
You can measure a point by using a Point Finder, Corner Finder Line Finder or Manual Crosshair Finder.

Never use a Circle Finder to define a point as the resulting Point will generally not be located at the center of the circle. If you want a point at the center of a circle, measure a Circle Feature. A Circle Feature can be used anywhere a point can be used.

1. Place the part to be inspected on the stage.
2. Click on the arrow on the **Point** button in the toolbar. From the submenu, select **Measure FOV** or **Measure Z** (to perform an Autofocus with the step). The *Measure Point* dialog box will be displayed. Input **Nominal** and **Tolerance** parameters in this box to determine whether the measurement of a Feature passes or fails. The **Nominal** value is the intended value to which all parts are compared and **Tolerance** specifies the acceptable range for part measurements. The Property Inspection check boxes determine which property inspections are reported by VMS when the measurement step is executed.



3. Click on the **Options** button to specify additional measurement parameters.



The Best-Fit-Method options are Standard, Simple Least-Squares (Data), and Simple Least-Squares (Centers).

Using the joystick, move to the edge or surface where the point is. Select a Finder from the *Video* window context menu and place it on the edge or surface. Note that if you use a Finder that returns several points, or if you add more than one Finder, the average of all the found points will be computed as the Point's position.

4. Double-click on the Finder to test-run the Finder. A blue cross is displayed over the point. The **Check Mark** on the *Video* window toolbar is gray prior to the test run, and should turn green after the test run. Click on the green **Check Mark** to signify that the Finder is acceptable. This also stores the information in the *Measure Point* dialog box.

5. Click **OK** in the *Measure Point* dialog box to define the point. Measure Point - P1 appears in the *Measurement Steps* window

Constructing a Point

The following methods can be used to construct a point in VMS:

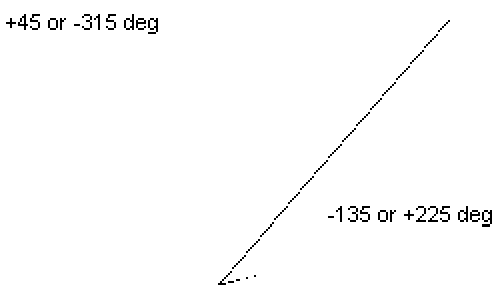
- **Best Fit** – several points, usually other previously defined Features, are used in a mathematical computation to define the current Point that most closely fits all the input points. When Best Fit is selected, the Best-Fit Method selection is enabled. When using another Feature as input to a Best-Fit construction, use the Features computed position. For example, use the center of several circles to compute a bolt-hole pattern. Occasionally, you may want to use the individual points from the input Feature's point buffer, for example, using all the edge points of a Blob, to make a circle. So, with the Best-Fit Method control, you can specify one of the following:
  - **Simple Least-Squares (Center)** – uses the input Features' computed positions, such as the center of Circles or Arcs or the centroids of Blobs.
  - **Simple Least-Squares (Data)** – uses the point buffers of the input Features. In order for this to work, the input Feature(s) must have had the **Point Data "Keep"** option On to keep the points in its Point Buffer after its step was finished. If a Feature does not have points in its buffer, then its computed position will be used instead.
  - **Standard** – uses the computed positions, except it uses the input Feature's point buffer points if the input Feature is a General Curve or a Blob.
- **At offset from Point/Circle/Arc** (Features that reduce to points)
- **Midpoint** – between two Features that reduce to Points
- **Intersection of two lines/slots** – if you select this option, VMS will automatically determine that the construction you want is the intersection Point of the two Lines.
- Intersection of Line and Plane
- Intersection of Line/Slot and Circle/Arc
- **Intersection of two Circles/Arcs** – This option has two possible intersection points. Supply nominal values to ensure the correct point is selected.
- Projection into a Plane

To construct a point:

1. Select **Measure/Point/Construct** from the menu. Enter the parameters in the *Construct Point* dialog box.
2. Click the **Options** button. Select a construction method from the drop-down list in the **Construct** box and click **OK**.
3. Click in the *Features* window on the Features to be used to construct the Point. The selected Features turn yellow, and their names also appear in the title bar of the dialog box for the Feature being constructed. You can also select a Feature by name in the *Object Names* dialog box (**Window/View/Object Names** menu). (This latter method is necessary when selecting Planes because they cannot be drawn in the *Feature* window. It may also be necessary if it is difficult to select one Feature in a crowded area.) To select all the Features in an array, enter the name of an array of Features without a subscript in the *Object Names* dialog box, e.g., CircArray instead of CircArray[3]. Click on a Feature in the *Features* window to deselect it.
4. Click **Run** in the *Define Point* dialog box. A Point (X-shape) appears in the *Features* window at the coordinates of the Point.
5. Click **OK** in the *Define Point* dialog box.

Lines

Data points returned by the Finders, or other points from other Features, are used to compute a best-fit Line. Other Features may be used in other geometric constructions to define a Line. The actual endpoints are both points on this line. They are either the points closest to the nominal endpoints (if both nominal endpoints are entered) or the points nearest the most extreme data points (if both nominal endpoints are not entered).



The order of the endpoints (and hence the line's direction) is determined from the nominal angle if it was entered, so as to make the actual angle closer to the nominal angle. For example, a nearly vertical line could be thought of as pointing up or down; if the nominal angle was +90° (or – 270°) we would know it should point up, so the endpoint with the more positive Y coordinate would be the end point. If neither the nominal endpoints nor the nominal angle are entered, the endpoint order will be chosen so that the line's direction is toward the right or upward (within the 180 degree arc between –45 and +135 degrees).

A line's angle can be negative or positive. The two numbers differ by 360°. If a line has no explicit direction (no angle is given), then the line could have two of four different angles and one of two different directions. Lines in VMS always have a direction. If you don't identify the direction by giving nominal values, VMS will choose a direction for you. The angle of a line is the angle from the PCS X axis to the projection of the line in the PCS XY plane. The X-axis pointing toward the right is 0° and positive angles are counter-clockwise from there. The direction of the line depends on the angle of the line. For example, a +45° line and a –135° line are parallel but point in opposite directions.

Measured Properties of Lines

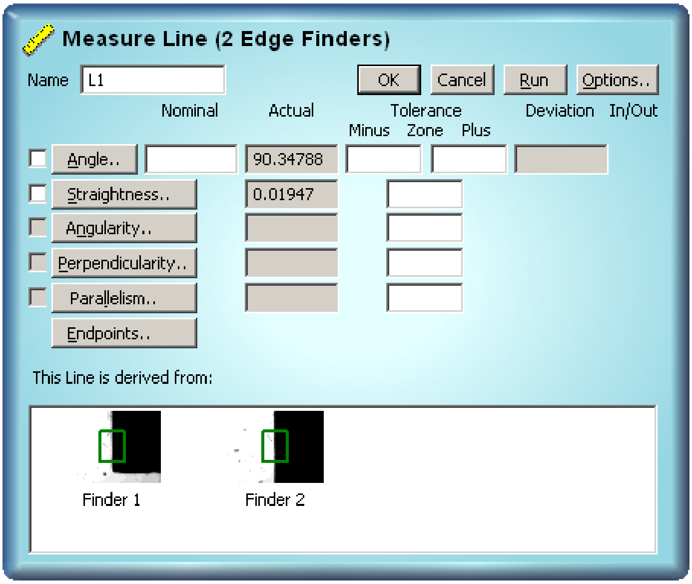
Property	Description
Angle	The actual angle will always be considered as the direction from the actual start point to the actual end point of a line.
Straightness	The width of the narrowest zone, bounded by two lines parallel to the best-fit (Actual) line itself, that contains all of the individual edge points measured.
Angularity	The width of the narrowest zone, bounded by two lines at a given angle relative to the given Line or Slot/Tab Datum (angle and Datum are specified in the *Property Inspection* dialog box), that contains all the individual edge points measured.
Perpendicularity	Same as Angularity but the angle is assumed to be 90 degrees.
Parallelism	Same as Angularity but the angle is assumed to be 0 degrees.

Property	Description
Endpoints	The direction of a line is taken from its start point to the end point You can also do a 2D positional tolerance of the endpoints, but this is not common because the endpoints are usually not precisely measured in the direction parallel to the line. Also, when entering a line's Nominal values, you can specify either both end points, or the start point and an angle, but not all three since you could then possibly enter self-contradictory data. If you enter one end point and an angle, the line does not have a second end and will not be drawn in the *Feature* window until it is actually measured.

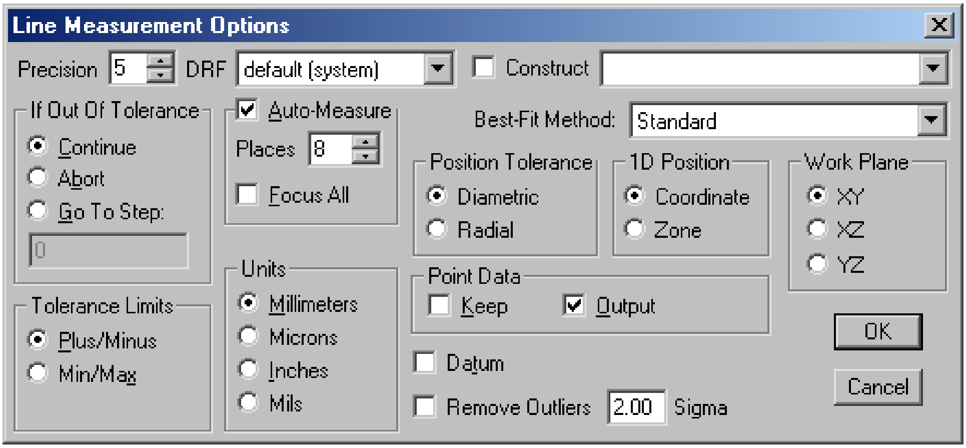
Measuring Lines

1. Using the joystick, position the part so that a straight edge is displayed in the *Video* window.

2. Click the **Arrow** next to the Line Feature button in the toolbar and select **Measure** from the list. The line you are about to measure is named L1 in the *Measure Line* dialog box. Input measurement parameters into the appropriate fields. Input **Nominal** and **Tolerance** parameters in this box to determine whether the measurement of a Feature passes or fails. The **Nominal** value is the intended value to which all parts are compared and **Tolerance** specifies the acceptable range for part measurements. To inspect certain properties of lines check the appropriate check boxes. The Property Inspection check boxes determine which property inspections are reported by VMS when the measurement step is executed.



3. Click on the **Options** button to specify additional measurement parameters.



Auto-Measure provides a method for using multiple edge finders to measure a Feature without manually placing all of the Finders. When this option is used, you place one or two Finders on the Feature, VMS will run these finders in multiple places as defined by the measurement step. Use Auto-Measure when measuring lines that are uniform, that is, that do not require different Edge Finder parameters or light settings on different sections of the Feature.

Remove Outliers will remove points that are greater than the specified number of pixels away from the primary edge. **Sigma** is the specified multiplier of standard deviation away from the primary edge.

Precision specifies the number of digits to the right of the decimal place you want in measurements. VMS will round values to the number of digits you specify. If the Feature is **Out-of-Tolerance**, you can choose to have the program **Continue**, **Abort** on the offending step or **Go to a Step** that you specify. The **Best-Fit-Method** options are **Standard**, **Simple Least-Squares (Data)**, and **Simple Least-Squares (Centers**). The **Datum** box allows you to use the Feature as a datum (reference point) while measuring other Features. **The Point Data "Keep"** box stores points in a point buffer for later use with advanced programming techniques (leave this option off unless you need it; it uses a lot of memory).

4. Select the **Line Finder** from the *Video* window context menu. Place the Finder on an edge of the part. Double-click on any open space in the *Video* window to run the Finder.

5. Click on the green **Check Mark** in the *Video* window toolbar to accept the Finder. VMS places the Finder's information in the *Measure Line* dialog box. In addition, the Finder appears in the *Features* window.

6. Measure additional segments of the same line, repeating steps 4 and 5, for greater accuracy. The title of the *Measure Line* dialog box will change to reflect the number of Edge Finders you are using to measure the line.

7. Once you have placed the desired number of Finders, click the **Run** button to add the measurement as a step in the *Measurement Steps* window. The Finders measuring the line are rerun. The line appears yellow in the *Features* window.

8. Click **OK**. The line appears green indicating that it has passed inspection. Unless **Tolerances** are input in the appropriate fields, the Feature will always pass inspection.

Constructing Lines

Construct a line using the following methods:

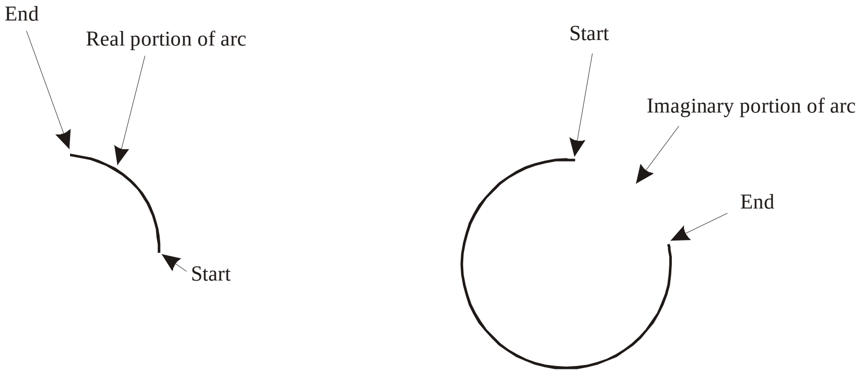
- **Best Fit** – several points, usually other previously defined Features, are used in a mathematical computation to define the current Point that most closely fits all the input points. When Best Fit is selected, the Best-Fit Method selection is enabled. When using another Feature as input to a Best-Fit construction, use the Features computed position. For example, use the center of several circles to compute a bolt-hole pattern. Occasionally, you may want to use the individual points from the input Feature's point buffer, for example, using all the edge points of a Blob, to make a circle. So, with the Best-Fit Method control, you can specify one of the following:
  - **Simple Least-Squares (Center)** – uses the input Features' computed positions, such as the center of Circles or Arcs or the centroids of Blobs.
  - **Simple Least-Squares (Data)** – uses the point buffers of the input Features. In order for this to work, the input Feature must have had its Point Data "Keep" option On to keep the points in its Point Buffer after its step was finished. If a Feature does not have points in its buffer, then its computed position will be used instead.
  - **Standard** – uses the computed positions, except it uses the input Feature's point buffer points if the input Feature is a General Curve or a Blob.
- Through Point/Circle/Arc at an angle
- Midline
- Intersection of two Planes
- Projection into a Plane

To construct a line:

1. Measure the Features that you will use to construct the line, for example, two points, two planes, etc.
2. Click on the drop-down arrow next to the **Line** Feature tool in the toolbar and select **Construct**. The *Construct Line* dialog box will be displayed. Enter the parameters of the Line you are constructing.
3. Click the **Options** button and enter the construction method. Click **OK**.
4. Locate the first Feature in the *Features* window that is to be used to construct the line and select it by single clicking on the Feature. The Feature will turn yellow. In the title bar of the *Features* window, the location of the Feature will be displayed along with its name. You can also select a Feature by name in the *Object Names* dialog box (**Window/View/Object Names** menu). (This latter method is necessary when selecting Planes because they cannot be drawn in the *Features* window. It may also be necessary if it is difficult to select one Feature in a crowded area.) To select all the Features in an array, enter the name of an array of Features without a subscript in the *Object Names* dialog box, e.g., CircArray instead of CircArray[3]. Click on a Feature in the *Features* window to deselect it.
5. Locate the second Feature to be used in the construction of the line in the *Features* window and select it. Click **OK** in the *Construct Line* dialog box. VMS will construct the line based on the method and Features selected.

Arcs

Arcs are Features similar to circles: they have a center and a radius, but they also have start and end angles. By definition, the Arc extends counter clockwise from the start angle to the end angle.



The start and end angles of an Arc are relative to the DRF that the Arc is measured in, that is, 0° points toward the direction of +X. In degrees, the real part of the Arc goes counterclockwise from the start to the end angle (therefore, the end angle is always greater than the start angle). Usually the precise endpoints of an Arc cannot be measured as part of the edge of the Arc, because it blends into or intersects some other straight edge of the part and the corner is not well defined.

If nominal start and end angles are supplied, then the "real" part of the Arc is defined. The nominal angles, if supplied, are always used as the actual angles. Do not place Finders over the ends of an Arc when nominals are given, or it could throw off the best-fit calculation. The fit determines the center point and radius actuals from the measured points (or other Features if it is a construction), but anything requiring actual start and end angles will use nominal angles instead. (If the exact position of the end of an Arc is needed, you can construct the intersection of a line or circle with the Arc.)

If you construct an Arc without giving the nominal angles, VMS will guess which points represent the endpoints and which part of the Arc you want. Always give nominals when possible.

Feature	Options	Application
Arc	Measure FOV	Assumes the Arc fits in the Field of View and automatically presents an Arc Finder.
	Measure Non-FOV	Assumes the Arc is much bigger than the Field of View and automatically presents a Line Finder; you'd want to place several of these over the length of the Arc.
	Construct	Constructs the Arc from other Features; possible construction methods are listed in Constructing Arcs.

Measured Properties of Arcs

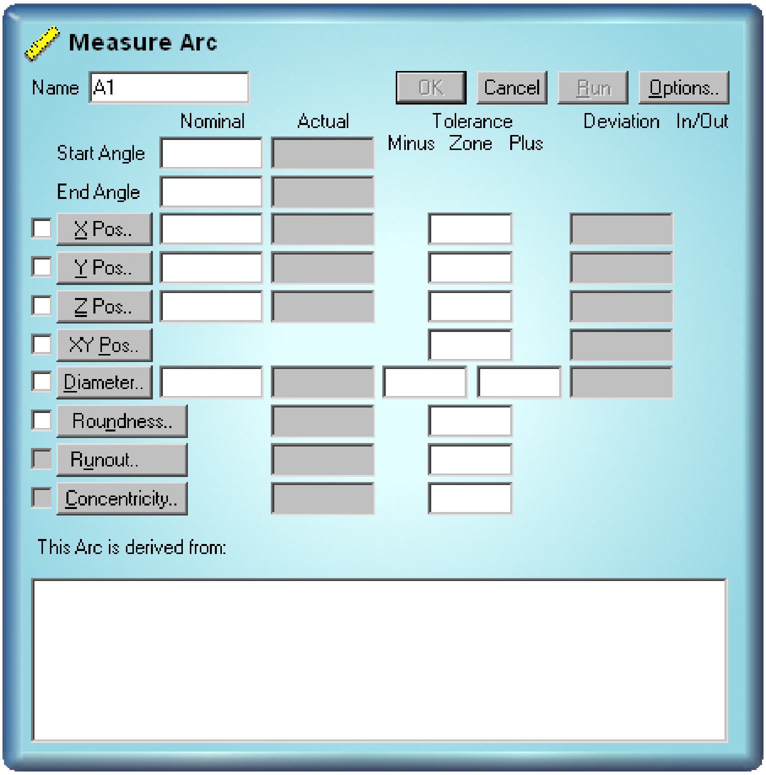
Property	Description
Position of the center (including bonus tolerancing)	This is the straight-line XY distance from the circle's nominal position to its actual measured position (computed by the selected fit algorithm). This value is how far out the position of the center is. Usually this number is shown as double the actual distance, to correspond with tolerance (specified as the diameter of the tolerance zone).
Diameter	Measures the diameter of an arc.
Roundness	The width of the narrowest zone, bounded by two circles concentric with the best-fit (Actual) arc that contains all the individual edge points measured.
Runout	Same as roundness but the two circles are concentric with the given Datum (Datum is specified in the *Property Inspection* dialog box).

Property	Description
Concentricity	ANSI concentricity is really only applicable to cylinders or other 3 dimensional shapes that have a length along which the axis position is measured in multiple places. For a 2D Arc or Circle, VMS reduces this to a simple distance measurement, from the Datum of the best-fit center, without computing the midpoints of all opposite surface points. The tolerance you enter is a zone diameter (i.e., twice the allowable position error). The returned concentricity measurement is therefore the distance doubled to match the Tolerance, i.e., the smallest Tolerance zone diameter that the Arc or Circle would pass.

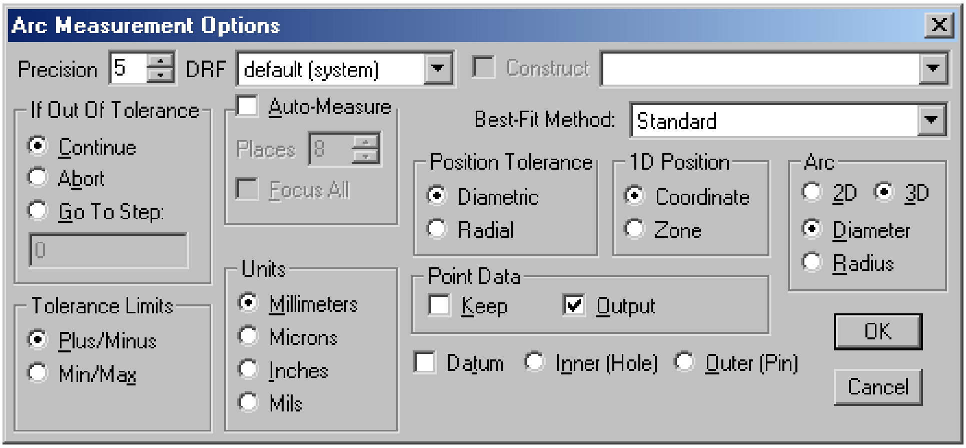
Measuring Arcs

1. Using the joystick, move the Feature to be measured so that it is visible in the middle of the Field of View in the *Video* window. If you are unable to move the Feature into the Field of View, select the Arc non-FOV option from the toolbar. This will display the Line Finder in the *Video* window.

2. Click the **Arc** button in the toolbar. The *Measure Arc* dialog box is displayed. Input measurement parameters into the appropriate fields. To inspect certain properties of arcs check the appropriate check boxes. Specify output results by clicking on the property buttons.



3. Click on the **Options** button to specify additional measurement parameters.



Auto-Measure provides a method for using multiple Edge Finders to measure a Feature without manually placing all of the Finders. When this option is used, place one or two Finders on the Feature, VMS will run these Finders in multiple places as defined by the measurement step. Use **Auto-Measure** when measuring Arcs that are uniform, that is, that do not require different Edge Finder parameters or light settings on different sections of the Feature.

Precision specifies the number of digits to the right of the decimal place you want in measurements. VMS will round values to the number of digits you specify. If the Feature is **Out of Tolerance**, you can choose to have the program **Continue**, **Abort** on the offending step or **Go To a Step** that you specify.

The **Best-Fit-Method** options are **Standard**, **Simple Least-Squares (Data)**, **Simple Least-Squares (Centers)**, **Inscribed** and **Circumscribed**.

The **Datum** box allows you to use the Feature as a datum (reference point) while measuring other Features. **The Point Data "Keep"** box stores points in a point buffer for later use with advanced programming techniques (leave this option off unless you need it; it uses a lot of memory).

In the Circle/Arc area, if **3D** is selected, the Z position of the Feature is set to the average Z value of the measured points. This gives a more intuitive, but less accurate, result for the Z position. If **2D** is selected, the Z position of the Feature is zero, i.e., it is projected onto the XY plane of the Datum Reference Frame. If the surface on which the Feature lies is used to establish the DRF before measuring the Feature, 2D gives a more accurate Z position of the Feature in relation to the Z positions of other Features.

4. Place the Arc Finder over the Feature being inspected in *Video* window. The scan direction of the Arc Finder is always from the inside to the outside. Double-click on any open space to run the Finder.
5. Click on the green **Check Mark** in the *Video* window toolbar to accept the Finder. VMS places the Finder's information in the *Measure Arc* dialog box. In addition, the Finder appears in the *Features* window.
6. Measure additional segments of the same Feature, repeating steps 5 and 6, for greater accuracy. The title of the *Measure Arc* dialog box will change to reflect the number of Edge Finders you are using to measure the Feature.
7. Once you have placed the desired number of Finders, click the **Run** button to add the measurement as a step in the *Measurement Steps* window. The Finders measuring the Feature are rerun. The Feature appears yellow in the *Features* window.
8. Click **OK**. The Feature appears green indicating that it has passed inspection. Unless **Tolerances** are input in the appropriate fields, the Feature will always pass inspection.

Constructing Arcs

Construction methods available are:

- **Best Fit** – several points, usually other previously defined Features, are used in a mathematical computation to define the current Point that most closely fits all the input points. When Best Fit is selected, the Best-Fit Method selection is enabled. When using another Feature as input to a Best-Fit construction, use the Features computed position. For example, use the center of several circles to compute a bolt-hole pattern. Occasionally, you may want to use the individual points from the input Feature's point buffer, for example, using all the edge points of a Blob, to make a circle. So, with the Best-Fit Method control, you can specify one of the following:
  - **Simple Least-Squares (Center)** – uses the input Features' computed positions, such as the center of Circles or Arcs or the centroids of Blobs.
  - **Simple Least-Squares (Data)** – uses the point buffers of the input Features. In order for this to work, the input Feature must have had its Point Data "Keep" option On to keep the points in its Point Buffer after its step was finished. If a Feature does not have points in its buffer, then its computed position will be used instead.
  - **Standard** – uses the computed positions, except it uses the input Feature's point buffer points if the input Feature is a General Curve or a Blob.
  - **Inscribed** or **Circumscribed** may be selected to perform a minimum or maximum circle fit.
- Tangent to two Lines/Slots/Tabs – computes the position of a Circle or Arc of the given Nominal diameter (so the Nominal is needed) that is tangent (just touching) to both Lines. When two lines cross there are four corners into which the Circle or Arc might fit. For VMS to give you the right one, you need to also supply the nominal XY coordinates so it can choose the closest one.

To Construct an Arc:

1. Measure the Features that you will use to construct the Arc, for example, two Lines, etc.
2. Click on the drop-down arrow next to the **Arc Feature** tool in the toolbar and select **Construct**. The *Construct Arc* dialog box will be displayed. Enter the parameters of the arc you are constructing.
3. Click the **Options** button and enter the construction method. Click **OK**.
4. Locate the first Feature in the *Features* window that is to be used to construct the Arc and select it by single-clicking on the Feature. The Feature will turn yellow. In the title bar of the *Features* window, the location of the Feature will be displayed along with its name.
5. Locate the second Feature to be used in the construction of the arc in the *Features* window and select it. Click **OK** in the *Construct Arc* dialog box.

Circles

Feature	Description	Options	Notes
Circle	Circles may be measured with Finders or constructed by selecting other Features that reduce to points for a best-fit circle. Circles can also be constructed as tangent to two Lines (used primarily to determine where a gauge ball or pin will rest in a notch). Construction of a circle tangent to two lines requires entry of the circle diameter or radius.	Measure FOV	Assumes the circle fits in the Field of View and presents Circle Finder.
		Measure Non-FOV	Assumes the circle does not fit in the Field of View and presents the Line Finder.
		Construct	Constructs the Circle from other Features; possible construction methods are listed in Constructing Circles.

Measured Properties of Circles

Property	Description
Position of the center (including bonus tolerancing)	This is the straight-line XY distance from the circle's nominal position to its actual measured position (computed by the selected fit algorithm). This value is how far out the position of the center is. Usually this number is shown as double the actual distance, to correspond with tolerance (specified as the diameter of the tolerance zone).
Diameter	Measures the diameter of a Circle.
Roundness	The width of the narrowest zone, bounded by two Circles concentric with the best-fit (Actual) Circle that contains all the individual edge points measured.
Runout	Same as roundness but the two circles are concentric with the given Datum (Datum is specified in the *Property Inspection* dialog box).
Concentricity	ANSI concentricity is really only applicable to cylinders or other 3 dimensional shapes that have a length along which the axis position is measured in multiple places. For a 2D Arc or Circle, VMS reduces this to a simple distance measurement, from the Datum of the best-fit center, without computing the midpoints of all opposite surface points. The tolerance you enter is a zone diameter (i.e., twice the allowable position error). The returned concentricity measurement is therefore the distance doubled to match the Tolerance, i.e., the smallest Tolerance zone diameter that the Arc or Circle would pass.

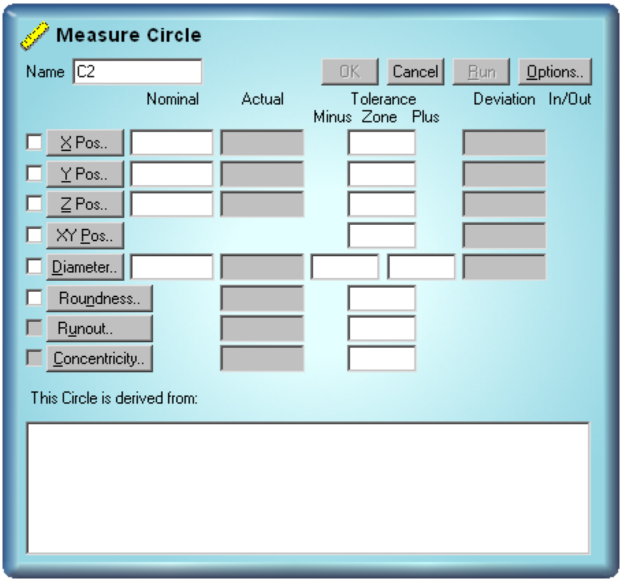
Measuring Circles

1. Using the joystick, move the Feature to be measured so that it is visible in the middle of the field of view in the *Video* window.

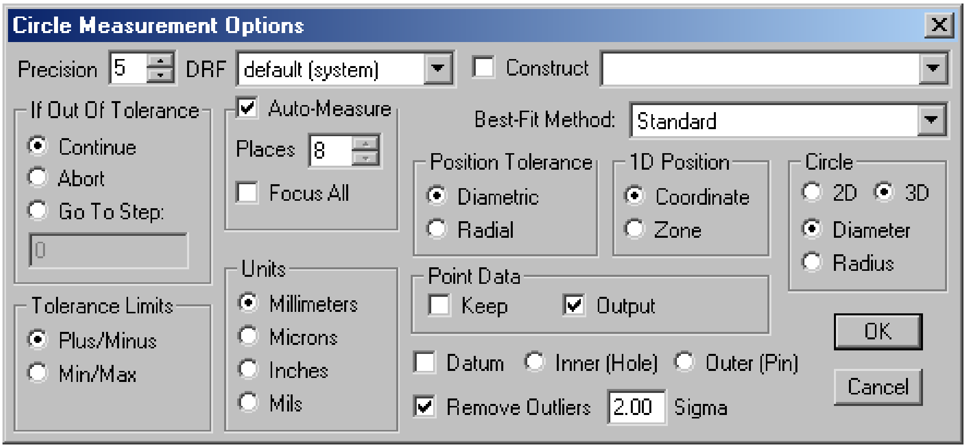
   - If you are unable to move the Feature into the field of view, click on the **Measure Non-FOV** option on the toolbar. This will display the Line Finder in the *Video* window or you can try the Arc Finder tool.
   - If you can display the entire circle in the *Video* window, click the **Measure FOV** option in the toolbar to display the Circle Finder. The *Measure Circle* dialog box is displayed.

2. Input measurement parameters into the appropriate fields. To inspect certain properties of circles check the appropriate check boxes. Specify output results by clicking on the property buttons. Input **Nominal** and **Tolerance** parameters in this box to determine whether the measurement of a Feature passes or fails.

   The **Nominal** value is the intended value to which all parts are compared and **Tolerance** specifies the acceptable range for part measurements. The Property Inspection check boxes determine which property inspections are reported by VMS when the measurement step is executed. Additional measurement parameters can be added by clicking on the **Options** button.



3. Click the **Options** button to specify additional parameters.



Auto-Measure provides a method for using multiple edge finders to measure a Feature without manually placing all of the finders. When this option is used, you place one or two Finders on the Feature, VMS will run these finders in multiple places as defined by the measurement step. Use Auto-Measure when measuring Circles that are uniform, that is, that do not require different Edge Finder parameters or light settings on different sections of the Feature.

Precision specifies the number of digits to the right of the decimal place you want in measurements. VMS will round values to the number of digits you specify. If the Feature is **Out-of-Tolerance**, you can choose to have the program **Continue**, **Abort** on the offending step, or **Go To** a **Step** that you specify.

The **Best-Fit-Method** options are **Standard**, **Simple Least-Squares (Data)**, **Simple Least-Squares (Centers)**, **Inscribed** and **Circumscribed**.

Remove Outliers will remove pixels that are greater than the specified number of pixels away from the primary edge. **Sigma** is the specified multiplier of standard deviation away from the primary edge.

The **Datum** box allows you to use the Feature as a datum (reference point) while measuring other Features. **The Point Data "Keep"** box stores points in a point buffer for later use with advanced programming techniques (leave this option off unless you need it; it uses a lot of memory).

In the Circle/Arc area, if **3D** is selected, the Z position of the Feature is set to the average Z value of the measured points. This gives a more intuitive, but less accurate, result for the Z position. If **2D** is selected, the Z position of the Feature is zero, i.e., it is projected onto the XY plane of the Datum Reference Frame. If the surface on which the Feature lies is used to establish the DRF before measuring the Feature, 2D gives a more accurate Z position of the Feature in relation to the Z positions of other Features.

4. Place the Circle Finder over the Feature being inspected in *Video* window so that the inner circle is completely inside of the Feature and the outer circle of the Finder is completely around the Feature being measured. The scan direction of the Circle Finder is always from the inside to the outside. Double-click on any open space to run the Finder.
5. Click on the green **Check Mark** in the *Video* window toolbar to accept the Finder. VMS places the Finder's information in the *Measure Circle* dialog box. In addition, the Finder appears in the *Features* window.
6. If the Feature does not fit in the field of view and you are using the Arc or Line Finder to measure the circle, measure additional segments of the Feature for greater accuracy. The title of the *Measure Circle* dialog box will change to reflect the number of Edge Finders you are using to measure the Feature.
7. Once you have placed the desired number of Finders, click the **Run** button to add the measurement as a step in the *Measurement Steps* window. The Finders measuring the Feature are rerun. The Feature appears yellow in the *Features* window.
8. Click **OK**. The Feature appears green indicating that it has passed inspection. Unless **Tolerances** are input in the appropriate fields, the Feature will always pass inspection.

Constructing Circles

Construction methods available are:

- **Best Fit** – several points, usually other previously defined Features, are used in a mathematical computation to define the current Point that most closely fits all the input points. When Best Fit is selected, the Best-Fit Method selection is enabled. When using another Feature as input to a Best-Fit construction, use the Features computed position. For example, use the center of several circles to compute a bolt-hole pattern. Occasionally, you may want to use the individual points from the input Feature's point buffer, for example, using all the edge points of a Blob, to make a circle. So, with the Best-Fit Method control, you can specify one of the following:
    - **Simple Least-Squares (Center)** – uses the input Features' computed positions, such as the center of Circles or Arcs or the centroids of Blobs.
    - **Simple Least-Squares (Data)** – uses the point buffers of the input Features. In order for this to work, the input Feature must have had its Point Data "Keep" option On to keep the points in its Point Buffer after its step was finished. If a Feature does not have points in its buffer, then its computed position will be used instead.
    - **Standard** – uses the computed positions, except it uses the input Feature's point buffer points if the input Feature is a General Curve or a Blob.
    - **Inscribed** or **Circumscribed** may be selected to perform a minimum or maximum circle fit.
- **Tangent to two Lines/Slots/Tabs** – computes the position of a Circle or Arc of the given Nominal diameter (so the Nominal is needed) that is tangent (just touching) to both Lines. When two lines cross there are four corners into which the Circle or Arc might fit. For VMS to give you the right one, you need to also supply the nominal XY coordinates so it can choose the closest one.

To Construct a Circle:

1. Measure the Features that you will use to construct the circle.
2. Click on the drop-down arrow next to the **Circle** Feature tool in the toolbar and select **Construct**. The *Construct Circle* dialog box will be displayed. Enter the parameters of the circle you are constructing.
3. Click the **Options** button and enter the construction method. Click **OK**.
4. Locate the first Feature in the *Features* window that is to be used to construct the circle and select it by single clicking on the Feature. The Feature will turn yellow. Also, in the title bar of the *Features* window, the location of the Feature will be displayed along with its name.

5. Locate the second Feature to be used in the construction of the circle in the *Features* window and select it. Click **OK** in the *Construct Circle* dialog box.

Ellipses

Feature	Description	Options	Notes
Ellipse	The Ellipse measurement is used to measure elliptical or circular Features. Because it provides information about the Feature's orientation (major axis) as well as its non-circularity (eccentricity) it can be more useful than a Circle measurement in process control of circular Features.	Measure	Automatically presents a Line Finder (although you can use a different Finder type if you prefer).
		Construct	Lets you choose a number of other Features to construct an Ellipse.

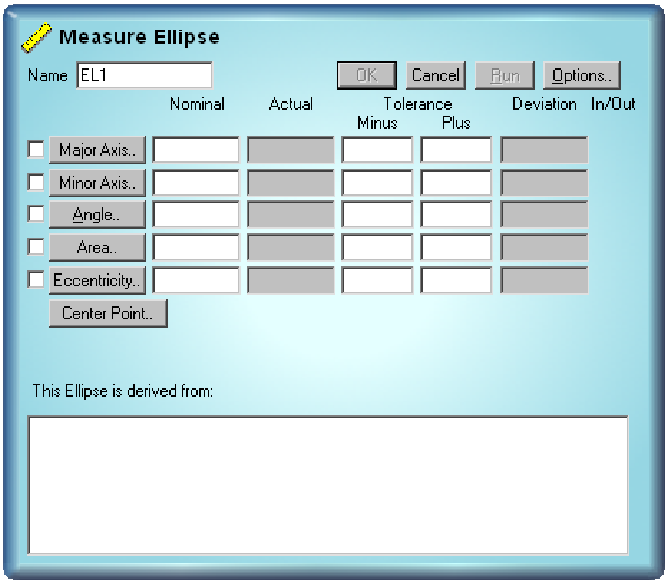
Measured Properties of Ellipses

Property	Description
Major Axis	Long diameter
Minor Axis	Short diameter
Angle	The Angle is the angle of the major axis to the DRF X axis.
Area	Area of the ellipse
Eccentricity	0.0 for a circle and increases toward 1.0 as the ellipse becomes narrower.
Center Point	Evaluates the X/Y/Z positions independently and whether the actual XY location lies in a circular tolerance zone.

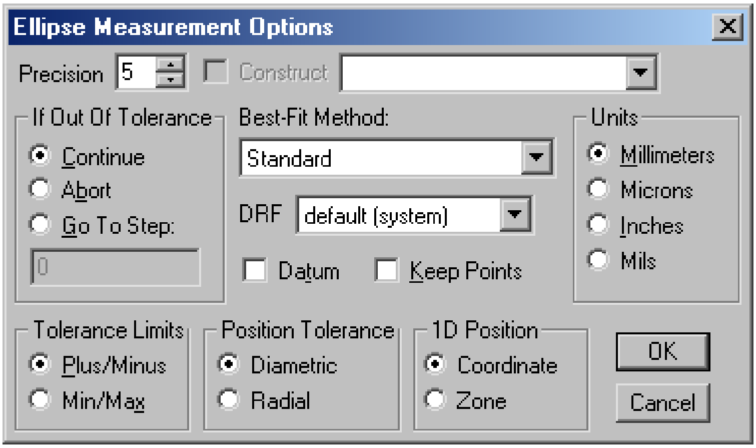
An ellipse Feature may be used in any construction/relation requiring a Feature that reduces to a point. The center point of the ellipse is used as the point in the construction/relation.

Measuring Ellipses

1. Using the joystick, move the Feature to be measured so that it is visible in the in the *Video* window.
2. Click the **Ellipse** button in the toolbar. The *Measure Ellipse* dialog box is displayed and the Line Finder is presented. Input measurement parameters into the appropriate fields. To inspect certain properties of Ellipses check the appropriate check boxes. Specify output results by clicking on the property buttons.



3. Click the **Options** button to specify additional parameters.



Precision specifies the number of digits to the right of the decimal place you want in measurements. VMS will round values to the number of digits you specify. If the Feature is **Out-of-Tolerance**, you can choose to have the program **Continue**, **Abort** on the offending step, or **Go To a Step** that you specify.

The **Best-Fit-Method** options are **Standard**, **Simple Least-Squares (Data)**, and **Simple Least-Squares (Centers**).

The **Datum** box allows you to use the Feature as a datum (reference point) while measuring other Features. **The Keep Points** box stores points in a point buffer for later use with advanced programming techniques (leave this option off unless you need it; it uses a lot of memory).

4. Place the Finder over the Feature being inspected in *Video* window. Double-click on any open space to run the Finder.

If the ellipse is nearly a circle that fits into the field of view, a Circle Finder could be used instead of the Line Finder.

5. Click on the green **Check Mark** in the *Video* window toolbar to accept the Finder. VMS places the Finder's information in the *Measure Ellipse* dialog box. In addition, the Finder appears in the *Features* window. Place additional Finders around the Feature as needed for the greatest accuracy.

6. Once you have placed the desired number of Finders, click the **Run** button to add the measurement as a step in the *Measurement Steps* window. The Finders measuring the Feature are rerun. The Feature appears yellow in the *Features* window.

7. Click **OK**. The Feature appears green indicating that it has passed inspection. Unless **Tolerances** are input in the appropriate fields, the Feature will always pass inspection.

Constructing Ellipses

The only type of construction allowed for an ellipse is regression best-fit using Features that reduce to a point or point buffers of other Features.

- **Best Fit** – several points, usually other previously defined Features, are used in a mathematical computation to define the current Point that most closely fits all the input points. When Best Fit is selected, the Best-Fit Method selection is enabled. When using another Feature as input to a Best-Fit construction, use the Features computed position. For example, use the center of several circles to compute a bolt-hole pattern. Occasionally, you may want to use the individual points from the input Feature's point buffer, for example, using all the edge points of a Blob, to make a circle. So, with the Best-Fit Method control, you can specify one of the following:
    - **Simple Least-Squares (Center)** – uses the input Features' computed positions, such as the center of Circles or Arcs or the centroids of Blobs.
    - **Simple Least-Squares (Data)** – uses the point buffers of the input Features. In order for this to work, the input Feature must have had its Point Data "Keep" option On to keep the points in its Point Buffer after its step was finished. If a Feature does not have points in its buffer, then its computed position will be used instead.
    - **Standard** – uses the computed positions, except it uses the input Feature's point buffer points if the input Feature is a General Curve or a Blob, or the Feature being constructed is a Plane.

To Construct an Ellipse:

1. Measure the Features that you will use to construct the Ellipse.
2. Click on the drop-down arrow next to the **Ellipse Feature** tool in the toolbar and select **Construct**. The *Construct Ellipse* dialog box will be displayed. Enter the parameters of the Ellipse you are constructing.
3. Click the **Options** button and enter the construction method. Click **OK**.
4. Locate the first Feature in the *Features* window that is to be used to construct the Ellipse and select it by single-clicking on the Feature. The Feature will turn yellow. Also, in the title bar of the *Features* window, the location of the Feature will be displayed along with its name.
5. Locate the second Feature to be used in the construction of the ellipse in the *Features* window and select it. Click **OK** in the *Construct Ellipse* dialog box. At least five (5) points are needed to construct an Ellipse, so you need at least 5 Features if you are using their computed positions or you must use the point buffer(s) of another Feature(s).

Planes

Feature	Description	Options	Notes
Plane	Planes may be measured with Finders or constructed by selecting Features that reduce to points or measured Features for which Point Data "Keep" was selected. In the latter case, the actual point data from the selected Features (e.g., a circle or two lines) is used to construct the plane.	Measure	Automatically presents an Autofocus Finder since you are usually measuring XY planes and the Z coordinates are important.
		Construct	Lets you choose a number of other Features to construct a Plane.

Measured Properties of Planes

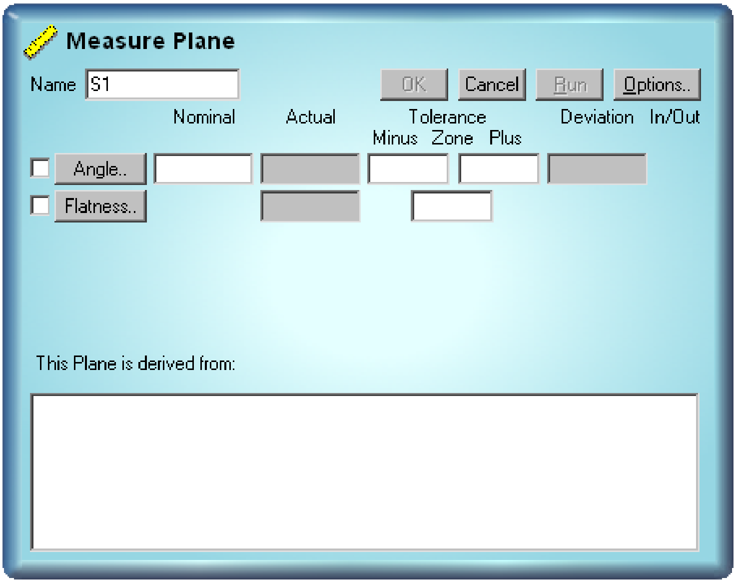
Property	Description
Angle	How tilted the plane is from the DRF XY plane.
Flatness	The distance between two planes, parallel to the best-fit (actual) plane, that contains all the individual edge points measured.

Measuring Planes

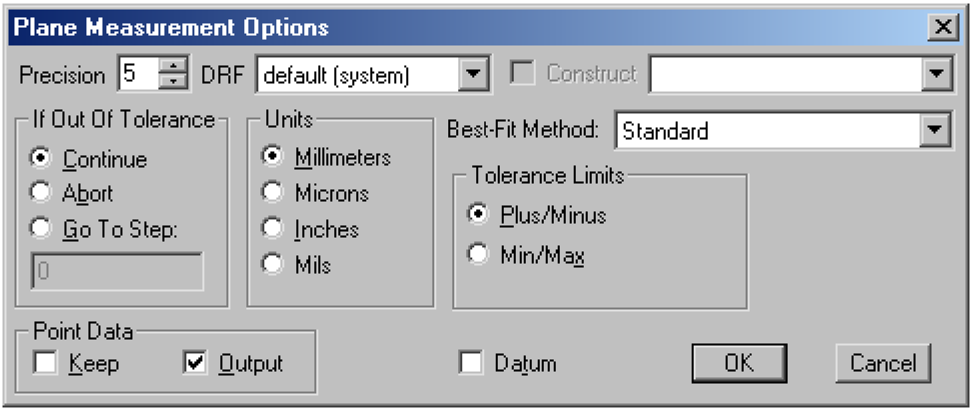
Measuring a plane can be part of defining the tilt of your alignment (see Note in Plane-Line-Point Part Alignment), but they can also be measured for other purposes, e.g., to use in constructions or to measure flatness. You must define a minimum of three points. When measuring a plane with Autofocus Finders, the angle of the plane must not be too steep for the camera to focus on it.

1. Secure the part on the stage.
2. Click on the Plane's **Arrow** button in the toolbar, and select **Measure**. The *Measure Plane* and *Autofocus* dialog boxes are displayed. The Autofocus

Finder is also displayed in the *Video* window. To inspect the properties of a plane, click on the appropriate check boxes in the *Measure Plane* window.



3. Click on the **Options** button to specify additional parameters.



4. Using the joystick, position the Autofocus over a desired area of the part, for example, the lower left corner. Select **Textured Surface** or **Smooth** (Ronchi grid) in the *Autofocus* dialog box, depending on the surface type of your part.

5. Click **Run** to test run the Autofocus Finder. When you are satisfied with the Autofocus, click **OK**.

6. Double-click in the *Video* window to test run the Finder. A blue cross is displayed over the point. Click on the green **Check Mark** to accept the Finder. This also stores the information in the *Measure Plane* dialog box. The title bar in the dialog box changes to indicate the number of Finders.

7. Using the joystick, place the Autofocus Finder over another areas of the part, for example, the upper right and upper left corners. Repeat steps 3 through 5 to define the second and third Autofocus points. Choosing focus points that are far away from each other increases the accuracy of the Plane measurement. If used for alignment, it will improve that also.

8. Click **OK** in the *Measure Plane* dialog box. A Measure Plane step will be added to the program in the *Measurement Steps* window.

Constructing Planes

The only type of construction allowed for a plane is best-fit using Features that reduce to a point or point buffers of other Features.

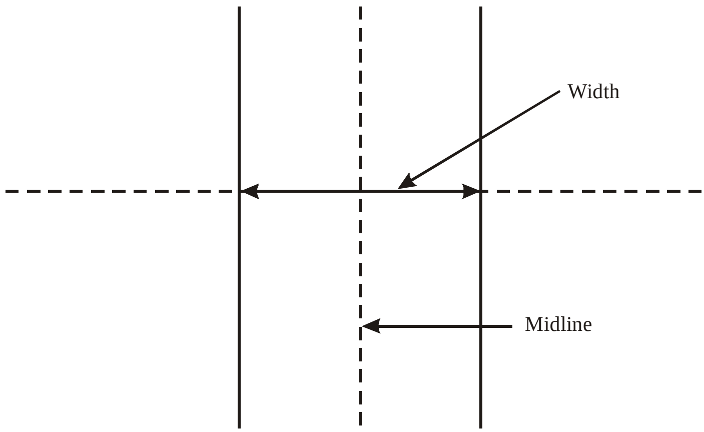
- **Best Fit** – several points, usually other previously defined Features, are used in a mathematical computation to define the current Point that fits all the input points. When Best Fit is selected, the Best-Fit Method selection is enabled. When using another Feature as input to a Best-Fit construction, use the Features computed position. For example, use the center of several circles to compute a bolt-hole pattern. Occasionally, you may want to use the individual points from the input Feature's point buffer, for example, using all the edge points of a Blob, to make a circle. So, with the Best-Fit Method control, you can specify one of the following:
    - Simple Least-Squares (Center) – uses the input Features' computed positions, such as the center of Circles or Arcs or the centroids of Blobs.
    - Simple Least-Squares (Data) – uses the point buffers of the input Features. In order for this to work, the input Feature must have had its Point Data "Keep" option On to keep the points in its Point Buffer after its step was finished. If a Feature does not have points in its buffer, then its computed position will be used instead.
    - Standard – for plane construction, this is the same as Simple Least-Squares (Data).
    - Seating Plane – computes a seating (coplanarity) plane above or below the point data.

To Construct a Plane:

1. Measure the Features that you will use to construct the Plane.
2. Click on the drop-down arrow next to the **Plane Feature** tool in the toolbar and select **Construct**. The *Construct Plane* dialog box will be displayed. Enter the inspection parameters of the plane you are constructing.
3. Click the **Options** button and enter the construction method (best-fit is the only option). Click **OK**.
4. Locate the first Feature in the *Features* window that is to be used to construct the plane and select it by single-clicking on the Feature. The Feature will turn yellow. In the title bar of the *Features* window, the location of the Feature will be displayed along with its name.
5. Locate the second Feature to be used in the construction of the plane in the *Features* window and select it. Click **OK** in the *Construct Plane* dialog box.

Slots/Tabs

Slots and Tabs are constructed from two Lines. VMS automatically determines if the Slot/Tab is vertical or horizontal; the Lines must be within +/-10° of vertical or horizontal with respect to the DRF if the position of the Slot/Tab is to be measured. A midline of the Slot/Tab is constructed, and the length of the Feature (perpendicular to its width) is computed from the midline endpoints if no nominal length is given.



Feature	Description	Notes
Slot/Tab	A Slot/Tab, represented by its midline, is the same as a line in constructions. It may also be used as a Line in defining coordinate systems.	Slots and Tabs are handy to use in more circumstances than one might think. They are a good way to measure the width between two parallel lines, or to find the center of a square or rectangular shape (define a vertical slot/tab and a horizontal slot/tab and intersect them). Slots and Tabs also allow the width of the Slot/Tab to earn bonus positional tolerancing (MMC/LMC).

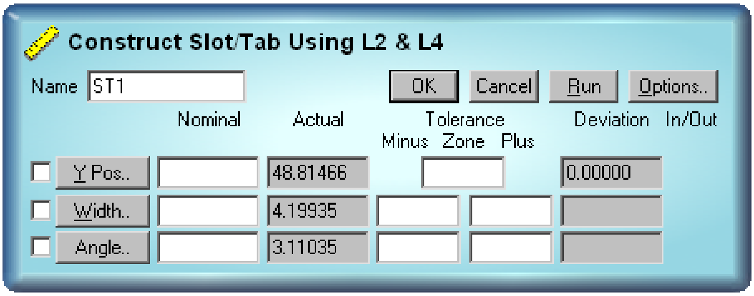
Measured Properties of Slots/Tabs

Property	Description
Position	One dimension only, e.g., X only for a vertical Slot/Tab. Bonus position tolerancing can be earned from the width (a Slot is "inner", a Tab is "outer") and is in one dimension only.
Width	The distance between the two lines at the midpoint of the midline, perpendicular to the mid line.
Angle	The angle of the midline with respect to the DRF's X or Y axis.

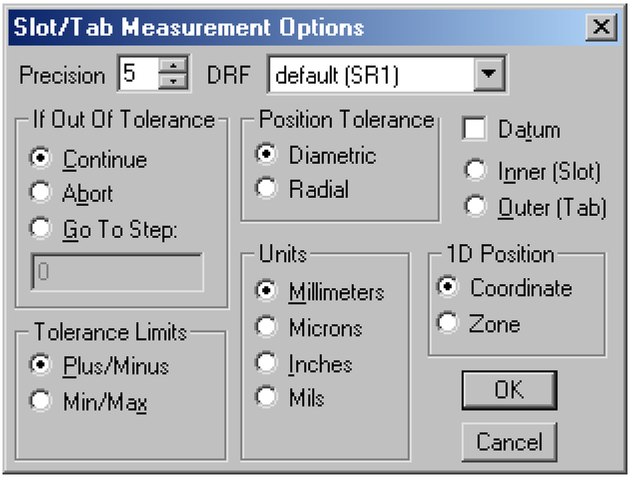
Constructing Slots/Tabs

Unlike other Features, Slots/Tabs cannot be measured with Finders. They can only be constructed from two previously defined Lines.

1. Select the **Slots/Tabs** button in the toolbar. The *Construct Slot/Tab* dialog box is displayed. Input the desired inspection parameters.
2. Select the two lines used for construction from the *Features* window or from the *Object Names* dialog box.

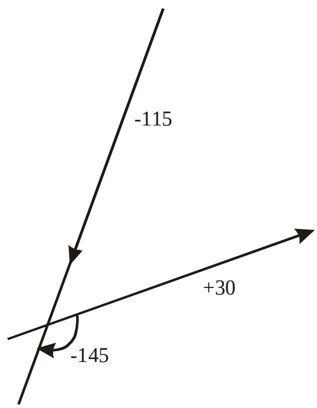


3. Click the **Options** button to specify additional parameters.



Angle Measurements

To determine the angle between two lines VMS uses the lines' directions and the order in which the lines are selected in the *Angle Measurement* dialog box. The actual angle of the first named line is subtracted from the actual angle of the second-named line. Then the nominal angle of the measurement is used to determine which angle to report.



If the +30° line was chosen first and the -115° line was chosen second, the angle between the two lines would be -145°: (-115)-(+30)=-145. A synonym for -145° is +215°. If the nominal is +220, then +215 will be reported as the actual instead of -145, since +215 is closer to +220 than -145.

To achieve a result of +145° or -215°, the lines would have to be chosen in the opposite order. Use the "swap" button (<=>) to exchange the lines.

If you were interested in the "narrower" angles as opposed to the "wider" ones, the nominals of one of the lines (not both) could be defined so that the line pointed in the other direction.

Alternatively, the nominal angle may be entered. VMS will check the complement of the angle (what you would get if one line pointed the other way) and its synonyms as well, and pick whichever one comes within 90° of the nominal.

When the angle is near +/-90° the actual angle will come out close to the nominal despite the order in which the lines were chosen. A general rule to remember here is:

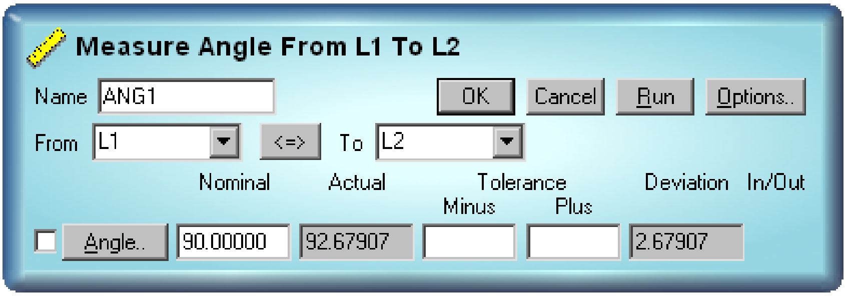
The angle will get larger when the first line tilts more clockwise.

The angle will get larger when the second line tilts more counterclockwise.

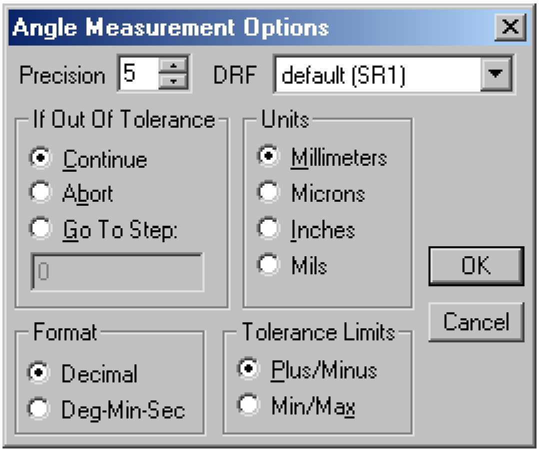
Measuring Angles

To measure the Angle between two Lines:

1. Construct or measure the Features that define the Angle.
2. Select the **Angle** button from the toolbar. The *Measure Angle* dialog box is displayed. Input the desired inspection parameters.



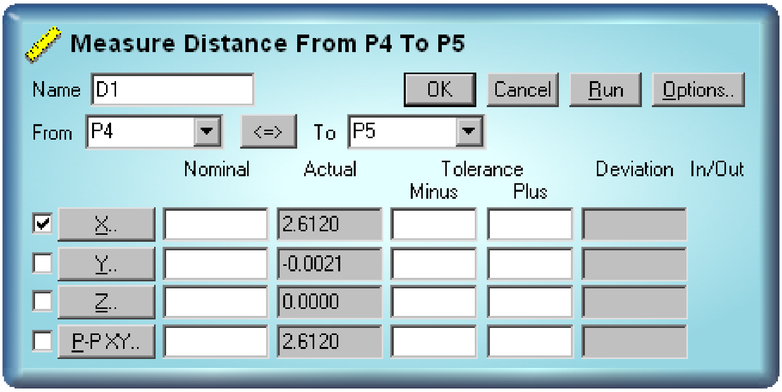
3. Click on the **Options** button to specify additional parameters.



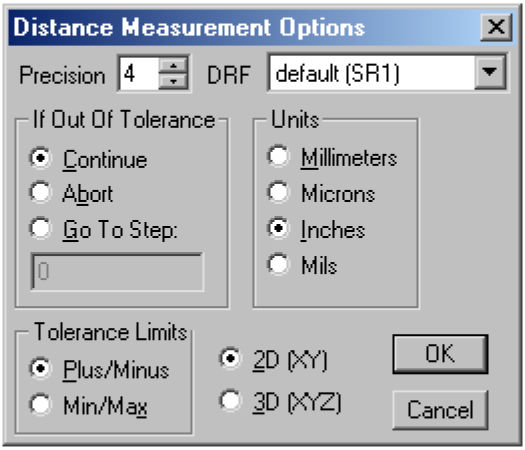
4. Select the two Features to be used in the angle measurement from the drop-down lists in the dialog box, from the *Features* window, or from the *Object Names* dialog box.
5. Click **OK** to add the step to the program.

Measuring Distance between Two Features

1. Define the Features between which you will be measuring the distance.
2. Click the **Distance** button in the toolbar. The *Measure Distance* dialog box is displayed. The Features to be used may be selected from the drop-down lists, from the *Features* window, or from the *Object Names* dialog box. Enter the desired inspection parameters.



3. Click on the **Options** button to specify additional parameters.



- 2D distance is the XY distance (square root of the sum of the squares of dX and dY.)
- 3D distance is the XYZ distance (square root of the sum of the squares of dX, dY, and dZ).

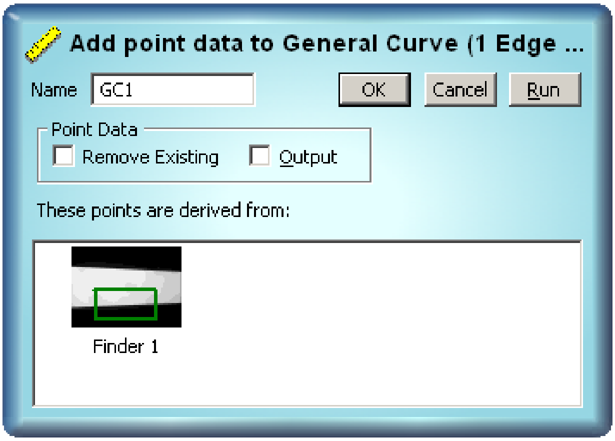
4. Select the two Features to be used in the distance measurement from the drop-down lists, the *Features* window, or the *Object Names* dialog box. VMS will calculate the distance between the two Features.
5. Click **OK** in the *Measure Distance* dialog box to accept the measurement.

General Curve

A General Curve Feature contains point data measured with one or more Finders using an Add/Remove Point Data step. After point data has been collected in a General Curve, Lines, Circles, or any other Features for which best-fit constructions are defined, may be constructed from the General Curve.

Using General Curve

A *General Curve* is a generic feature type for collection of point data. It is created by selecting **Add/Remove Point Data** from the **Measure** menu and adding finders to the step as with any Feature. After the point data has been collected in a General Curve, Features such as Lines, Arcs, etc. may be constructed from the General Curve. An example of adding point data from a Line Finder to a General Curve is shown below.



A *General Curve* may be used when more than one measurement of a Feature is desired. For example, to measure an inscribed circle and a circumscribed circle for a hole, collect the Finder data for the hole in a General Curve and construct the two types of circles from the General Curve:

Add point data to General Curve 'GC1'

Construct Best Fit Circle 'C1' Using GC1

Construct Best Fit Circle 'C2' Using GC1

Note the image in the example above. Suppose both lines are to be measured at several stage positions and both lines are in the field of view at these positions. Instead of measuring both lines separately, it will be more efficient to create two General Curves, one containing the data for the upper line and the other containing the data for lower line. At each stage position, create two steps: one that runs a finder on the lower line and adds the point data to GC1, and another that runs a finder on the upper line and adds the point data to GC2. After all finders are run, create two Line construction steps, one using GC1 and the other using GC2:

Clear General Curve 'GC1'

Clear General Curve 'GC2'

Step & Repeat Begin 'SR1'

Add point data to General Curve 'GC1'

Add point data to General Curve 'GC2'

Step & Repeat End

Construct Best Fit Line 'L1' Using GC1

Construct Best Fit Line 'L2' Using GC2

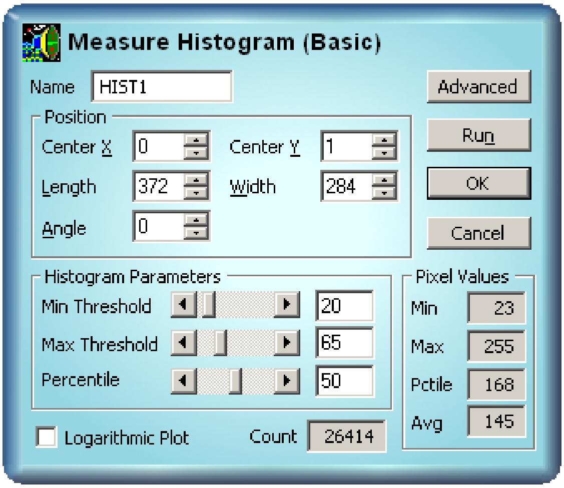
Removing Point Data

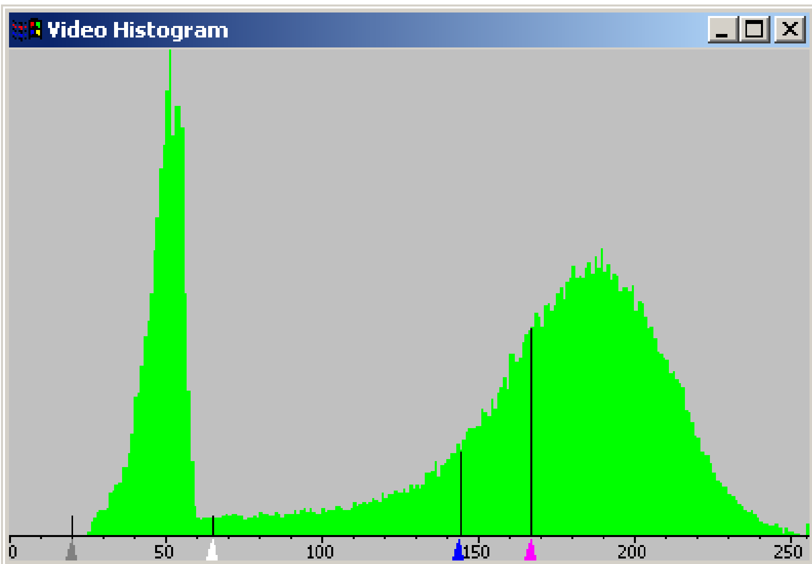
The **Point Data "Keep"** measurement option that may be specified for any feature is always in effect for a General Curve. Later in the program when the point data for a General Curve (or for any other feature that used the **Point Data "Keep"** option) is no longer needed, the point data may be removed to free the associated memory. To do this, select **Add/Remove Point Data** from the **Measure** menu, type the name of the Feature, select **Remove Existing** under Point Data, and click **OK**.

Histograms

Histogram Measurement calls up a dialog box to create a special step containing a single Histogram Finder that it runs as part of that step. The finder results are put into a Histogram structure variable, e.g., HIST1. This variable is similar to Feature variables, but there are no nominals or tolerances associated with the step, no pass/fail judgments are made, and no results are output. A histogram measurement is often used to determine whether or not a particular Feature exists in the field of view before performing other measurements. VMS tallies up how many pixels there are of each brightness value from 0 to 255. In this case, it will actually draw the resulting histogram in the dialog box. You give the upper and lower limits (thresholds) for the absolute brightness to consider. That way VMS will ignore the darkest and brightest parts of the image. Pixels that fall within this range will appear blue in the *Video* window on older systems with ISA imaging hardware (listed during installation).

On current systems, the pixels in the threshold range are drawn in pseudo-color, while the rest of the image is slightly dimmed. The pseudo-color shows a blue color for darker pixels in the range, transitioning through the spectrum to red for the lighter pixels in the range. Each time the Histogram Finder is re-run, the Histogram is redrawn and lines indicate where the thresholds have been set. To make small peaks more visible, the intensities may be plotted logarithmically by selecting the **Logarithmic Plot** checkbox.





Histogram Inputs

Input Parameter	Description
Min Threshold	The minimum pixel intensity to count when counting pixels.
Max Threshold	The maximum pixel intensity to count when counting pixels.
Percentile	The percentile (0-99%) for which a pixel intensity value is to be computed.

Histogram Outputs

Output Parameter	Description
Count	The number of pixels in the Finder area whose intensity was at least that specified by Min Threshold but not greater than that specified by Max Threshold.
Min	The minimum pixel intensity of all pixels in the Finder area.
Max	The maximum pixel intensity of all pixels in the Finder area.
Avg	The average pixel intensity of all pixels in the Finder area.
Nth Percentile	The intensity that “N%” of the pixels in the Finder area is at or below.

VMS finds the pixel intensity of a given percentile of the pixels in the Finder area. For example, if you specify 25, it will find the point on the histogram where 25% of the pixels (25% of the area under the histogram curve) are darker than the other 75%. This brightness value is reported in the "n Pctile" field in the dialog box, and is shown by a blue line on the histogram.

The Histogram also reports the "Count" of how many pixels fell within the limits, what the Min and Max brightnesses were within the limits (which will often be equal to the limits), and the average brightness. These results are then available in VMS as Histogram Variable Structure Components, for example: HIST1.PTILE, HIST1.COUNT, HIST1.MIN, HIST1.MAX, HIST1.AVG. The complete histogram array is available as HIST1.HBUF, an array of floating point values.

About Blob Analysis

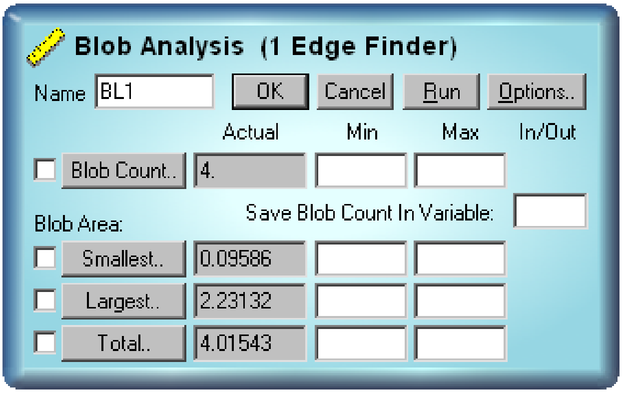
Blob analysis is a common technique in image processing. The basic operation consists of finding a threshold brightness and classifying each pixel as black or white, then identifying the "blobs" of white pixels against the black background, or black blobs on a white background (user selects the polarity). A "blob" is an isolated group of pixels of the selected color that touch each other, surrounded by pixels of the other color. The process of classifying the pixels into groups is also called Segmentation.

Various kinds of information about the blobs can be determined, and these can then be used for a variety of purposes. Blob analysis does not have a specific, narrow purpose; rather, it is a tool that may be very useful in various situations, in a very wide variety of ways. Blob Analysis is useful in, but not limited to, the following areas:

- **Flaw detection** - detecting defects (scratches, cracks, dust chips) in otherwise clear areas.
- **Counting randomly placed objects** - as in a container where a certain number of objects are supposed to be present.
- **Categorizing objects** - once the blobs are obtained, the attributes of the blobs can be used to determine whether it is a square, circle, scratch, hair, dust, etc.

Blob Analysis Step

A Blob Analysis step extracts the results from one or more Blob Finders into an array of Blob Features. Each Blob Feature contains the position, area, and other information about one of the blobs located. There may be one or several Blob Finders associated with each Blob Analysis step, and Blob Finders cannot be used in other types of steps. The *Blob Analysis* dialog box is shown below.



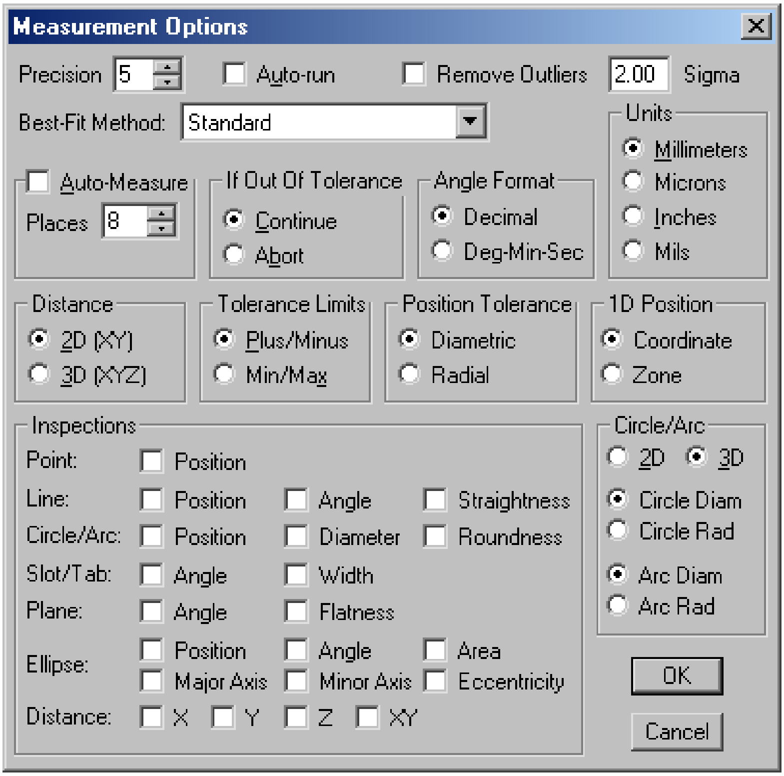
The Blob Analysis step will make up a name for, or allows you to name, a Blob array. It must be an array, not a single Blob, because the Blob Finder returns a list of Blobs that it finds, not just one. If the software makes up a Blob array, it will be dimensioned to hold 100 Blobs. If it is declared the dimension can be as large or as small as desired. In the Blob Analysis step, an integer variable can be specified that will be set to receive the Blob count (the number of Blobs found).

Each Blob Feature in the array has the following structure components:

FLAGS	Integer	Various bit flags, depends on type; not for normal use
NPTS	Integer	Number of edge points in the following buffer
PBUF	Simple Point[]	Array of Simple Points containing all of the points around the edge of the blob, in MCS.
AREA	Float	Area of the blob in square millimeters
POS	Simple Point	X, Y, Z coordinates (mm) of the centroid of the blob
NPIX	Float	Area of the blob in pixels

Default Measurement Options

The *Default Measurement Options* dialog box is shown below. It is accessed from the **Setup/Options/Measurement** menu.

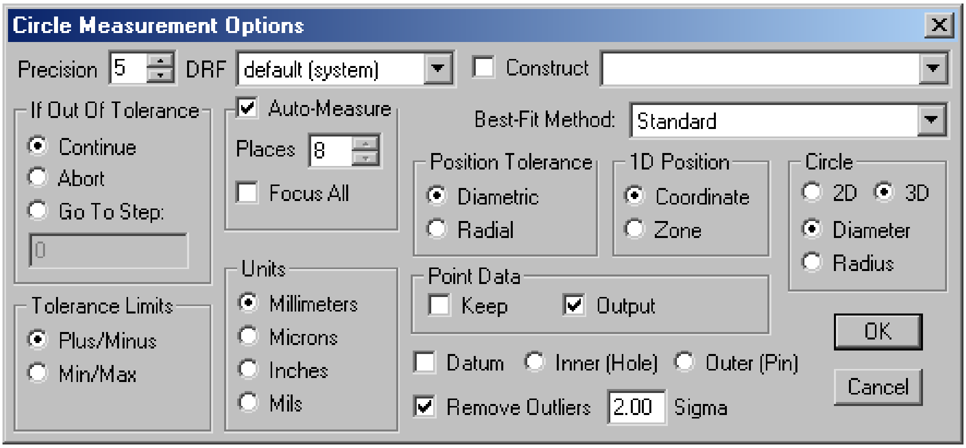


- **Precision** specifies the default number of digits to the right of the decimal place you want in measurements. VMS will round values to the number of digits you specify.
- **Auto-run** specifies that the step should be rerun and results recomputed when a Finder is added, removed, or edited in a measurement step.
- **Remove Outliers** specifies default removal of point data that is outside the specified limit.
- The **Best-Fit Method** list allows selection of the default fit method from the fit methods allowed for all Feature types.

- The Auto-Measure section allows automatic finder placement for measuring a Point, Line, Arc or Circle Feature.
- If a property of the Feature is **Out of Tolerance**, you can choose to have the program, by default, **Continue** or **Abort** on the offending step.
- **Angle Format** specifies a default of decimal or degrees-minutes-seconds for angle display and output.
- **Units** of millimeters, inches, microns, or mils may be selected as the default.
- **Distance** measurements may be either 2D (XY) or 3D (XYZ) by default.
- **Tolerance Limits** may be input and output as Plus/Minus or Min/Max. Note that because this may be changed for each step, the output headers will always display -Tol and +Tol even if Min/Max tolerances are used exclusively.
- **Position Tolerance** allows specification of the default tolerance zone as a diameter or as a radius.
- **1D Position** (e.g., X Position) may be reported either as the actual value (X) or as the size (diameter or radius) of the zone containing the nominal and actual positions.
- **Inspections** checkboxes specify which inspections will be performed by default for the indicated measurement types.
- **Circle/Arc** buttons specify whether Circle and Arc measurements are 2D or 3D by default, and whether a radius or a diameter inspection is the default for each.

Feature Options

A sample *Measurement Options* dialog box for a Feature is shown below.



Many of the options are described in Section 7.13.1, *Default Measurement Options*. The others are:

- The **DRF** list allows selection of an alignment or measurement DRF to create the Feature with respect to. The default is the immediately enclosing alignment DRF or Part (not Feature) Step & Repeat loop, if any.

- The **Point Data "Keep"** option stores points in a point buffer for later use with advanced programming techniques (leave this option off unless you need it; it uses a lot of memory).

- The **Point Data Output** option tells VMS to output the point data via DDE and/or COM to another program that accepts it as input. An excellent example of such a program is ICAMP®, which can perform Profile Analysis after doing a generalized translation/rotation determination that most closely matches the actual data points to a nominal profile from a CAD file. The Profile Analysis allows checking any 2D shape for conformance, without having to break it up into lines and arcs. It also handles spline curves. Details about the DDE and COM parameters are available upon request.

- The **Datum** box marks the Feature as a datum (reference feature) for display in datum lists. Any Feature may be used as a datum for measuring other Features (usually by typing its name instead of selecting from a list), but only Features marked as datums are displayed in datum lists. This makes it easier to find the desired datum Features in those lists.

Understanding Positional Tolerance

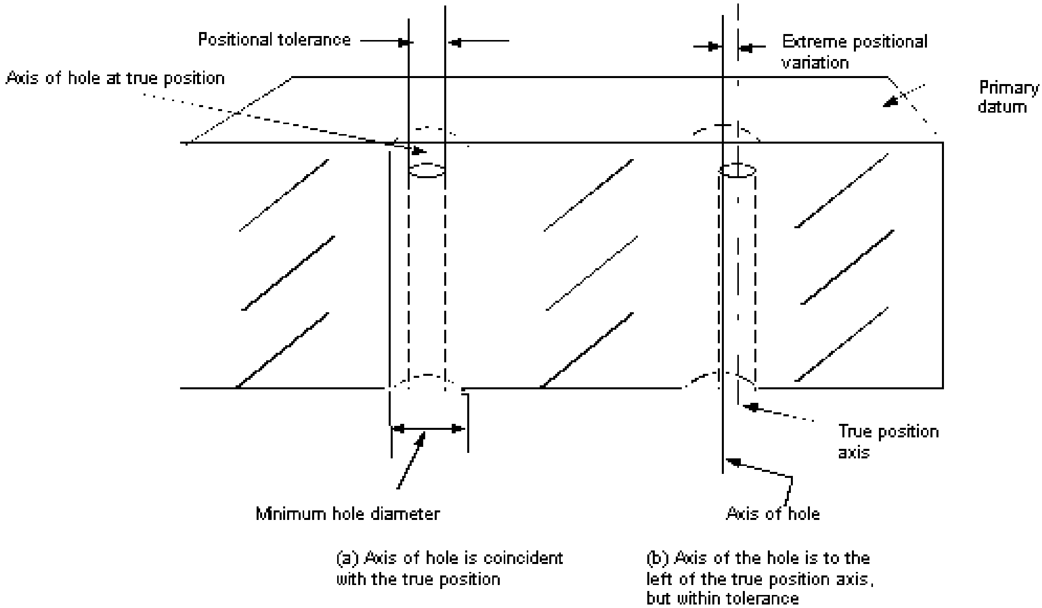
Positional tolerance defines a zone within which the center or center plane of a Feature is permitted to vary from true position. The true position is established from the basic dimensions of specified datum Features and between interrelated Features. A positional tolerance is indicated by the position symbol, a tolerance, and appropriate datum references placed in a Feature control frame.

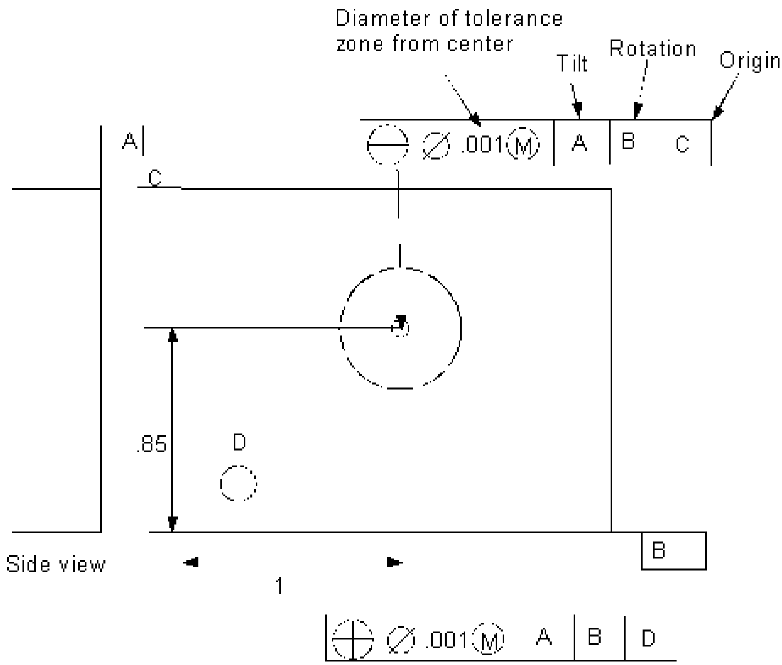
⌖	⌀ 0.4 Ⓜ	A	B	C

Positional tolerance is applied on either a Maximum Material Condition (MMC), Regardless of Feature Size (RFS) or Least Material Condition (LMC) basis. These are identified by the symbols M, S or L respectively. The appropriate symbol follows the specified tolerance and applicable datum reference in the Feature control frame.

MMC means a Feature of a finished product contains the maximum amount of material permitted by the toleranced size dimension for that Feature. For holes, slots and other internal Features, MMC is where these Features are at their minimum allowable sizes. For shafts, lugs, tabs and other external Features, MMC is where these Features are at their maximum allowable sizes. It is only when the Feature is at MMC that the specified positional tolerance applies.

Examples of Positional Tolerancing:





The MMC is equal to the nominal diameter value minus the tolerance. If the size of the hole is greater than the MMC or the shaft is smaller than the MMC the additional room is called bonus tolerance. The bonus tolerance increases the tolerance zone. Therefore, a part with bonus tolerance will still be within specifications.

In order to measure a Feature's position, you need to define a DRF and a nominal position with respect to that DRF. Often, several DRFs are needed for a part with many of them different in origin only, that is, their orientations use the same Datums, but their origins use different Datums. As a shortcut, you do not need to construct a full DRF for each one of these. In each *Feature Measurement* dialog box, you must specify a DRF, even if it is just the default one. But you might also specify one Datum as the origin. This creates a temporary DRF for this step. You give the nominal positions with respect to the DRF origin, or the Datum if you named one. (VMS versions prior to 3.0 required the nominals to be with respect to the DRF, even if you named a positional Datum.)

The named positional Datum can also contribute a bonus tolerance to the position evaluation if all of the following are true:

- it is a "Feature of size"
- it was properly measured before (including nominals and you specified "inner" or "outer" for it)
- you select MMC or LMC where you name it as the positional Datum for the measurement

Settings in the *Default Measurement Options* dialog box are used to handle certain aspects of position measurements.

When printing the results of a 1D position measurement, if you want to the Actual to be shown as a deviation amount (which will always be positive), select **Zone** for 1D Position measurements. If you want the Actual to be shown as the actual coordinate, select **Coordinate** (this is the default.) The former is consistent with how 2D positions are reported, but the latter is more informative.

Note that many Features' position measurements have separate 1D position measurements available in addition to 2D position measurements.

The Position Tolerance mode always affects 2D position measurements. It only affects 1D position measurements if "Zone" mode is selected. The size of a position tolerance zone is always given, by the user, as a diameter (or width, for 1D). For example, if the position can be off by 0.002 either way, we give a zone size of 0.004. If the position is off by 0.003 it fails, but in the default "Diametric" mode we print that deviation as 0.006 so it can be easily compared to the 0.004. However, if desired, the Position Tolerance mode can be set to "Radial" to show the real deviation amount (0.003 in this example). This obviously is not applicable to 1D position measurements in coordinate mode.

For positional measurements, two lines or slots can be used as separate position and rotation datum. MMC/LMC bonus tolerancing can be obtained from the Feature being measured and from the Datum(s). When measuring the position of a point, circle or arc, you can use one linear Feature (slot/tab or line), or two non-parallel linear Features as datums, establishing a temporary DRF in a way similar to when a circle or point is chosen as a datum. (If two datums are used they must both be linear Features.)

With two linear Features, their intersection becomes the origin of the DRF, and the XY rotation of the DRF is set to match the first linear Feature selected. If the Feature being measured is a circle, the separate X and Y bonus tolerances gained from slot/tab datums may also be combined with a radial bonus tolerance from the circle itself, giving a rounded-rectangle shaped tolerance zone.

Auto-Measure

Auto-measure provides a method for using multiple Finders to measure a Feature without manually placing all of the finders. When auto-measure is used, the user places one or two Finders and they are run in multiple places by the measurement step.

Auto-measure is useful for measuring any circles, arcs, or lines that are uniform: i.e., that have no gaps that must be avoided when placing edge finders, and that do not require different Edge Finder parameters or light settings on different sections of the Feature.

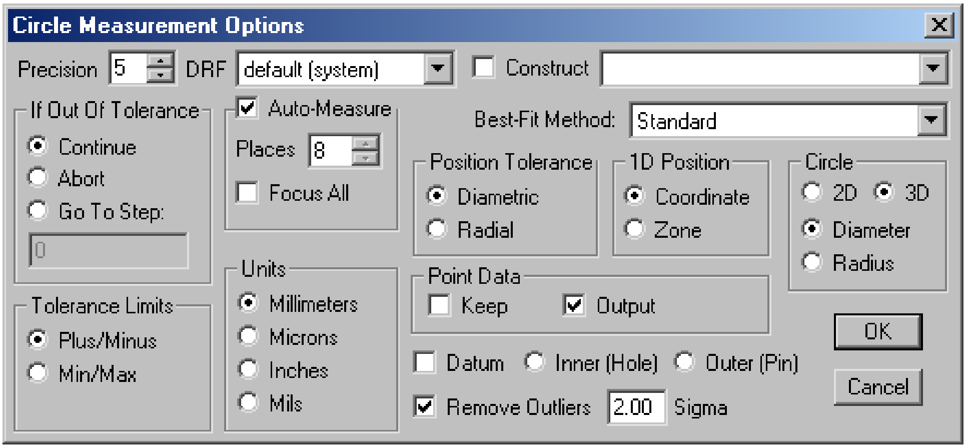
The Smart Finder may be used to generate a Finder for Auto-measure. This is especially useful for generating an Arc Finder with the correct radius. However, for Circle or Arc Auto-measure, the generated Finder must be manually reduced in size (after clicking off the "Smart Finder" button) because the tracer generates a Finder that is too large to rotate as required to run in different orientations.

Auto-measure may be used to quickly create measurements for Features imported from CAD data.

Enabling Auto-Measure

To enable auto-measure:

1. Click the **Options** button in the arc, circle, or line *Measurement* dialog box.



2. Select the **Auto-Measure** check box, and adjust the number of places where Edge Finders are to be run as desired (2 to 60 places may be specified.)
3. If **Auto-Measure** will be used for most circles, arcs, and lines, make it the default setting by selecting it in the *Default Measurement Options* dialog box.

Auto-Measure a Circle

To use Auto-Measure for a circle:

1. Enable Auto-Measure as described in Enabling Auto-Measure.
2. Run a Finder on the edge of the circle and add it to the step,
3. Click the **Run** button in the *Measurement* dialog box.

If the circle was imported from CAD data, its nominal position and diameter are known and the Auto-measure is run in the specified number of places. Otherwise, the first time the Auto-measure is run, it will be run in a "teach phase" that attempts to track the edge of the circle by running finders at close intervals. This usually works if the points located by the finder form a smooth edge, but may fail if they do not. If the edge is rough or the diameter is large (over 20 times the field of view), it may be necessary to adjust the Edge Finder parameters and/or enter the nominal diameter of the circle.

If a rectangular Finder is used for a circle that nearly fits in the field of view, the radius of the circle may be smaller than the width of the finder. In this case, the auto-measure teach phase will fail. A radius smaller than the Finder width typically happens when a circle fit is performed on a very rough edge, so VMS rejects the fit to prevent moving to incorrect locations. Either an Arc Finder or a smaller rectangular Finder should be used instead for smaller circles.

Auto-Measure an Arc

Auto-measure for Arcs is the same as for Circles (see *Auto-Measure a Circle*), except that the ends of the arc must be indicated. To do this:

Place two Finders, one at each end of the Arc, before running the Auto-measure teach phase.

> A single Finder may be used if the user enters the nominal start and end angle for the Arc or the Arc was imported from CAD data. Because the teach phase proceeds from the Finder location to the more distant end of the Arc, it is best to place the Finder near one end when nominal angles are entered.

Auto-Measure a Line

To use Auto-measure for a Line, the ends of the Line must be indicated. To do this:

1. Place two Finders, one at each end of the Line, before running the Auto-Measure. A single Finder may be used if the user enters the nominal start and end point for the Line or the Line was imported from CAD data.
2. The first Finder may contain an Autofocus addition so that VMS will focus on the edge of the Feature before performing the Auto-measure.
3. If high magnification is used and the height of the Feature is not uniform, it may also be necessary to check the Auto-measure **Focus All** option, so that VMS will Autofocus before running the Finder at each location. This is very slow, and should be avoided if possible.

This page was left blank intentionally.

Section 7

Finders

Finders are VMS's on-screen tools, used to collect point data for measuring Features on parts. Select a Finder in the *Video* window that is similar to the shape of the Feature you want to measure. For example, select a Line Finder to define straight edges or an Arc Finder to define a curve. A Finder returns (finds) one or more points from a Feature that you are teaching VMS about. These returned points help you measure parts in the measurement programs you create.

You can rotate, resize and move Finders to help you define a Feature. The flexibility and accuracy of Finders is an important part of VMS' accurate measurement capability.

Finders are selected from the toolbar in the *Video* window or by clicking on the right mouse button in the *Video* window. The Line Finder is probably the most commonly used Finder, although they are all useful. The Finders you use in the measurement programs you create are based on the Features you are measuring and the parts you wish to inspect.

When you program a Finder, the Finder is saved as a collection of numeric values taken from how you had everything set up: the Finder type, the position and angle relative to the current FCS , the height and width, Autofocus parameters, the light levels, and image processing parameters. These are all saved as numbers that do not change. The only way to set these parameters is by editing the Finder, which must include properly establishing the FCS (alignment) then actually moving the stage to the right place.

You can supply expressions for those values. The expressions can be simple numbers but they can also be variables, or any mathematical expression. So, for example, the measurement program could open a disk file, and based on information in it, Finders could be positioned and adjusted differently for different parts. Another possible use might be to adjust all light levels by a common factor.

To do this, click on the **Advanced** button in the *Finder Editing* dialog box. This brings up a dialog box containing fields for entering the expressions. Related parameters are grouped together.

When the program is run, the expressions will be evaluated before the Finder is run, and their values will *replace* the constant values that were derived from how the Finder was programmed in the normal way. That is, even when you use expressions, **the Finder still contains** the stage position, light settings, Finder size and angle, etc. that you set up when you programmed it. The programmed values can now be replaced at run time. But even the values that get replaced can be very useful.

The expressions can raise a predicament if you forget to execute the steps that set variables used in the expressions. The expressions of particular concern are those that specify the Finder's stage position and/or PRL position. The stage and/or PRL could move to the wrong place, possibly causing a collision. To help prevent this problem, all expressions are *disabled* (the Finders use the constant values that they were programmed with) whenever a Finder is test run and whenever a step is test run from the dialog box. To test the effect of the expressions on the Finders, the step must be added to the program, then run (e.g., using "single step"). Before doing this, you need to be sure that all variables used by the Finders, *especially those that control the stage and PRL position*, contain correct values for the current part. Also remember that before you create or edit a measurement step, VMS will execute all the Align Define sections affecting that step. That means it will run the Finders that measure the Datums. If any of those Finders have expressions, they may work incorrectly (i.e., crash) if the expressions do not have the correct values for the part.

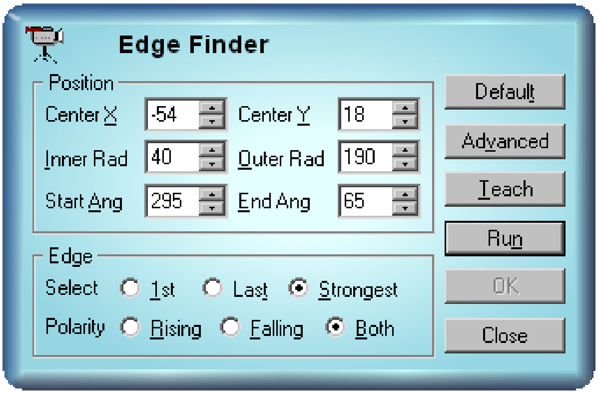
To ensure that variables have correct values, make sure you have the right data file(s) or other inputs that go with the part. Also, it is a good idea to make sure that *all* of the steps that contribute to Finder parameter expressions (opening, reading and closing files, getting user inputs, doing computations and assignments) are located within an Align Define section that defines the alignment context for Finders that use those variables.

Finder parameter expressions allow variables or expressions to be entered for the parameters of non-dynamic Edge Finders and Focus Finders only. Dynamic Edge Finders (i.e., the old kind), Histogram Finders, and Centroid Finders will not allow expressions for parameters at this time.

The parameters are divided into groups containing several items in each group. Entering a variable or expression for any item in a group causes expressions to be generated for all items in that group. These generated expressions are simply the current constant values for those items. (For example, if you enter something for the X stage position, the Y and Z positions will be filled in with the current FCS-relative stage position.) You can then change these to something else if you want. If a whole group is left blank, the constant values of those items will be used without modification.

Basic Edge Finder Parameters

To view the parameters for a finder in the *Video* window, right-click in the *Video* window and select **Edit the Finder Parameters** or click the **Edit Finder Parameters** button on the *Video* Window Toolbar. An example of a basic *Edge Finder Parameters* dialog box is shown below. This example is intended to show the appearance of the parameter controls; the dialog box for a particular edge finder may differ slightly.



Basic Edge Finder dialog boxes may contain the following parameters. Other parameters unique to a particular Finder are described in the section defining that Finder.

Position Parameter	Description
Center X	The Center X parameter specifies the X location of the center handle in the Finder with respect to the center of the field of view (FOV). Zero represents the center of the *Video* window. Positive numbers move the center handle and the Finder to the right; negative numbers move the center handle to the left.
Center Y	The Center Y parameter specifies the Y location of the center handle in the Finder with respect to the center of the field of view (FOV). Zero represents the center of the *Video* window. Positive numbers move the center handle and the Finder up; negative numbers move the center handle down.

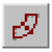
Position Parameter	Description
Length	The length parameter specifies the size in scan direction, perpendicular to the edge. Making this bigger allows for more variation in the edge location. The arrows in the Finder point to the relevant edges. For the Centroid Finder, which does not have arrows, the length is measured from the left edge to the right edge based on the original orientation of the Finder.
Width	The width parameter specifies the size perpendicular to the scan direction, parallel to the edge. Making this bigger gets more of the edge in the Finder. For the Centroid Finder, which does not have arrows, the width is measured from the top edge to the bottom edge based on the original orientation of the Finder.
Angle	The angle parameter specifies the angle of rotation of the Finder in degrees. 0 represents no rotation. All other numbers up to 359 degrees rotate the Finder counterclockwise.
Inner Radius	The inner radius parameter specifies the radius of the inner circle of the Arc and Circle Finders.
Outer Radius	The outer radius parameter specifies the radius of the outer circle of the Arc and Circle Finders.
Start Angle	The start angle parameter specifies the angle of rotation of the first edge of the Arc Finder, looking counterclockwise from 0 degrees rotation.
End Angle	The end angle parameter specifies the angle of rotation of the second edge of the Arc Finder, looking counterclockwise from 0 degrees rotation.
Select	The first, last, or strongest edge in the finder area may be selected for detection.
Polarity	A **Rising** edge (dark-to-light) only, a **Falling** edge (light-to-dark) only, or **Both** rising and falling edges in the finder area may be selected for detection.

Teaching Edge Finder Parameters

There is a **Teach** button for automatic Finder parameter optimization in Line, Circle, and Arc Finder parameters dialog boxes. To “teach” finder parameters, position the center “handle” of the Finder close the edge to be detected and press the “Teach” button in the Finder parameters dialog box. For Line Finder teaching, the arrows of the Finder need to be placed perpendicular to and toward the edge to be taught. For Circle and Arc Finders, the radii of the Finder need to be adjusted so that the inner radius and outer radius are equally distant from the edge. For Arc Finders, the curvature of the Finder must match the curvature of the edge. In simpler words, make the Finder shape symmetrically match the edge as closely as possible.

The Advanced parameters that are automatically set by this are Form, Sharpness, Total #, Selected, Strength, Contrast, Polarity and Outliers. (If the *Basic* dialog box is used, these parameters are still set even though you can’t see them.) These parameters, once set, will be saved with the Finder as if you had set them manually.

This function assumes there is not another edge within 20 pixels of the edge you are working with. If there are other edges closer, it may not work correctly.

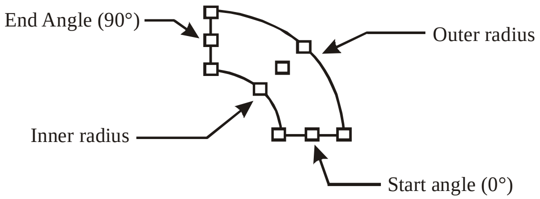
Arc Finder 

The Arc Finder defines all points on a radius. It contains an Inner and Outer radius and Start and End angles.

Position the Arc Finder in the *Video* window so that it matches the curve of the Feature your are measuring.

If you are having difficulty adjusting the Arc Finder, try using the Smart Finder tool in the *Video* window to position the Arc Finder for you. Double-click on any open space in the *Video* window to run the Finder and find the edge.

Arc Finder

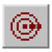


To manipulate the Arc Finder's position:

- Rotate the Finder by clicking the right mouse button on the center or center side handles, rather than any outside handle.
- Adjust the radius of the Finder by clicking the right mouse button on any outside handle.
- The scan direction of the Arc Finder is always from the inside to the outside.
- You can also manipulate the Arc Finder by clicking on the **Edit Finder Parameters** button in the *Video* window Toolbar. **Position** parameters specific to the Arc Finder are:

Parameter	Description
Inner Radius	The inner radius parameter specifies the radius of the inner circle of the Arc and Circle Finders.

Parameter	Description
Outer Radius	The outer radius parameter specifies the radius of the outer circle of the Arc and Circle Finders.
Start Angle	The start angle parameter specifies the angle of rotation of the first edge of the Arc Finder, looking counterclockwise from 0 degrees rotation.
End Angle	The end angle parameter specifies the angle of rotation of the second edge of the Arc Finder, looking counterclockwise from 0 degrees rotation.

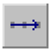
Circle Finder 

The Circle Finder defines all of the points on a circle. The entire Circle being measured must fit within the *Video* window, if it does not, use the Arc Finder. Typical Features that the Circle Finder measures best are pins or holes. To use the Circle Finder, place the inner circle of the Finder well inside of the Feature, and then encircle the Feature with the outer circle of the Finder. The scan direction of the Circle Finder is from the inside to the outside.

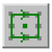
Click and hold the Circle Finders handles to enlarge or reduce it. Click and drag the center handle to move the Finder. Adjust the Circle Finder's parameters by clicking on the **Edit Finder Parameters button** in the *Video* window toolbar.

Position parameters specific to the Circle Finder are:

Parameter	Description
Inner Radius	The inner radius parameter specifies the radius of the inner circle.
Outer Radius	The outer radius parameter specifies the radius of the outer circle.

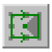
Point Finder 

This Finder defines a single point. Place the point handle over the image in the *Video* window so that the middle handle is on the edge. The arrow should point from the "clean" surface to the "dirty" or "noisy" surface. Use the handles on the ends of the Finder to rotate it. To adjust Point Finder parameters, click on the **Edit Finder Parameters Button** in the *Video* window toolbar.

Line Finder 

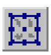
The Line Finder is used for finding multiple points along a straight or slightly curved edge. Position the center point of the finder on the edge of the Feature being measured. The arrows should point from the "clean" surface to the "dirty" or "noisy" surface.

Use the handles to resize, rotate and reposition the Finder in the *Video* window. To adjust Line Finder parameters, click on the **Edit Finder Parameters** button in the *Video* window toolbar.

Corner Finder 

The Corner Finder defines a minimum or maximum point on an edge in the *Video* window. This Finder is useful if you are interested in the location of a corner or tip, but do not need to measure the lines that make up the sides of it.

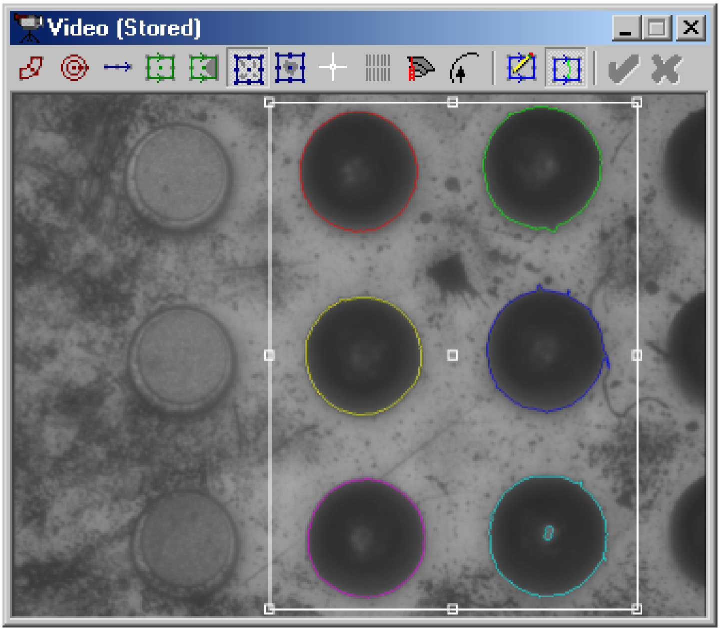
Use the handles to resize, rotate and reposition the Finder. To adjust Corner Finder parameters, click on the **Edit Finder Parameters** button on the *Video* window toolbar.

Blob Finder 

The Blob Finder locates regular or irregular shapes in the Finder area by finding connected sets of pixels within a given brightness range, i.e., forming contiguous blocks of pixels. It returns the number of Blobs found in the Finder area, and for each Blob, the centroid point and the number of pixels inside (i.e., its area). Blobs whose areas are outside the minimum and maximum specified by the finder are ignored.

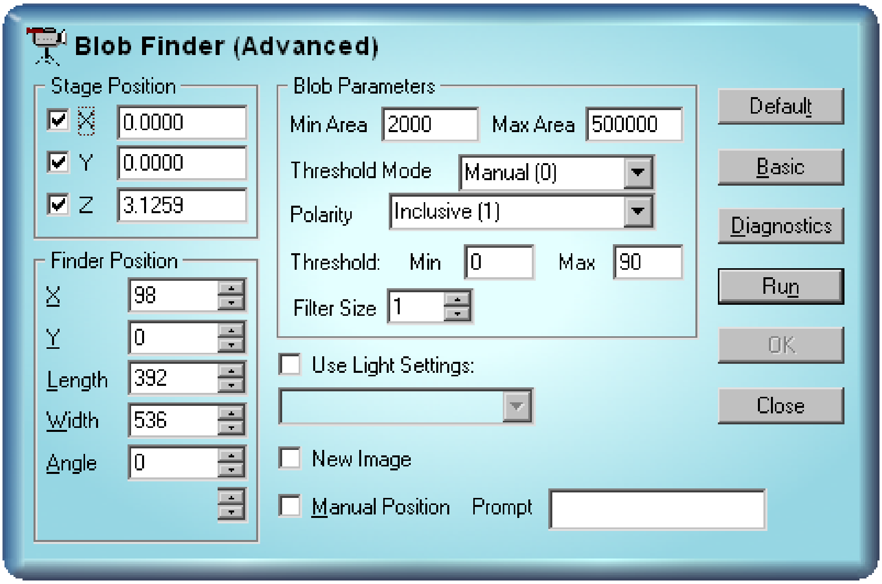
Unlike Edge Finders, the Blob Finder returns an array of blob results. Each blob result contains edge points similar to those returned by an Edge Finder. Because the Blob Finder returns more than one Feature result, it requires a special Blob Analysis Step that extracts the results from a Blob Finder into an array of Blob Features. Each Blob Feature contains the position, area, and other information about one of the blobs located. There may be one or several Blob Finders associated with each Blob Analysis step, and Blob Finders cannot be used in other types of steps. The Blob Finder is accessed from the *Video* window toolbar.

Running a blob finder produces a *Video* window display like the following:



Blob Finder Parameters

The *Blob Finder Parameters* dialog box is shown below.



Parameters specific to the Blob Finder are shown below. The **Min Area**, **Threshold Mode**, and **Polarity** are most often adjusted to obtain the desired results. Parameters specific to the selected threshold mode are described in the section for that mode.

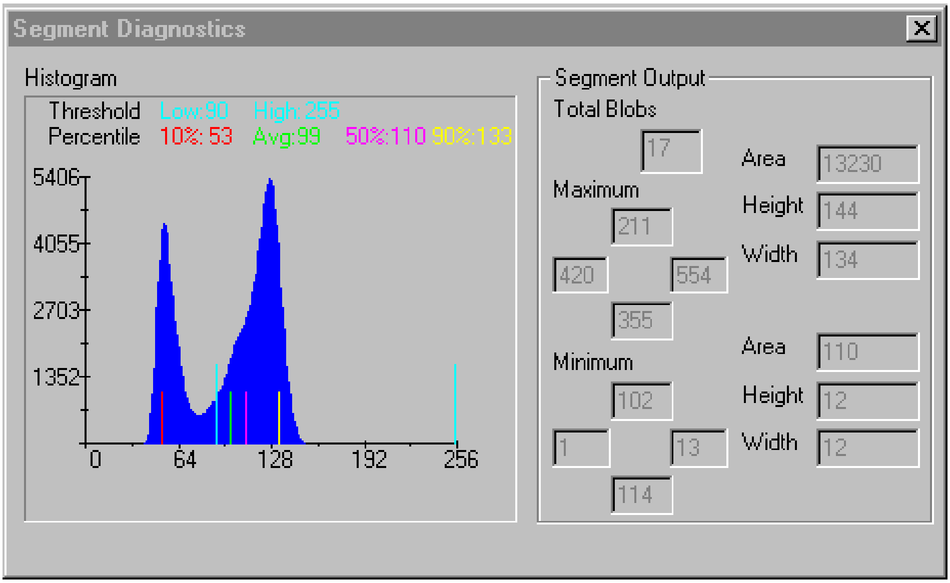
Parameter	Description
Min Area (pixels)	Removes blobs smaller than this from the results.
Max Area (pixels)	Removes blobs larger than this size from the results.
Threshold Mode	Select Automatic Threshold Mode or Manual Threshold Mode.
Filter Size	Number of pixels to average together at each point in the image (see note below).

Sometimes the view of a scene will contain lots of little details or grain that will be interpreted as small blobs. The Blob Finder can ignore these by using the **Min Area** parameter, but that is only applied to kick out the small ones after all blobs have been found. When there are many blobs, it causes the blob algorithm to take more time, sometimes very noticeably. A different way to ignore speckles or grain is to smooth or blur the image before doing the blob analysis. This might help in other ways too; for example, a blob that had a jagged, irregular edge would have a smoother edge.

The **Filter Size** parameter tells how many pixels to average together at each point. For example, if it is set to 3, then each pixel is averaged with all the pixels within a 3 pixel radius. You will not see the smoothed image on the screen, but you can see the effect it has on the results.

Automatic Threshold Mode

The Automatic threshold mode analyzes the selected area of the image and chooses a threshold based on the histogram. This mode works best when the objects to be identified are shaded consistently different and distinct from a relatively uniform background.



The picture on the previous page is from the *Segment (Blob) Diagnostic* window illustrates a Histogram that would likely work well in Automatic Mode. The threshold would be chosen between the two peaks.

Parameter	Description
Polarity	Select **Light on Dark** if the brighter peak in the histogram represents the blobs. Select **Dark on Light** if the darker peak in the histogram represents the blobs.

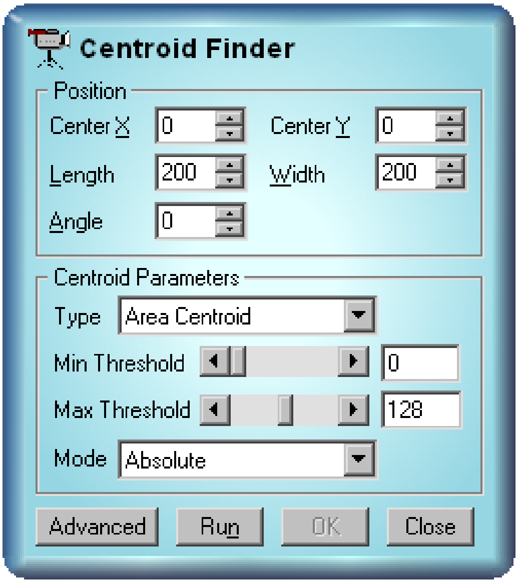
Manual Threshold Mode

In manual mode, simply specify a range of two absolute brightness levels (thresholds) between 0 and 255. See the picture in *Automatic Threshold Mode*.

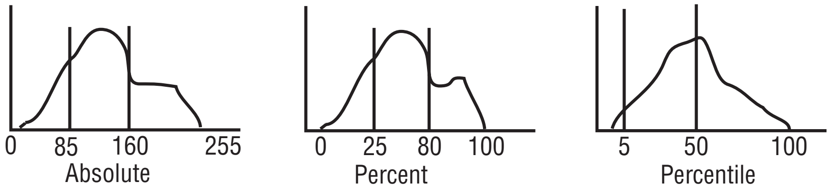
Parameter	Description
Polarity	If **Inclusive** is selected, the Blobs will be the areas of pixels that fall between the two brightness thresholds. If **Exclusive** is selected the Blobs will be the areas of pixels that do *not* fall between the two brightness thresholds.
Min Threshold	Minimum brightness value for the range.
Max Threshold	Maximum brightness value for the range.

Centroid Finder

The Centroid Finder gives back a single point that is at the centroid (similar to center of gravity) of all the pixels that fall within a given brightness range. For example, it could find the location of a blob of light gray paint on a dark gray background, or the location of a white stamped marking on black plastic. Position the Finder in the desired location. Specify which brightness range to use as part of the Feature, by giving upper and lower limits (thresholds) in one of several modes.



In the following graphs, the horizontal axis is brightness and the vertical axis is how many pixels in the Finder area had that brightness. Remember, the result of the Finder is the centroid of all the pixels whose brightness falls between the two thresholds.



Graphs Under the Centroid Finder

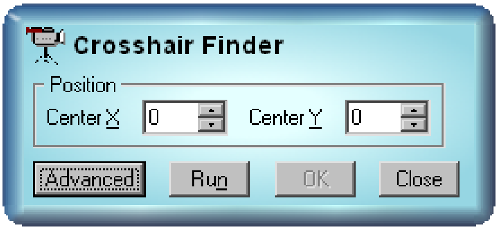
Mode	Description	Use When
Absolute	You give the threshold numbers as absolute brightness values between 0 (darkest) and 255 (brightest). This may be useful in some situations, but the absolute brightness levels can be affected by changes in the light bulbs' brightness and by the presence of different things in the camera's field of view, which will affect its Automatic Gain Control (AGC).	Absolute mode works best if there are any tiny specs of high-contrast brightnesses (relative to the brightness of the main areas of interest).
Percent	You give the threshold numbers as a percentage between the darkest and lightest brightness values seen in the area. Thus if the whole picture gets brighter or darker, this mode will continue to pick out the same pixels. However, "darkest" and "lightest" can be determined by even one stray pixel that is unusually dark or bright, so even this may have some problems in some situations.	Use when there is a variation in the presence or absence of significant areas of contrasting color.
Percentile	This is most easily explained by the graph. The lower limit of 5% means that the darkest 5% of the pixels are excluded. The upper limit of 50% means that the brightest 50% of the pixels are excluded. Note that the area of the "brightest" region is 50% of the total area under the curve, yet because the curve is not symmetrical, the 50th percentile line is not exactly halfway on the horizontal axis.	This is the most useful mode in all circumstances, except when there is a variation in the presence or absence of significant areas of contrasting color, in which case the percent or absolute modes would work better.

Centroid Types are:

- **Area Centroid** takes into account internal pockets of brightness ("holes" or "flaws") not within the range.
- **Outline Centroid** ignores internal "holes" or "flaws", as if the outermost outline of the areas within the range were completely filled.

Crosshair Finder 

The Crosshair Finder defines a single point on an edge. The Crosshair Finder does not find the edge using image processing, it only allows the operator to locate a point in the *Video* window manually. It is most useful when identifying changes in texture rather than brightness or items that because of loose manufacturing or fixturing tolerances do not always fall within the FOV.



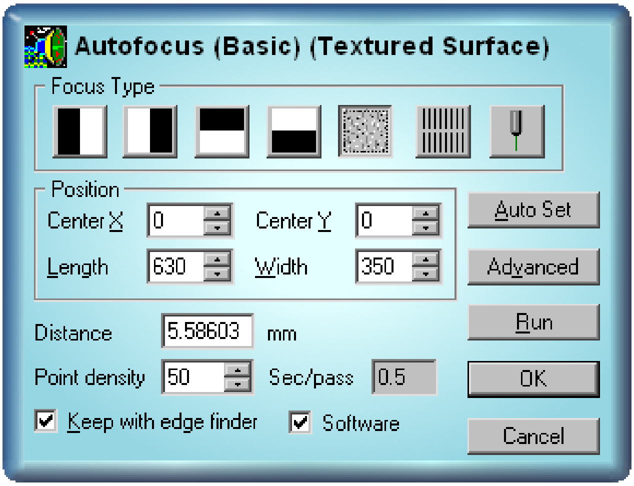
Position	Description
Center X	The Center X parameter specifies the X location of the center handle in the Finder. 0 represents the center of the *Video* window (the field of view). Positive numbers move the center handle (and the entire Finder) to the right; negative numbers move the center handle to the left.
Center Y	The Center Y parameter specifies the Y location of the center handle in the Finder. 0 represents the center of the *Video* window (the field of view). Positive numbers move the center handle (and the entire Finder) up; negative numbers move the center handle down.

Because the Crosshair Finder is not an Edge Finder, there are no edge parameters.

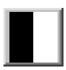
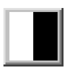
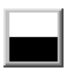
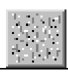
Autofocus Finder 

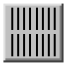
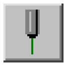
Autofocus is a capability in which VMS automatically finds the optimal focus point (the Z point) for the part in the *Video* window. To accomplish this, VMS moves the Z-axis through the approximate point of focus and notes precisely where the best focus occurs.

1. Select Autofocus from the *Video* window. The Autofocus returns one or multiple points, with different XY coordinates taken from the main Finder and Z coordinates (all the same) taken from the Autofocus.



2. Select the Focus Type based on the image. Use the following table:

Button	Name	Application
	Vertical B/W	Back lit (silhouette) vertical edge, dark side on left.
	Vertical W/B	Back lit vertical edge, dark side on right.
	Horizontal W/B	Back lit horizontal edge, dark side on bottom.
	Textured	Top lit textured surface.

Button	Name	Application
	Ronchi Grid	(Optional) For surfaces with little or no texture, use the Ronchi grid. A grid pattern is projected through the lens onto the surface. This is good for highly polished surfaces, and also for ceramics.
	TTL Laser	(Optional) For fast and accurate focus within a limited range, an optional through-the-lens laser may be used.

3. Size the Autofocus Finder in the *Video* window. Use the large default size of the Autofocus Finder for focusing on a surface (Textured Surface or Ronchi Grid focus types). The following should be noted when using hardware autofocus:
   - For horizontal edge modes, the window is automatically set to be very narrow horizontally, so it intersects only a small portion of the edge.
   - For backlit edge modes, only the sharpest edge in each scan is considered.
   - For toplit modes, the overall sharpness of the entire area is considered, since it assumes you are focusing on a surface, not an edge.
4. Set the **Distance** parameter. Distance signifies the distance the lens moves along the Z-axis to find the optimal focus. VMS computes a default value for Distance. The default value may work fine. If it does not, you can increase or decrease the Distance to improve your results by changing the number in the Distance box. Before doing this, consider that the pass length must be long enough to make sure it includes all expected variations in the part or its placement. Increase the Distance for a greater capture range; decrease the Distance to focus more quickly.
5. Systems with newer (MuTech) imaging hardware are capable of using a more sophisticated software image analysis to obtain focus readings. This option (the **Software** item checked) is the default on such systems.
6. To return to VMS's default value for Distance, click the **Auto Set** button.
7. Autofocus establishes a Z coordinate for all points returned by the Finder in the *Video* window. Click the **Run** button in the *Autofocus* dialog box to test if Autofocus brings the part into sharp focus.

Edge Finder With Autofocus 

You can add an automatic focus to any Finder. It is remembered as an extra action associated with the Finder and it runs every time the Finder runs. To add Autofocus to a Finder you must:

1. Select a Finder that you want to use to detect an edge.
2. Right-click in the *Video* window and select **Edge+Autofocus finder** from the *Video* context menu. The Autofocus returns one or multiple points, with different XY coordinates taken from the main Finder and Z coordinates (all the same) taken from the Autofocus. The Finder you were using to detect an edge will temporarily be disabled and the *Autofocus* dialog box is displayed.
3. You can change various elements in the *Autofocus* dialog box and test it using the **Run** button. Make sure the **Keep with Finder** option is checked.
4. Click **OK**. VMS remembers where the edge detecting part of your Finder was before you brought up the Autofocus Finder. VMS will return you to the original Finder when you are done setting up Autofocus.

Accelerating Autofocus

VMS allows users to accelerate focus via the **Point Density** field in the *Autofocus* dialog box. Point Density is a percentage of the standard density, which is 50 points (one second) per default pass distance. The default pass distance (established by Auto Set) is a function of the pixel size, which decreases with magnification; the assumption being that part variations are smaller if higher magnification is being used.

A point density of 50% reduces the time of the data-gathering phase from 1 second to one-half second for the default **Distance**. This means there will be only half as many data points from which the Z position is calculated, and thus there will be a slight decrease in accuracy and repeatability. If you have an application where Autofocus takes up a large part of the total inspection time and the **Distance** cannot be reduced because of part variations, edit the Autofocus Finder(s) and reduce the point density. Look at the repeatability you get using this option. It should still be within a factor or two of the repeatability you got before. If the results are acceptable, the increased speed will allow you to get more throughput in your program.

One note of caution applies to increasing Autofocus speed. The initial pass distance you set in the *Autofocus* dialog box is for the first, or search, pass. The first pass is usually made longer than the depth of field in order to find the Z height of the subject within a zone, anticipating that a variation in height may have occurred due to manufacturing tolerances of the part. If you make this pass too long, the camera will speed through the point of focus so fast it will not be able to reliably measure the depth of field for the second pass. The result of this is that any one of several Autofocus errors may occur, even though the subject appears to be well-defined. The usable search depth, therefore, is only half as large when you use a point density of 50%.

TTL Laser Autofocus

Through-the-lens (TTL) Laser Auto Focus is a special type of Autofocus Finder. It can be selected in the same manner as edge, surface, and Ronchi focus. (The Ronchi option is not available with the microscope optics.) There are two types of TTL Laser devices, designated TTL-G and TTL-U.

When Laser Autofocus focus is run using the TTL-U Laser device, the X, Y, and Z axes move to the programmed position as with any other Finder. But then the laser's signal is connected to the Z servo amplifier, and the Z-axis will quickly "home in" to the best focus. This motion requires Autofocus to wait for the TTL-U Settle-time. After it is finished, the Z position is recorded and the motion controller is put back in charge of the Z-axis. The point measured is returned to the Feature measurement step it is part of.

For the TTL-G Laser device, the laser does not control Z-axis motion as with the TTL-U laser. The TTL-G laser measures the focus point directly from the programmed position and the point measured is returned to the Feature measurement step it is part of. As with edge and surface focus, the Z axis only moves to the focus position if:

- Finders are being displayed in the *Video* window (the Z axis moves immediately)
- The next Finder to run is run at the Z position located by the focus (the Z axis moves when the next Finder runs)

TTL-U Settle-time

How long does it take for the TTL-U laser to home in and stabilize? It's a damped oscillation. If you don't wait long enough, the position will be recorded while the Z-axis is still moving, most likely at the wrong place. If you wait too long, you'll get a good reading but waste time. Therefore, you can program the settle-determination parameters in the *Focus Finder* dialog box. When the laser focus type is selected, the controls at the bottom of the dialog box change to "Settle time" in milliseconds, and "Settle range" in millimeters.

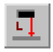
There are two different settling modes:

- The first mode waits the given amount of time from when the laser is activated, regardless of what the Z-axis is doing. This mode is selected by leaving the settle range distance zero or blank.
- The other mode waits until the Z position has changed less than the given settle range distance for the given settle time: "It stayed this still for this long." For example, suppose a typical settle action is that it oscillates for 400 milliseconds, and within that oscillation there are times where it holds within 0.2 microns for no more than 80 milliseconds. You could detect that it is settled by waiting until it has held within 0.2 microns for 100 ms (to exclude those temporary holds) so it would finally realize it was settled after 500ms. If it sometimes oscillates for 600 ms, you'd be safe because it would wait until it got that 100ms of holding. But if you're sure about that profile (you're sure it will always be settled after 400ms), it's faster to wait just the flat 400ms.

The maximum allowed total settle time is 2.5 seconds.

So good questions are: How does it settle for a given situation (lens, subject, gain)? And how do you find out? Trial and error may work, but there is a way to get a detailed look.

When the laser focus mode is selected, the **Auto Set** button in the dialog box changes to **Laser Setup**. When you click this button, the laser is activated permanently (unless it goes out of range) until you turn it off by clicking the button again (it changes to **Setup Done**). The Z-axis position is monitored for 2.5 seconds. During that time, it checks for when the Z-axis has held to within 0.2 micron for at least 300ms. The time it took to the **start** of the 300ms is reported in the *Results* window as an approximation of how long it takes. Also, the time and Z position data collected from the moment the laser was activated until settling was detected (one pair of data are collected about every 50ms) are written out to a text file LASRDUMP.TXT, if the Focus Trace option is enabled (in the Registry). This file is overwritten every time Laser Setup is activated. The data in this file are in two columns: Milliseconds since Laser Activation, and the Corresponding Z Raw Scale reading. The file can be read into Excel or a similar program to produce a plot of Z position vs. time.

Point Laser Scan Finder 

The Point Laser Scan Finder appears to the user as a Finder whose measured points will be put into the point buffer of whichever Feature is being measured. It is called up from the *Video* window pop-up menu or *Video* toolbar. A laser scan returns multiple points in which the Z coordinate is actually measured; it's similar to many Autofocuses done quickly.

The Finder appears in the *Video* window as a 1-D arrow. The arrow depicts, against the video image, the scan path. Because the scan can be longer than the field of view, and could put the arrowhead outside the *Video* Window, it has an arrowhead in the center, too, so you can tell which way it is pointing. Its color is whatever color is selected for Circle Finders (usually red). Unlike other Finders, dragging the graphic in the *Video* window cannot change the Point Laser Finder's Position parameters. So the *Finder Editing* dialog box comes up automatically. Use the dialog box to specify the various scan parameters (see *Basic Laser Finder Parameters*).

While the Point Laser Scan Finder parameters determine the length and angle of the path, its **location** is programmed like a Video Finder. You never actually place the laser spot on the part to teach its position, because it's often too difficult to see. Instead, you look at the image from the camera in the *Video* window, and place the **center** of the desired scan path in the center of the camera's field of view, under the arrow center. For very long scans over Featureless surfaces, this position may be more easily set up by using the *Features* window, with its ability to control-click to move the stage. Because you indicate the center of the scan, the scan will start half of the given length before the programmed position and proceed to half the given length after it.

The axes do not start moving at the given velocity instantaneously; they must accelerate up to the given speed, scan at that speed, and then decelerate to a stop. Specify the acceleration in the *Laser Setup* dialog box, not on a per-Finder basis. VMS computes the distance required for the accel/decel and adds these "ramps" onto the total move. (Data are not returned from these portions.) So, the constant-speed part of the scan will be the length you requested, but the total move will be longer.

When the *Features* window displays a Point Laser Scan Finder (remember, it displays all of the Finders that are part of a step that is being edited), the entire path of the scan will be depicted. The accel and decel portions of the move (which do not return points) are drawn in a contrasting color.

When the Finder is run (either a test-run while programming, or during inspection) the laser will be moved into position and it will perform the scan. If several laser scans are performed consecutively, the camera will not move back into position between the laser scans when the program is running. However, for convenience, the camera *will* be moved back to the center point after the laser scan if it was a test-run of a Finder while editing the Finder (but not after a test-run of the whole step).

🗎 The coordinates in the DRO are never the coordinates of the laser, since you never program by placing the laser on the part.

The safety shutter (required for some laser models) is automatically opened whenever a scan begins. It does not close automatically at the end, because another scan may be next. But it will close automatically when a Video Finder is run, a step has finished a test-run, or the Measurement Program has ended. You may also control the shutter with the VMS **outport** function. To open: outport(0,0,1). To close: outport(0,0,0). (See Appendix A, *Advanced Programming*, if you have that option.)

Laser Usage

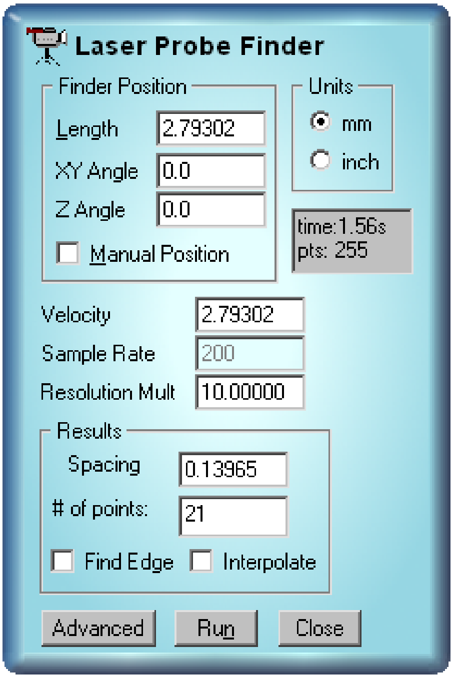
A typical use of the laser would be to make several linear scans over a surface to collect a 2-D grid of points to construct a plane (surface). This could then be tested for flatness.

A single linear scan could be used to define a line, but for an XY working plane, the X and Y coordinates returned will have no relation to the part. So only the Z component would be meaningful, and the line would not be very useful as a line. (It could be more useful in the XZ or YZ working planes.) This is not to say the machine has a shortcoming; rather, that even though the basic scan is in a "line", there is seldom a reason to actually construct a line from it. However, putting the points in the point buffer of a line might be the most logical first step if subsequent VMS steps were to do some kind of special analysis of the buffer points. More commonly, several scans will be done to establish a plane.

The system is also able to acquire an XYZ point without performing a scan.

Basic Laser Finder Parameters

The *Basic Laser Finder Parameters* dialog box is shown below:



The following parameters are in the *Basic Laser Finder Parameters* dialog box:

Parameter	Description
Length	Floating point scan length in mm or inches.
XY Angle	Floating point scan angle in the XY plane, in degrees with respect to the current FCS. Zero is +X, +90 is +Y.
Z Angle	Floating point scan angle in the Z axis, in degrees with respect to the current FCS. Zero is in the XY plane, +90 is straight up. Of course, straight up or down is not useful. Enter a constant floating point value in the dialog (will be rounded to 0.1 degree).

Parameter	Description
Manual Position	Check this only if you want to position the Finder at run-time.
Units	Select mm or inches.
Velocity	Floating point scan velocity in mm/sec or inches/sec.
Sample Rate	Integer value for number of raw points per second. It must be between 10 and 2000. On some systems it is fixed at 200.
Resolution Multiple	Desired ratio of Result Spacing to sample spacing (floating point 1.0 to 1000.0) For example, if the desired Result Spacing is 0.1 mm and the Resolution Multiple is 10.0, raw data points will be sampled every 0.01 mm.
Result Spacing	Floating point value in mm or inches. The returned point data will have this spacing along the scan direction.
Result # of points	Integer number of points to be returned as point data for the desired Feature. Points will be selected from the raw laser data points.
Find Edge	Check this box to indicate that VMS should return a single data point at the location in the raw data where a Z edge transition was found.
Interpolate	Check this box to indicate that VMS should interpolate between raw laser data points for greater accuracy when selecting result points.

There is a field that displays the time in the number of seconds that the entire scan will take, including acceleration and deceleration, but not including moving from one Finder to another. It also displays the number of raw points the scan will use, and will put three exclamation points (!!!) after that number if it is bigger than the buffer size of 2000. If you ignore that, and try to run the Finder, you will get an error (see *Point Laser Scan Exceptions*).

Laser Parameter Interactions

There is a fixed-size buffer that holds the raw data points accumulated during the acceleration and constant-speed portions of the scan. The data fill this buffer at a fixed rate. Therefore there is a maximum time limit for the combined acceleration and constant-speed portions of the scan. An “overflow test” checks the specified acceleration, velocity, and scan length against the time limit and determines if the buffer would be overflowed. (The buffer will not be allowed to actually overflow.)

In addition, systems with Aerotech motion controllers have a limit of 1024 scale positions that can be saved, so the number of raw laser data points cannot exceed this number.

You are not prevented from entering such values, because you may be trying to enter several values and it is only temporarily out of range. But if the values would create an overflow, three exclamation points (“!!!”) appear to the right of the length and speed to alert you that the current combination is not acceptable. If you try to run the Finder or close the dialog box, a message will explain this to you. But the scan will proceed anyway, however the extra data will be discarded and a message “Scan Truncated” will appear in the *Results* window.

Here are the numbers and equations to show exactly how the limitations can be calculated.

The buffer size is 2000 data points.

The acquisition rate is r points per second (fixed at 200 on some systems).

Therefore the maximum combined time of the accel and constant-speed portions of the scan is 2000/r seconds.

d = distance of constant-speed phase (scan length)

v = speed

a = acceleration

Time of accel phase = v/a

Time of const speed phase = d/v

Therefore v/a + d/v must not exceed 2000/r

There are more Laser Finder parameters than are necessary to specify the desired scan. This allows you to specify the scan in a number of different ways. Often you will want to specify the scan **Length**, **Sample Rate**, and **Result Spacing**.

Because the parameters are interdependent, if you change one, others may have to be changed automatically. Often there are several possible changes that would maintain consistency, but the dialog box chooses one based on assumptions of what you probably want. One common assumption is that the Sample Rate must remain the same, because on some systems it is fixed.

Here is what happens when you change each value:

- **Length:** VMS assumes that you want the Number of Returned Points to remain the same, so the *Result Spacing* must be changed. Next, it assumes that you want the Sample Rate and Resolution Multiple to remain the same, so the *Velocity* must be changed.
- **Number of Returned Points:** VMS assumes that you want the Length to remain the same, so the *Result Spacing* must be changed. Next, it assumes that you want the Resolution Multiple and Sample rate to remain the same, so the *Velocity* must be changed.
- **Result Spacing:** VMS assumes that you want the Length to remain the same, so the *Number of Returned Points* must be changed. Next, it assumes that you want the Resolution Multiple to remain the same, so the *Velocity* must be changed. Bear in mind that the number of points returned is *one more than* the number of intervals between the points. This means, for example, that for a 10mm scan with 2mm result spacing, you get *six* result points (at 0, 2, 4, 6, 8, and 10mm), not the 10/2 = 5 you might have thought. So if you want to enter a number of points like 50, but you want the spacing to be a nice round number, 51 would be better.
- **Velocity:** VMS assumes that you want the Length, Number of Returned Points, and Sample Rate to remain the same, so the *Resolution Multiple* must be changed.
- **Resolution Multiple:** VMS assumes that you want the Result Spacing and Sample Rate to remain the same, so the *Velocity* must be changed.
- **Sample Rate:** VMS assumes that you want the Result Spacing and Velocity to remain the same, so the *Resolution Multiple* must be changed.

These automatic changes are not always intuitive, but the changes made when **Velocity** and **Sample Rate** are entered will generally be what you expect. Here is a suggested order for entering parameters:

- Because VMS never changes the **Length**, it should be entered first.
- The **Sample Rate** should be set to the desired value.

A high sample rate and velocity combined with a low acceleration (see *Laser Calibration*) may generate an excessive number of points during acceleration, causing the total number of points to exceed system limits. The number of extra points is equal to **Sample Rate** x **Velocity / Accel**. For example, if a scan rate of 1000 points/sec and velocity of 20 mm/sec are used with an acceleration of 20mm/sec2, 1000 points will be generated during acceleration alone.

- Next, the desired **Result Spacing** or **Number of Returned Points** should be entered and **Interpolate** should be checked.
- If the **Velocity** needs to be specified, it should be entered at this point, because doing so will not change the values previously entered.
- Finally, the **Resolution Multiple** should be adjusted if necessary so that it is not less than the length of any filters being used (see *Laser Calibration*). This ensures enough oversampling to give meaningful result points after filtering.

Result Spacing

Some systems allow user-specified raw sampling rates, while others are fixed at 200 per second. In the following discussion, we will assume a raw sampling rate of 200 points per second. The scan electronics always collect points at a rate of 200 per second. Thus the spacing between raw data points will be speed/200. So the speed and spacing are directly related. See *Laser Parameter Interactions*.

It should be noted that the positions of the points along the path are not guaranteed to have any relationship to the start and end points of the path, because the 200 samples per second sampling clock is not tied to position. For example, if you specify a path length of 1mm and a speed of 50mm/sec, you would expect to collect about 5 points spaced 0.25mm apart (50/200). Conceivably, you would get points at 0, 0.25, 0.5, 0.75, and 1.0 mm. But the first point is not guaranteed to be taken exactly at the beginning of the constant-speed phase; it can be +/-0.125mm on either side. For many applications the locations are not important as long as the readings are taken along the desired path, so this is acceptable.

Sometimes the user wants to get a few points that are taken at certain positions. To do this, take many more points more densely, and look through them for the ones closest to the desired positions. In the previous example we could have increased the density tenfold by reducing the speed tenfold. So we would use a speed of 5mm/sec to get about 41 points spaced 0.025 mm apart. From these raw points we could select the 5 points that came closest to the positions of 0, 0.25. 0.5, 0.75, and 1.0 mm. Their locations would be off by at most +/-0.0125mm, one tenth as much as before. So our Result Spacing is 0.25 mm with a **“Scan Resolution”** of 0.0125 mm. The ability to do this "selecting" is built into the Point Laser Finder. In the *Finder Editing* dialog box, the “Result Spacing” entry allows the user to enter the desired spacing for the point selector, the desired spacing between the result points returned from the Finder. This number must, of course, be no smaller than the Scan Resolution. In other words, once you have chosen the desired Result Spacing, you should realize that the actual positions could be off by as much as +/-1/2 of the Scan Resolution, so you should check the Scan Resolution and change it if needed (the speed will change accordingly).

While the position along the path may be somewhat uncertain, the coordinates of the points returned are accurately measured on the part surface, that is, the returned XYZ point is indeed a point somewhere on the surface. The accuracy of the points found is based on the scale and laser resolutions, etc. But whether the XYZ **scale** readings used correspond to the **laser** reading at the “same” point is related to the scan speed and direction. As the laser is scanning, 200 times a second (every 5 milliseconds) the scales are latched and the laser is read. Unfortunately, the laser cannot be read at the same instant the scales are latched, but rather about 1 millisecond later. The delay means the laser reading will not be from precisely the same point where the scale was latched. So we can consider this to be an error in the scale positions, which is roughly proportional to the speed (about 1ms/1sec = 0.1% of the speed). (It is exactly the speed times the delay, but the delay is slightly variable due to the random nature of what is being interrupted.) At 1 inch per second the error would therefore be about 0.001 inch.

For scans in the XY plane, the Z speed is zero so this error would not apply to the Z component of the reading, which is probably the one of interest if a laser is being used. A slower speed can always be used if more accuracy is needed in the direction of the scan.

The above discussion pertains to how the Finder selects, from the actual raw points, those points that are closest to the desired positions. That is how it works when Interpolation is off. If you use Interpolation, the returned points will be created from the raw data to be at the desired positions. This means the returned points are not “real” data, but in most cases this is not a problem.

Interpolate Mode

Data points are not collected at precisely the locations along the scan path that the Position and Result Spacing would seem to imply. Rather, points are taken at a denser spacing (the old Path Resolution; now, Result Spacing divided by Resolution Multiple) and a “point selector” algorithm selects the point closest to each desired location.

The Interpolate mode will pick the two points closest to each desired location (one on either side of it) and do a linear interpolation between them. The result is that the XY coordinates will be exactly where they are expected to be. The fact that the Z value is also interpolated should not be of concern if the density is of an scale appropriate to the type of surface being scanned. (In other words, you should be sampling at a density higher than surface variations anyway, so a linear interpolation between data points should closely approximate the true surface.)

Laser Data Filtering

The Z coordinates of the raw data (before the "point selector" algorithm selects results) may be filtered. The filter parameters are specified in the *Laser Calibration* dialog box. There are two filters.

- The **Averaging** (or "smoothing") filter replaces each raw data Z with the average of it and its N neighbors on either side. So for example if you enter 5, each Z value will be the average of 11 Z values.
- The **Median** filter replaces each Z value with the median of it and its N neighbors on either side. Additionally for Median, there is a threshold: if the difference between the original value and the computed median is less than a certain distance, then the original value will not be replaced by the median. This is because the purpose of the median filter is to remove noise spikes; small differences may most often represent real data. The median filter is applied before the averaging filter.

For data at the beginning or end of the scan, in order to come up with enough neighbors, data values are "reflected" out. For example, if the first data point is index 0, then if a neighbor is needed at –1, it will use the value at 1; if a neighbor is needed at –2, it will use the value at 2.

A special case is when the scan length is zero. In this case, the speed, number of points, and result spacing values are not meaningful and are ignored. Instead, it will collect 2N+1 values at the programmed location, where N is the larger of the two filter lengths. This allows the filters to do their jobs. All of the points, after filtering, will be returned from the Finder to the Feature being measured.

About Scan Velocity

The scan Velocity will be computed to agree with your other parameters. But the system will have to reduce the Velocity if it comes out too high. (The maximum allowed velocity may be seen, but not changed, by choosing the menu items Setup/Options/System. The X axis velocity is used as the maximum.) If it has to reduce the Velocity, it will also reduce the Sample Rate if it can, to maintain the requested Resolution Multiple and to prevent a Scan Buffer Overflow. (If it can't reduce the Sample Rate, the resulting Resolution Multiple will be *higher* than you requested. This shouldn't be a problem; it's an improvement. But this might cause a Scan Buffer Overflow.) You get the data points you wanted but it happens more slowly.

Reduce Velocity by increasing the Number of Returned Points, decreasing the Sample Rate, or increasing the Resolution Multiple. Beware, these actions may also increase the buffer usage over its limit. If you find you are unable to get what you want, bumping into one restriction (Velocity) or the other (buffer usage), then you have probably hit the system's limitations. You'll need to break the scan up into smaller parts.

If the potential for an automatic velocity reduction is a problem, perhaps because of throughput issues, you can test if this is happening. If the "Debug Log Messages" option is turned on, it will print a message in the *Result* Window (and result output file): "Laser scan velocity exceeds max; reducing."

Point Laser Scan Exceptions

“Scan truncated”

The given parameters would require too many raw points to fit in the raw scan buffer. The number of raw points must be less than the buffer size of 2000:

Buffer points required = R * (V/A + L/V)
where R = sample rate, V = velocity, A = acceleration, L = length.

The V/A is there because data are taken during acceleration too. The *Basic* dialog box is able to check this for you, and displays "!!!" if this will happen. The *Advanced* dialog box cannot do this because it cannot predict what values the expressions will have.

A simpler formula for the number of buffer points (which does not take the acceleration into account) is Number of Returned Points * Resolution Multiple.

This is not an error that can be handled by the error handling capability of VMS. If it occurs, it merely adds the error message into the result output. This error basically happens when you try to collect too much data with one scan. Possible remedies:

- Increase the Acceleration in the *Point Laser Calibration* dialog box. Values of 10 or 20 mm/sec2 are often fine, and the acceleration does not have to be low just because the velocity is low or the scan is short!
- Lower the Resolution Multiple.
- Lower the Number of Returned Points.
- If all else fails, you must decrease the Length. This is hard to accept sometimes because it will mean that you must do multiple scans instead of one.

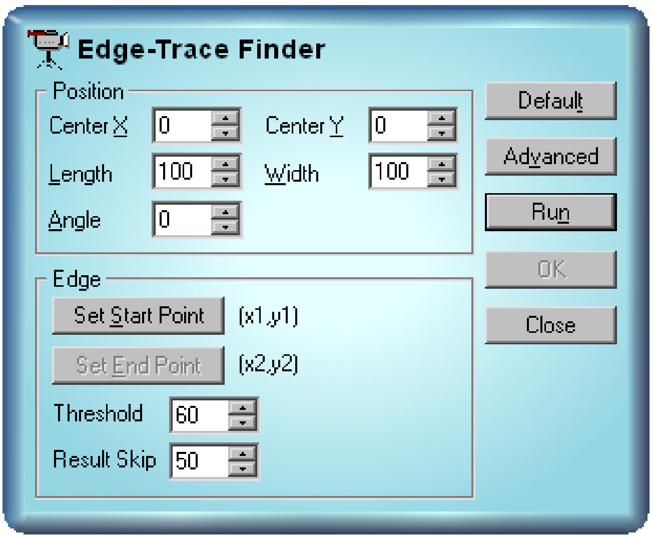
Smart Finder 

The Smart Finder button allows you to create a Finder automatically when you click on the edge of a Feature in the *Video* window. To use this tool:

1. Click the **Smart Finder** button from the *Video* toolbar. The cursor will turn into the Smart Finder cursor when you move it to the *Video* window.
2. Place the top left point of the Smart Finder near an edge of a part. Click the left mouse button once. The Smart Finder will automatically find the edge of the part and the edge will be drawn.
3. Click on the **Smart Finder** button again to turn it off.

Edge Trace Finder 

The Edge Trace Finder is taught to start at a particular place, like any other Finder – but then, it takes off on its own, finding its way along an edge wherever the edge goes. It will continue to try to follow the edge until it comes back to where it started, or until it reaches a predetermined stop point. So, it could trace anything from a simple circular path around the inside of a large hole, to the convoluted form of an extrusion in cross-section. An example of the *Edge Trace Finder* dialog box is shown below.



Please note that it can only trace along well-defined edges, preferably back-lit silhouettes. You specify a Threshold between 0 (darkest) and 255 (lightest); the Finder follows the edge between pixels that are darker and lighter than this. This is not an absolute value; it is adjusted according to the brightness range seen in the first FOV. So 128 will always be midway between the darkest and lightest parts of the image, regardless of light levels.

> If the edge is a little “fuzzy”, you can use this parameter to tune where it finds the edge within the gradient.

“Result Skip” tells it how many pixels to skip between output points. If you are tracing a large shape, outputting every single pixel along the edge would generate a huge number of points; with Result Skip you can reduce the number of points you must contend with.

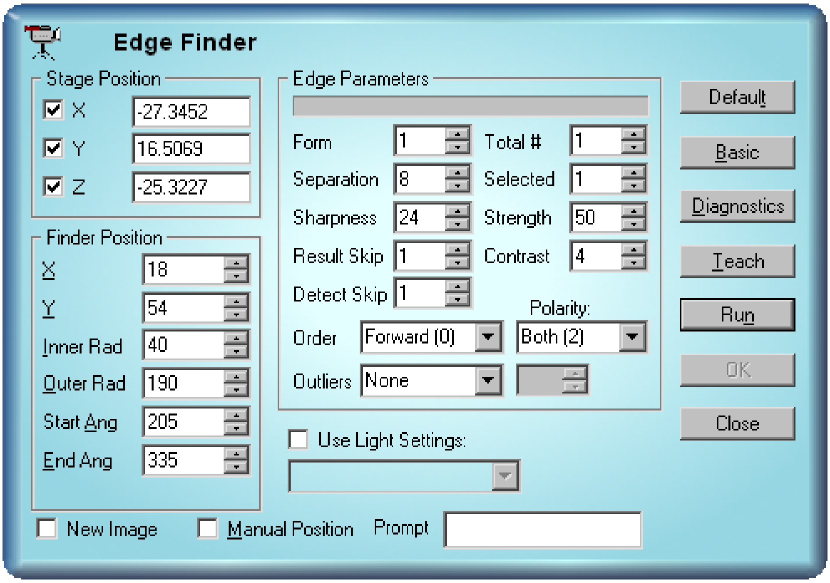
You start out by putting the Finder on the edge where you want to start tracing, and rotate it so the arrows point in the direction you want it to go. Then there are two ways to use this Finder: You can specify the Start and Stop points, or not. If you don’t, it will start from wherever it happens to be when you test-run it, and it will stop when it gets back to where it started. If you only want to trace part of a shape, you can put the Finder on the start point, click Set Start Point, then put the Finder on the end point and click Set End Point. Then when you test-run it (and when it runs as part of the program) it will run from the Start point to the End point.

You should always let the Finder run until it stops (either when it gets back to where it started, or when it gets to the end point). If you need to stop it for some reason, you can hit the Stop button that normally stops a measurement program, but in this case, the Edge-Trace Finder may not be set up correctly and should not be added to a Measurement step.

Speaking of Measurement steps: Although this can be used to trace along Lines, Circles and Arcs, and so could be used in those kinds of Feature Measurement steps, it is also very useful for tracing out complex shapes. In such cases, you would probably want to use the Finder in an Add/Remove Point Data (General Curve) step. And, you would probably want to select **Point Data Output** for the step so that the system will output the coordinates of all the points to an external Profile Analysis program such as ICAMP.

Advanced Finder Parameters Dialog Box

An example of an *Advanced Finder Parameters* dialog box is shown below.



For most finders, the dialog box contains sections for stage Position, Finder Position and Edge Parameters or Focus Parameters. These are described in the following sections.

Stage Position

This group in the *Advanced Finder Parameters* dialog box contains three fields for the FCS-relative X, Y, and Z stage position **in millimeters**. These are the coordinates of the center of the field of view. By default, they display the current FCS-relative stage position. Enter constants or expressions for these stage positions. Next to each field there is a check box. Clearing the check box for an axis means that the stage will not move in that axis (the corresponding edit control will then be grayed out.) Normally this should be done only for the Z-axis (e.g., to run the Finder at a Z position determined by a previous focus) or for all axes (e.g., to create a Manual Finder that does not first move the stage.) Setting all axes to zero means that the Finder will have no offset relative to the Finder Coordinate System (FCS).

Finder Position

This group in the *Advanced Finder Parameters* dialog box contains fields for the position, dimensions and orientation of the Finder within the video image. The parameters vary from one type of Finder to another. These parameters are described in Basic Edge Finder Parameters.

For Edge Finders, these are the X, Y positions within the FOV in pixels, the angle (2 angles used for Arc Finders) in degrees, and the two size dimensions in pixels. These parameters are relative to the FCS; the Finder's angle and center position will be rotated around the center of the field of view by the angle of the FCS at run time. Be aware that depending on the initial position and FCS angle, this rotation might cause part of the Finder to go outside the FOV, which will cause an error.

For Focus Finders, the fields are the XY position of the center and the X and Y size, in pixels. Focus Finders remain in the given place in the FOV and are not adjusted by the FCS angle.

Because the positions and sizes are in pixels, you may want to convert from inches or millimeters by using the VMS calibration functions to get the pixel size (see *About Lens Calibration*).

Advanced Edge Parameters

This group in the *Advanced Finder Parameters* dialog box contains fields that control how the edge is located by image processing. Currently, only the edge processing parameters are likely to require modification when the orientation of the Finder or the polarity of the Feature (light on dark or vice-versa) changes can accept expressions. These are **Order** (Forward/Reverse), **Polarity** (Rising/Falling/Both), and **Mode** (Min/Max Point.) These fields have drop-down lists so you can see the integer values that represent the different choices. The other Finder parameters can only contain constant values.

Parameter	Description
Form (1-32)	Averages the adjacent pixels to smooth noise or other image anomalies. When computing the edge profile, the pixel data is smoothed over the finder width by the number of pixels specified.
Separation	Minimum number of pixels between edge transitions found along a profile.
Sharpness (1-31)	This sets the smoothing filter length along each profile. The pixel data is smoothed over the finder length by the number of pixels specified. The actual filter length is: Filter Length = 31 - Sharpness.
Result Skip	The number of lines that are skipped in between lines used for edge point calculations. This essentially determines sampling frequency of the lines to calculate edge points. The primary purpose of this parameter is to save computation time (larger numbers will yield faster edge calculations).
Detect Skip	The number of pixels skipped in between pixels used for calculating the edge profile along a line only in the pre-filter pass. The primary purpose of this parameter is to save computation time (larger numbers will yield faster edge calculations).
Total #	Specifies the total number of edges to search for. If searching for the strongest edge, this should be set to 1. If searching for the first or last edge, it should be set to 2. A large number can slow down the computation.
Selected	Specifies which edge to select out of the total number. This should normally be set to 1.

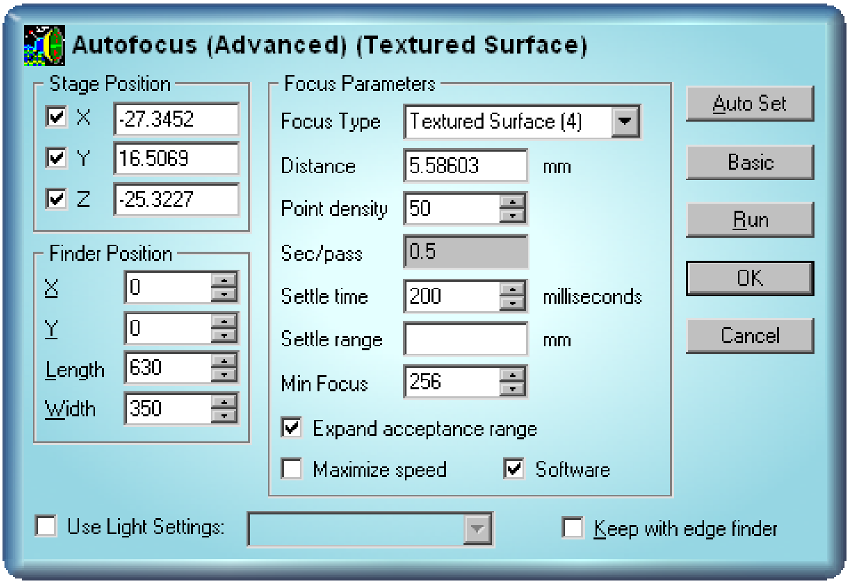
Parameter	Description
Strength (0-100%)	The minimum percentage of the maximum edge strength to be considered when counting edges along each profile. Only edges that meet this criterion are included in the profile.
Contrast (0-100%)	The minimum absolute edge strength to be considered when counting edges along each profile. Only edges that meet this criterion are included in the profile.
Order	If Forward is selected, the first edge in each profile is the one nearest the beginning of the profile; if Reverse is selected, the first edge is the one nearest the end.
Polarity	A Rising edge (dark-to-light) only, a Falling edge (light-to-dark) only, or Both rising and falling edges in the finder area may be selected for detection.
Outliers	Specifies how close edge points must be to a fit through all edge points found. The type of fit performed depends on the finder type. If Delta is selected, the numeric parameter is the zone in pixels outside of which points are removed. If Sigma is selected, the numeric parameter is a sigma value applied to the edge point data to determine the width of the zone.

Here is how the simplified choices in the "Edge" section of the basic *Edge Finder* dialog boxes correspond to the underlying advanced parameters:

- Selecting **Strongest** will set **Total#** to 1; other edge selection parameters (**Selected**, **Strength** and **Order**) are irrelevant in this case.
- Selecting **1st** will set **Total#** to 2, **Selected** to 1, and **Order** to Forward (0).
- Selecting **Last** will set **Total#** to 2, **Selected** to 1, and **Order** to Reverse (1).

Advanced Focus Parameters

This group in the *Advanced Autofocus Options* dialog box contains a combo box for focus type (vertical edge B/W, surface, etc.) The integer values used for the focus type can be seen in the drop-down list. Also shown are fields for pass distance (mm), point density (percent), laser hold range (mm) and laser wait time (milliseconds).

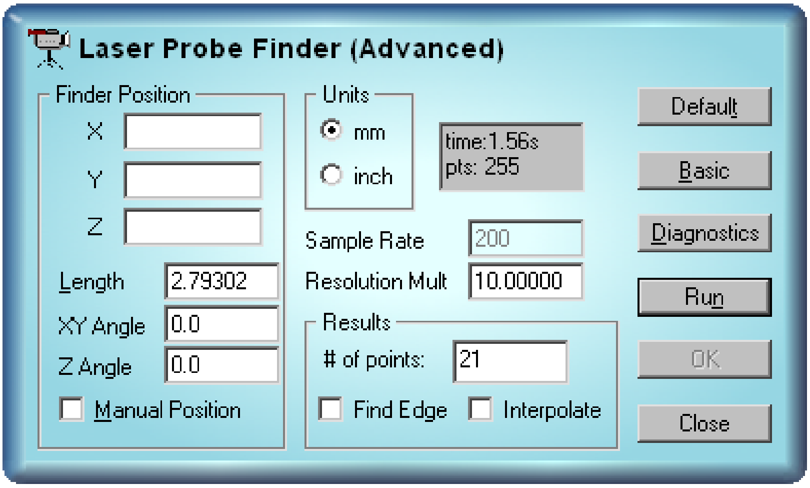


The **Focus Parameters** group also contains additional options for maximizing speed and expanding the autofocus acceptance range:

- Reduce the **Min Focus** value to reduce the number of "Focus too weak" errors (this is not needed for software autofocus).
- Select **Expand Acceptance Range** to reduce the frequency of errors due to skewed or multi-peak focus data.
- Select **Maximize Speed** to reduce the number of passes used to bring the hardware focus values into the desired range (this is not needed for software autofocus).

Advanced Laser Finder Parameters

All of the parameters in the *Basic Laser Finder Editing* dialog box are constants. The only thing about the scan that can change is the position and angle of the scan, which changes automatically based on alignment (FCS). It is useful to determine some parameters when the program is running, so that a single Finder can be used in different situations. For example, if you had a "family" of related parts, one program could measure any part by basing the scan parameters on a data file. To accommodate such situations, VMS provides the ability to choose an *Advanced Laser Finder Parameters* dialog box where you can enter expressions in some of the fields. This dialog box is shown below.



As with other Finders with expressions, the expressions are only used when the whole step is run as part of the program. When you test-run a Finder, or when you test-run a whole step by pressing the **Run** button in the *Measurement Step* dialog box, it will run according to the constants in the *Basic* dialog box, so **be sure to set them up first!**

The *Point Laser Scan Advanced* dialog box will allow you to enter expressions for the following values:

- **Stage X, Y, Z position of center of scan** - user enters a floating point expression in the dialog, mm relative to the Finder Coordinate System (FCS).
- **Length** - user enters a floating point expression in the dialog, mm (or constant inches).
- **XY and Z angles** - user enters a floating point expression in the dialog, in degrees, relative to the FCS.
- **Result Number of Points** - user enters an integer expression in the dialog.

You can still enter constants for any of these values (except stage position). The constant will also appear in the *Basic* dialog box if you switch to it, and will interact with the other constant parameters there. In other words: **Entering a constant in a box that can accept an expression, is the same as entering it in the Basic dialog box and not entering an expression.** The *Advanced* dialog box also includes controls to enter *only constant values* for Sample Rate and Resolution Multiple.

You cannot use expressions for all of the parameters. If all the parameters had expressions, the values might conflict with each other. This problem is solved in the Basic box by automatically changing conflicting constant parameters, but we can't do that with expressions. Here is why some parameters do not accept expressions:

- The Length (L), Result Spacing (S), and Number of Returned Points (N) are related: **S=L/(N-1)**. So, to prevent the possibility of conflicting expression values, **you cannot enter an expression for the Result Spacing; it is derived from Length and Number of Returned Points**. You can always choose the other two to get a desired Result Spacing.
- The Velocity (V), Sample Rate (R), Result Spacing (S) and Resolution Multiple (M) are related: **V=(R*S)/M**. So, to prevent the possibility of conflicting expression values, **you cannot enter an expression for the Velocity; it is derived from other parameters**. You can always choose the other parameters to get a desired Velocity. See below for information about Velocity limits.
- **Sample Rate and Resolution Multiple do not accept expressions**, because these parameters will seldom or never need to be varied at run time.
- Expressions for light values are not included in the *Point Laser Scan Advanced* dialog box, because the lights are only used to help you see while you set up the scan center during programming. This should seldom if ever need parameterization, and it is better to not have them cluttering up the dialog box. The light levels can still be programmed at constant values using the *Stage and Lights* dialog box.

We strongly suggest that you try out the full range of your computed values by entering the highest and lowest intended values as constants in the *Basic* dialog box. Check for buffer overflows (!!! after the number of buffer points) and for excessive Velocities (see *About Scan Velocity*).

Finder Diagnostics

Finder Diagnostics are available with the optional Advanced Programming component of VMS. They help diagnose the proper function of the Finder as described below.

To enable the *Finder Diagnostic* window:

1. Run the Finder.
2. While in the *Video* window, simultaneously press the **Alt-D** keys on the keyboard *or* click the **Diagnostics** button in the *Advanced Finder Parameters* dialog box.

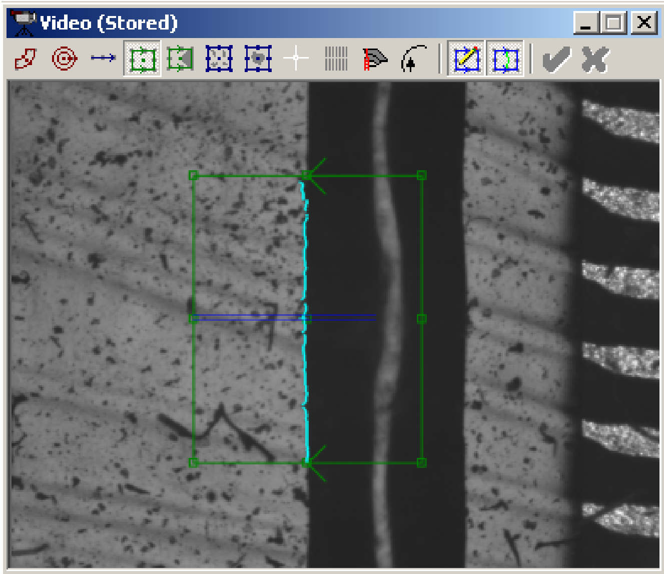
The *Diagnostics* window will not close itself. To close the window, left click the close box at the top right corner of the window, or right click the title bar on the *Diagnostic*s window and select **Close**.

Edge Finder Diagnostics

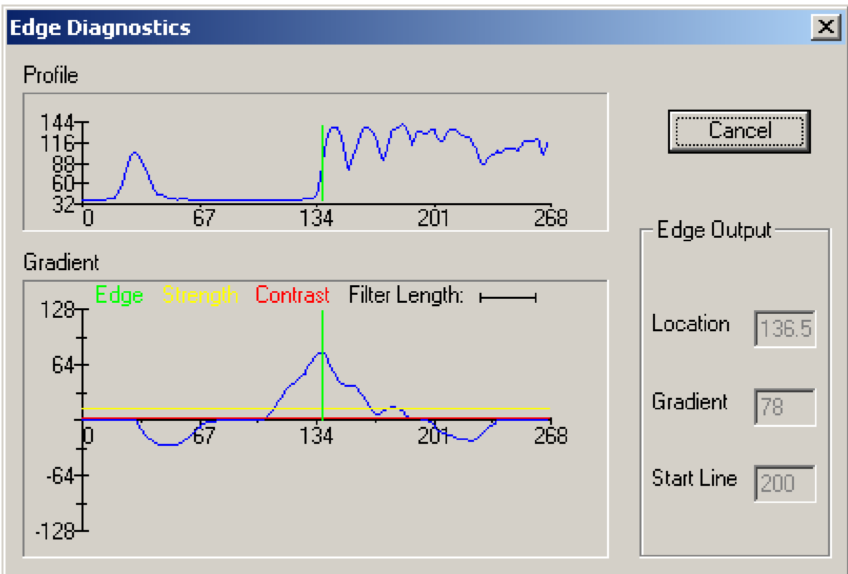
The *Edge Diagnostics* window shows one scan line from the Finder. The scan line runs across the edge perpendicular to it. For Line Finders, this is in the direction of the arrows. For Circle and Arc Finders, this is from the center outward. There are many scan lines in a Finder, but you look at only one at a time, changing which scan line you are viewing by using the keyboard. A pair of blue trace lines is drawn in the *Video* window showing which scan line is being analyzed in the *Edge Diagnostic* window. In the following picture, dark blue lines show the location of the scan line and the light blue cross shows the edge of the point found on that scan line. In a Line Finder like this, there would normally be many edge points returned along the edge; with the Diagnostic running, you are looking at one edge point at a time.

Notice that the two dark blue lines have some space between them. This space visually shows the area of the image being used due to the Form parameter – the larger the Form value, the farther apart the lines. The pixel values are averaged across the width in a direction parallel to the edge (perpendicular to the scan line), thus deriving a filtered row of pixels along the scan line. This row of filtered pixels along the scan line and across the edge is called the Profile. It is shown in the top of the *Finder Diagnostic* window. The vertical scale is brightness, from 0 (black) to 256 (white).

The horizontal scale is pixels relative to the start of the scan line. Here is the image used for the diagnostics example:



The corresponding *Edge Diagnostics* window is shown below, followed by a description of its contents.

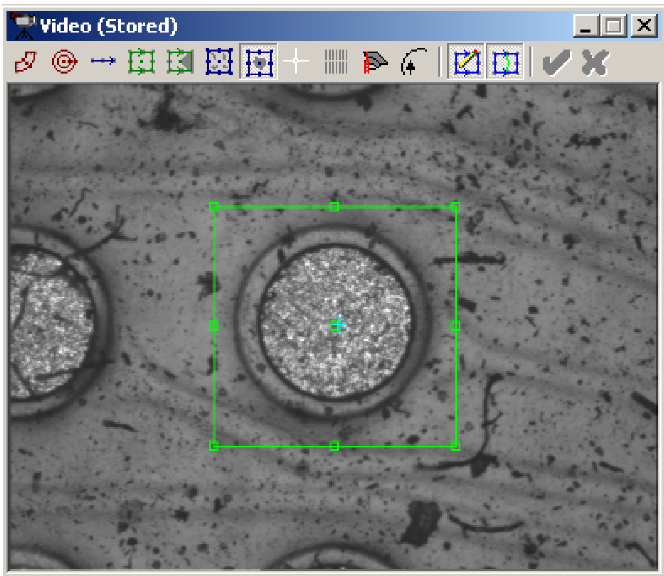


Window	Description
Profile	This shows the average gray level value of the probe line along each pixel in the length of the Finder. The probe line can be moved within the Finder shown in the *Video* window using the Alt-Arrow (moves one line), Alt-Page (moves ten lines), Alt-Home (moves to first line) and Alt-End keys.
Gradient	Shows the gradient of the profile data. The gradient is essentially the first derivative of the profile for that set of scan lines. Every positive or negative peak in the Gradient is a potential Edge. **Note:** It is not exactly the simple derivative of the Profile; a filter based on the Sharpness parameter obtains it.
Designator Lines	**Edge** (Green) – Shows the position of the actual edge point found for that probe location. It occurs at one of the peaks in the Gradient. **Strength** (Yellow) – Shows the minimum value that the gradient can have to qualify as an edge location when there are multiple edges. This is computed relative to the highest value of any peak in the gradient graph. The yellow line is not shown when the “Total #” parameter is 1, since then the Strength parameter is irrelevant. **Contrast** (Red) – Shows the minimum value the gradient can have to qualify as an edge location. This is a fixed amount computed as 256*(contrast/100). It is an absolute value and gradient peaks (edges) weaker than it are not considered. **Filter Length** (Black) – Shows the size of the filter length relative to the graph X-axis for reference. This is the filtering that takes place in a direction parallel to the scan, across the edge. (Filter Length = 31 - Sharpness.)

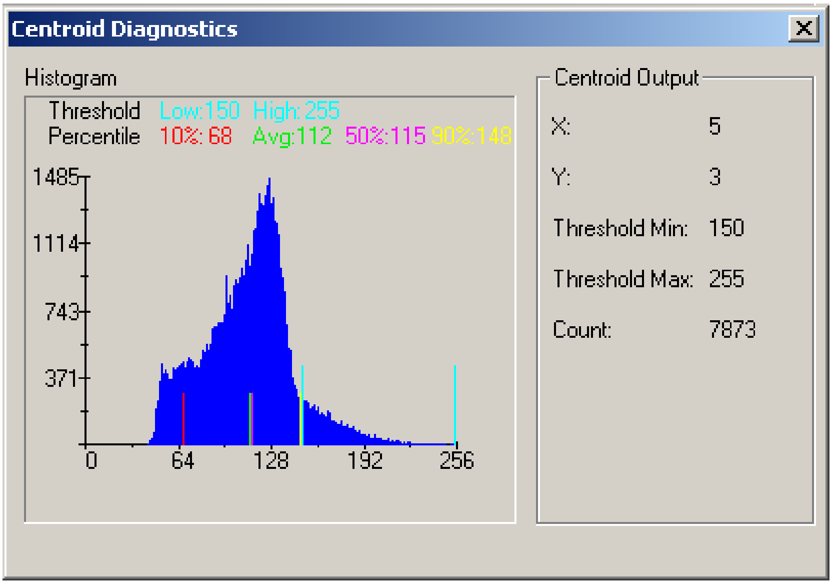
Window	Description
Edge Output	**Location** – The value of the Edge Position found along the length of the Finder. This should correspond to the Edge Position shown along the X-axis of both the profile and gradient graphs. **Value** – The value of the edge found. This is a gradient value and should correspond to the Y-Axis value of the edge found in the gradient graph. **Start Line** – The position of the probe line within the Finder (in pixel units).

Centroid Finder Diagnostics

Run a Centroid Finder on an image, and then press **Alt-D** in the *Video* window.



A histogram of pixels inside the Centroid Finder will be drawn on the *Centroid Diagnostics* window.

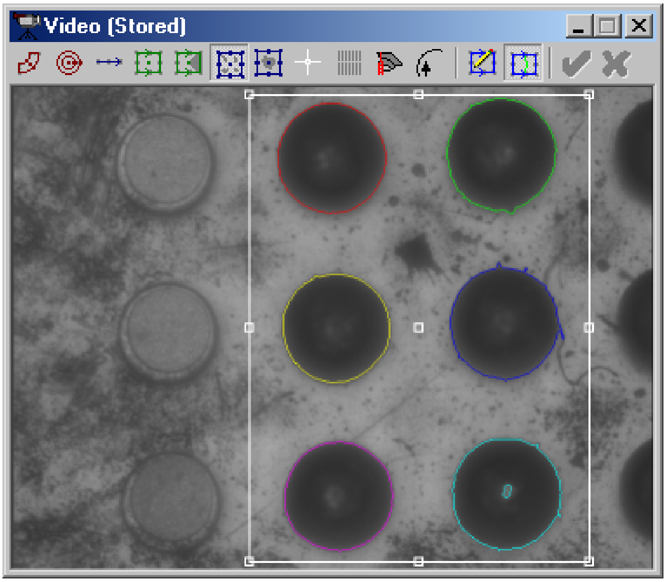


Vertical bars are drawn to represent some important brightness values. Light blue bars represent the two Threshold values specified in the Finder parameters. The brightness values in the range of 0-255 correspond to the given thresholds when in the Absolute mode. In the percentile mode, they show the absolute brightness corresponding to the given percentile values in this particular histogram. There is also a green bar for the average brightness.

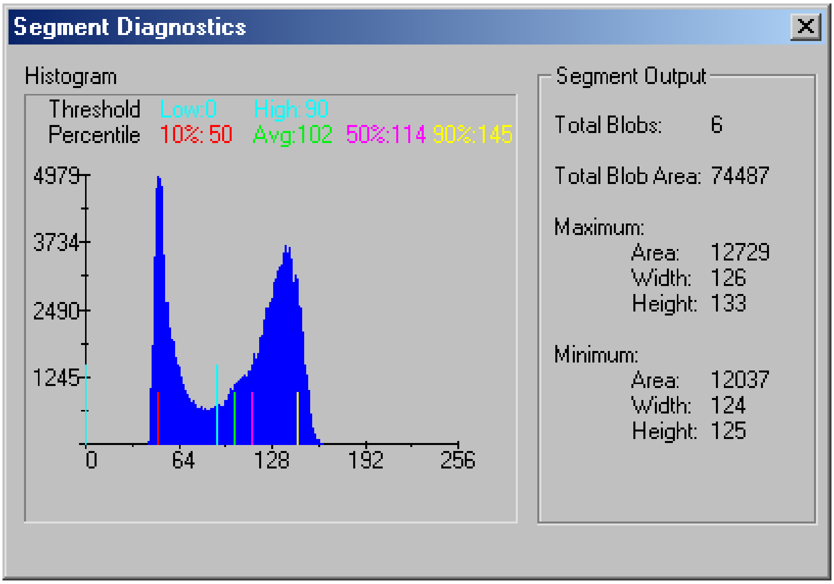
Window	Description
Histogram	Threshold: The range of pixel intensities to be counted when counting pixels.
Centroid Output	**X:** x coordinate of the centroid (in pixels, relative to the center of the Finder) **Y:** y coordinate of the centroid (in pixels, relative to the center of the Finder) **Threshold Min/Max**: The minimum/maximum values to be considered included (or excluded) in the centroid calculation. The values are in gray level units (along the X-axis of the histogram graph). **Count**: The number of pixels in the Finder area whose intensity was at least that specified by Min Threshold, but not greater than Max Threshold.

Blob Finder Diagnostics

This is a picture of Blobs with edges drawn on the *Video* window. Each Blob is drawn in a different color (from a limited set of colors) to help tell the Blobs apart.



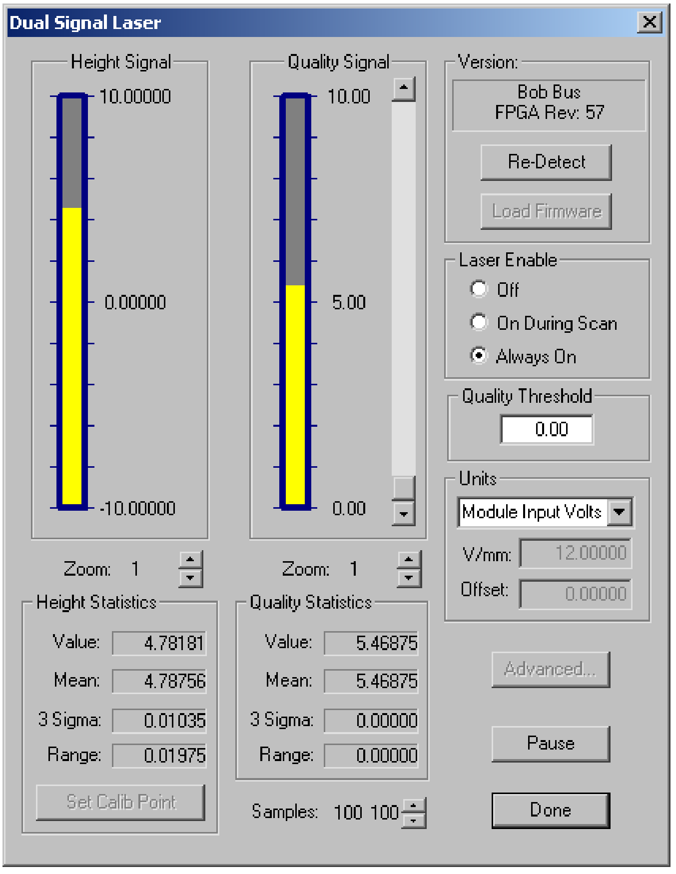
A histogram of the pixels inside the Blob Finder will be drawn on the *Segment Diagnostics* window. There are also the Total Blobs, Total Blob Area, and the Min/Max Blob information displayed on the *Centroid Diagnostics* window.



Window	Description
Histogram	**Threshold Min/Max**: The minimum/maximum values to be considered included (or excluded) in the blob calculation. The values are in gray level units (along the X-axis of the histogram graph). **Percentile**: The percentile for which a pixel intensity value is to be computed.
Segment Output	**Total Blobs**: The total number of Blobs found. **Total Blob Area**: Sum of the area of all Blobs found. **Maximum**: The Blob with the maximum area. **Minimum** The Blob with the minimum area.

Laser Finder Diagnostics

A/D diagnostics are available for point scan lasers attached to the CISP board by pressing the **Diagnostics** button in the *Advanced Laser Finder Parameters* dialog box. The *Laser Diagnostics* dialog box for a dual-signal TTL-G Laser is displayed below.



Section 8

Step and Repeat Programming

Step and repeat is a capability available from the **Program** menu that allows you to use programming techniques to re-perform portions of measurement programs. For example, if you have several identical parts in a fixture, you can write program steps to measure one part and then repeat the steps to measure additional parts after specifying an offset value to help VMS find the additional parts. You can perform a step and repeat for *parts* (multiple parts measured by the same steps) or for *Features* (repetitive Features on a single part measured by the same steps).

Step & Repeat creates a *loop* containing the program steps to be repeated. Step & Repeat loops are related to Align Blocks (see *Align Blocks*). Every Step & Repeat loop has a coordinate system associated with it, with a name like SR1, SR2, etc. This is a special type of PCS. It is the same as the FCS in effect outside the loop, except it is moved according to the step and repeat parameters. It is the FCS in effect within the loop, except of course inside other Align Blocks or Step & Repeat loops within the loop.

Feature Step & Repeat vs. Part Step & Repeat

There is a difference between when there are multiple Features on a part, and when there are multiple parts on the stage.

When there are multiple Features on a part, they usually have their own individual positions with respect to the same DRF. We want the Finders to move over, so this is done by moving the FCS on each iteration of the loop. But we don't want the DRF to move, so the default DRF remains what it was outside the loop. And because the Features' positions with respect to the DRF are all different, you have an option to automatically adjust the Nominal positions of all Features within the loop. So you just program the Finders and Nominals for the first Feature, and the loop takes care of the rest.

When there are multiple parts on the stage, we typically want the DRF to move over to each part in turn. This means that Nominals should not be adjusted, and the Step & Repeat coordinate system is not only the FCS within the loop but it is also the default DRF. In practical terms, of course, the DRF should be re-established on each part because the position of each part is probably not precisely related to the position of the first part.

Part Step and Repeat Loop

A *Part* step and repeat loop is useful when you have several parts in a fixture. Within the loop, there is a coordinate system set up by the loop, identified by a name like SR1. It is used as both the FCS and the default DRF within the loop.

This coordinate system starts out equal to the enclosing FCS (the FCS in effect at the point in the program where the loop is). It is then offset by the amount you specify, relative to that enclosing FCS. For example, to step though a tray full of parts, you want the enclosing FCS to be aligned with the tray. Within the loop, you might want to do an alignment to the individual part itself.

Feature Step and Repeat Loop

A *Feature* step and repeat loop is useful when you have a pattern of repeated Features on a single part. The coordinate system set up by the loop is used as the FCS only; the default DRF remains what it was before the loop was entered. Consequently, the DRF for the part remains in place, and only the Finders get shifted to new positions. Again, the shifting of the FCS is relative to the enclosing FCS.

Another difference between Feature and part step and repeat is that the nominal positions of all Features measured within the loop can be adjusted according to the offsets from the loop. So you program a Feature with the nominal positions of the first Feature in the pattern, but the nominal position will be automatically adjusted during the second and subsequent iterations of the loop. When you create or edit a Feature step and repeat loop you can select whether it will apply the nominal adjustments or not.

Feature step and repeat loops may be nested, and you can even have an FCS-only align between two loop levels (i.e., a Feature step and repeat loop nested in an FCS-only align nested in a Feature step and repeat loop.) The adjustments to the nominals only depend on the exact spacings you specified for the loops, and do not take into effect any offset compensated for by the FCS align(s).

Creating an XY Feature Step and Repeat

1. Select **Step and Repeat /XY Step and Repeat** from the menu. In the *XY Step & Repeat* dialog box fill in the spacing between the repeated Feature in the X and/or Y directions as well as the number of repeat positions. Enter a float expression for spacing and an integer expression for number of repetitions. A negative or positive spacing expression determines the step direction. Note the name of the Step & Repeat variable; in our example we will assume it is SR1.
2. Press **OK**. A **Step and Repeat Begin** line will be inserted into the program.
3. Select **Window/View Object Names** from the menu. The *Object Names* dialog box is displayed. Create an array of Features instead of using a single Feature. This gives each Feature its own identity.
4. Type in a name for the Feature being measured in the Name field (for example, GridHole), and select the Feature type from the Type drop-down list. For an array of Point Features, select Point, not Simple Point.
5. Set the dimension of the array. Entering a value in "Dim 2" makes the array 2-dimensional. You must make 2-dimensional (X & Y grid) arrays as big as they will need to be; it's OK to make them larger than necessary. 1-dimensional arrays can be smaller that needed. When the index on a 1-dimensional array reference is larger than the current size, the array will automatically be dimensioned larger to include the referenced element. However, the array will not extend beyond 32768 elements. The new elements are initialized the same as when the array is newly created. Rather than relying on VMS to automatically resize an array, it is best to create an array that is larger than your current need. For example, if the program you are creating has 15 repeated elements, create an array for 20 elements. Dynamically resizable arrays only apply to 1-dimensional arrays.

Once an object (such as an array) is named the name can not be changed. It can be deleted only if no other steps refer to it.

6. Measure the first repeated Feature. This will be the starting point for the step and repeat. Do not use the default Feature name supplied by the system. Rename the Feature in the *Measure Feature* dialog box. The name of the Feature must be exactly the same as the declared variable, but include the array index. To follow the prior example, you could type GridHole[SR1.I]. The SR1.I identifies which Feature in the array is to be measured. If the repeatable Features are in the Y direction the array should be SR1.J instead of SR1.I. If the grid is two-dimensional, enter both array indices, for example GridHole[SR1.I, SR1.J]. The array indices, and therefore the measurement results, will start with 0 and go up to N-1 where N is the number of Features in the pattern. For example, if there are 10 Features, you program the loop to go 10 times and the index will go from 0-9.

7. Test run the program by selecting the **Restart** and **Run** buttons from the tool bar.

Creating a Polar Feature Step and Repeat

Polar step and repeat can be used when a group of things is repeated in a circular pattern around a center Feature, but only if the group also rotates. For example, you could use polar step and repeat for the tick marks going around an analog clock face because they are all at different angles. But you could not use it for the numbers because they are all vertical.

1. Define the Feature that will be the center of rotation.
2. From the menu select **Program/Step & Repeat/Polar**.
3. Leave **Multiple Features** selected because the DRF will remain fixed while multiple Features are measured at different locations relative to it.
4. Leave **Adjust Nominals** selected, since the nominal XY coordinates for each Feature are different and it will automatically calculate them.
5. Specify the Feature that is the center of rotation and set up the desired Angle Increment (an expression may be used) and Repeat Count. Use only floating constants (numbers with a decimal point in them) or floating point expressions (containing at least one floating point constant or variable) for an angle. If the wrong type of expression is used it won't be accepted. Click **OK**. The Step and Repeat line will be inserted into the program
6. Create an array in which to store Features once they are measured. Select **Window/View Object Names** from the menu. The *Object Names* dialog box is displayed.
7. Type in a name for the Feature being measured in the **Name** field (for example, GridHole), and select the Feature type from the **Type** drop-down list. (For an array of Point Features, select Point, not Simple Point.)
8. Set the dimension of the array. Entering a value in “Dim 2” makes the array 2-dimensional. You must make 2-dimensional (X & Y grid) arrays as big as they will need to be; it’s OK to make them larger than necessary. 1-dimensional arrays can be dimensioned smaller than needed. When the index on a 1-dimensional array reference is larger than the current size, the array will automatically be dimensioned larger to include the referenced element. However the array will not extend beyond 32768 elements. The new elements are initialized the same as when the array is newly created. Rather than relying on VMS to automatically resize an array, it is best to create an array that is larger than your current need. For example, if the program you are creating has 15 repeated elements, create an array for 20 elements. Dynamically resizable arrays only apply to 1-dimensional arrays.

Once an object (such as an array) is named the name can not be changed. It can be deleted only if no other steps refer to it.

9. Measure the first repeated Feature. This will be the starting point for the Step & Repeat. Do not use the default Feature name supplied by the system. Rename the Feature in the *Measure Feature* dialog box. The name of the Feature must be exactly the same as the declared variable, but include the array index. To follow the prior example, you would type GridHole[SR1.I]. The array index, and therefore the measurement results, will start with 0 and go up to N-1, where N is the number of Features in the pattern. For example, if there are 10 Features, you program the loop to go 10 times and the index will go from 0 to 9.
10. In the *Measure Feature* dialog box, fill in the nominal position of one of the Features. Select the XY position and fill in Tolerances. If it is a Circle Feature, click on **Options** and select Hole, Pin, Inner, and/or Outer, depending on the Feature you are measuring. Then close the *Options* dialog box.
11. Program the Finders needed to measure the Feature, and test run the step.

List Input Programming

List input is a step that reads ASCII numeric data from a disk file and puts it into a Simple Point array. Although this can also be accomplished by using the I/O functions, this step is much faster. The list it creates is for use by List Step & Repeat explained below. The only restriction is that the data file must conform to a certain format. The numbers in the file must be ASCII decimal (with or without decimal points) separated by one or more delimiting characters (space, comma etc.). Each line of the file must have two numbers (XY) or three numbers (XYZ). For example:

```
20.0, 20.0, 3.0
50.0, 30.0, 3.0
70.0, 40.0, 3.0
80.0, 50.0, 3.0
110.0, 60.0, 3.0
```

The *List Input* dialog box contains a field for entering the file name (it may be an expression).

There is a field for the name of the Simple Point array. For a new step, a name is generated in a fashion similar to the way Feature names are created: SP1, SP2, etc. You can choose any name you'd like. There is a count field where you can specify the maximum number of points allowed. The first time a new name is used and OK is selected, VMS will create the array using the Count as the dimension. One-dimensional arrays now auto-resize when needed. If a file with fewer points is read, it will fill in the rest of the array with a special value that indicates “unused.” The List Step & Repeat function will recognize these special values and skip them.

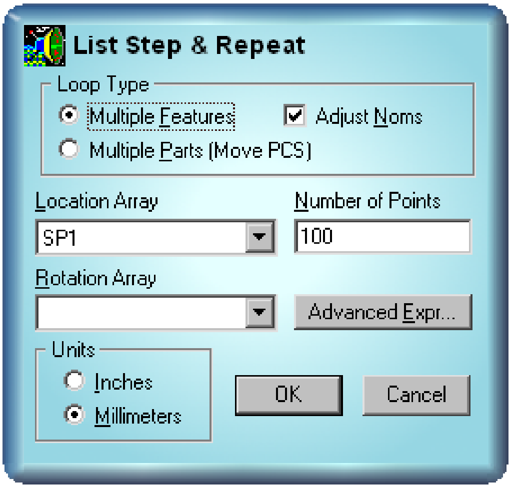
There is also a field where you may enter the name of an integer variable to receive the actual count determined from the file each time a file is read, but it is not required.

List Input generates a block of two steps, beginning and end (similar to an If/Endif). You place steps inside this block that measure a point based on Features on the part. This point is then copied into the first element of the array. This block of steps is only needed and run when you don't have an actual file. So, if the file is not found, it will put up a message box: "File not found. Click **OK** to continue with setup block." When you click OK, it will execute all the steps inside the block. Therefore, when you don't have a file, these steps allow you to program and test run on the first group. When a file is available, it will use the values from the file and skip the steps in the block without stopping for a message block.

More than one step can be placed in a block. Constructing the origin of a group may take several steps. In order to identify which point is the group origin, add a special step, **Set First List Item**. Then you can specify which Feature (it must be a point, circle or arc) to use. Its location will be copied into the first slot of the array associated with the List Input.

List Step & Repeat Programming

List Step & Repeat allows you to repeat a group of Features (or parts) in irregular locations and/or orientations, based on a list of those locations. The list contains the X, Y, and Z offsets of each group of Features (or part) from the current (enclosing) FCS. To access the *List Step & Repeat* dialog box, select **Program/Step & Repeat/List Expr** from the menu.



In the dialog box choose the name of a one dimensional array of Simple Points from a drop- down list of all such defined arrays. The array chosen could be created by the List Inputfunction.

The dimension of the array is automatically entered into the **Number of Points** field, but you can clear the expression and leave this field blank. In this case it will use only the “valid” entries in the array. You can also enter a number here or an expression if desired. Using a variable number of iterations would be useful if the program is intended for a family of parts rather than one specific part.

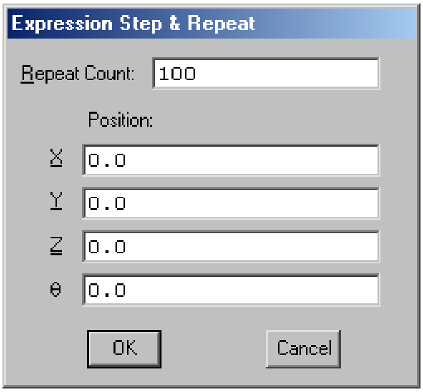
You can select a 1D array of Points or Circles as the list of positions. Their nominal positions must already be initialized. A blank value for the Number of Points is not allowed.

A button labeled “Advanced Expr” in the *List Step & Repeat* dialog box takes you to another dialog box for *Expression Step & Repeat.* (See *Expression Step & Repeat*).

Before you can create or edit any steps containing Finders within the loop, the first location in the array must contain the correct coordinates of the group on which you will be teaching the Finders. This means you must program and execute a step or steps that fill in the array, or at least set up the first location by defining a point at the desired group origin. The List Input step helps you do both of these.

Expression Step & Repeat

This is a more generalized and flexible form of Step & Repeat. It is the underlying method by which List Step and Repeat works, but by allowing you to access the offsets directly, you can solve more advanced inspection problems.



Supply expressions for X, Y, Z and angle offsets for each iteration, and a number of iterations. These offsets are not relative to the last iteration (step amount), but are complete offsets from the enclosing FCS. They could be calculations based on anything, or items from an array that was read from a file.

The coordinate system generated for each iteration is the enclosing FCS, shifted by the given offset, and then rotated around its origin if an angle is given. Note that this is different than the way rotation is handled in Alignment, where the rotation is done first.

Be sure to supply expressions that are correct for the first time through the loop, prior to programming Finders within the loop.

Stepping at Any Angle

Suppose you have a row of 20 holes that are laid out in a diagonal row going up and to the right at 18°, each hole is 0.5 inch diagonally from the last one.

1. The X spacing is cos(18°)*0.5 or 0.4755 inch, and the Y spacing is sin(18°)*0.5 or 0.1545 inch. Select **List/Expr** from the **Step & Repeat** menu and select the **Advanced Expr** button.
2. Enter **20** for the number of repetitions:
3. Enter **SR1.I* cos(18.0)*0.5*25.4** for the X position expression.

> 🗎 The 25.4 is necessary because an expression is not automatically converted from inches to mm.

4. Enter **SR1.I*sin(18.0)*0.5*25.4** for the Y position expression.
5. Define an array of circles called Circ with a dimension of 20 (or more, if the part could change later).
6. Create a Circle measurement step inside the Expression Step & Repeat loop. Change the name of the circle to **Circ[SR1.I]** and measure the first hole.

Stepping in Z

Suppose you have a part that looks like a staircase: each step is 0.81 inch in X and 0.37 inch in Z.

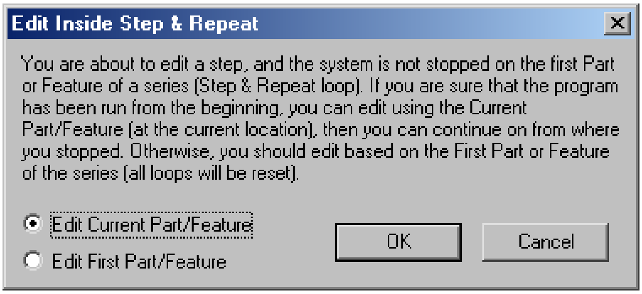
1. Select **List/Expr** from the **Step & Repeat** menu and select the **Advanced Expr** button.
2. Enter **10** for the number of repetitions.
3. Enter **SR1.I*0.81*25.4** for the X position expression. This is the iteration count times the step distance, yielding the total offset of each iteration.
4. Enter **0.0** for the Y offset.
5. Enter **SR1.I*0.37*25.4** for the Z position.
6. Define arrays for the Features to be measured on each step (i.e., a line in X and a line in Z).
7. Inside the Expression Step & Repeat loop, create measurement steps for the Features on the first step.

> 🗎 In the above two examples, the step and repeat loop is assumed to use SR1; this should be changed to whatever variable the Step & Repeat uses.

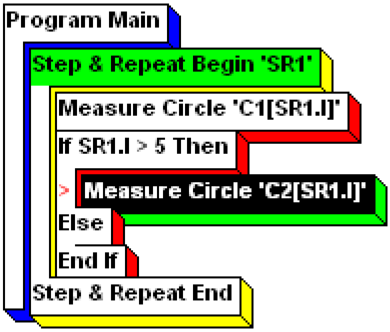
Editing Inside a Step & Repeat Loop

Normally, when you create or edit an alignment-dependent step (containing finders) inside a step and repeat loop, VMS will set the conditions to be as if the loop were executing for the first time. This means you must program measurements on the first part or Feature only.

An exception is when you have paused program execution inside the Step & Repeat loop on some part or Feature other than the first. In this case, it is possible to edit either the first Part/Feature as usual or edit the *current* Part/Feature. VMS will display the following dialog box:



Editing the current Part/Feature may be necessary if Finders have a problem on that particular Part/Feature but not on the first one. It may also be necessary if the step that requires editing is inside a conditional statement and is not executed each time through the loop. For example, to edit the selected step in the following program, set a breakpoint at that step, run the program from the beginning, and edit the step when the program stops at that point. Select **Edit Current Part/Feature** when the dialog box above appears.



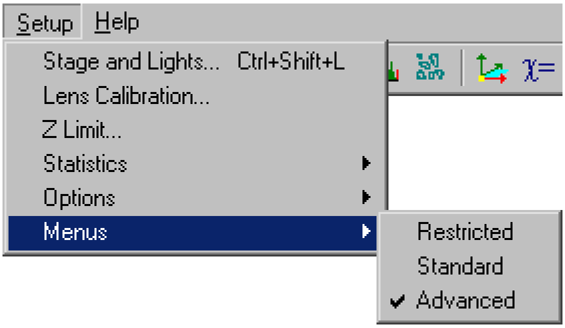
You should only edit the current Part/Feature if the program has run all necessary steps to establish alignments before the Step & Repeat loop and is the stage is correctly positioned on the Part/Feature that you want to edit.

Appendix A

Advanced Programming

Advanced Programming allows you to build programming routines into your VMS measurement programs. To display the Advanced Programming menus:

1. Start the VMS application.
2. Select the **Setup** menu and highlight **Menus**.



3. Select **Advanced** from the **Menus** menu. This will activate all Advanced Programming items in VMS.

Typical Advanced Programming uses include:

- User-defined Procedures and Procedure Libraries
- Conditionals
- Error Handling
- Formatting result output
- Generating an Average Measurement
- Program to Inspect a Family of Parts
- Storing Results in a File
- Creating Steps to interface with QC-Calc or VIP Programs

Assumed Knowledge

We assume that the reader understands basic fundamentals for creating computer programs. If you do not, you will need to get basic training in programming fundamentals before you can understand this documentation or implement programming routines on the system.

Programming Elements

Constants

VMS integer constants can be between -32,768 and 32,767, with an optional leading plus. Or you can type a hexadecimal constant starting with 0x such as 0x10DC. Floating point constants can have a decimal point and an exponent, for example 6.02E23. String constants are enclosed in double-quotes, for example "This is a string."

Program Variables

A VMS program variable name must start with a letter, contain only letters, digits, and/or underscores, and should be 31 characters or less in length. Variable names are case-sensitive, so CenterPoint is a different variable than Centerpoint. Each variable is given one of the predefined VMS data types when it is declared.

VMS automatically declares variables when it needs to, for example, Features, Step & Repeat, and PCS variables. But you can also declare your own variables with the Declare Variable capability.

Operators

Operators are symbols embedded within an expression that indicate an operation to be performed on the two operands on either side of it. The expressions on both sides of an operator must have compatible types; for example, an integer cannot be added to a string.

The standard arithmetic operators are:

+	addition
-	subtraction
*	multiplication
/	division
**	raise to power

Integer division truncates the fractional part of the result. With integers, the modulo operator, **mod**, is available which divides one number into another and returns the remainder. For example: 53 mod 10 produces 3.

Relational operators return 0 if the relation is false and 1 if it is true. They can be used with integer, floating point, or string expressions. They are:

==	equal
!=	not equal
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to

Logical operators are used only with integer expressions for bit-wise logical arithmetic. They are:

&	and
|	or
^	exclusive or

Shift operators are also used only with integer expressions. They are:

<<	left shift
>>	right shift

For example, 8 >> 1 is 4.

There is only one operator for use with strings: + means to concatenate two strings. There are, however, many string functions.

The dot operator '.' is used to indicate selection of a structure component for a variable of a structure type. For example, C1.RADIUS denotes the component named RADIUS in the C1 structure variable.

Operator Precedence

Here are the operators in order of decreasing precedence (operators on the same line are equal):

** * / mod

+ -

>> <<

> < >= <=

== !=

&

^

|

To force a different precedence, use parentheses. For example, 4+5*3 gives 19 because the multiplication is done first; but (4+5)*3 gives 27.

Function Calls

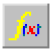
A function call is used to execute a user-defined procedure or a VMS function. A function call in an expression must have parentheses for parameters even if it has no parameters. For example, **nl()** returns a 2-character string but takes no parameters. If there is more than one parameter, they are separated by commas.

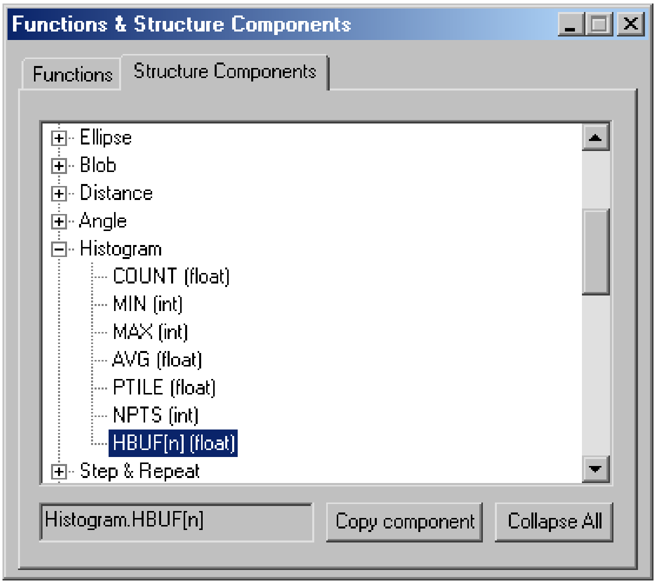
Expressions

An expression is a series of Constants , Variable Names, Function Calls, and Operators that represent the calculation of a single value. The calculated value can be an integer, a float, or a string. A mixture of float and integer operands produces a float result.

The use of expressions is not limited to Assignment steps. Many fields in VMS dialog boxes allow the entry of expressions. As in the *Assignment* dialog box, these fields have a white background when the expression is correct and a yellow background when it is not.

Structure Components

The *Functions & Structure Components* dialog box, accessed via the toolbar button  or the **Help** menu, shows all types of VMS structures in a tree-view. Expand or collapse an item with sub-components by clicking on the + and - boxes. The clicked item will be highlighted and shown in the left bottom text box along with its parent components.



VMS variables that are structures have internal components you can access by dot notation, as in the C and Basic programming languages. For example, if you have a circle named C3, you can add 0.5mm to its radius with the following statement:

```
C3.RADIUS = C3.RADIUS + 0.5
```

In these structures, all measurements of position, length, or size are always maintained in millimeters. You can print the diameter of a circle in inches with this statement:

```
fprt(C3.RADIUS * 2.0/25.4)
```

VMS Data Types

Data Types

Each program variable declared in VMS must be given a data type. There are many data types available in VMS. They are grouped into three categories:

- Basic Data Types (Integer, Float and String)
- Structure Types
- Arrays

Basic Data Types

While there are over 30 different variable types, the basic three appear in the following table.

Type	Description
Integer	A whole number with a range of -32,768 to 32,767.
Float	A floating point number with approximately 17 significant digits, base 10 exponent +/-300.
String	Character data. A string can be up to 65,534 characters but there are more practical limits on the length of string constants, etc. The first character is considered to be in position 0.

Structure Types

A structure variable is a variable with a single name (e.g., C1) that contains several fields of different types within it. You can reference the fields (called components) using dot notation. For example, C1.RADIUS is a floating-point field within C1.

Structure variables may be declared in the *Object Names* dialog box. The full list of structure types can be accessed by scrolling through the **Type** drop-down list in the dialog box. It is as follows.

Simple Point	Default Tolerance
Point	Plus/Minus Tolerance
Line	Zone Tolerance
Plane	Angle of Tolerance
Circle	Angle-Between Tolerance
Arc	Angularity Tolerance
Slot/Tab	Concentricity Tolerance
General Curve	Diameter Tolerance
Ellipse	Distance Tolerance
Blob	Flatness Tolerance
Distance	Parallelism Tolerance
Angle	Perpendicularity Tolerance
Histogram	Position Tolerance
PCS	Roundness Tolerance
Step & Repeat	Runout Tolerance

File	Straightness Tolerance
Light Settings	Radius Tolerance
	Width Tolerance
	Area Tolerance

Arrays

To declare an array in the *Object Names* dialog box, you select a name, a data type, and enter numbers for one or two dimensions. As a matter of convention, if you have a two-dimensional array where the dimensions correspond to X and Y directions, use the first subscript for X.

Below is an example program using Step and Repeat logic and an Array data structure.

Programming Details:

- Multiple features box checked, spacing and repeat count entered for X in Step 15
- Centroid finder used to locate the feature in the field of view in Step 18
- WIDTH array is declared in the *Object Names* dialog box. Enter WIDTH for the name, Distance for the type, 10 for Dim1 box and click on Add button – Step 23
- Since it is stepping across in X axis, SR1.I is used for indexing which will takes the values from 0 through 9 (total 10). For stepping along the Y axis, we would use SR1.J (use uppercase)

Program Listing:

```
 0 Program Main
 1   Align Define 'XY_ORIGIN'
 2     Measure Point 'P1'
 3     Define Align PCS 'XY_ORIGIN' Using P1
 4   Align Use 'XY_ORIGIN' As Default DRF
 5     Align Define 'Z_ORIGIN'
 6       Measure Point 'Z1'
 7       Define Align PCS 'Z_ORIGIN' Using Z1
 8     Align Use 'Z_ORIGIN' As Default DRF
 9       Align Define 'ROTATION'
10         Measure Point 'P2'
11         Construct Best Fit Line 'X_AXIS' Using P1, P2
12         Define Align PCS 'ROTATION' Using X_AXIS
13       Align Use 'ROTATION' As Default DRF
14         "Inspect 10 Pockets, at 0.1 inch interval, and measure each width"
15         Step & Repeat Begin 'SR1'
16           Align Define 'PCS1'
17           "Locate pocket in the field of view - to accommodate variation in
              pitch"
18             Measure Point 'POCKET'
19             Define Align PCS 'PCS1' Using POCKET
20           Align Use 'PCS1'
21             Measure Line 'LEFT_EDGE'
22             Measure Point 'RT_PT'
```

```
        Measure Distance 'WIDTH[SR1.I]' From LEFT_EDGE To RT_PT
      Align End
    Step & Repeat End
  Align End
 Align End
Align End
```

One-dimensional arrays are automatically enlarged if an attempt is made to access beyond the current last element. Once a two dimensional array has been declared, its dimensions cannot be changed. The array can be changed by the following method:

1. Cut a block containing all steps that reference the array to the clipboard (the entire main program or procedure may be cut if necessary).

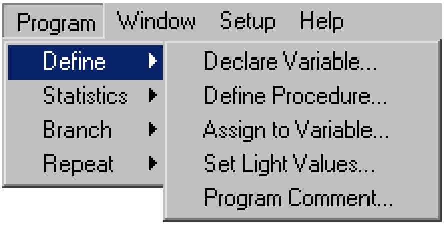
2. Delete the array and re-declare it with the desired new dimensions.

3. Paste the block of steps back into the program.

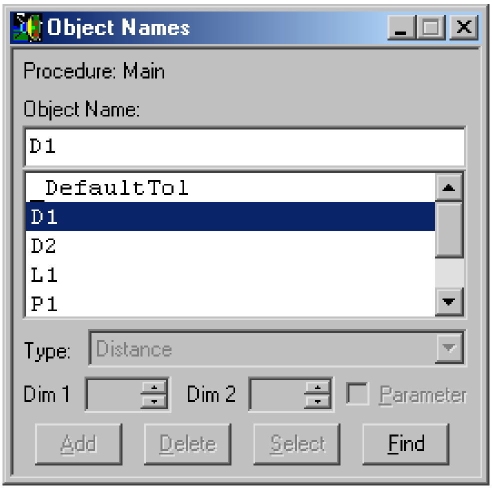
This method cannot be used to change a 1D array to 2D or vice-versa.

Declaring Variables

Variables are declared in VMS using the *Object Names* dialog box.



When you select **Program/Define/Declare Variable**, the *Object Names* dialog box is displayed.



Specify the variable attributes:

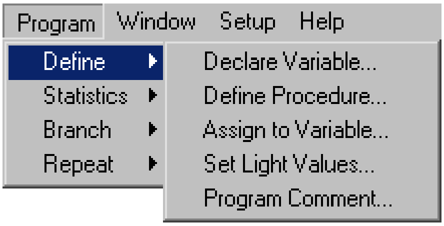
- Variable Name
- Data Type
- Array dimensions (if declaring an array; see *Arrays*)
- Whether or not it is a parameter (see *Defining Procedures*)

A declared variable that is no longer needed may be deleted by selecting **Delete**. This is allowed only if there are no steps in the program that use the variable; otherwise the **Delete** button will be grayed out.

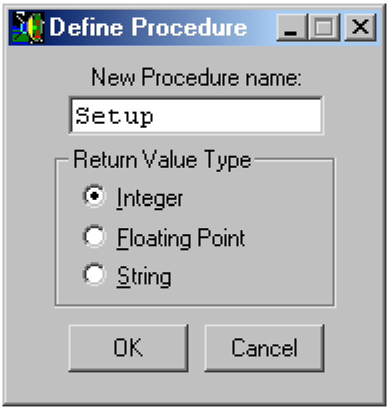
Defining Procedures

You can create your own subroutine-like procedure with the **Define Procedure** command. To define a procedure:

1. Select Program/Define/Define Procedure.

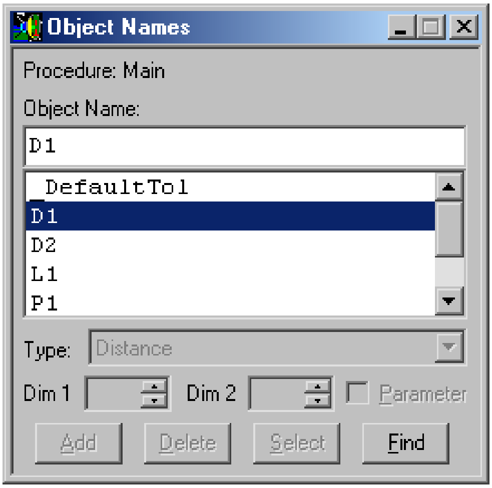


The *Define Procedure* dialog box is displayed.



2. Enter a descriptive name for the procedure you are creating. The naming rules are the same as for variables -- the name must start with a letter, contain only letters, digits, and/or underscores, and be 31 characters or less in length.
3. Specify a return value type based on the kind of result you want returned. If you don't want to return anything, choose integer to return 0. Click **OK**. The empty procedure will be placed after the procedure that currently has the selected (black) step.
4. Click on the procedure in the *Measurement Steps* window.

5. Select **Program/ Define/ Declare Variable** from the menu. The *Object Names* dialog box is displayed.



6. In the *Object Names* dialog box, declare a parameter to this procedure by filling in the appropriate information (name, type, and, if necessary, dimensions) and checking the **Parameter** check box. To avoid confusion, local variables (variables you declare within a procedure) are not allowed to have the same name as variables in the Main program. But you can have local variables with the same name in different procedures; they are completely different variables. **Note for advanced programmers**: local variables are dynamically allocated and you can do recursion. Click the **Add** button. Notice the name of the parameter appears in the first step of the procedure in the *Measurement Steps* window.

You must define all the parameters to the procedure before you declare any non-parameter local variables. Once you have declared a non-parameter local variable in the procedure, you cannot add any more parameters.

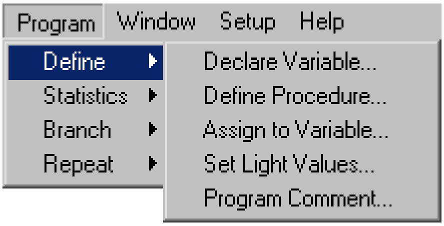
7. Repeat step 6 for each additional parameter. Place a Return step in your procedure, usually at the end, that returns control to where the procedure was called. Provide an expression for what value to return; usually this would be a result calculated in the procedure. If a procedure falls off the end when it runs, it will cause an error message and the program will return all the way back to the main program.

For an example of user-defined procedures, see *Library Procedure Example*.

Setting Light Values

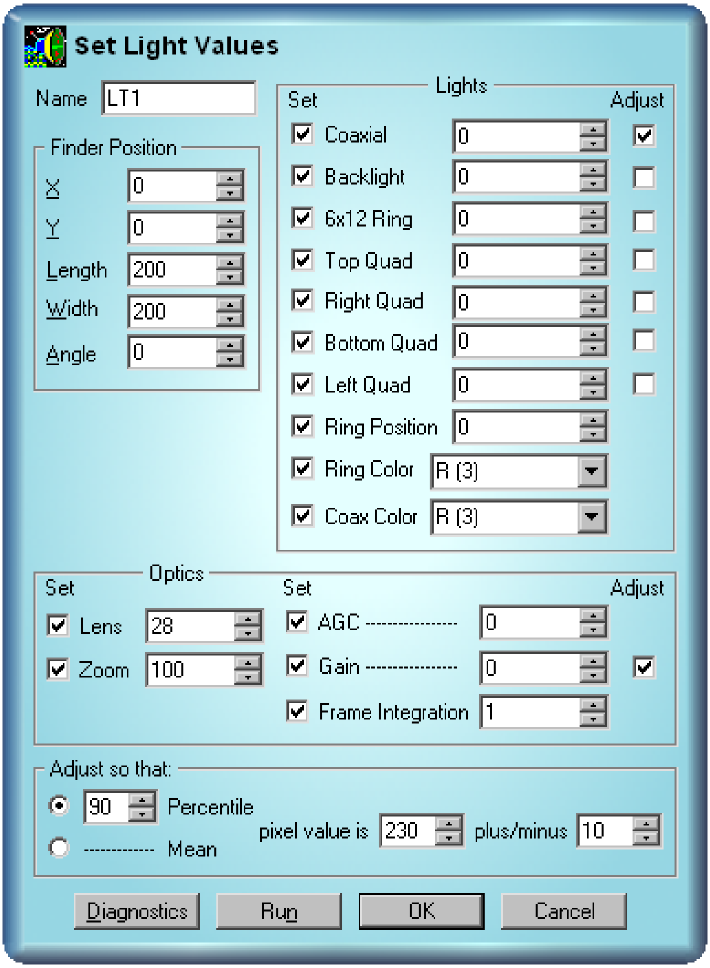
The “Set Light Values” step type and the “Light Settings” VMS structure type are new features of VMS Advanced Programming. The “Light Settings” structure contains light level, AGC, Gain, and Frame Integration values. These are named structure components that can be accessed in VMS (e.g., LT1.BACK, LT1.COAX, etc.)

Select **Program/Define/Set Light Values** or click .



The dialog box displayed for creating or editing a “Set Light Values” step is shown below.

“LT1” is the default name given to the Light Settings variable (structure) that this step fills in.



In the “Adjust so that” section, the user specifies the criteria that determine when the values are adjusted correctly. If the specified criteria cannot be met, a Finder error ("Set Light Values could not obtain target value", error number 98) is produced by the step.

Variables/expressions may be used in any of the edit or combo-box controls in the *Set Light Values* dialog box.

Execution of the “Set Light Values” step first adjusts the specified lights. It then adjusts the camera gain only if the light adjustment failed to meet the specified criteria. In each phase, a histogram is run and its values are compared to the user-specified criteria. If the criteria are not met, the user-specified items are adjusted up or down using a binary-search strategy and a new histogram is run. This is repeated iteratively in each phase until the criteria are met or a value is adjusted to its upper or lower limit.

“Adjust” may be off in all of the checkboxes. In this case, the values, if any, in the “Adjust so that” section are not used and the Light Settings structure created by the step simply names a group of light settings (possibly containing expressions) that can be used in Finders.

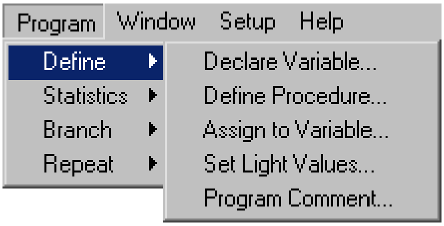
More than one “Set Light Values” step may be used to construct a desired set of light values in a single Light Settings structure. For example, the backlight in LT1 could be adjusted using one “Set Light Values” step and the coaxial or ring light in LT1 could be adjusted using another step. The “Set” checkboxes are normally all on. However, if one or more values in the Light Settings structure have already been set by another step, then the checkboxes for those values may be turned off to preserve the existing settings. In this case, the “Set” checkbox for the backlight would be turned off in the second step to preserve the value set by the first step. The combined set of values may then be used in Finders that need both lights set correctly.

The capability to specify expressions for lights, AGC, Gain, and Frame Integration values in a Finder is made obsolete by the new Light Settings structure and “Set Light Values” capability. Instead, a “Light Settings” structure may be selected in the *Advanced Finder Parameters* dialog for use by the Finder. Existing Measurement Programs that contain individual expressions for the lights within a Finder will still work and will bring up the old style dialog box with the individual expressions.

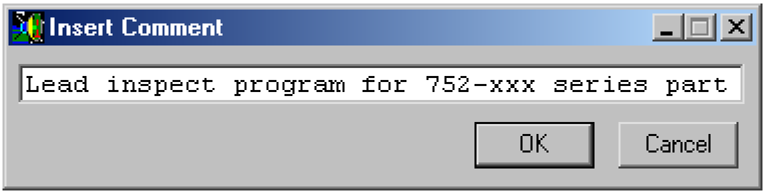
Adding Comments

Comments may be added to the program to aid you or others in understanding what various parts of the program do.

Select **Program/Define/Program Comment**.



The following dialog box is displayed:



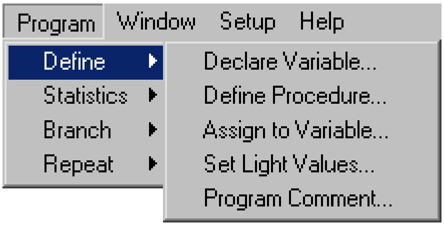
Enter the text for the comment. When you select **OK**, the comment will be added to the program on a line that begins with several asterisks to distinguish it from other program steps.

Existing program steps may also be “commented out”; see *Inserting, Moving and Deleting Steps*.

Advanced Programming Steps

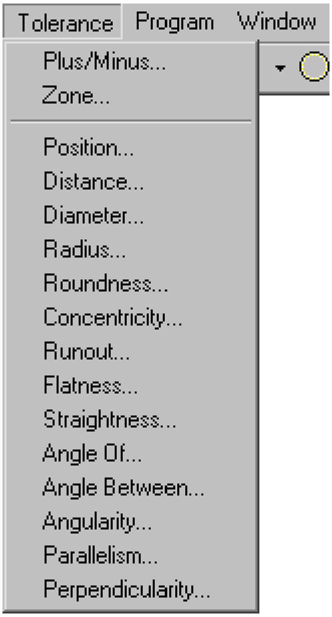
Program Menu

Many of the Advanced Programming step types are available through the **Program** menu.

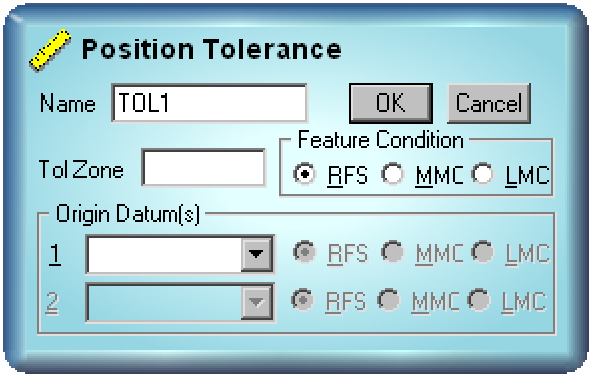


Tolerance Menu

1. Select the **Tolerance** menu. The **Tolerance** menu and all its options are only available with the Advanced Programming component.



Each selection under the tolerance menu has a dialog box that contains selections for feature condition and origin datums. An example of one is shown here:

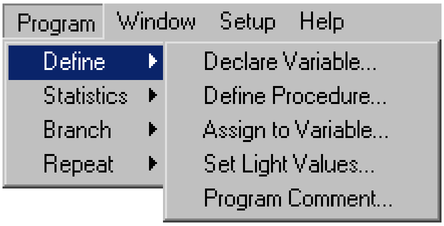


Condition	Description
RFS	Regardless of feature size.
MMC	Maximum material condition.
LMC	Least material condition.

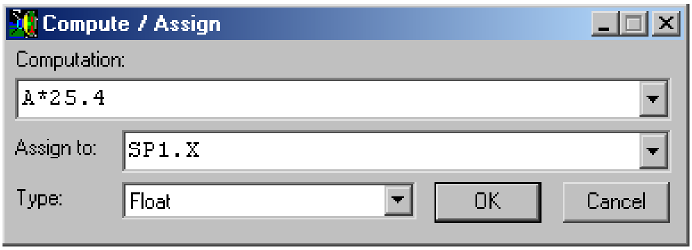
Assignment

The Assign to Variable step lets you enter an expression and a variable name. The expression is evaluated and the resulting value is assigned, or put into, the variable. For example, you can have the value of A*25.4 assigned to X. This shows in the *Measurement Steps* window as X = A * 25.4.

Select **Program/Define/Assign To Variable**.



The **Compute/Assign** dialog box is displayed.



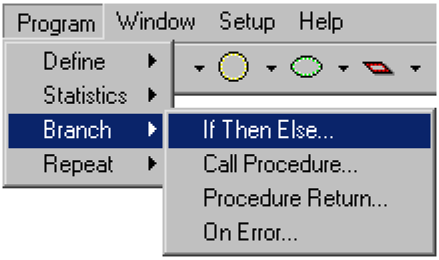
The **Computation** and the **Assign to** variable must be of the same type (integer, float, or string). **Type** may be selected first to show only variables of that type in the **Assign to** box. To aid the programmer in entering expressions, the **Computation** and **Assign to** fields have a white background when the expression is correct and a yellow background when it is not.

The **Assign to** target can also component of a structure variable or an array element expression. For example, you can assign A*25.4 to SP1.X (where SP1 is a Simple Point) or to Length[n] (where Length is an array of floating point values and n is an Integer.)

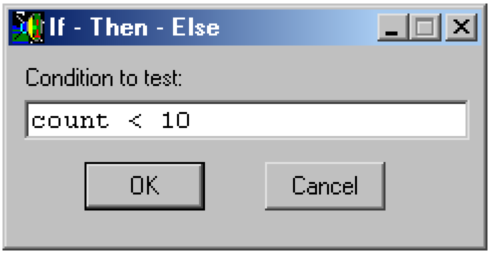
You can also assign the value of one structure to another, as in C3 = C4. However, in the case of Feature variables like this, the point buffer is not copied (C3 would have no points in its buffer even if C4 did.) If the variable you name appears to be a simple variable but has not been declared, VMS asks if you want to declare it automatically.

If-Then-Else

Access the *If-Then-Else* dialog box from the **Program/Branch** menu.



To create an **If-Then-Else** block, you must enter a Boolean expression. You get a 3-step block (**If Then**, **Else**, and **End If**) into which you can insert steps. At run-time, if the result of the Boolean expression is non-zero (true), the steps between **If** and **Else** will be executed; if the result is zero (false), the steps between **Else** and **Endif** will be executed.



A simple example:

If N < 0 Then

resultout("N was negative"+nl())

Else

resultout("N was zero or positive"+nl())

End If

For examples using If-Then-Else programming see:

- Random Selection Example
- Ball Grid Step & Repeat Example
- Library Procedure Example

While-End While

To create a **While-End While** block, you enter a Boolean expression. You get a 2-step block into which you can insert steps. At run-time, if the result of the expression is non-zero (true), the steps between **While** and **End While** will be executed, then control will jump back to the top and test the condition again. It will jump to the first step after the **End While** when the expression evaluates to zero (false).

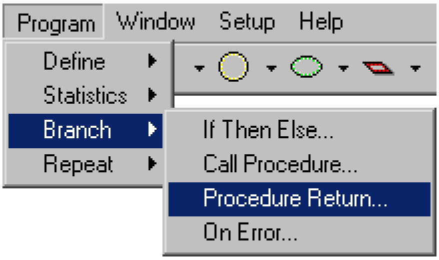
```
BottlesOfBeer = 99

While BottlesOfBeer > 0

    Bottle = TakeDown(1)

    PassAround(Bottle)

    BottlesOfBeer = BottlesOfBeer - 1

End While
```

For examples using While-End While programming, see:

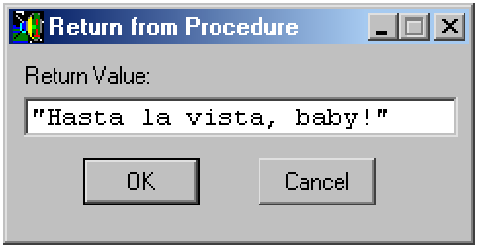
- Random Selection Example
- Library Procedure Example

Procedure Return

To return from a procedure to the main program or another procedure, you can use the *Return From* dialog box. It is accessed via the **Program/Branch** menu.

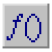


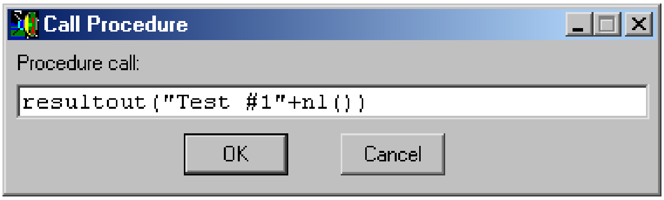
An Expression for the return value is entered. It must match the type selected when the procedure was declared.



An example of a Return from Procedure step is given in the Library Procedure Example.

Call Procedure

Sometimes you want to call a VMS Function or a User-Defined Procedure but you don't want to use the result value it returns. One example of this is the resultout() function. It allows you to send something to the *Results* window, but it does not have any relevant value to return (it just returns the integer 0). A Call Procedure step allows you to give an expression that will be evaluated, but its value will be discarded. To access the *Call Procedure* dialog box select **Program/Branch/Call Procedure** from the menu or click the **Call Procedure** button  on the toolbar.

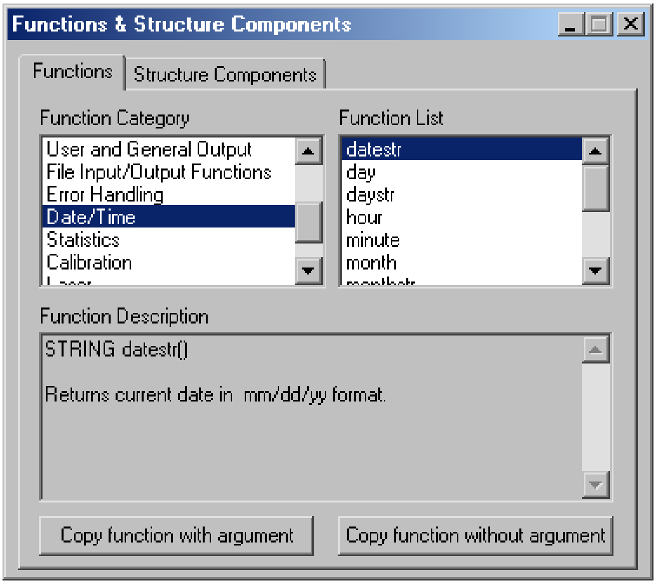


For examples, see:

- See *File I/O Examples* (calling VMS functions)
- See *Library Procedure Example* (calling user-defined functions)

VMS Functions

A description of all VMS functions can be accessed via the **Help** menu or by clicking on the function help button  in the *General Assistance Group* of the toolbar. The *Function* window shows the function category, function list, function description and examples. It also provides buttons to copy the function name, with or without an argument to the clipboard. You may then paste the function to any editable text box. You may browse the function list by scrolling up or down. The items are listed alphabetically. You may also search for specific functions by typing in the first letter of the function name in the **Function List**.



Algebraic

Function	Description
INTEGER **abs**(INTEGER number)	Returns absolute value of number.
FLOAT **fabs**(FLOAT number)	Returns absolute value of number.
FLOAT **float**(INTEGER number)	Returns number as a FLOAT.
INTEGER **round**(FLOAT number)	Returns the closest whole number to number (that is, a fractional value of 0.5 or higher is treated as a whole number).
INTEGER **chop**(FLOAT number)	Chops off fractional part of number and returns what's left.
FLOAT **log**(FLOAT number)	Returns base 10 logarithm of number.
FLOAT **exp**(FLOAT number)	Returns 10 raised to the power of number (inverse of log).
FLOAT **sqrt**(FLOAT number)	Returns square root of number.

Trigonometric

Function	Description
FLOAT **sin**(FLOAT angle)	Returns sine of angle (angle is in degrees).
FLOAT **cos**(FLOAT angle)	Returns cosine of angle (angle is in degrees).
FLOAT **tan**(FLOAT angle)	Returns tangent of angle (angle is in degrees).
FLOAT **asin**(FLOAT value)	Returns the arcsine of value, that is the angle whose sin is value.
FLOAT **acos**(FLOAT value)	Returns the arccosine of value, that is the angle whose cos is value.

Function	Description
FLOAT **atan**(FLOAT X, FLOAT Y)	Returns the arctangent of (Y/X), that is the angle whose tan is Y/X. This form of the function is handier, since it gives desired results if Y is 0.

Coordinate Transformation

Function	Description
INTEGER **PCStoMCS**(SIMPOINT point, PCS pcsvar)	This changes point in place, converting it from PCS to MCS according to the PCS in pcsvar. If you want to keep the original value of point, make a copy of it and pass the copy to this function. Returns 0.
INTEGER **MCStoPCS**(SIMPOINT point, PCS pcsvar)	This changes point in place, converting it from MCS to PCS according to the PCS in pcsvar. If you want to keep the original value of point, make a copy of it and pass the copy to this function. Returns 0.

String

Function	Description
STRING **upper**(STRING source)	This returns a string identical to source except that all alphabetic characters have been made uppercase according to the rules of the operating system.
STRING **lower**(STRING source)	This returns a string identical to source except that all alphabetic characters have been made lowercase according to the rules of the operating system.
STRING **left**(STRING source, INTEGER numchars)	This works like the Basic Left$function, returning the first numchars characters of source.

Function	Description
STRING **right**(STRING source, INTEGER numchars)	This works like the Basic Right$function, returning the last numchars characters of source.
STRING **mid**(STRING source, INTEGER start, INTEGER numchars)	This works like the Basic Mid$function, returning the numchars characters of source beginning at the zero-based offset start.
STRING **trim**(STRING source)	This returns source with any leading or trailing white space removed.
STRING **ltrim**(STRING source)	This returns source with any leading white space removed.
STRING **rtrim**(STRING source)	This returns source with any trailing white space removed.
INTEGER **strlen**(STRING target)	This returns the number of characters in target.
INTEGER **instr**(STRING target, STRING search)	This returns the zero-based offset in target where you can find the string search. This must be an exact case-sensitive match. If search is not found instr returns -1.
INTEGER **asc**(STRING source)	This returns the numeric equivalent of the first character in source. The coding will be determined by the operating system. In Windows, this will be the ANSI equivalent of the character.
STRING **chr**(INTEGER charcode)	This returns a one-character string that is the character equivalent of charcode. The number will be interpreted according to the operating system. In Windows, charcode will be treated as an ANSI code.

Function	Description
STRING **fill**(INTEGER count, INTEGER charcode)	This returns a string consisting of the same result as chr(charcode) repeated count times, except that if charcode is 0, chr does nothing, but fill will return a string with count null characters (but if something is appended to the string later, the null characters will disappear). If count is not positive, fill will return an empty string and will set a global error variable that can be retrieved with the error functions.
STRING **iformat**(INTEGER number, STRING format)	This returns a string representation of the number as formatted, according to the format string. The format string options are a subset of the format string options in the Basic programming language. If the format string is not correct, an empty string will be returned and a global error variable will be set that can be retrieved with the error functions. For information on the formal syntax of the format strings, see *Numeric Formatting with iformat and fformat*.
STRING **fformat**(FLOAT number, STRING format)	This returns a string representation of the number as formatted, according to the format string. The format string options are a subset of the format string options in the Basic programming language. If the format string is not correct, an empty string will be returned and a global error variable will be set that can be retrieved with the error functions. For information on the formal syntax of the format strings, see *Numeric Formatting with iformat and fformat*.
INTEGER **ival**(STRING numericstring)	This returns the integer represented by the string. If the string doesn't parse correctly into an integer, it returns zero and will set a global error variable that can be retrieved with the error functions.

Function	Description
FLOAT **fval**(STRING numericstring)	This returns the float represented by the string. If the string doesn't parse correctly into an integer, it returns zero and will set a global error variable that can be retrieved with the error functions.
STRING **nl**()	This returns a string consisting of a carriage return followed by a new line ("\r\n").
STRING **np**()	This returns a string consisting of a carriage return followed by a form feed or new page ("\r\f").
==, !=, >, <, >=, and <=	These comparison operators make case-insensitive comparisons according to the string sorting rules defined by the language settings in the international category of the Windows Control Panel.
STRING **programname**()	Returns the name of the part program.
STRING **headeritem**(STRING itemname)	Returns a header data item entered by the user in the *Results Header* dialog box (that can be displayed when the program starts). The standard item names are: "Company", "Machine", "User:", "Part Name:", "Part #", "Lot #", and "Comment:". The last five item names are configurable in the *Edit Results Header* dialog box. The configurable item names are available as items "Header1"..."Header5". So the assignment \`item1 = headeritem("Header1")' will set item1 to either "User:" or whatever that item name was changed to in the *Edit Results Header* dialog. Then headeritem(item1) will return the data entered by the user in that field of the *Results Header* dialog box.

User and General I/O Functions

Function	Description
INTEGER **iprt**(INTEGER value)	This sends value to the result window and/or file and returns value. To specify a format for the integer, use iformat (above) with resultout (below). Returns the input value.
FLOAT **fprt**(FLOAT value)	This sends value to the result window and/or file and returns value. To specify a format for the integer, use fformat (above) with resultout (below). Returns the input value.
INTEGER **timprt**()	This sends the current time (for example, 13:24:59) to the result window and/or file and returns 0.
INTEGER **crlf**()	This sends a CR/LF (new line) to the result window and/or file and returns 0. See STRING **nl**().
INTEGER **resultout**(STRING output)	This sends output to the result window and/or file and returns 0.
INTEGER **ddeout**(STRING output)	This sends output to DDE (Dynamic Data Exchange protocol) via the SPC_ASCII item and returns 0. (This allows the system to communicate with other Windows applications.)
INTERGER **msgbox**(STRING message)	This displays ‘message’ in a message box and returns 1.

Function	Description
INTEGER **msgboxex**(STRING output, STRING title, INTEGER style)	This sends output to a message box with the title ‘title’ and a message box style determined by ‘style’ and returns an integer representing the button the user pressed to dismiss the message box. The value of ‘style’ is the sum of values from the following categories:

Category	**Value**	**Description**
Buttons	0	Display OK button only
	1	Display OK and Cancel buttons
	2	Display Abort, Retry, and Ignore buttons
	3	Display Yes, No, and Cancel buttons
	4	Display Yes and No buttons
	5	Display Retry and Cancel buttons
Icons	16	Display a Stop Sign icon
	32	Display a Question Mark icon
	48	Display an Eclamation Point icon
	64	Display an Information icon
Default Button	0	First button
	256	Second button
	512	Third button

<table>
<tr><th>Function</th><th colspan="3">Description</th></tr>
<tr><td rowspan="8"></td><td>Modality</td><td>0

4096</td><td>VMS cannot continue until the message box is dismissed

Nothing on the computer can continue until the message box is dismissed</td></tr>
<tr><td colspan="2">Return Value</td><td>Description</td></tr>
<tr><td colspan="2">1</td><td>OK button selected</td></tr>
<tr><td colspan="2">2</td><td>Cancel button selected</td></tr>
<tr><td colspan="2">3</td><td>Abort button selected</td></tr>
<tr><td colspan="2">4</td><td>Retry button selected</td></tr>
<tr><td colspan="2">6</td><td>Yes button selected</td></tr>
<tr><td colspan="2">7</td><td>No button selected</td></tr>
<tr><td>STRING userinput(STRING prompt)</td><td colspan="3">This function opens a dialog box to allow the user to input a string, and then returns the string. The prompt is displayed in the dialog box. This function is also used to input a string sent by a client program using COM or DDE.</td></tr>
<tr><td>INTEGER show_userinput(INTEGER show)</td><td colspan="3">This function shows or hides the userinput dialog box. If show is zero, the userinput dialog box is hidden for use with COM or DDE. If show is nonzero, the userinput dialog box is displayed for the user.</td></tr>
<tr><td>INTEGER inbit(INTEGER address, INTERGER bit)</td><td colspan="3">This function reads one bit of an input port.

address – port address (0 through 255)

bit – which bit (0 through 7). Bit 0 is the least significant bit, 7 the most.

Return value is the bit read from the port, 0 or1.</td></tr>
</table>

Function	Description
INTEGER **outbit**(INTEGER address, INTEGER bit, INTEGER value)	This function changes the value of one bit on one of the output ports. The other bits on that port will retain their values. So this function saves you from having to remember what you last output on the other bits. address - port address (0 through 255). bit - which bit (0 through 7). Bit 0 is the least significant bit, 7 the most. value - 0 or 1, what to change the bit to. Return value is an error code; 0 means no error. **Note:** This function used to be named outport. If you have any programs that used that function, you will see that its name is automatically changed to outbit under this version of software.
INTEGER **inbyte**(INTEGER address)	This function reads all 8 bits of an input port; address – port address (0 through 255). Returns the value read from the port, 0-255.
INTEGER **outbyte**(INTEGER address, INTEGER value)	This function sets all 8 bits of an output port at once. address - port address (0 through 255). value - value to output on the selected port (0 through 255). Return value is an error code; 0 means no error.

File Input/Output Functions

<table>
<tr><th>Function</th><th colspan="2">Description</th></tr>
<tr><td>INTEGER close(FILE filevar)</td><td colspan="2">This closes a file.</td></tr>
<tr><td>INTEGER eof(FILE filevar)</td><td colspan="2">Returns non-zero if EOF has been encountered.</td></tr>
<tr><td rowspan="7">STRING excel(STRING argument)</td><td colspan="2">Data transfer to Excel, using command keywords CREATEBOOK, CREATESHEET, SETCELL, APPEND, CLOSEBOOK.
Excel function usage:</td></tr>
<tr><td>CREATEBOOK book</td><td>create or open an Excel workbook</td></tr>
<tr><td>CREATESHEET sheet</td><td>create or open a Excel worksheet</td></tr>
<tr><td>SETCELL row col data</td><td>set cell value</td></tr>
<tr><td>APPEND</td><td>reset row index for appending</td></tr>
<tr><td>CLOSEBOOK</td><td>save and close a workbook</td></tr>
<tr><td colspan="2">APPEND command also returns the last used row index</td></tr>
<tr><td>FLOAT getpos(FILE filevar)</td><td colspan="2">Gets the file read/write position. The position is numbered from 0 thru (file length-1). Integer types take up 2 bytes, float 8 bytes and strings per string size.</td></tr>
</table>

Function	Description
INTEGER **imageio**(STRING filename, INTEGER save, INTEGER quality)	If save = 0, loads a video image from a bmp file filename. If save = 1, saves a video image to a bmp file filename. If save = 2, loads a video image from a jpg file filename. If save = 3, saves a video image from a jpg file filename. The 3rd parameter quality is only used in jpg file saving process.
INTEGER **iostatus**(FILE filevar)	Returns error code of last operation on file.
INTEGER **open**(FILE filevar, STRING namestring, STRING modestring)	namestring - name of the file. modestring: “r” for reading, “w” for overwriting, or “a” for appending. This opens a file.
INTEGER **readchar**(FILE filevar)	Reads 1 byte (to int).
FLOAT **readfloat**(FILE filevar)	Each float type variable is made up of 8 bytes. Reads 8 bytes from the file.
INTEGER **readint**(FILE filevar)	Each integer has two bytes. This function therefore will read in 2 bytes from the file.
STRING **readitem**(FILE filevar)	Reads to next space, comma, tab, null, or newline. Useful with text files only.
STRING **readline**(FILE filevar)	Reads to next newline only. Useful with text files only.
STRING **readstring**(FILE filevar, INTEGER bytecount)	Reads given number of bytes, but the returned string will be shorter if there were embedded nulls.
INTEGER **setpos**(FILE filevar, FLOAT pos)	Set the file read/write pointer position. The position is numbered from 0 thru (file length-1) bytes. Integer types take up 2 bytes, float 8 bytes and strings per string size.

Function	Description
INTEGER **view3dio**(STRING filepath, INTEGER mode)	If save = 0, loads a View3D file to View3D window. If save = 1, saves View3D data in memory into a file. If fails, it will return 1. If succeeds, it will return 0.
INTEGER **writechar**(FILE filevar, INTEGER char)	Writes 1 byte from ‘char’.
INTEGER **writefloat**(FILE filevar, FLOAT float)	Each float type variable is made up of 8 bytes. Writes 8 bytes to the file.
INTEGER **writeint**(FILE filevar, INTEGER int)	Each integer has two bytes - this will write 2 bytes to the file.
INTEGER **writestring**(FILE filevar, STRING string)	Writes a string.

Error Handling

Function	Description
INTEGER **abort**(INTEGER errcode)	This function causes an error with the error code ‘errcode’ on purpose and always returns 0.
INTEGER **errline**()	This function returns the line number that the last error occured on.
STRING **errmsg**(INTEGER errno)	This returns an error message string that explains the error represented by the error code in errno.
INTEGER **errno**()	This returns the integer code represented the last error. If there was no error, the returned value will represent a non-error condition.

Function	Description
INTEGER **get_error_mode**(INTEGER category)	This returns the current error mode for a given error category.
INTEGER **restore_error_mode**()	These allow you to save everything (all categories, all modes including GoTo) and restore it, so that the error handling may be temporarily changed and put back. These can be nested 10 deep. Returns 0.
INTEGER **save_error_mode**()	These allow you to save everything (all categories, all modes including GoTo) and restore it, so that the error handling may be temporarily changed and put back. These can be nested 10 deep. Returns 0.
INTEGER **set_error_mode**(INTEGER category, INTEGER mode)	category: 0 = Computation, 1 = File I/O, 2 = Finders, 3 = Fail. mode: 0 = Abort, 1 = Continue, 2 = Abandon. This function is an alternative to the On Error step. The GoTo mode may not be set by this function because it would require a step number, and as a general function it would not be practical for Voyager to find it to update the step number when editing caused the target step to move. The Abandon mode is only with with with the Finders error type. Returns 0.

Date/Time

Function	Description
STRING **datestr**()	Returns current date in mm/dd/yy format.
INTEGER **day**(FLOAT utctime)	Returns 1 through 31.
STRING **daystr**(FLOAT utctime)	Returns "Monday", "Tuesday", ... (may be abbreviated with substr).
INTEGER **hour**(FLOAT utctime)	Returns 0 through 23.
INTEGER **minute**(FLOAT utctime)	Returns 0 through 59.
INTEGER **month**(FLOAT utctime)	Returns 1 through 12.
STRING **monthstr**(FLOAT utctime)	Returns "January", "February", ... (may be abbreviated with substr).
FLOAT **second**(FLOAT utctime)	Returns 0.0 through 59.999.
FLOAT **time**()	Returns the system time (UTC) in seconds since Jan. 1, 1970 including fractional seconds. This can be used for timing intervals in the program.
STRING **timeformat**(FLOAT utctime, STRING format)	Returns date/time components formatted according to the format string using the standard “C” strftime() function (see a “C” programming manual for details about the format string).
STRING **timestr**()	Returns current time in 24 hr hh:mm:ss format.
INTEGER **year**(FLOAT utctime)	Returns the year (for example, 1997).

Statistics

Function	Description
FLOAT **statmax**(STRING label)	Returns the largest of the measurements.
FLOAT **statmean**(STRING label)	Returns the mean of the measurements.
FLOAT **statmin**(STRING label)	Returns the smallest of the measurements.
INTEGER **statN**(STRING label)	Returns the number of measurements.
FLOAT **statsd**(STRING label)	Returns the standard deviation (1X) of the measurements.

Calibration

Function	Description
INTEGER **calibrate_lens**(STRING lensname, INTEGER lenstype, FLOAT xpixel)	Creates a lens calibration of the specified name and type if it does not exist. If 'lensname' is an empty string, the type description string is used as the name. The X pixel size of the lens is set to ‘xpixel’ (the nominal value for the specified lens type is used if ‘xpixel’ is zero.) The Y pixel size is set to the X pixel size multiplied by the camera aspect ratio. This function works only on single-lens systems and will return an error code if called when the system is configured with a lens turret.
FLOAT **compute_x_pixel**(FLOAT known_x_size, FLOAT measured_x_size)	Computes the current X pixel size using a feature's known size in X and its size in X as measured with the current calibration.
INTEGER **get_image_size**(INTEGER axis)	Gets the current image size for the specified axis (0=X, 1=Y).
FLOAT **get_pixel_size**(INTEGER axis)	Gets the current pixel size for the specified axis (0=X, 1=Y).

Function	Description
INTEGER **get_zoom_id**()	Gets zoom id for the current optics. Possible return values: 1 (standard optics) 2 (turret) 100 (dual mag, low mag view) 101 (dual mag, high mag view)
STRING **lensname**()	Returns the name of the lens currently in use.
INTEGER **lenstype**(STRING lensname)	Returns the integer type of the named lens (zero if not found).
INTEGER **select_lens**(STRING lensname, INTEGER lenstype)	Selects a lens (makes it the current lens.) If 'lensname' is an empty string, the lens is selected by type; otherwise, the lens is selected by name and 'lenstype' is ignored. The return value is zero if the specified lens calibration exists and can be selected. This function works only on single-lens systems and will return an error code if called when the system is configured with a lens turret.
INTEGER **set_pixel_size**(FLOAT xpixel)	Sets the X pixel size of the current lens to the given value. The new size is saved in the registry for the named lens calibration that is currently being used, but not as the default for lenses of that type. The Y pixel size is set to the X pixel size multiplied by the camera aspect ratio. The return value is always zero.

Laser

Function	Description
FLOAT **get_laser_offset**(INTEGER axis)	Gets the offset of the current laser sensor from the camera FOV center in the specified axis (0=X, 1=Y, 2=Z).
FLOAT **get_laser_res**()	Returns theoretical laser resolution in mm per A/D unit.
INTEGER **select_laser**(STRING lasername)	Select named laser calibration for further use. Name must match exactly. The return value is zero if the specified laser calibration exists.
INTEGER **set_laser_offset**(INTEGER axis, FLOAT offset)	Sets the offset of the current laser sensor from the camera FOV center in the specified axis (0=X, 1=Y, 2=Z) The return value is always zero.
INTEGER **set_laser_res**(FLOAT cal)	Sets theoretical laser resolution in mm per A/D unit. The return value is always zero.
INTEGER **SetScanParams**(FLOAT X, FLOAT Y, FLOAT Z, FLOAT Len, FLOAT Vel, FLOAT XYAngle, FLOAT ZAngle, FLOAT ResultSpacing, INTEGER Mode)	The function returns an integer that is always zero. All the parameters are floating point expressions except Mode is an integer. X, Y, Z: The FCS-relative coordinates of the center of the scan, in millimeters. Len: The length of the active part of the scan, in millimeters.

Miscellaneous

Function	Description
FLOAT get_4th_axis()	Returns the current position of the 4th axis in degrees (including fractions of a degree).
INTEGER get_lamp_type(INTEGER group, INTEGER channel)	group - which type of light is being checked 0=Coaxial Light 1=Ring Light (single or PRL) 2=Back Light channel - 0-2 Incandescent are on channel 0 only. LED lights may have 2 or 3 colors on channels 0-2. Returns lamp type code 0 = no lamp 1 = Incandescent 3 = Red LED 6 = Green LED 7 = Blue LED 9 = Red LED (backlight only) 900 = single ring incandescent
INTEGER launch(STRING application, STRING arguments, INTEGER winmode, INTEGER timeout)	This function launches an application outside of VMS.

Function	Description
INTEGER move_4th_axis(FLOAT angle)	Makes the 4th axis move to the commanded angle. The parameter is in degrees. For example, to move to 7 and a half degrees, or 7 degrees 30 minutes, use a parameter of 7.5. Returns 0 if OK, 1 if error or timed out (over 10 seconds).
FLOAT rnd()	Returns a random number between 0.0 and 1.0.
INTEGER set_AMF(INTEGER mode, INTEGER resultmode, INTEGER xsize, INTEGER ysize, INTEGER xspacing, INTEGER yspacing, INTEGER medianfiltsize, INTEGER smoothfiltsize)	This function sets parameters for Area Multi-Focus. When AMF is on, autofocus finders return Area Multi-Focus results. mode - on or off 0 = AMF off 1 = AMF on resultmode - type of results returned 0 = text for View3D 1 = general purpose text 2 = binary xsize - width of summation area in pixels ysize - height of summation area in pixels xspacing - X spacing of result grid in pixels yspacing - Y spacing of result grid in pixels medianfiltsize - number of neighboring pixels to use for the median filter (0 = no filter) smoothfiltsize - number of neighboring pixels to average for smoothing (0 = no filter)
INTEGER sleep(FLOAT time)	This function makes the program pause 'time' seconds. It always returns 0.

Function	Description
FLOAT temp(INTEGER axis,INTEGER conv)	Returns a temperature. Axis values are: 0=X, 1=Y, 2=ambient. If conv is 0, the returned value is in degrees Centigrade; if conv is 1, the value is in Fahrenheit.

Using Serial (COM) Ports

If the file name you give to an open() function is COM1, COM2, COM3, or COM4 (do not use a colon, e.g., COM3:) then the indicated serial port is opened instead of a disk file. It is opened for both read and write operations. The File variable is then associated with the serial port, so all the read functions that use that File variable input bytes from the serial port, and all the write functions output bytes to the serial port. Here is an example:

IOError = open(File1, "COM2", "9600,e,7,1,CD")

The mode string has a different syntax when opening a serial port. It is used to specify various serial communication parameters. There are five items, separated by commas. You can leave items off the end (e.g., "9600,e,7") but you cannot leave them out of the middle. Items you leave off the end will be given default values. The five items are:

Baud rate: default 9600

Parity: Can be N (none, default), E (even), O (odd), M (mark) or S (space)

Data bits: default is 8; usually 7 is used if parity is E, O, M, or S

Stop Bits: default is 1; the only other value allowed is 2, which is uncommon.

Handshaking: default is none; C=CTS/RTS, D=DTR/DSR, X=Xon-Xoff. You can use any combination. If C is not used, RTS will simply be left on; if D is not used, DTR will be left on.

Thus, this statement

IOError = open(File1, "COM2", "") (you must supply at least an empty mode string)

is equivalent to:

IOError = open(File1, "COM2", "9600,N,8,1")

- Do not use setpos() and getpos() functions on a serial port.
- If the mouse is on COM1, you can't use COM1. The extra serial port is COM2.

Handshaking Notes

Originally, serial ports were used to connect dumb terminals to modems. When you looked at a 25-pin connector, whether a pin was an input or an output depended on whether you were looking on the terminal or the modem. In the lingo, a terminal was a Data Terminal Equipment (DTE) and a modem was a Data Communication Equipment (DCE). Computer serial ports, being often connected to modems, are thus wired as DTEs. If you want to hook two computers together, you can’t connect pin-to-pin because they are both DTEs. So you connect through a simple gadget called a "null modem" which swaps the connections around. The following discussion refers to a computer talking to a modem, but it could be any DTE talking to any DCE.

Either the computer or the modem may be too busy to handle any more input at the moment, so each needs a way to tell the other to shut up for now. This is called "handshaking". There are two separate hardware handshake systems, which are really redundant. They are DTR/DSR and CTS/RTS. There is also a software handshake system called Xon/Xoff.

- The computer activates the "Data Terminal Ready" (DTR) line when it is ready. The modem isn't supposed to send anything unless DTR is active.
- The modem activates the "Data Set Ready" (DSR) line when it is ready. The computer isn't supposed to send anything unless DSR is active.
- The computer activates the "Request To Send" (RTS) line when it is ready. The modem isn't supposed to send anything unless RTS is active.
- The modem activates the "Clear To Send" (CTS) line when it is ready. The computer isn't supposed to send anything unless CTS is active.

Either side can send an Xoff character to tell the other side to stop. This character has the value of 19 decimal, same as control-s. This is why typing control-s is often a way to stop a scrolling output. When it is ready to receive again, it sends an Xon character, which is 17 decimal, same as control-q. Because these character values would not be found in ordinary plain text, it is safe to use this method when only text is being transferred or when dealing with a person typing at a terminal. But whenever binary data are being transferred, some of the bytes may happen to have the value 17 or 19, and so they would trigger the handshaking incorrectly. So Xon/Xoff handshaking must not be used when transferring binary data.

Using CISP Ports

The CISP port I/O functions allow you to input or output data over various digital I/O ports that may be present on the system.

> This does NOT give you access to the I/O port space of the CPU – you cannot control just any device on the PC. This only addresses ports that are located on the **Connector Interface Sub-Panel (CISP)**.

On some systems, port 1 bit 0 is used to control a pneumatic fixture mechanism.

On machines with microscope turret optics and/or through-lens laser focus , the following bits in port 1 control the hardware. **You should not attempt to control this hardware via this function.** It could cause damage to the system. The information is given here so you know what NOT to do!

Bit 1 controls the through-lens-laser.

Bits 4 and 5 are the commanded turret address.

Bit 6 commands a turret move.

Bit 7 commands a turret initialize.

On select systems, the following ports may be present:

Port	Function
03	"Aux" outputs
04	"Opto" outputs on P3
05	"Opto" inputs on P1
06	"Digital" outputs on P14
07	"Digital" inputs on P2
08-15	**For internal use, do not access!**

I/O Time-outs

Any read or write operation will return with an error (error code #54, I/O Incomplete) if more than 10 seconds elapses. When reading, this may be caused by the sending device not sending. When reading or writing, this can be caused by the handshake lines not being connected or used properly. A common cause of this problem when initially setting up a system is forgetting to use a "null modem" when connecting two computers directly to each other.

When using the readitem() or readline() functions, the 10 second limit is reset every time a character is received. For all other functions, all the characters specified must be sent or received in 10 seconds. Note that if you are using an unusually slow baud rate or are sending/receiving an unusually long string, the total time could be more than 10 seconds and you would get an error. For example, sending 10000 bytes at 9600 baud (which is 960 chars/sec) takes 10.4 seconds. But it is unusual to exchange that many bytes in one function call. You could break it up into smaller pieces.

If your situation is such that you expect an "indefinite" wait for receiving data, then you should have a loop that repeats the read operation if it times out. That way, if the data never arrive, you can click on VMS's STOP button in order to stop the program from waiting forever. Or you could put a limit on the loop; for example, after 12 times (which would be two minutes) it could break out and show an error message like "Acme Widget Feeder Not Responding!"

Numeric Formatting with iformat and fformat

The following syntax is used for numeric formatting strings. It is very similar to the syntax used by Basic.

<format string>	:= [<pos template>[;<neg template>[;<zero template>]]]
<pos template>	:= <template>
<neg template>	:= <template>
<zero template>	:= <template>
<template>	:= [<literal string>][<numeric format>][<literal string>]
<literal string>	:= <literal token> [<literal string>]
<literal token>	:= + or - or space or any characters surrounded by double-quotes
<numeric format>	:= [<integer format>][.][<fraction format>]
<integer format>	:= 0 repeated any number of times
<fraction format>	:= [0 repeated any number of times][# repeated any number of times]

- Positive numbers are formatted by the <pos template>.
- Negative numbers are formatted by the <neg template>.
- Zeroes are formatted by the <zero template>.
- If there is no <zero template>, zeroes are formatted by the <pos template>.
- If there is no <neg template>, negative numbers are formatted by the <pos template> with a - prepended to the entire result, including any leading <literal string>.
- If there is no <pos template>, default formatting is used.

A format string may have one, two or three format templates separated by semicolons, for example: "+000.0#;-000.0#;0". The first template will be used to print positive numbers, the second template will be used for negative numbers, and the third template will be used for zero.

Default formatting (using "") prints as many as 15 significant digits with no leading or trailing zeroes. If the number cannot be represented in fixed notation with 15 digits, it is represented in scientific notation. The decimal point is only used if there would be something after it. If the number is between 1 and -1 and is represented in fixed notation, a single zero is put before the decimal point. A - is prepended to negative numbers but a + is not prepended to positive numbers.

With explicit format strings, a decimal point is printed if and only if there is a decimal point in the format template. Any number before the decimal point is printed in full without truncation. If the number of zeroes in the <integer format> is greater than the number of digits before the decimal point, it is padded with leading zeroes. The number of digits after the decimal point is never greater than the total number of zeroes and #s after the decimal in the format template. If there are fewer significant digits after the decimal point than there are zeroes in the format template, the number is padded with trailing zeroes. Explicit format templates never result in scientific notation.

In other words, the format you provide is an "example" for the function to follow. For example, "#.##" would tell it you want to round the number to show no more than two places after the decimal point. With the pound sign (#) format, the number of leading digits is not relevant; it will use as many digits as it takes to show the number, and no more. So "#.##" is equivalent to "###.##". Leading and trailing zeroes are not shown. For example, the number 23.4567 would print as "23.46"; 1.3002 would print as "1.3"; 6234 would print as "6324.". This type of format is useful when embedding numbers in text.

The alternative is to use zeroes, for example "000.00". This example guarantees that there will be three digits, a decimal point, and exactly two digits. 23.4567 would print as "023.46"; 1.3002 would print as "001.30"; and 6324 would print as "6324.00". The last example shows that it will give more leading digits if the number requires it, so you ought to anticipate what the largest number might be in order to be sure how many characters you will get. This type of format is useful when creating tables where you want the columns to line up.

You can combine zeroes and pound signs. For example, 0.0004 printed with "#.##" would give you just a decimal point! You would probably want to use "0.0#", which would print it as "0.0"; this format would print 345.669 as "345.67". You can put pluses, minuses, and blanks before or after the number format characters, e.g., "++ 00.00--". They will simply be printed. You can also supply an empty string, "". In this case it will show as many significant digits as there are, but limit the entire string to no more than 15 digits. If the number is too large or small for that, it will switch to scientific notation.

Area Multi-Focus (AMF)

Area Multi-Focus (AMF) is a new technique in which the Z axis is moved through a distance, as with regular auto focus, but instead of finding the Z height of one surface, it finds the Z heights of many small areas of the image simultaneously. The field of view is divided into a 2 dimensional grid of small areas. The result of AMF is a 2D array of Z values, thus giving a 3D map of the FOV. This should have many applications in areas where laser range sensors have traditionally been used, but it should be faster and in some other ways better. Area Multi-Focus requires a system capable of performing software autofocus.

Area Multi-Focus also returns “strength” values for the areas. The strength value says how much “sharpness” there was in that small area, when it was in focus. This can help to distinguish between “good” and “bad” Z values, and also between different surface textures.

At the time of this writing, AMF has not been developed completely. What is described here is the current implementation in VMS and DGS software.

You run an AMF Finder by first calling the VMS Function set_AMF() to set up the special AMF parameters, including turning on AMF mode. All Autofocus Finders that you run afterwards (commanded in the usual way) will run as AMF Finders, because AMF mode has been turned on. You can make another call to set_AMF() to turn AMF mode off, so that “regular” auto-focus Finders can be run. The call has the following form:

```
set_AMF(INTEGER mode, INTEGER resultmode,

        INTEGER xsize, INTEGER ysize,

        INTEGER xspacing, INTEGER yspacing,

        INTEGER medianfiltsize, INTEGER smoothfiltsize)
```

AMF Parameters

An AMF Finder has two sets of parameters. The first set of parameters for an AMF Finder are those set by the call to set_AMF():

- **mode**: This enables (1) or disables (0) the AMF mode for all subsequent focus Finders. Other values may be defined in the future.
- **resultmode**: This determines the way the AMF results are returned. See the *AMF Results* section below.
- **xsize, ysize**: The dimensions (in pixels) of the small areas. Bigger areas will give answers with less “noise”, but with less XY resolution. Note that the “strength” values are larger if the areas are larger, on the same texture.

- **xspacing, yspacing**: The spacing (in pixels) of the small areas. This, plus the size of the image, will determine the number of points in the returned data. These numbers must be at least 2, in order to keep the returned data to a manageable size. For example, with the MuTech MV-510, the image is 768x574, and if the results are every 2 pixels in X and Y, there will be 110208 points returned! If the spacing is every 8 pixels, a quite useable spacing, there will be 6888 points.

  Note that it is possible to make the small areas overlap or to have gaps between them. Only if the size and spacing are equal will the areas just touch each other without overlapping. Also, note that the first small area is always in the upper left corner of the FOV, so the top and left edges of the FOV are always covered. Depending on the parameters, the last small areas may end before the right or bottom edges of the FOV, so a small space on the right or bottom may not be covered.

- **medianfiltsize**: A median filter removes "spike" noise from the Z data, without seriously reducing resolution. If this parameter is zero, there is no median filtering. If it is 1, then the filter will combine 1 point on each side of each result point; so most points will have been combined with 8 of their neighbors, a 3x3 grid. If it is 2, it will use 2 points on each side, giving a 5x5 grid, etc. This number can be bigger if desired (maximum 10) but it may not be more helpful. No filter is applied to the "strength" data.

- **smothfiltsze**: The smoothing filter simply averages each point with its neighbors (equally weighted). This does reduce the resolution of sharp vertical edges. This filter is run after the median filter. The size value works as with the median filter.

The second set of parameters for an AMF Finder are the usual ones for any auto focus, and are part of the command to actually run the Finder:

- XYZ position of Finder in MCS
- Length of pass
- Velocity of pass (or, in VMS, density)
- Lights, camera gain, etc.

These parameters are sent as in a normal auto focus Finder. Note that the type (edge, surface, grid) and the area (left, right, top, bottom) are not relevant to AMF Finders. It always does the whole FOV, using a mode similar to surface focus.

AMF Results

Because the number of points returned is potentially so huge, AMF does not return its results in the same way as other Finders. For DGS clients, it does not return an IPR_IPU_RESULT structure. For VMS programmers, it does not return them to the Feature's point buffer. The set_AMF() **resultmode** parameter selects one of three ways to return the results.

resultmode=0: Text mode, designed to be displayed in the *View3D* window.

AMF will generate two data files, named AMFResultZ.dat and AMFResultS.dat. Both files contain the same 1-line header:

Number of points in X Number of points in Y Spacing in X (mm) Spacing in Y (mm)

After that, there is one line of text for each row of data. Each line contains one number for each data point in the row. The difference in the files is that AMFResultsZ.dat contains Z positions in mm, *relative to the mid-Z height,* and AMFResultsS.dat contains "strength" values. The format of these files is such that they can be opened by the *View3D* window for a useful visualization of the data.

resultmode=1: Text mode, with absolute positions included.

AMF will generate a single text file, named AMFResultZ.dat. It contains a 2-line header:

Number of points in X Number of points in Y Spacing in X (mm) Spacing in Y (mm)

CornerX (mm) CornerY (mm) Mid-Z (mm)

The extra values are the XY coordinates of the upper-left hand corner point in MCS millimeters, and the mid-Z value (halfway between the lowest and highest points) in MCS millimeters. After the header, the Z data follow as in Text Mode 0, but these are immediately followed by the strength data, also as in text mode 0 but in the same file, without another header.

resultmode=2: Binary mode (for DGS clients only; not to be used from VMS)

Procedure Libraries

About Procedure Libraries

A VMS Procedure Library is a collection of user-defined Procedures (subroutines) that is saved as a separate file, for example “MyHandyProcs.vml”. Measurement programs that you then write can include one or more Library files. The libraries are found, and the procedures in them are loaded, whenever the program is loaded. Thus, a change in a Library Procedure will automatically be effective in all measurement programs that use it when they are later loaded. That is much more efficient than making the same change to every copy of a common Procedure included separately inside many measure programs.

Creating and Saving Library Files

VMS Libraries are created in the same manner as VMS Measurement Programs with Procedures. A VMS Library may or may not have statements in its Main program. If it does, these can be used to initialize global variables and to test the library procedures during development, but **the Main program is ignored when the Library is included by another Measurement Program**.

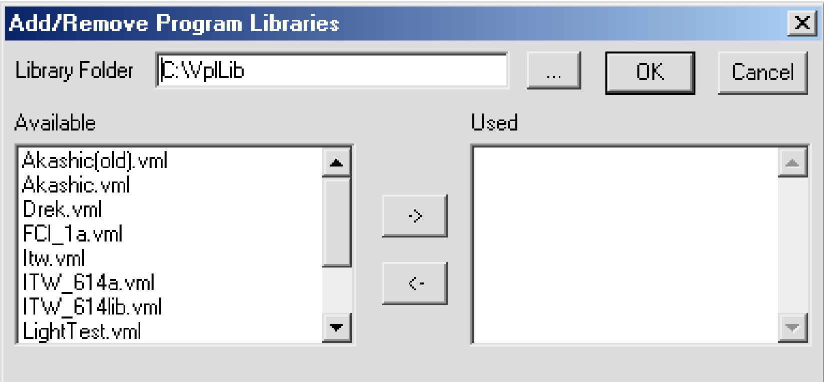
VMS Libraries must be saved with the file extension “.vml”. The new item “Library (*.vml)” has been added to the “Save as type” list in the *Save As* dialog box. Because library file names are added to the names of their components when they are used (explained in the next section), VMS Library file names must follow the rules for VMS identifiers.

Remember, you may have more than one Library (vml file), and each Library may contain more than one Procedure.

Including Libraries in a Program

The following figure shows the *Library Configuration* dialog box. It is displayed from the menu:

File/Add/Remove Program Libraries.



This dialog box serves two functions. The first function of this dialog box is to tell VMS where to look for Libraries. (You may have more than one folder on your system that contains Libraries, but you can only use one folder at a time so you probably will not want to have more than one folder.) In this dialog box, you specify the folder that you want to use.

The second function of this dialog box is to choose which Libraries to include when you are creating or editing a Measurement Program. A list of the Libraries available in the library folder is displayed on the left; by selecting one on the left and clicking the -> button you move it to the right side, which means to actually use that Library in your Measurement Program. Of course, you can use more than one Library. After you change the selection of which Library files to use and close the dialog box, the chosen Libraries will be added to your current Measurement Program. Remember that each Library file may contain several Procedures. You will see their Procedure definition steps appear in your program. The detailed steps of each Library Procedure will be added if and when that Procedure is actually used by something else in your Program; if a Library Procedure is not used you will see only its definition step.

Also, you will notice that the original names of the Procedures have been changed. They now start with the name of the Library file they came from, for example “MyHandyProcs:GetMaximum”.

Why? Well, suppose you used two Libraries, and they both contained a Procedure named “ABC”. That is allowed. VMS changes the name of each Procedure by prepending the name of the Library it came from. Now each Procedure has a unique name (for example, File1:ABC and File2:ABC). That way you can specify which one you are calling.

After libraries have been included, you may use their procedures in Function Calls and Call Procedure statements. If the procedure name exists in only one library, you do not need to type the library name. VMS will add the library name to the procedure name when the calling step is saved.

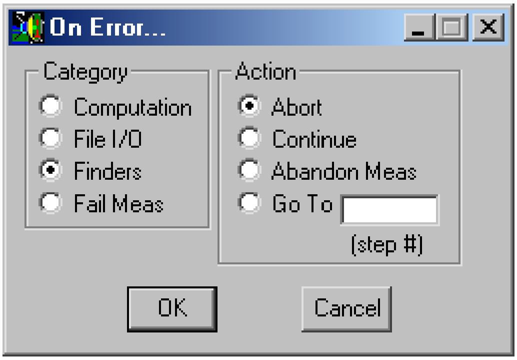
Using Global Variables in Libraries

A Library can be created that uses global variables, that is, variables that are defined as being part of the Main procedure of the Library, not local variables that only exist within a Procedure. There are two reasons you would use a global variable: to share it between Procedures in a Library, or to share it with whatever Main program the Library gets used by. Also, only global variables are “persistent”, and have their values saved when you save a program or Library and restored when you load it later.
If you want to use a global variable only for sharing between Procedures in a Library, and you do not intend for any program to use it, then just declare it in the Main procedure of the Library and give it a normal name, for example MyGlobalVar. It is possible for several Libraries to all have a variable of this sort, with the same name, and they will actually be separate variables so you don’t have to worry that a Procedure from one Library will accidentally change a global variable that belongs to another Library.

If you want to use a global variable and you DO want to access it from programs that use the library, and share it with programs and between Libraries, then declare it in the Main procedure of each Library as above, but give it a name that **starts with a colon**, such as **:MyGlobalVar**. This kind of global variable is shared between the Main program and all Procedures from all Libraries. Of course, there is a possible problem: What if one Library says :MyGlobalVar is an Integer, but another one says it is a Float? Well, then you have a conflict and you will get an error. You must be careful when you use shared global variables!

Error Handling With 'On Error'

Error conditions that arise during a measurement, including Finder problems like "no points found" and focus errors, normally cause the program to stop execution with an error message. To prevent this, you can insert an **On Error** step into the program. The *On Error* dialog box is shown below.



The *On Error* dialog box allows selection of an Error Category and how to handle errors for that error category: Abort, Continue, Abandon Measurement, and Go To step. After an **On Error** step is executed, the chosen error handling mode will be in effect until another **On Error** step is executed.

Error Categories

There are four categories of errors: Finder errors, File I/O errors, Computation errors, and Fail Measurement. The error handling mode is set separately for each category. For example, File I/O errors can be set to Continue mode and Computation errors can be set to Abort mode. A separate **On Error** step is used to set the handling mode for each error category. So, you create steps that look like **On File I/O Error Continue**.

Category	Description
Computation	This includes string errors like trying to extract the 8th character of a 4 character string, arithmetic errors like dividing by zero, etc. A properly written program would normally not incur these errors.
File I/O	This includes attempting to read past the end of a file, couldn't find a given file name, trying to write a read-only file, etc.
Finder	This includes if no points are found, if auto focus could not successfully find a focus point, if the stage position was out of bounds, or if some internal error prevented the Finder from running.
Fail Meas	This is when a Feature or Relation is successfully measured (no Finder errors) but a measured property is outside of specified tolerances. There is only one error code in this category.

Error Handling Actions

One of the following error handling actions may be selected in an **On Error** step:

Mode	Description
Abort	The program never gets to the next step. VMS will display an error message and stop. This is what VMS does by default.
Continue	When an error occurs in a step, the rest of the step is skipped and the next step in the program is executed. The next step should call **errno()** to find out if there was an error and what the error was, to do whatever it likes about the error.
Abandon Meas	This only applies to Finder errors, which can only happen in Feature measurement steps. If a Finder error occurs, then all the measurements on the affected Feature will be considered "Unknown". Their Actual, Deviation, and Out of Tolerance values will be blank in the result output, and "Unknown" will appear instead of the word Pass or Fail. Such measurements will not be counted in VMS's statistics. **Note:** The Feature will internally retain its last successfully measured Actual value(s). If this Feature is subsequently used for other things in the program, such as constructing other Features or Alignments, then it is up to you to add logic to the program to avoid using the Feature further.

Mode	Description
Go To step	When an error occurs in a step, the rest of the step is skipped and it jumps to the step number indicated by the user. The target step would generally be an error handling section of the program. Although you do have to indicate the step-by-step number, the number will be updated automatically as the target step's step number changes due to insertions and deletions. If you cut and paste the error handling code, VMS can't keep track and you need to fix it yourself. But when an **On Error Go To** step is pasted via editing, it will be changed into an **On Error Abort** step; you will have to edit it and supply a valid step number to jump to. This (as with any GoTo in any programming language) should be used carefully, since there are no safeguards against jumping into a different align block, which could have bad consequences. Also, the user should be careful not to create a tangled program organization with this; this is commonly called "spaghetti code".

Old Programs

The default "If Out Of Tolerance" settings are no longer available in the *Default Measurement Options* dialog box entered from the Setup menu. All new steps will be initially created with Continue chosen, so that the global mode will be in effect. And unless overridden by an **On Error** step or a **set_error_mode()** function, all error categories default to Abort, except the Fail Measurement category defaults to Continue. This means old programs will still work the same way they did before.

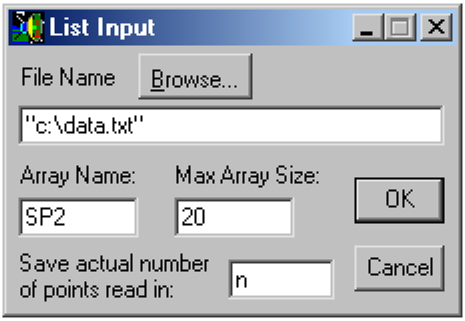
The existing "If Out of Tolerance" section of Measurement Options for a single step will still work to some extent. This affects what happens if a Fail Measurement category error occurs. If the Abort, Pause, or Go To Step mode is selected for a step, it will supersede the current global mode set for the Fail Meas category. (Pause was really the same as Abort. It will no longer appear as a choice.) There are separate error codes (22 & 23) for aborting or jumping because of a per-step action on Fail Meas (different from global Fail Meas error 7).

If Continue was selected for a step, then it will use the current global mode.

List Input Examples

Example 1

1. Select List Input from the Program/Repeat/Step & Repeat menu.
2. Input the appropriate text file name as a string, e.g., “data.text” (this is a string expression, so you must include the quotes) and input the integer value for maximum array size, for example, 10. Also specify an integer variable to receive the number of items in the file, for example, i.



3. Highlight the line that says *Input List 'SP1' from file “data.txt”*, and then measure the desired Feature to set up the first item in the list. In this case the Feature being measured is A1. Double click the line “Set First List Item SP1[0]=unknown” and input the group origin point -- in this example A1.
4. Next select **List/Expr** from the **Program/Repeat/Step & Repeat** menu. Input the location array (SP1). This will move the Arc and all related Features in a group to positions in the array. The SP1 in the array signifies a Simple Point. A Simple Point is a single structure-type variable that contains X, Y, and Z fields. You can declare an array of Simple Points.

5. Input the Features that are being measured in the step and repeat loop. The program will read the offsets of each group in the array from the .txt file. An example file format showing four locations with X, Y, and Z for each is listed below:

   3.876, 4.384, 0.034
   3.976 4.384 0.034
   4.076 6.321345 0.177
   4.176 7.39 0.222

Notice that columns need not line up, and spaces and commas are not consistent. You only need two or (in this case) three numbers on each line.

Example 2

Programming Details:

Select Program, Repeat, Step & Repeat and List Input menus. In the *List Input* window, type **Data.txt**.

For the File Name box, a number for Max Array Size box (maximum) and a variable name e.g., NUM for the Save actual number of points read in box. You may leave the step ‘Set First List Item SP1[0] = <unknown>’ as it is or delete it as shown in the program listing. The data file will be assumed to be in the same folder as this Example.voy program file – if not, you may provide the path e.g., “C:\coord\data.txt” (Step 15). You may set a break point (F9 when the *Measurement Steps* window is active) at Step # 16 Input List End and run the program up to that point which will read in the data from the data.txt file and store them in SP1 (simple) point array.

Next pick Program, Repeat, Step & Repeat and List/Expr menu items. At the *List Step & Repeat* window, pick SP1 from the drop down list of Location Array, enter NUM in the Number Of Points box and pick Inches from the Units button and select OK. This will create the Step & Repeat step in the program.

Now, move the lens to the first feature location called out in the data.txt file (1.2,1.0) and create the Measure Circle step. The CIR is an array of type Circle and in List Step & Repeat, we always use .I index as we do in Polar Step & Repeat. This completes the creation of the List Expression Step & Repeat logic.

Program Listing:

Data.txt (X, Y Coordinates file)

```
1.2,1.0
1.3,1.0
1.7,1.0
2.1,1.2
2.5,1.9
3.4,2.5
```

and so on…

Example5.voy

```
Program Main
Align Define 'XY_ORIGIN'
  Measure Point 'P1'
  Define Align PCS 'XY_ORIGIN' Using P1
Align Use 'XY_ORIGIN' As Default DRF
  Align Define 'Z_ORIGIN'
    Measure Point 'Z1'
    Define Align PCS 'Z_ORIGIN' Using Z1
  Align Use 'Z_ORIGIN' As Default DRF
    Align Define 'ROTATION'
      Measure Point 'P2'
      Construct Best Fit Line 'X_AXIS' Using P1, P2
      Define Align PCS 'ROTATION' Using X_AXIS
    Align Use 'ROTATION' As Default DRF
```

```
14        "Inspect 100 holes with their X,Y locations available in a data file
          (data.txt)"
15        Input List 'SP1, NUM' From File "Data.txt"
16        Input List End
17        Step & Repeat Begin 'SR2'
18          Measure Circle 'CIR[SR2.I + 1]'
19        Step & Repeat End
20      Align End
21    Align End
22  Align End
```

Random Selection Example

Below is an example program that inspects randomly selected part locations using variable assignment, branching, looping, **rnd()**, and **msgboxex()** functions.

Programming Features Used:: Step and Repeat, If-Then-Else, Assignment, While-End While, arithmetic, logical and relational Expressions.

Programming Details:

The **msgboxex()** function at Step # 1 uses 36 as the last argument of the function. This number is a sum of

Values from four categories: buttons (Display Yes and No button with a value of 4), Icons (Display A Question Mark Icon – value 32), Default Button (First button is default – value 0) and Modality (VMS cannot continue until the message box is dismissed – value 0). If Yes is selected by the user, the variable prompt has the value 6. If No is selected, the prompt contains 7.

Steps 3 through 39 implement the logic wherein 5 randomly generated numbers are created and stored in

Ascending order in an integer array called random[].

Steps 57 through 80 use these random numbers in inspecting five parts when so selected by the user.

If not, all 10 parts are inspected.

Program Listing:

```
Program Main
 prompt = msgboxex("Do you want to measure all 10 parts ??", " Measure
 Mode", 36)
 If prompt == 7 Then
  "Randomly Picked 5 parts"
  ************************************************************
  note = "For a random number between 1 and 10, the list is 10"
  note = "number = 5 will inspect randomly 5 parts out of 10. Change it as
  needed."
  list = 10
  number = 5
  ix = 1
   While ix < list + 1
    vals[ix] = 0
    ix = ix + 1
   End While
   ix = 1
   While ix < number + 1
    Rem = "Generate a random number between 1 and ct - reject 0"
    ct = list
    rxx = 0
    While rxx == 0
      xx = rnd() * ct
      rxx = round(xx)
```

```
      If rxx != 0 & vals[rxx] != rxx Then
        vals[rxx] = rxx
      Else
        rxx = 0
      End If
    End While
    ix = ix + 1
  End While
  ix2 = 1
  ix = 1
  While ix < list + 1
    If vals[ix] != 0 Then
      random[ix2] = vals[ix]
      ix2 = ix2 + 1
    Else
    End If
    ix = ix + 1
  End While
  **********************************************************"
Else
  "Insp. All Parts"
End If
Align Define 'XY_ORIGIN'
  Measure Point 'P1'
  Define Align PCS 'XY_ORIGIN' Using P1
```

```
Align Use 'XY_ORIGIN' As Default DRF
  Align Define 'Z_ORIGIN'
    Measure Point 'Z1'
    Define Align PCS 'Z_ORIGIN' Using Z1
  Align Use 'Z_ORIGIN' As Default DRF
    Align Define 'ROTATION'
      Measure Point 'P2'
      Construct Best Fit Line 'X_AXIS' Using P1, P2
      Define Align PCS 'ROTATION' Using X_AXIS
    Align Use 'ROTATION' As Default DRF
      COUNT = 1
      Step & Repeat Begin 'SR1'
        If prompt == 7 Then
          "Randomly Picked Parts"
          If COUNT <= number Then
            PARTNUM = random[COUNT]
          Else
            "Make PARTNUM  something that doesn't equal to SR1.I+1"
            PARTNUM = 11
          End If
        Else
          "Inspect All Parts"
          PARTNUM = SR1.I + 1
        End If
        If PARTNUM == SR1.I + 1 Then
```

```
          "Inspect"
          Measure Line 'L2'
          Measure Line 'L3'
          Measure Angle 'ANGLE[SR1.I + 1]' From L2 To L3
          COUNT = COUNT + 1
        Else
          "Skip"
        End If
      Step & Repeat End
    Align End
  Align End
Align End
```

Ball Grid Inspection Example

Below is a Step and Repeat program example that defines a 2-dimensional array (grid) of Point Features and measures a Ball Grid that has an empty interior in rows and columns 4 through 7.

Programming Features Used:: Step and Repeat, Assignment, If-Then-Else, arithmetic, logical and relational Expressions, defining a plane using an array of points

Programming Details:

SR1.I indexes the Features along the X axis; SR1.J along the Y-axis. Select Program, Define and

Assign To Variable..menu items. In the Compute/Assign dialog box, type **SKIP** in the **Assign To** field and **SR1.I > 2 & SR1.I < 7 & SR1.J > 2 & SR1.J < 7** in the **Computation** field.

When the condition is evaluated to be true, SKIP (integer variable) is set to 1. When the condition is false, SKIP is set to zero.

Program Listing:

```
0 Program Main

1   Align Define 'XY_ORIGIN'

2     Measure Point 'P1'

3     Define Align PCS 'XY_ORIGIN' Using P1

4   Align Use 'XY_ORIGIN' As Default DRF

5     Align Define 'Z_ORIGIN'

6       Measure Point 'Z1'

7       Define Align PCS 'Z_ORIGIN' Using Z1

8     Align Use 'Z_ORIGIN' As Default DRF

9       Align Define 'ROTATION'

10         Measure Point 'P2'

11         Construct Best Fit Line 'X_AXIS' Using P1, P2
```

```
 Define Align PCS 'ROTATION' Using X_AXIS
Align Use 'ROTATION' As Default DRF
"Inspect 10x10 Grid of balls with rows and columns 4,5,6 and 7 missing"
 Step & Repeat Begin 'SR1'
  "Check for row or column 4-7 (index 2-6) - ball is not present there"
  "For row or column 4-7, SKIP = 1 (true) else SKIP = 0 (false)"
  SKIP = SR1.I > 2 & SR1.I < 7 & SR1.J > 2 & SR1.J < 7
  If SKIP == 1 Then
   "Skip"
  Else
   "Inspect"
   "Look for damaged or missing ball"
   Measure Histogram 'HIST1'
   If HIST1.COUNT > 2000.0 Then
    "Ball OK"
    Align Define 'BALL'
     Measure Point 'BALL_TOP'
     Define Align PCS 'BALL' Using BALL_TOP
    Align Use 'BALL'
     Measure Point 'PTS[SR1.I,SR1.J]'
    Align End
   Else
    "Ball damaged or missing"
    resultout(nl())
```

```
36             resultout("Missing Ball " + iformat(SR1.I + 1, "") + "," +
               iformat(SR1.J + 1, ""))

37             resultout(nl())

38           End If

39         End If

40       Step & Repeat End

41       "Compute seating plane (co-planarity)"

42       Construct Best Fit Plane 'S1' Using PTS

43     Align End

44   Align End

45 Align End
```

Pocket Inspection Example

Below is an Example of a Step & Repeat program that handles missing Features by using the If-Then-Else capability.

Programming Features Used: Step and Repeat, If-Then-Else, arithmetic and logical Expressions.

Programming Details:

The Histogram step is created by selecting Measure/Histogram from the menu (Step 17).

To teach the Finder, adjust the minimum and maximum threshold values (for a given light intensity) such that the Feature is represented by white (on) pixels. Typically, when dealing with a white feature, the Max threshold value may be set at 255 and Min threshold value adjusted such that the feature appears as a group of connected white (on) pixels. For black features, try setting the Min value to zero and adjusting the Max threshold until the feature appears as a white (on) group of pixels and the background pixels are off.

Select Program, Branching and If Then Else menus items and type in the condition HIST1.COUNT > 5000.0. (Step 18)

See the array data structure example (Step 27).

Select Program, Branching and Call Procedure.. menu items and type in the code shown in Step # 31 to output the message about partial or missing features.

Program Listing:

```
0 Program Main
1   Align Define 'XY_ORIGIN'
2     Measure Point 'P1'
3     Define Align PCS 'XY_ORIGIN' Using P1
4   Align Use 'XY_ORIGIN' As Default DRF
5     Align Define 'Z_ORIGIN'
6       Measure Point 'Z1'
7       Define Align PCS 'Z_ORIGIN' Using Z1
8     Align Use 'Z_ORIGIN' As Default DRF
9       Align Define 'ROTATION'
10        Measure Point 'P2'
11        Construct Best Fit Line 'X_AXIS' Using P1, P2
12        Define Align PCS 'ROTATION' Using X_AXIS
13      Align Use 'ROTATION' As Default DRF
14        "Inspect 10 Pockets along X-axis, at 0.1 inch interval, and measure each
           width"
15        Step & Repeat Begin 'SR1'
16          "Look for plugged or missing pocket"
17          Measure Histogram 'HIST1'
18          If HIST1.COUNT > 5000.0 Then
19            "Pocket OK"
20            Align Define 'PCS1'
```

```
          "Locate pocket in the field of view - to accommodate variation in
          pitch"
          Measure Point 'POCKET'
          Define Align PCS 'PCS1' Using POCKET
        Align Use 'PCS1'
          Measure Line 'LEFT_EDGE'
          Measure Point 'RT_PT'
          Measure Distance 'WIDTH[SR1.I + 1]' From LEFT_EDGE To
          RT_PT
        Align End
      Else
        "Pocket plugged or missing"
        resultout(nl() + "Pocket Number " + iformat(SR1.I + 1, "") + "Not
        Inspected" + nl())
      End If
    Step & Repeat End
  Align End
 Align End
Align End
```

File I/O Examples

This program simply prints a text file in the *Result*s window:

Program Listing:

```
set_error_mode(1, 1)

i = open(File1, "words.txt", "r")

if i == 0

    While iostatus(File1) == 0

        str = readline(File1)

        If iostatus(File1) == 0 Then

            resultout(str+nl())

        Else

        End If

    End While

    close(File1)

Else

Endif
```

Below is an example program for Storing Inspection Results in a User-defined Format in a Text File from within a Measurement Program.

Programming Details:

Program listing contains many comments on program steps.

Define File1 variable as type File by selecting Program, Define, Declare Variable menu items.

“a” designation is for append mode; “w” for overwrite and “r” for read only mode.

Step # 19 is created by selecting Program, Branching, Call Procedure and typing in the code listed in the step.

Programming Features Used: Step and Repeat, Assignment.

Program Listing:

```
Program Main
 Align Define 'XY_ORIGIN'
  Measure Point 'P1'
  Define Align PCS 'XY_ORIGIN' Using P1
 Align Use 'XY_ORIGIN' As Default DRF
  Align Define 'Z_ORIGIN'
   Measure Point 'Z1'
   Define Align PCS 'Z_ORIGIN' Using Z1
  Align Use 'Z_ORIGIN' As Default DRF
   Align Define 'ROTATION'
     Measure Point 'P2'
     Construct Best Fit Line 'X_AXIS' Using P1, P2
     Define Align PCS 'ROTATION' Using X_AXIS
    Align Use 'ROTATION' As Default DRF
     "Inspect 10 Pockets, at 0.1 inch interval, and measure each width"
     "Store Results In Text File In User-defined Format"
     "Open File (in the same folder as the part program) using 'append'
      mode"
     "You May Specify Path For File e.g., C:\data\Myresults.txt"
     status = open(File1, "Myresults.txt", "a")
     writestring(File1, "Feature #   Length   Width " + nl())
     Step & Repeat Begin 'SR1'
      writestring(File1, iformat(SR1.I + 1, "") + "   ")
```

```
        Align Define 'PCS1'
          "Locate pocket in the field of view - to accommodate variation in pitch"
          Measure Point 'POCKET'
          Define Align PCS 'PCS1' Using POCKET
        Align Use 'PCS1'
          Measure Line 'LEFT_EDGE'
          Measure Point 'RT_PT'
          Measure Line 'TOP_EDGE'
          Measure Point 'BT_PT'
          Measure Distance 'LENGTH' From TOP_EDGE To BT_PT
          Measure Distance 'WIDTH' From LEFT_EDGE To RT_PT
          "Distance Measurement Has 4 Components : X, Y, Z Distances and Absolute Distance"
          "The Following Steps Use Absolute Distance Which Is Indicated By .P2PDIST"
          "Note : When Accessing These Structure Components Directly, The Results Are Always In MM."
          "To Output Them In Inches, Use e.g., LENGTH.P2PDIST/25.4"
          writestring(File1, fformat(LENGTH.P2PDIST, "#.000000") + "  ")
          writestring(File1, fformat(WIDTH.P2PDIST, "#.000000") + nl())
        Align End
      Step & Repeat End
      status = close(File1)
    Align End
  Align End
Align End
```

Library Procedure Example

The following two listings give an example of a VMS library and a program that uses it. The library contains a single procedure called 'ColFormat'. The procedure formats a number in a string whose width is specified by the 'width' parameter. The width to the right of the decimal point is determined by counting characters to the right of the decimal point in the format string. The string is padded with blanks on both ends as required.

Programming Features Used:: Assignment, If-Then-Else, While-End While, Defining Procedures, Procedure Libraries, arithmetic, string, logical and relational Expressions.

The first listing is for the "ColFormat" library file.

Library Listing:

ColFormat.vml

```
0  Program Main

0  Procedure ColFormat(Float num, String format, Integer width) : String

    0   Var add_left: Integer

    1   Var add_right: Integer

    2   Var fmt_len: Integer

    3   Var fmt_right: Integer

    4   Arg format: String

    5   Var ix: Integer

    6   Var newfmt: String

    7   Arg num: Float

    8   Var str: String

    9   Var str_len: Integer

    10   Var str_right: Integer

    11   Arg width: Integer

1   **** Format a number in a string of specified width.
```

```
**** The width to the right of the decimal is computed from the format.
str = fformat(num, format)
str_len = strlen(str)
**** Count the decimal & chars to the right in the 1st format
fmt_right = 0
fmt_len = strlen(format)
ix = instr(format, ".")
While ix < fmt_len & mid(format, ix, 1) != ";"
  fmt_right = fmt_right + 1
  ix = ix + 1
End While
**** If the string has fewer chars to the right than the format,
**** pad the right end with that many spaces
str_right = str_len - instr(str, ".")
add_right = fmt_right - str_right
If add_right > 0 Then
  str = str + fill(add_right, asc(" "))
  str_len = str_len + add_right
Else
End If
**** If the string has fewer chars than the width,
**** pad the left end with that many spaces
add_left = width - str_len
If add_left > 0 Then
  str = (fill(add_left, asc(" "))) + str
```

27 Else

28 End If

29 Return str

The second listing is a program that includes library "ColFormat."

Program Listing:

ColForm.voy

0 Program Main

0 Var form: String

1 Var wid: Integer

1 **** This program uses the ColFormat function to print numbers so that the decimal points line up

2 form = "0.0##;-0.0##"

3 wid = 8

4 resultou t("" + ColFormat:ColFormat(10.2, form, wid) + "" + nl())

5 resultout("" + ColFormat:ColFormat(1.12, form, wid) + "" + nl())

6 resultout("" + ColFormat:ColFormat(0.0, form, wid) + "" + nl())

7 resultout("" + ColFormat:ColFormat(-12.345, form, wid) + "" + nl())

```
Procedure ColFormat:ColFormat(Float num, String format, Integer width) : String

    Var add_left: Integer
    Var add_right: Integer
    Var fmt_len: Integer
    Var fmt_right: Integer
    Arg format: String
    Var ix: Integer
    Var newfmt: String
    Arg num: Float
    Var str: String
    Var str_len: Integer
    Var str_right: Integer
    Arg width: Integer

**** Format a number in a string of specified width.
**** The width to the right of the decimal is computed from the format.
str = fformat(num, format)
str_len = strlen(str)
**** Count the decimal & chars to the right in the 1st format
fmt_right = 0
fmt_len = strlen(format)
ix = instr(format, ".")
While ix < fmt_len & mid(format, ix, 1) != ";"
```

```
  fmt_right = fmt_right + 1
  ix = ix + 1
End While
**** If the string has fewer chars to the right than the format,
**** pad the right end with that many spaces
str_right = str_len - instr(str, ".")
add_right = fmt_right - str_right
If add_right > 0 Then
  str = str + fill(add_right, asc(" "))
  str_len = str_len + add_right
Else
End If
**** If the string has fewer chars than the width,
**** pad the left end with that many spaces
add_left = width - str_len
If add_left > 0 Then
  str = (fill(add_left, asc(" "))) + str
Else
End If
Return str
```

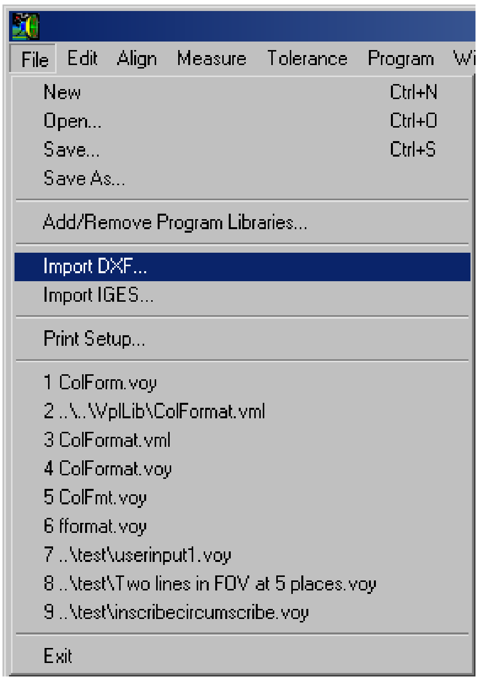
Appendix B

CAD Import Option

The CAD Import Component

CAD Import capability is an optional component of the VMS software. It allows you to import CAD drawings into the VMS software. Once imported, you can use the physical specifications in the CAD drawing to help you build measurement programs. The CAD Import component is especially useful if you want to use the optional Offline Programming Component to develop measurement programs.

CAD import is accessed from the **File** menu.



Working with CAD Files

Computer-Aided Design (CAD) files are computer-generated drawings containing precise information about a physical entity. CAD drawings relevant to VMS would be drawings of parts you want to inspect. Each CAD file would contain a precise drawing of a part you want to inspect with precise measurement information about the part.

You can import two types of CAD files -- DXF format files and IGES format files -- into VMS. When you import a CAD file into an existing program, it does not delete any existing Features or steps, it just adds more Features to what you already have.

Here is how CAD file entities are translated into VMS Features:

- Generally, lines, arcs, and circles are imported directly. Four-sided surfaces are translated into four lines.
- Polylines are translated into a series of lines, but "bulge" (in DXF) is not considered, except in the special case of a "donut" which is translated into two circles.
- Wide lines are translated into two lines representing the two edges of the wide line; if the line was just wide for visual impact, VMS has no way to know this.
- Spline curves and polygonal meshes are not translated to anything, because VMS does not have any Features corresponding to such items.
- Three dimensional parts with arcs and circles that do not lie in the XY plane will give confusing looking pictures, because VMS generally does not draw such things in perspective.

Since CAD files do not tell VMS how to measure, you must create measurement steps, just as you would with any other part. The difference is, you can use Features in the *Features* window to help create your measurement steps. In measurement steps for circles and lines, you can create a single Finder and use the **Auto-measure** option (click **Option**s in a *Feature Measurement* dialog box) to run it in multiple places on the Feature.

Preparing to Import Cad Files

Before you import a CAD file, you need to know whether the drawing was stored in millimeters or inches. In addition, the file must contain measurement information.

There are two main goals in importing CAD files:

- Eliminate extraneous information contained in the CAD file (borders, notations, views, etc.) so that only the actual Features of the part remain.
- Relocate the origin of the part in the CAD file to coincide with the intended origin of the part in VMS, so that the PCS nominal positions of Features are correct.

Eliminating Extraneous CAD Information

In some CAD files, lines that are extraneous to the part (e.g., an arrow or a border around a drawing or notation) are identical to lines that make up the part. There are often various views of the part and only the view as VMS will see it is desired. If VMS imports such a file in its entirety, there will be more line Features than you really want. VMS ignores text items during import, but extraneous lines and other views may be imported. There are three techniques to eliminate unwanted elements:

Technique	Description
Edit the CAD File	If you have access to the CAD application that was used to create the CAD file, you can do one of a couple of things: Make a copy of the file and open the copy in the CAD program. Edit the file to eliminate extraneous elements. While you are editing the drawing, note where the drawing origin is. Often, it is in the corner of the page, or somewhere else on the drawing that is not related to the picture of the part. If possible, reset the drawing origin to coincide precisely with where you want the DRF origin to be in VMS (i.e., the main Datums). If you can do this, you can skip the first import of the CAD file and just use zeroes for the offsets when importing the file. Open the original file and select the basic view (as it will be seen on the VMS stage), and copy only those parts to a new file.

Technique	Description
Use Layer Capability	Elements in a CAD drawing can sometimes be divided into different layers by the person who creates it. If the file was created using layers, notations and other extraneous elements should be in a layer different than the one containing the Features of the part. During import, VMS allows you to identify one, two, or three layers you want, and ignore other layers. If you do not identify any layers in the *Import* dialog box, VMS imports all layers. Ask the supplier of the CAD file for the layer names, or open the file in the CAD program to identify them. Then, during import, enter the names of the relevant layer(s) in the **Layers** text box(es) in the *Import Control* dialog box.
Edit Imported CAD Drawing in VMS	Delete Features in VMS's *Feature*s window. This may be more tedious than the first two methods, but it can be a useful technique, especially if you don't have access to the application that created the CAD file. To delete Features in the *Features* window: 1) Complete the process of importing the CAD file into VMS. Make sure you complete both the first (Importing a CAD File: Part 1) and second (Importing a CAD File: Part 2) phases of the import process. 2) Select the unwanted Features by clicking on them with the left mouse button. The selected Features turn yellow (or another color if you altered color schemes in the *Select Color* dialog box). 3) Click on the right mouse button to display the *Features* window pop-up menu and select **Delete**. Make sure you don't accidentally delete an element that is a Feature of the part.

Importing a CAD File: Part 1

To successfully import a CAD file, you need to import the CAD file twice: the first time to gain data on the part's origin; the second time to use this data in the import process so that you can synchronize the part's origin in the *Features* window with the part's PCS origin.

After you relocate the origin in the CAD file so that it is synchronized with the part's PCS origin in VMS, you still need to synchronize this origin with where the part is placed in a fixture on the stage. Do this by establishing a Fixture Alignment after the CAD file is imported.

1. Transfer the CAD file to the VMS computer's hard drive.
2. If you are not familiar with how to transfer files from one location to another, see your Microsoft Windows documentation. Look for the section on moving and copying files in Explorer.
3. Place the file in the VMS subdirectory named **PROG**.
4. Location of Prog Directory in *Windows Explorer.*
5. If the VMS application is not yet open, open it.
6. Select **System Alignment/Align** from the menu.
7. In the *System Alignments* dialog box, select **Stage** and click **Apply**.
8. Later, you'll establish a fixture alignment for the part. For now, **Stage** is an appropriate choice for system alignment.
9. Select **File/ Import DXF** or **Import IGES** from the menu, depending on the type of CAD file you are importing. DXF file names have a DXF extension and IGES file names have an IGS extension, for example, TRIANGLE.DXF or WIDGET.IGS. Select the file you want to import. Click the **Open** button. The *Import Control* dialog box is displayed.
10. Select Metric or English, dependent on the units used in the CAD drawing.
11. If you don't know the units used in the CAD drawing, select Metric and, after import, select Zoom Factor from the *Features* window pop-up menu and click 1 to 1. Compare an actual part to the part in the *Features* window. If they are the same size, Metric was the correct unit. If they are not the correct size, begin this process again and select English as the unit.
12. If you are importing a DXF file and it contains X, Y, and Z data, select **3D**. Otherwise, select **2D**. The *IGES Import* dialog box does not ask for this data. Leave the remaining default values the way they are and click the **OK** button.

13. The CAD drawing is imported so that the CAD origin aligns with the MCS origin. The lower left corner corresponds to the zero points on the X and Y axes of the stage. You may have to select **Part View** from the *Features* window pop-up menu (right click on the mouse button in the *Features* window) to see all the CAD Features. If you want to zoom out further, select **Zoom Factor** from the *Features* window pop-up menu, enter a value in the text box smaller than the default value offered, and press [Enter].
14. The CAD drawing is converted into points, lines, and circles. In addition, each Feature is assigned a name, such as C2 or L3.

> The origin of the CAD drawing is placed at the lower left corner (origin) of the stage. If the origin of the CAD drawing was not set at the lower left corner of the drawing, portions of the drawing will be outside the *Features* window's yellow rectangle (stage limits). This is not a problem. The drawing will be moved within the rectangle in a subsequent step of the import process.

15. Zero the stage if it has not already been zeroed.
16. In the *Features* window, locate the origin of the part. For improved accuracy, you may want to zoom in on the origin in the *Features* window. You can enlarge the part by selecting **Part View** from the *Features* window pop-up menu or by clicking and dragging the mouse ("rubber banding") around the part
17. In the *Features* window, double-click on the Feature that represents the origin, such as a circle. If the origin is not a single Feature, create a temporary Feature that represents the origin. For example, if the part's origin is based on the intersection of two lines, you can construct a Feature from the intersection of two lines and get the X, Y, and Z offset information see *Constructing a Point*. If you construct a Feature to define an origin, use the **Actual** values in the *Feature Measurement* dialog box.
18. From the *Feature Measurement* dialog box that appears, write down the X, Y, and Z values listed under **Nominals**.

Importing a CAD File: Part 2

1. Select **File/New** from the menu.
2. Select **File/Import DXF** or **Import IGES** from the menu.
3. In the *Import* dialog box, select the file you want to import and click **Open**.
4. In the *Import Control* dialog box, select **Metric** or **English**, depending on the units used in the CAD drawing. Enter the negatives of the X, Y, and, if relevant, Z values you wrote down previously and click **OK**.

   For example, if the numbers you wrote down for X and Y previously were 6.574 and 3.682, enter -6.574 and -3.682 in the appropriate boxes. If the CAD origins were already fixed to coincide with the part origin and you skipped part 1, just leave the zeroes.

   The CAD drawing is imported into VMS so that the part's origin in the *Features* window is synchronized with the part's intended PCS origin. Make sure that the origin arrows in the *Features* window are superimposed on the Feature you specified as the origin in your CAD drawing.
5. Apply the appropriate fixture alignment for the part, using the **Align/System Alignment** command in the menu, see *System Alignment*.
6. In the *Features* window in **Stage View**, your CAD drawing may have been inappropriately placed in relation to the stage. Applying the appropriate fixture alignment will put the Features from the imported CAD file in the proper position.

Now you can use the nominal values of Features in the *Features* window to help you create measurement programs. Here are a couple of useful tips for doing this:

- Pressing [Ctrl] + click (left button mouse click) in the *Features* window moves the stage to the place you click.
- Double-click a Feature in the *Features* window to begin creating a measurement step for the Feature. VMS displays a *Feature Measurement* dialog box with nominal values already filled in.

Measuring Features Manually

1. Make sure that all measurement steps, except those defining the Part Alignment, follow the Align Use step in the *Measurement Steps* window. To define a Feature:
2. Double-click on a Feature in the *Features* window.
3. A *Feature Measurement* dialog box is displayed. For example, if you double-clicked a straight line, the *Measure Line* dialog box is displayed
4. Add one or more Finders to the Feature as needed. See *About Finders*.

> 🗎 Keep in mind that Auto-Measure is available to simplify this process. For more information see Section 7.13.4, *About Auto-Measure*.

5. Save the program.

Automatic Measurement Step Generation

After importing Features using the CAD Import option, the Features exist as variables in your program and they have nominal values for position and size etc. so they appear in the *Features* window. But at this point there is no alignment and there are no steps to measure the Features. The next step after import is to create the alignment steps based on the actual part on the stage. How to do this is covered in other material.

But after that, it would still remain to create, in the Align Use section, the steps that measure the bulk of the Features. This is what takes a lot of time and this is where this new capability comes in. You will be automatically creating steps to measure Features that you select in the *Features* window. Follow these steps:

1. Go into the *Measurement Options* dialog box (menu: Setup/Options/Measurement) and review the settings to make sure they are what you would want when creating the Feature Measurements steps.

> 🗎 Auto-Measure will be used (even if the checkbox is off) so the “Places” value will be used.

2. Set up and run the Line, Arc, and Circle Finders on the kinds of edges that you want to measure. The last-used values you create in this way will be used in the Finders that are created automatically.

3. Select multiple Features in the *Features* window – either individually (using Shift-click to select each Feature and keep the already selected ones), or by holding down Shift and doing a rubber-band box, which is a new capability. VMS will be automatically generating steps to measure the Features you select.
4. Select the step after which you want the new steps to be placed. If the Align Use section is empty, this would probably be the Align Use step itself. If some other Feature Measurement steps are already in the Align Use section, you will probably want to select the last of these so the new steps will come after them.
5. Right-click in the *Features* window and click on **Auto build steps for Features**. VMS will create steps that measure the selected Features.

Just before those steps, VMS also creates Define Tolerance steps that define Tolerance variables for the different measurements that were checked on in step 1 above. That is, for example, if the default is to measure the angle of a Line, then a Define Angle Tolerance step would be created, etc. This single Angle Tolerance variable would be used in all the created Line measurement steps. You can edit the Define Tolerance steps to put in the desired tolerances. The handy part of this is that, for example, if you create steps to measure a hundred Circles, you can set the diameter tolerances for all of them by editing the single Define Diameter Tolerance step.

If you have some Features that require one set of tolerances, and other Features that require a different set of tolerances, then you can follow the above procedure more than once. Each time you would select, and create steps for, all the Features that share a given set of tolerances, and there will be Define Tolerance steps for each batch of them.

The Feature Measurement steps each contain only one Finder, and they use Auto Measure to position the Finder in several places around the Feature according to the Feature's nominal values (which came from the CAD file).

This page was left blank intentionally.

Appendix C

Offline Programming Option

Offline Programming is an optional component of the VMS. It allows you to create measurement programs without being physically connected to the hardware (i.e., the unit that includes the stage, camera/lens assembly, etc.).

VMS, with the Offline Programming component, should only be installed on a stand alone PC. Installing a QVI system with Offline Programming will not work because Offline Programming prohibits the VMS from communicating with the stage, lights, and other hardware. Conversely, if you install VMS without Offline Programming on a stand alone computer, running VMS will not work and may crash your computer since the software will attempt to communicate with the hardware that is not connected to your computer.

Offline Programming allows you to generate Measurement Programs without tying up a real system. Since you cannot see an actual image of the part when you program offline, you will probably need to take the program you created offline to a real system and adjust Finders to complete your program. Much of the work in creating a measurement program, however, can be accomplished offline. The Offline Programming option is especially useful if you use CAD files to develop measurement programs.

Understanding How Offline Programming Works

The offline version of the VMS is almost identical to the software you use on a real system. The differences are in the lower levels that normally control the stage and image processing functions. In offline mode, these parts of the program pretend to do what they normally do. The stage control routines move an imaginary stage, and the image processing routines always find perfect edges in the centers of the Finders.

Since you don't have a joystick, you move the stage by holding down the **Control** key and clicking the left mouse button in the *Features* window (this also works with online systems). Because you can't see the part in the *Video* window, you position the camera relative to features seen in the *Features* window. You can see features in the *Features* window either by importing a CAD file into VMS or by entering the nominal values of your part in *Feature Measurement* dialog boxes. See *Viewing a Part in the Features Window*.

Getting Information from the System

You need to know certain things about the real system(s) before you begin programming offline. Take pen and paper to the real system and gather the following pieces of information:

- The version of VMS installed. To check the version number on your system, select **About VMS** from the **Help** menu.

> Make sure that the version of the VMS you're using offline is the same as or earlier than the version of VMS you're using on the real system. We cannot guarantee that a program written under a later version of VMS will run on an earlier version of VMS.

- The lens name, type, and pixel size.
- What light levels, and maybe non-default Finder settings, give good results on the different places on the part you will be programming.

On the offline system, create a new Lens Calibration. Enter the lens type, lens name and pixel size based on the lens used on the real system.

Set the light levels for each Finder you program using the *Stage and Light Control* dialog box.

Setting Up Alignment

Set up a Fixture Alignment on the offline system (see *Creating a Fixture Alignment*). You do not have to use the same Fixture Alignment as the one you will use on the real system, just something that puts the imaginary part in a good place on the imaginary stage.

Programming the alignment is just as critical offline as it is online. Think out your alignment strategy thoroughly before you start, just as you would when programming a real system. For more information on setting up alignment, see *Introduction to Alignment and Coordinate Systems*.

Viewing a Part in the Features Window

Before you can program Finders for a part's Features, you need to be able to view the part's nominal Features in the *Features* window. If you have a CAD file, this will create nominal features for you. If you don't have a CAD file, then as you program each feature, type in its nominal values first and run the step; this will create the nominal Feature. Then you can add Finders on the edge of the Feature and add tolerances, etc.

To program a Finder, you need to set the Z position of the imaginary stage, since you can't twist the joystick handle. To do this:

1. In the *Features* window, click the right mouse button to display the *Features* window pop-up menu.
2. Select **Part View**.
3. From the *Features* window pop-up menu, select **Set Mouse Z**.
4. Select **Manual Entry Z**.
5. Enter the **Height** in PCS at which you want the Finder to run. You should be able to determine this from the drawing.

As an alternative, instead of selecting **Manual Entry Z** in the *Set Mouse Z* dialog box, you can select **Closest Feature Z**. This option will often work well, setting the Z position automatically when you click near a Feature. This only works if either:

- the CAD file you are using has 3D data or
- you entered Z nominal positions for features.

Programming Measurement Steps

After setting the Z position, you can begin programming measurement steps. As with normal programming, begin offline programming by establishing a part alignment (see *Part Alignment*); then measure Features (see Section 7, *Measurements*). This section details the procedural steps for offline programming of both part alignment and Feature measurement.

1. In the *Features* window, zoom in as much as you can on the feature where you want to place the Finder. Use the Zoom Factor command from the *Features* window pop-up menu.
2. Hold down the **Ctrl** key and click the left mouse button on the desired edge. The stage position crosshair moves to where you click. In the *DRO* window in PCS mode you should see the coordinates come up correctly.
3. Set the light levels in the *Stage and Light Control* dialog box.

> Before you begin to measure Features, make sure you have set the Z position and the light levels. Otherwise, the Z position and light levels may be wrong for all measurement steps.

4. From the **Measure** menu, select the type of Feature you want to measure.
5. In the *Video* window, select the type, position, and angle of the Finder as you normally do in online mode (see Section 8, *Finders*).
6. Test run the Finder by double-clicking in an open area in the *Video* window.
7. Click the green **Check Mark** to see the Finder in the *Features* window.
8. In the *Features* window, double-click the just-created Finder. Now, if you move the mouse arrow to the *Video* window, the arrow has turned into a pencil, indicating you can edit the on-screen Finder.
9. Using the mouse (pencil) in the *Video* window, adjust the Finder and test run it again and re-accept it (green check) until you are satisfied with the location and angle of the Finder as it appears in the *Features* window.
10. Notice that the *Feature Measurement* dialog box states "Measure Feature (1 Edge Finder)."
11. To enter additional Finders for this feature, repeat steps 5-9.
12. Notice that the *Feature Measurement* dialog box states "Measure Feature (2 Edge Finders)" for the second Finder you accepted. The number of Finders increments for each additional Finder you add.
13. Click **OK** in the *Feature Measurement* dialog box.

14. After all the Finders are programmed, you can test-run the whole step and see the measurement results. They will probably not be perfect (zero deviation) because your positioning of the Finders was probably not perfect in offline mode.

Remember to save the program frequently as you work.

Using Offline Program on a System

Using Your Program on a Real System

After you have completed your offline programming, you can take it to a real system to test, make necessary adjustments, and run the program.

1. Copy to a floppy disk the program you created offline.

   The program file is saved with a name you gave it followed by a .VOY extension, for example, WIDGET.VOY. Unless you specify a different location, the file will typically be saved in VMS' PROG directory.

2. Take the floppy disk to the system and copy it to the system's hard drive. Copy the file to the directory on the system where your other have other VMS program files are stored. If no other program files exist, we recommend that you copy it to VMS Prog directory.

You can copy files to and from floppy disks by dragging files from the floppy and dropping them in the appropriate folder in Windows Explorer. For more information on copying files, see your Windows documentation.

3. Start VMS on the system.
4. Select **File/Open** from the menu. The *Open* dialog box is displayed: Locate the file you created offline and open it by double-clicking on it.
5. Set up a Fixture Alignment that establishes an initial origin for the part at the same place on the part as the Fixture Alignment on the offline system. See *Creating a Fixture Alignments*.

6. Test the program.

   You may want to test the program with one hand on the E-stop, just to be sure that it really moves to the right place. If you set up your Fixture alignments properly, all Finders should be in approximately the right place.

   Some of the Finders may need to be edited for light levels or image processing parameters. If you think a Finder needs to have its position edited, consider that it may be the part, not the Finder that is not at nominal.

7. Run the program.

Appendix D

Error Messages

This section helps you identify the cause of possible problems with the system. Descriptions of problems and potential solutions are offered for each of the listed problems. There are three sets of error tables in this section:

- All messages that could appear on the VMS screen most of which are messages to the user during VMS setup/programming.
- Errors that can happen while running a program, that can be handled automatically if the user writes the program to do so.
- Extended Windows I/O messages (only listed for advanced debugging purposes).

Error messages are listed in alphabetical order. If you find a problem you cannot address on your own, contact your authorized service representative.

Common Error Message Descriptions

The description column often contains one of the following two messages:

- system failure
- out of memory

System failure means the error is not expected to occur under any circumstances. If an error in this category occurs, save your program under a new name (to keep the previously-saved version intact) and restart your system. If the same error occurs repeatedly, record the precise sequence of operations, from starting your system to receiving the error message, and report it to your authorized service or sales representative. The exact sequence of operations and the error message can give us a clue toward solving the problem.

Out of memory means that you have exceeded the random access memory (RAM) capabilities of your system. These errors can occur if:

- other Windows or DOS programs are run simultaneously
- the measurement program is extremely large
- a large number of Features has been declared or imported from a CAD file

System Error Messages

The following error messages are issued to indicate a user error or system error.

Error Message	Description
3D to 2D Divide by Zero	Error during translation of a point when trying to draw a Feature.
Array index not integer type	Array indices must be of integer type.
Array index on non-array	A name had an open-bracket ([) after it, but it was not the name of an array.
Bad Var Token	VMS cannot find a Feature variable.
Best fit to points failed	You need at least two points to do a best fit to a line, and three non- colinear points for a circle or plane.
Block cannot cross procedure boundaries	An end-block was selected (by shift-click) in a different procedure than where the block started.
Block end must be after block start	To select a block of steps, you must select the first step in the block then shift-click on the last step in the block.
Block must end at same level as start	A marked block must start and end at the same nesting level. Otherwise, for example, you could cut an IF without the ELSE and ENDIF, or a loop start without the loop end.
Block must not cross level higher than start	A marked block can include nested levels but cannot, for example, run from inside an IF-THEN section to inside an ELSE section, because the block would contain an isolated ELSE step.

Error Message	Description
Block start has not been defined	You selected an end-block (shift-click) before you selected the start of the block.
Can't add procedure - no room	More than 200 user-defined procedures, or out of memory.
Can't have more than one decimal point	Syntax error in a floating point constant.
Can't put decimal point in hex number	Hex constants start with 0x and cannot contain a decimal point (integers only).
Can't use structure-type operand(s)	Operators like + and * never take a reference to an entire structure as an operand. For example, if you want to add the coordinates of two points you can't say P1+P2, you must do the coordinates individually, P1.X + P2.X.
Cannot add parameters after adding non-parameters	When creating variables for a procedure, you must create all parameters (passed to the procedure) before creating any other variables. You might have hit the "Parameter" check box accidentally.
Cannot begin a new program while a program is running	You must stop the currently running program if you want to start a new one.
Cannot delete parameter because it is not the last	The only time a parameter variable can be deleted is if it was the last variable declared. The easiest thing may be to leave it in and pass a dummy value to it wherever the procedure is called.
Could not create Program List window	System failure (see *Common Error Message Descriptions*).
Could not create window	System failure or out of memory (see *Common Error Message Descriptions*).
Could not initialize module. Program aborted.	System failure or out of memory (see *Common Error Message Descriptions*).
Could not initialize user program.	System failure or out of memory (see *Common Error Message Descriptions*).

Error Message	Description
Could not initialize VMS Instance. Program aborted.	System (see *Common Error Message Descriptions*).
Could not register (name) class. Program aborted.	System failure (see *Common Error Message Descriptions*).
Couldn’t add first & last step	System failure or out of memory (see *Common Error Message Descriptions*).
Detected 2D Features, Import Again to Set 2D Features Z	The attempted import file has 2D entities, not 3D entities. To avoid a bad translation, the import was stopped. Set DXF Import to 2D, and set the Features Z Height.
Didn’t find] where expected	Last array index ended but was not followed by a]
Dimensions are illegal	The 2D array cannot have first dimension = 1; or, array size takes more then 65536 bytes. Remember that each Feature in an array of Features takes many bytes of memory. For example, a circle takes over 100 bytes.
Divide by zero	System failure (see Common Error Message Descriptions), or you used an expression that resulted in a divide by zero. In the latter case, check the program logic before attempting the divide.
Dot notation on non-structure	A name had a dot (.) in it but it is not the name of a variable that has a structure type, e.g., C1.X.
DXF Import, Couldn’t Open DXF File	Could not obtain a file handle from the operating system. Or the file could have a bad name.
DXF Import, Procedure is buried by another active procedure	Main Procedure is buried. Cannot create Feature variables from CAD entities.
Error %d adding measurement variable ‘%s’	System failure (see *Common Error Message Descriptions*).
Error adding step to program	System failure (see *Common Error Message Descriptions*).
Error adding variable	System failure (see *Common Error Message Descriptions*).

Error Message	Description
Error changing variable name	System failure (see *Common Error Message Descriptions*).
Error creating dialog box	System failure (see *Common Error Message Descriptions*).
ERROR, DDE Init	DDE Initialization Error. No operating system resources available. Close other applications, if possible
Error deleting step from program	System failure (see *Common Error Message Descriptions*).
Error deleting variable	System failure (see *Common Error Message Descriptions*).
Error, Get Print FileName returned Error #	Cannot open *Common File Open* dialog box, or file selection failed. VMS application error.
Error getting Feature position offsets	System failure (see *Common Error Message Descriptions*).
Error getting var list	System failure or out of memory (see Common Error Message Descriptions).
Error locating measurement variable '%s'	System failure (see *Common Error Message Descriptions*).
Error Opening File (name)	Possible wrong file name.
Error Reading File (name)	You may have a corrupted disk. Run CHKDSK or SCANDISK from a DOS prompt.
Error running Finder	System failure (see *Common Error Message Descriptions*).
Error Writing File (name)	You may have a write-protected file or floppy. Or you may have a full floppy or hard disk. It's also possible you have a corrupted disk. To check for disk corruption, run CHKDSK or SCANDISK from a DOS prompt.
Execution past end of procedure	You may have forgotten to put a Return step at the end of a procedure. Only the Main program automatically restarts if it "falls off the end."

Error Message	Description
Expression has wrong type	A floating point expression has been entered where only an integer type may be used, or vice-versa. You might use the float, round, or chop functions on the expression.
Features do not intersect	An intersection or circle tangent to two lines construction has been attempted but the two Features were parallel and so did not intersect.
File (name) is not a VMS program	The file you specified to load as a measurement program is not recognized as a VMS file. Perhaps you may have saved some other data under that name, or the file has become corrupted. Run CHKDSK or SCANDISK from a DOS prompt.
File Error, Could Not Open VRESULTS.TXT	Could not obtain a file handle from the operating system. Read only File attribute. Increase the number of open files in your CONFIG.SYS file.
Float Error, Denormalized	CPU Error Flag DENORMAL
Float Error, Divide by Zero	CPU Error Flag ZERODIVIDE
Float Error, Invalid Not a Number(NAN)	CPU Error Flag INVALID
Float Error, Overflow	CPU Error Flag OVERFLOW
Float Error, Precision	CPU Error Flag INEXACT
Float Error, Underflow	CPU Error Flag UNDERFLOW
Focus pass would take too much time	We force the speed to match the time to give 1 second always. System failure (see *Common Error Message Descriptions*).
Focus point not centered enough in pass	If this is due to variation of the part, the first pass length should be increased to include all expected part variation.
Focus too strong. (Reduce area or lower light)	For Textured or Ronchi focus, the area can be reduced. For others, you must reduce the light level.

Error Message	Description
Focus too weak	For Textured or Ronchi focus, try increasing the area. For others, try increasing the light (but not to saturation.) Another possibility is that the Feature was not in the field of view at all.
Function not defined	You used an undefined function name in an expression. A function name is indicated by an open parentheses for the argument list. You may have typed a (by accident.
IGES Import, Couldn't Open IGS File	Could not obtain a file handle from the operating system. Or bad file name.
Illegal operator for string type	The only operator that applies to strings is + for concatenation.
Invalid block in heap at <address> (follows <address>)	Produced when "Debug Heap" is selected in *Setup/System* dialog. System failure (see *Common Error Message Descriptions*).
Invalid block type %s (%d) should be %s (%d)	System failure (see *Common Error Message Descriptions*).
Invalid block type (%d)	System failure (see *Common Error Message Descriptions*).
Invalid Feature selection	You entered the name of a *Feature in a distance or angle* dialog box or selected a Feature for a construction by name in the *Object Names* dialog box, but the Feature is not the correct type for use in the distance, angle, or construction.
Invalid Feature type (%04X)	System failure (see *Common Error Message Descriptions*).
Invalid Feature type (%d)	System failure (see *Common Error Message Descriptions*).
Invalid measurement type (%d)	System failure (see *Common Error Message Descriptions*).
Invalid proc token	System failure (see *Common Error Message Descriptions*).

Error Message	Description
Invalid selection of Feature to use	The Feature selected has a type that cannot be used for the intended purpose. For example, trying to measure the angle between two Features, you cannot pick a circle as one of the Features.
Invalid subtype (%d)	System failure (see *Common Error Message Descriptions*).
Invalid tolerance type (%d)	System failure (see *Common Error Message Descriptions*).
Invalid Type	System failure (see *Common Error Message Descriptions*).
IPU couldn’t allocate memory	System failure or out of memory (see *Common Error Message Descriptions*).
Measurement out of tolerance	A measurement was out of tolerance and you specified that program execution should stop if this happened.
Measurement steps must be in the main procedure	Steps that measure Features or Relations can only be used in the main program. Otherwise, the alignment in effect for the step would be indeterminate.
Mem Alloc failed	Out of memory (see *Common Error Message Descriptions*).
Missing comma after parameter	Syntax error
MMC/LMC on datum of wrong type	You can only do MMC/LMC on Features that have a “size” (currently only circles).
MMC/LMC on datum with unknown inner/outer	The MMC and LMC size of a datum cannot be computed if you have not told the system if this is an “inner” (hole) or “outer” (pin) datum.
MMC/LMC on Feature with unknown inner/outer	The MMC and LMC size of a Feature cannot be computed if you have not told the system if this is an “inner” (hole) or “outer” (pin) Feature.
Name is not a variable	You have given a variable name that is not really a variable. Check the spelling.

Error Message	Description
Name is used by a variable of a different type	A name was entered for a Feature or Relation (to replace the default), but a that name is already in use for a different type of Feature or variable (e.g., “N1” was entered for a distance measurement name, but a point Feature named N1 exists).
No Message Memory	IGES Import cannot obtain FAR Memory from the operating system. Check information in VMS’s help menu.
No Message Memory HREALLOC	IGES Import cannot obtain FAR Memory from the operating system. Check information in VMS’s help menu.
No points found by Finder	Either the edge was not in the Finder area or the image processing parameters were not right for the edge(s) in the Finder area.
No points in Point Buffer	An operation on a Feature’s point buffer discovered there were no points in it.
No position in program is currently selected	You have tried to perform some operation, such as defining a variable or creating a step, that requires some position in some procedure to be selected in the *Measurement Steps* window, yet no such position is selected.
No room for step	16384 steps per procedure maximum
No Xform Index Memory	IGES Import cannot obtain FAR Memory from the operating system. Check information in VMS’s help menu.
No Xform Memory HREALLOC	IGES Import cannot obtain FAR Memory from the operating system. Check information in VMS’s help menu.
Not enough change in focus. (Use longer pass?)	For autofocus to measure accurately, it needs to have areas of poor focus at both ends of the pass. If the pass is too short the focus will be good through the entire pass.
Null pointer to variable	System failure (see *Common Error Message Descriptions*).

Error Message	Description
Numeric constant too large	20 characters max, or integer outside range of -32768/+32767 or 0x0000/0xFFFF
Out of memory	System failure or out of memory (see *Common Error Message Descriptions*).
Parameter can’t be an array	Passing an array as a parameter is not supported.
PostMessage failed	System failure (see *Common Error Message Descriptions*).
Printer Error, Error #	Your serial port had a communications error when trying to print by line. Select print mode of Print by Page or Print by Part in the *Results Control* dialog box.
Printer Warning, Could Not Load Driver	Could not load printer device driver. No printer handle was returned from the operating system. Out of system resources; delete other applications, if possible.
Printer Warning, No Driver Selected	No printer device driver is available. Install printer device driver.
Problem creating declare step	System failure (see *Common Error Message Descriptions*).
Problem getting Finder results	System failure or out of memory (see *Common Error Message Descriptions*).
Problem sending Finder	System failure or out of memory (see *Common Error Message Descriptions*).
Problem with declare step	System failure (see *Common Error Message Descriptions*).
Problem with step (number) in proc (name)	A step read from a measurement program file contained data that were not in the proper format. Perhaps the program was written with an earlier, incompatible version of software. More likely, the file has become corrupted; run CHKDSK or SCANDISK from a DOS prompt.
Problem with step contents	System failure (see *Common Error Message Descriptions*).

Error Message	Description
Proc doesn't exist (impossible!)	System failure (see *Common Error Message Descriptions*).
Procedure already exists	Attempt to create a procedure with the same name as one that already exists
Procedure cannot be deleted	System failure (see *Common Error Message Descriptions*).
Procedure doesn't exist	System failure (see *Common Error Message Descriptions*).
Procedure name does not follow rules	Names must start with a letter and must not exceed 32 alphanumeric characters.
Program aborted	This message is normal when the Debug Output is turned on and you select New or Open from the File menu. It means the previous measurement program — even an empty one — is being stopped so a new one can be created.
Program Advise Error at Meas	DDEML Post Advise failed on transaction measurement. Restart SPC or other client application.
Program Advise Error1	DDEML Post Advise failed on transaction PROGRAM. Restart SPC or other client application.
Second array index on 1-D array	You may have hit a comma by accident.
Step number invalid	System failure (see *Common Error Message Descriptions*).
Steps could not be pasted because step (number) is invalid	System failure (see *Common Error Message Descriptions*).
Steps could not be pasted because variables used by the steps conflict with the variables used at the current location.	At least one variable in the step(s) being pasted has the same name as, but differs in type from, a variable that already exists at that location. The step(s) cannot be pasted at such a location.

Error Message	Description
Stopped Part Program, Save Program if Desired	Auto Clear's Float Error caused a bad program step. Message sent to VMS Monitor Model to stop measurement program.
Symbol Table full	16352 variables per procedure maximum
Syntax error - bad character	This could be due to unusual punctuation marks, unbalanced parentheses, or a bad numeric character in your program.
Syntax error - expected)	Syntax error —parentheses don't match. You must have a close parenthesis for each open parenthesis.
Syntax error: unexpected operator	Strange character?
The clipboard is currently opened by another program	The clipboard is opened for a short time when any application writes to it or reads from it. You tried to access it during such a time. Try again.
The Emergency Stop switch has been hit! You may need to re- zero after you turn it back on.	In case you accidentally hit the Emergency Stop button and didn't realize it.
The part program has been modified. Do you wish to save the changes?	This message prompts you to save the currently open file.
The probe hit something and the stage could not back off. You must fix the problem and re-zero the stage.	This message usually appears when people keep hitting the touch probe. It also can appear if the probe hits something at high speed, or hits something flexible.
The Stage and/or the Image Processor has not shut down properly. Do you want to stop the main VMS application anyway?	VMS starts two programs that perform some of its tasks: VMSTAGE and VMIPU. When shutting down, VMS tells them to stop and waits for them to acknowledge that they have stopped. If this acknowledgment does not occur within 6 seconds, VMS shows this message so you can allow the main VMS program to stop anyway. One or both of the two programs must have had an error. System failure (see *Common Error Message Descriptions*).

Error Message	Description
The stage has stopped unexpectedly! It may have hit a limit switch inside the software limits. You must re-zero the stage.	The motion control system expects certain problems and can automatically recover from them. But some situations, usually caused by mechanical problems or obstructions, are unexpected and the motion will be disabled for safety until a zeroing operation is commanded.
The system ran out of timers. You should close one or more applications.	There are too many Windows applications running that use “timers.” VMS needs a timer during shutdown.
There is not enough memory to complete the requested operation. You should save your work in a new file and restart the VMS application.	System failure or out of (see *Common Error Message Descriptions*).
Too many Features selected	System failure (see *Common Error Message Descriptions*).
Too many measurement elements	You attempted to add more than 50 Finders in a measurement or more than 50 Features in a construction.
Too many steps to define alignment - you must get current alignment yourself.	The mechanism to automatically execute align define sections for you has a limit of 40 steps. If more than this need to execute, you will get this message. Use the GoTo button and the Single Step button to execute the Align Define sections of all the Align blocks enclosing the place in the program you are trying to edit (or until the remaining ones have fewer than 40 steps).
Type mismatch between computation and target	If assignment is to an integer variable then the expression must be type integer, and the same for floating point. You might use the float, round, or chop functions on the expression.
Type mismatch on parameter	Integer used as a parameter where floating point was required, or vice- versa.
Type mismatch: operator takes integer only	The bitwise operators &, |, ^, <<, and >> are only used for integer operations.

Error Message	Description
Type mismatch: string/nonstring	A string was used where a numeric item was required, or vice-versa.
Unary minus on non-number	Syntax error
Unknown program format in file (name)	The file you specified to load says it is a VMS measurement program but with a version number the current VMS software does not understand. You may have a corrupted file.
Unknown Structure Component	You have specified a component name after a dot that is not a valid component of that structure type, e.g., C1.XNORM instead of C1.XNOM.
Unspecified	Some execution error happened that has not been classified with a specific message.
Value out of range	Attempt to ask for more than 8 decimal places of precision, or jump on error to step number that does not exist, or auto-measure places less than 0 or more than 5000.
Variable already exists	Attempt to create a variable with the same name as one that already exists.
Variable cannot be added until the currently executing procedure is exited	You have tried to add a variable to a procedure that has made another procedure call which has not returned. A procedure cannot have variables added while it has any outstanding calls.
Variable has not been declared	A name is used like a variable but no such variable exists.
Variable name does not follow rules	Names must start with a letter and must not exceed 32 alphanumeric characters.
Variable still in use	Somewhere there is a step that still refers to the variable you are trying to delete. You must delete all references to a variable before you can delete it.
Variable token does not exist	System failure (see *Common Error Message Descriptions*).

Error Message	Description
View API, Error Sending VM Command	VMS stage did not process Mouse stage move message.
VOY DDEML, Program Advise Error at Stop	DDEML Post Advise failed on transaction Measurement. Restart SPC or other client application.
Warning, Could Not Access Printer	Bad printer ID when printing by line. Change your print mode in the *Results Control* dialog box.
You can't assign a value to a function	Function reference was outermost syntactical element of assignment target.
You can't dimension a string	Arrays of strings are not supported.
You cannot put this step here. You are in the Align Define section.	A step to create a Fixture Alignment must go in an Align Use section, where the desired alignment that was just derived is actually in effect.
You must select a Type	Attempt to click on "Add" button to add a variable but you have not chosen a type for it.
You must terminate the VMS application before terminating the Windows session.	VMS, unlike most Windows applications, will not shut itself down automatically if you try to close Windows. Go back to the VMS application and select Exit from the File menu.
You're not in an Align Define section.	Steps that define an alignment may only be placed in an Align Define section.
Zoom Box Divide by Zero	Display Zoom Box Area is Zero. Try again.

Computation Errors

The following errors are in the Computation group. If Error Handling is used, the error codes shown are those returned by **errno()**. Otherwise, the error message will be displayed.

Error Code	Error Message
1	Unspecified error
2	Problem sending Finder
3	Problem getting Finder results
4	No points in point buffer
5	Execution past end of procedure
6	Out of memory
8	Best fit to points failed
9	MMC/LMC on feature with unknown inner/outer
10	MMC/LMC on datum with unknown inner/outer
11	MMC/LMC on datum of wrong type
12	Divide by zero
13	Features do not intersect
14	Too many points for measurement
15	Arithmetic overflow
16	Zero may not be raised to the power of zero
17	Returning to Main
18	Attempt to execute a non-existent function

Error Code	Error Message
19	Undefined PCS used
20	Bad parameter value (e.g., invalid file open mode string)
21	Nesting limit exceeded (error mode or Align)
22	Program aborted by failing measurement
23	Program jumped by failing measurement
30	More than three formats, separated by semicolons
31	Literal characters embedded in number
32	More than one decimal point within a number
33	No close quote to end the literal character string
34	Illegal character in the format string
35	String cannot be interpreted a number
36	Number is out of the range for this data type
37	Requested length is negative

File I/O Errors

The following errors are in the File I/O group. If Error Handling is used, the error codes shown are those returned by **errno()**. Otherwise, the error message will be displayed.

Error Code	Error Message
45	File not found
46	Write protected
47	End of file reached
48	Access not allowed (protected or shared)
49	Disk full
50	File already exists
51	Bad file or path name
52	Hardware or medium problem
53	Miscellaneous I/O problem
54	I/O Incomplete
55	Incorrect file format (list input)
56	File too long (list input)
57	Improper file variable status (already in use, etc.)

Finder Errors

The following errors are in the Finder group. If Error Handling is used, the error codes shown are those returned by **errno()**. Otherwise, the error message will be displayed.

Error Code	Error Message
61	IPU couldn't allocate memory
62	IPU couldn't create Finder
63	IPU tried to run non-created Finder
64	Internal IPU error
65	Bad IPU parameters were sent
66	Error running Finder
67	Bad Video Bank ID sent to IPU
68	Bad Video Frame ID sent to IPU
69	Bad Video Camera ID sent to IPU
70	No points found by Finder
71	Focus too strong. (Reduce area or lower light)
72	Focus too weak
73	Not enough change in focus. (Use longer pass?)
74	Focus pass would take too much time
75	Focus point not centered enough in pass
76	The bottom of the focus Finder was too low...
77	IPU Timeout

Error Code	Error Message
78	Error testing IPU
79	Attempt to move beyond travel limit
80	Attempt to move X axis beyond travel limit
81	Attempt to move Y axis beyond travel limit
82	Attempt to move Z axis beyond travel limit
83	Edge point scatter exceeded limit
84	Focus supersaturated
85	Invalid light number
86	Laser was initially out of range
87	Laser went out of range
88	Laser did not settle (in time)
89	Lens turret malfunction
90	The lens requested by a finder is not mounted
91	Can't execute command because TTL laser is on
92	Problem setting requested frame integration
93	Problem selecting requested magnification
94	Problem obtaining video bank
95	Problem with image acquisition
96	Problem getting latched scale position
97	Point laser sensor not configured correctly
98	Set Light Values could not obtain target value

Error Code	Error Message
99	Invalid laser or A/D type for TTL laser focus

Measurement Fail Errors

The following errors are in the 'Fail Meas' group. If Error Handling is used, the error codes shown are those returned by **errno()**. Otherwise, the error message will be displayed.

Error Code	Error Message
7	Measurement out of tolerance

Windows (WIN32) Error Codes

The only place you normally will see these codes in VMS is if you have Debug output turned on and you get an I/O error (even if you handle it). Knowing a more precise definition of an error in this way might help in unusual circumstances.

Error Code	Description
-8	LZERROR UNKNOWNALG
-7	LZERROR BADVALUE
-6	LZERROR GLOBLOCK
-5	LZERROR GROR_LOBALLOC
-4	LZERWRITE
-3	LZERROR_READ

Error Code	Description
-2	LZERROR_BADOUTHANDLE
-1	LZERROR_BADINHANDLE
0	NO_ERROR
0	ERROR_SUCCESS
1	ERROR_INVALID_FUNCTION
2	ERROR_FILE_NOT_FOUND
3	ERROR_PATH_NOT_FOUND
4	ERROR_TOO_MANY_OPEN_FILES
5	ERROR_ACCESS_DENIED
6	ERROR_INVALID_HANDLE
7	ERROR_ARENA_TRASHED
8	ERROR_NOT_ENOUGH_MEMORY
9	ERROR_INVALID_BLOCK
10	ERROR_BAD_ENVIRONMENT
11	ERROR_BAD_FORMAT
12	ERROR_INVALID_ACCESS
13	ERROR_INVALID_DATA
14	ERROR_OUTOFMEMORY
15	ERROR_INVALID_DRIVE
16	ERROR_CURRENT_DIRECTORY
17	ERROR_NOT_SAME_DEVICE

Error Code	Description
18	ERROR_NO_MORE_FILES
19	ERROR_WRITE_PROTECT
20	ERROR_BAD_UNIT
21	ERROR_NOT_READY
22	ERROR_BAD_COMMAND
23	ERROR_CRC
24	ERROR_BAD_LENGTH
25	ERROR_SEEK
26	ERROR_NOT_DOS_DISK
27	ERROR_SECTOR_NOT_FOUND
28	ERROR_OUT_OF_PAPER
29	ERROR_WRITE_FAULT
30	ERROR_READ_FAULT
31	ERROR_GEN_FAILURE
32	ERROR_SHARING_VIOLATION
33	ERROR_LOCK_VIOLATION
34	ERROR_WRONG_DISK
36	ERROR_SHARING_BUFFER_EXCEEDED
38	ERROR_HANDLE_EOF
39	ERROR_HANDLE_DISK_FULL
50	ERROR_NOT_SUPPORTED

Error Code	Description
51	ERROR_REM_NOT_LIST
52	ERROR_DUP_NAME
53	ERROR_BAD_NETPATH
54	ERROR_NETWORK_BUSY
55	ERROR_DEV_NOT_EXIST
56	ERROR_TOO_MANY_CMDS
57	ERROR_ADAP_HDW_ERR
58	ERROR_BAD_NET_RESP
59	ERROR_UNEXP_NET_ERR
60	ERROR_BAD_REM_ADAP
61	ERROR_PRINTQ_FULL
62	ERROR_NO_SPOOL_SPACE
63	ERROR_PRINT_CANCELLED
64	ERROR_NETNAME_DELETED
65	ERROR_NETWORK_ACCESS_DENIED
66	ERROR_BAD_DEV_TYPE
67	ERROR_BAD_NET_NAME
68	ERROR_TOO_MANY_NAMES
69	ERROR_TOO_MANY_SESS
70	ERROR_SHARING_PAUSED
71	ERROR_REQ_NOT_ACCEP

Error Code	Description
72	ERROR_REDIR_PAUSED
80	ERROR_FILE_EXISTS
82	ERROR_CANNOT_MAKE
83	ERROR_FAIL_I24
84	ERROR_OUT_OF_STRUCTURES
85	ERROR_ALREADY_ASSIGNED
86	ERROR_INVALID_PASSWORD
87	ERROR_INVALID_PARAMETER
88	ERROR_NET_WRITE_FAULT
89	ERROR_NO_PROC_SLOTS
100	ERROR_TOO_MANY_SEMAPHORES
101	ERROR_EXCL_SEM_ALREADY_OWNED
102	ERROR_SEM_IS_SET
103	ERROR_TOO_MANY_SEM_REQUESTS
104	ERROR_INVALID_AT_INTERRUPT_TIME
105	ERROR_SEM_OWNER_DIED
106	ERROR_SEM_USER_LIMIT
107	ERROR_DISK_CHANGE
108	ERROR_DRIVE_LOCKED
109	ERROR_BROKEN_PIPE
110	ERROR_OPEN_FAILED

Error Code	Description
111	ERROR_BUFFER_OVERFLOW
112	ERROR_DISK_FULL
113	ERROR_NO_MORE_SEARCH_HANDLES
114	ERROR_INVALID_TARGET_HANDLE
117	ERROR_INVALID_CATEGORY
118	ERROR_INVALID_VERIFY_SWITCH
119	ERROR_BAD_DRIVER_LEVEL
120	ERROR_CALL_NOT_IMPLEMENTED
121	ERROR_SEM_TIMEOUT
122	ERROR_INSUFFICIENT_BUFFER
123	ERROR_INVALID_NAME
124	ERROR_INVALID_LEVEL
125	ERROR_NO_VOLUME_LABEL
126	ERROR_MOD_NOT_FOUND
127	ERROR_PROC_NOT_FOUND
128	ERROR_WAIT_NO_CHILDREN
129	ERROR_CHILD_NOT_COMPLETE
130	ERROR_DIRECT_ACCESS_HANDLE
131	ERROR_NEGATIVE_SEEK
132	ERROR_SEEK_ON_DEVICE
133	ERROR_IS_JOIN_TARGET

Error Code	Description
134	ERROR_IS_JOINED
135	ERROR_IS_SUBSTED
136	ERROR_NOT_JOINED
137	ERROR_NOT_SUBSTED
138	ERROR_JOIN_TO_JOIN
139	ERROR_SUBST_TO_SUBST
140	ERROR_JOIN_TO_SUBST
141	ERROR_SUBST_TO_JOIN
142	ERROR_BUSY_DRIVE
143	ERROR_SAME_DRIVE
144	ERROR_DIR_NOT_ROOT
145	ERROR_DIR_NOT_EMPTY
146	ERROR_IS_SUBST_PATH
147	ERROR_IS_JOIN_PATH
148	ERROR_PATH_BUSY
149	ERROR_IS_SUBST_TARGET
150	ERROR_SYSTEM_TRACE
151	ERROR_INVALID_EVENT_COUNT
152	ERROR_TOO_MANY_MUXWAITERS
153	ERROR_INVALID_LIST_FORMAT
154	ERROR_LABEL_TOO_LONG

Error Code	Description
155	ERROR_TOO_MANY_TCBS
156	ERROR_SIGNAL_REFUSED
157	ERROR_DISCARDED
158	ERROR_NOT_LOCKED
159	ERROR_BAD_THREADID_ADDR
160	ERROR_BAD_ARGUMENTS
161	ERROR_BAD_PATHNAME
162	ERROR_SIGNAL_PENDING
164	ERROR_MAX_THRDS_REACHED
167	ERROR_LOCK_FAILED
170	ERROR_BUSY
173	ERROR_CANCEL_VIOLATION
174	ERROR_ATOMIC_LOCKS_NOT_SUPPORTED
180	ERROR_INVALID_SEGMENT_NUMBER
182	ERROR_INVALID_ORDINAL
183	ERROR_ALREADY_EXISTS
186	ERROR_INVALID_FLAG_NUMBER
187	ERROR_SEM_NOT_FOUND
188	ERROR_INVALID_STARTING_CODESEG
189	ERROR_INVALID_STACKSEG
190	ERROR_INVALID_MODULETYPE

Error Code	Description
191	ERROR_INVALID_EXE_SIGNATURE
192	ERROR_EXE_MARKED_INVALID
193	ERROR_BAD_EXE_FORMAT
194	ERROR_ITERATED_DATA_EXCEEDS_64k
195	ERROR_INVALID_MINALLOCSIZE
196	ERROR_DYNLINK_FROM_INVALID_RING
197	ERROR_IOPL_NOT_ENABLED
198	ERROR_INVALID_SEGDPL
199	ERROR_AUTODATASEG_EXCEEDS_64k
200	ERROR_RING2SEG_MUST_BE_MOVABLE
201	ERROR_RELOC_CHAIN_XEEDS_SEGLIM
202	ERROR_INFLOOP_IN_RELOC_CHAIN
203	ERROR_ENVVAR_NOT_FOUND
205	ERROR_NO_SIGNAL_SENT
206	ERROR_FILENAME_EXCED_RANGE
207	ERROR_RING2_STACK_IN_USE
208	ERROR_META_EXPANSION_TOO_LONG
209	ERROR_INVALID_SIGNAL_NUMBER
210	ERROR_THREAD_1_INACTIVE
212	ERROR_LOCKED
214	ERROR_TOO_MANY_MODULES

Error Code	Description
215	ERROR_NESTING_NOT_ALLOWED
230	ERROR_BAD_PIPE
231	ERROR_PIPE_BUSY
232	ERROR_NO_DATA
233	ERROR_PIPE_NOT_CONNECTED
234	ERROR_MORE_DATA
240	ERROR_VC_DISCONNECTED
254	ERROR_INVALID_EA_NAME
255	ERROR_EA_LIST_INCONSISTENT
259	ERROR_NO_MORE_ITEMS
266	ERROR_CANNOT_COPY
267	ERROR_DIRECTORY
275	ERROR_EAS_DIDNT_FIT
276	ERROR_EA_FILE_CORRUPT
277	ERROR_EA_TABLE_FULL
278	ERROR_INVALID_EA_HANDLE
282	ERROR_EAS_NOT_SUPPORTED
288	ERROR_NOT_OWNER
298	ERROR_TOO_MANY_POSTS
317	ERROR_MR_MID_NOT_FOUND
487	ERROR_INVALID_ADDRESS

Error Code	Description
534	ERROR_ARITHMETIC_OVERFLOW
535	ERROR_PIPE_CONNECTED
536	ERROR_PIPE_LISTENING
994	ERROR_EA_ACCESS_DENIED
995	ERROR_OPERATION_ABORTED
996	ERROR_IO_INCOMPLETE
997	ERROR_IO_PENDING
998	ERROR_NOACCESS
999	ERROR_SWAPERROR
1001	ERROR_STACK_OVERFLOW
1002	ERROR_INVALID_MESSAGE
1003	ERROR_CAN_NOT_COMPLETE
1004	ERROR_INVALID_FLAGS
1005	ERROR_UNRECOGNIZED_VOLUME
1006	ERROR_FILE_INVALID
1007	ERROR_FULLSCREEN_MODE
1008	ERROR_NO_TOKEN
1009	ERROR_BADDB
1010	ERROR_BADKEY
1011	ERROR_CANTOPEN
1012	ERROR_CANTREAD

Error Code	Description
1013	ERROR_CANTWRITE
1014	ERROR_REGISTRY_RECOVERED
1015	ERROR_REGISTRY_CORRUPT
1016	ERROR_REGISTRY_IO_FAILED
1017	ERROR_NOT_REGISTRY_FILE
1018	ERROR_KEY_DELETED
1019	ERROR_NO_LOG_SPACE
1020	ERROR_KEY_HAS_CHILDREN
1021	ERROR_CHILD_MUST_BE_VOLATILE
1022	ERROR_NOTIFY_ENUM_DIR
1051	ERROR_DEPENDENT_SERVICES_RUNNING
1052	ERROR_INVALID_SERVICE_CONTROL
1053	ERROR_SERVICE_REQUEST_TIMEOUT
1054	ERROR_SERVICE_NO_THREAD
1055	ERROR_SERVICE_DATABASE_LOCKED
1056	ERROR_SERVICE_ALREADY_RUNNING
1057	ERROR_INVALID_SERVICE_ACCOUNT
1058	ERROR_SERVICE_DISABLED
1059	ERROR_CIRCULAR_DEPENDENCY
1060	ERROR_SERVICE_DOES_NOT_EXIST
1061	ERROR_SERVICE_CANNOT_ACCEPT_CTRL

Error Code	Description
1062	ERROR_SERVICE_NOT_ACTIVE
1063	ERROR_FAILED_SERVICE_CONTROLLER_CONNECT
1064	ERROR_EXCEPTION_IN_SERVICE
1065	ERROR_DATABASE_DOES_NOT_EXIST
1066	ERROR_SERVICE_SPECIFIC_ERROR
1067	ERROR_PROCESS_ABORTED
1068	ERROR_SERVICE_DEPENDENCY_FAIL
1069	ERROR_SERVICE_LOGON_FAILED
1070	ERROR_SERVICE_START_HANG
1071	ERROR_INVALID_SERVICE_LOCK
1072	ERROR_SERVICE_MARKED_FOR_DELETE
1073	ERROR_SERVICE_EXISTS
1074	ERROR_ALREADY_RUNNING_LKG
1075	ERROR_SERVICE_DEPENDENCY_DELETED
1076	ERROR_BOOT_ALREADY_ACCEPTED
1077	ERROR_SERVICE_NEVER_STARTED
1078	ERROR_DUPLICATE_SERVICE_NAME
1100	ERROR_END_OF_MEDIA
1101	ERROR_FILEMARK_DETECTED
1102	ERROR_BEGINNING_OF_MEDIA
1103	ERROR_SETMARK_DETECTED

Error Code	Description
1104	ERROR_NO_DATA_DETECTED
1105	ERROR_PARTITION_FAILURE
1106	ERROR_INVALID_BLOCK_LENGTH
1107	ERROR_DEVICE_NOT_PARTITIONED
1108	ERROR_UNABLE_TO_LOCK_MEDIA
1109	ERROR_UNABLE_TO_UNLOAD_MEDIA
1110	ERROR_MEDIA_CHANGED
1111	ERROR_BUS_RESET
1112	ERROR_NO_MEDIA_IN_DRIVE
1113	ERROR_NO_UNICODE_TRANSLATION
1114	ERROR_DLL_INIT_FAILED
1115	ERROR_SHUTDOWN_IN_PROGRESS
1116	ERROR_NO_SHUTDOWN_IN_PROGRESS
1117	ERROR_IO_DEVICE
1118	ERROR_SERIAL_NO_DEVICE
1119	ERROR_IRQ_BUSY
1120	ERROR_MORE_WRITES
1121	ERROR_COUNTER_TIMEOUT
1122	ERROR_FLOPPY_ID_MARK_NOT_FOUND
1123	ERROR_FLOPPY_WRONG_CYLINDER
1124	ERROR_FLOPPY_UNKNOWN_ERROR

Error Code	Description
1125	ERROR_FLOPPY_BAD_REGISTERS
1126	ERROR_DISK_RECALIBRATE_FAILED
1127	ERROR_DISK_OPERATION_FAILED
1128	ERROR_DISK_RESET_FAILED
1129	ERROR_EOM_OVERFLOW
1130	ERROR_NOT_ENOUGH_SERVER_MEMORY
1131	ERROR_POSSIBLE_DEADLOCK
1132	ERROR_MAPPED_ALIGNMENT
1140	ERROR_SET_POWER_STATE_VETOED
1141	ERROR_SET_POWER_STATE_FAILED
1150	ERROR_OLD_WIN_VERSION
1151	ERROR_APP_WRONG_OS
1152	ERROR_SINGLE_INSTANCE_APP
1153	ERROR_RMODE_APP
1154	ERROR_INVALID_DLL
1155	ERROR_NO_ASSOCIATION
1156	ERROR_DDE_FAIL
1157	ERROR_DLL_NOT_FOUND
1200	ERROR_BAD_DEVICE
1201	ERROR_CONNECTION_UNAVAIL
1202	ERROR_DEVICE_ALREADY_REMEMBERED

Error Code	Description
1203	ERROR_NO_NET_OR_BAD_PATH
1204	ERROR_BAD_PROVIDER
1205	ERROR_CANNOT_OPEN_PROFILE
1206	ERROR_BAD_PROFILE
1207	ERROR_NOT_CONTAINER
1208	ERROR_EXTENDED_ERROR
1209	ERROR_INVALID_GROUPNAME
1210	ERROR_INVALID_COMPUTERNAME
1211	ERROR_INVALID_EVENTNAME
1212	ERROR_INVALID_DOMAINNAME
1213	ERROR_INVALID_SERVICENAME
1214	ERROR_INVALID_NETNAME
1215	ERROR_INVALID_SHARENAME
1216	ERROR_INVALID_PASSWORDNAME
1217	ERROR_INVALID_MESSAGENAME
1218	ERROR_INVALID_MESSAGEDEST
1219	ERROR_SESSION_CREDENTIAL_CONFLICT
1220	ERROR_REMOTE_SESSION_LIMIT_EXCEEDED
1221	ERROR_DUP_DOMAINNAME
1247	ERROR_ALREADY_INITIALIZED
1248	ERROR_NO_MORE_DEVICES

Error Code	Description
1300	ERROR_NOT_ALL_ASSIGNED
1301	ERROR_SOME_NOT_MAPPED
1302	ERROR_NO_QUOTAS_FOR_ACCOUNT
1303	ERROR_LOCAL_USER_SESSION_KEY
1304	ERROR_NULL_LM_PASSWORD
1305	ERROR_UNKNOWN_REVISION
1306	ERROR_REVISION_MISMATCH
1307	ERROR_INVALID_OWNER
1308	ERROR_INVALID_PRIMARY_GROUP
1309	ERROR_NO_IMPERSONATION_TOKEN
1310	ERROR_CANT_DISABLE_MANDATORY
1311	ERROR_NO_LOGON_SERVERS
1312	ERROR_NO_SUCH_LOGON_SESSION
1313	ERROR_NO_SUCH_PRIVILEGE
1314	ERROR_PRIVILEGE_NOT_HELD
1315	ERROR_INVALID_ACCOUNT_NAME
1316	ERROR_USER_EXISTS
1317	ERROR_NO_SUCH_USER
1318	ERROR_GROUP_EXISTS
1319	ERROR_NO_SUCH_GROUP
1320	ERROR_MEMBER_IN_GROUP

Error Code	Description
1321	ERROR_MEMBER_NOT_IN_GROUP
1322	ERROR_LAST_ADMIN
1323	ERROR_WRONG_PASSWORD
1324	ERROR_ILL_FORMED_PASSWORD
1325	ERROR_PASSWORD_RESTRICTION
1326	ERROR_LOGON_FAILURE
1327	ERROR_ACCOUNT_RESTRICTION
1328	ERROR_INVALID_LOGON_HOURS
1329	ERROR_INVALID_WORKSTATION
1330	ERROR_PASSWORD_EXPIRED
1331	ERROR_ACCOUNT_DISABLED
1332	ERROR_NONE_MAPPED
1333	ERROR_TOO_MANY_LUIDS_REQUESTED
1334	ERROR_LUIDS_EXHAUSTED
1335	ERROR_INVALID_SUB_AUTHORITY
1336	ERROR_INVALID_ACL
1337	ERROR_INVALID_SID
1338	ERROR_INVALID_SECURITY_DESCR
1340	ERROR_BAD_INHERITANCE_ACL
1341	ERROR_SERVER_DISABLED
1342	ERROR_SERVER_NOT_DISABLED

Error Code	Description
1343	ERROR_INVALID_ID_AUTHORITY
1344	ERROR_ALLOTTED_SPACE_EXCEEDED
1345	ERROR_INVALID_GROUP_ATTRIBUTES
1346	ERROR_BAD_IMPERSONATION_LEVEL
1347	ERROR_CANT_OPEN_ANONYMOUS
1348	ERROR_BAD_VALIDATION_CLASS
1349	ERROR_BAD_TOKEN_TYPE
1350	ERROR_NO_SECURITY_ON_OBJECT
1351	ERROR_CANT_ACCESS_DOMAIN_INFO
1352	ERROR_INVALID_SERVER_STATE
1353	ERROR_INVALID_DOMAIN_STATE
1354	ERROR_INVALID_DOMAIN_ROLE
1355	ERROR_NO_SUCH_DOMAIN
1356	ERROR_DOMAIN_EXISTS
1357	ERROR_DOMAIN_LIMIT_EXCEEDED
1358	ERROR_INTERNAL_DB_CORRUPTION
1359	ERROR_INTERNAL_ERROR
1360	ERROR_GENERIC_NOT_MAPPED
1361	ERROR_BAD_DESCRIPTOR_FORMAT
1362	ERROR_NOT_LOGON_PROCESS
1363	ERROR_LOGON_SESSION_EXISTS

Error Code	Description
1364	ERROR_NO_SUCH_PACKAGE
1365	ERROR_BAD_LOGON_SESSION_STATE
1366	ERROR_LOGON_SESSION_COLLISION
1367	ERROR_INVALID_LOGON_TYPE
1368	ERROR_CANNOT_IMPERSONATE
1369	ERROR_RXACT_INVALID_STATE
1370	ERROR_RXACT_COMMIT_FAILURE
1371	ERROR_SPECIAL_ACCOUNT
1372	ERROR_SPECIAL_GROUP
1373	ERROR_SPECIAL_USER
1374	ERROR_MEMBERS_PRIMARY_GROUP
1375	ERROR_TOKEN_ALREADY_IN_USE
1376	ERROR_NO_SUCH_ALIAS
1377	ERROR_MEMBER_NOT_IN_ALIAS
1378	ERROR_MEMBER_IN_ALIAS
1379	ERROR_ALIAS_EXISTS
1380	ERROR_LOGON_NOT_GRANTED
1381	ERROR_TOO_MANY_SECRETS
1382	ERROR_SECRET_TOO_LONG
1383	ERROR_INTERNAL_DB_ERROR
1384	ERROR_TOO_MANY_CONTEXT_IDS

Error Code	Description
1385	ERROR_LOGON_TYPE_NOT_GRANTED
1386	ERROR_NT_CROSS_ENCRYPTION_REQUIRED
1387	ERROR_NO_SUCH_MEMBER
1388	ERROR_INVALID_MEMBER
1389	ERROR_TOO_MANY_SIDS
1390	ERROR_LM_CROSS_ENCRYPTION_REQUIRED
1391	ERROR_NO_INHERITANCE
1392	ERROR_FILE_CORRUPT
1393	ERROR_DISK_CORRUPT
1394	ERROR_NO_USER_SESSION_KEY
1395	ERROR_LICENSE_QUOTA_EXCEEDED
1400	ERROR_INVALID_WINDOW_HANDLE
1401	ERROR_INVALID_MENU_HANDLE
1402	ERROR_INVALID_CURSOR_HANDLE
1403	ERROR_INVALID_ACCEL_HANDLE
1404	ERROR_INVALID_HOOK_HANDLE
1405	ERROR_INVALID_DWP_HANDLE
1406	ERROR_TLW_WITH_WSCHILD
1407	ERROR_CANNOT_FIND_WND_CLASS
1408	ERROR_WINDOW_OF_OTHER_THREAD
1409	ERROR_HOTKEY_ALREADY_REGISTERED

Error Code	Description
1410	ERROR_CLASS_ALREADY_EXISTS
1411	ERROR_CLASS_DOES_NOT_EXIST
1412	ERROR_CLASS_HAS_WINDOWS
1413	ERROR_INVALID_INDEX
1414	ERROR_INVALID_ICON_HANDLE
1415	ERROR_PRIVATE_DIALOG_INDEX
1416	ERROR_LISTBOX_ID_NOT_FOUND
1417	ERROR_NO_WILDCARD_CHARACTERS
1418	ERROR_CLIPBOARD_NOT_OPEN
1419	ERROR_HOTKEY_NOT_REGISTERED
1420	ERROR_WINDOW_NOT_DIALOG
1421	ERROR_CONTROL_ID_NOT_FOUND
1422	ERROR_INVALID_COMBOBOX_MESSAGE
1423	ERROR_WINDOW_NOT_COMBOBOX
1424	ERROR_INVALID_EDIT_HEIGHT
1425	ERROR_DC_NOT_FOUND
1426	ERROR_INVALID_HOOK_FILTER
1427	ERROR_INVALID_FILTER_PROC
1428	ERROR_HOOK_NEEDS_HMOD
1429	ERROR_GLOBAL_ONLY_HOOK
1430	ERROR_JOURNAL_HOOK_SET

Error Code	Description
1431	ERROR_HOOK_NOT_INSTALLED
1432	ERROR_INVALID_LB_MESSAGE
1433	ERROR_SETCOUNT_ON_BAD_LB
1434	ERROR_LB_WITHOUT_TABSTOPS
1435	ERROR_DESTROY_OBJECT_OF_OTHER_THREAD
1436	ERROR_CHILD_WINDOW_MENU
1437	ERROR_NO_SYSTEM_MENU
1438	ERROR_INVALID_MSGBOX_STYLE
1439	ERROR_INVALID_SPI_VALUE
1440	ERROR_SCREEN_ALREADY_LOCKED
1441	ERROR_HWNDS_HAVE_DIFFERENT_PARENT
1442	ERROR_NOT_CHILD_WINDOW
1443	ERROR_INVALID_GW_COMMAND
1444	ERROR_INVALID_THREAD_ID
1445	ERROR_NON_MDICHILD_WINDOW
1446	ERROR_POPUP_ALREADY_ACTIVE
1447	ERROR_NO_SCROLLBARS
1448	ERROR_INVALID_SCROLLBAR_RANGE
1449	ERROR_INVALID_SHOWWIN_COMMAND
1500	ERROR_EVENTLOG_FILE_CORRUPT
1501	ERROR_EVENTLOG_CANT_START

Error Code	Description
1502	ERROR_LOG_FILE_FULL
1503	ERROR_EVENTLOG_FILE_CHANGED
1700	RPC_S_INVALID_STRING_BINDING
1701	RPC_S_WRONG_KIND_OF_BINDING
1702	RPC_S_INVALID_BINDING
1703	RPC_S_PROTSEQ_NOT_SUPPORTED
1704	RPC_S_INVALID_RPC_PROTSEQ
1705	RPC_S_INVALID_STRING_UUID
1706	RPC_S_INVALID_ENDPOINT_FORMAT
1707	RPC_S_INVALID_NET_ADDR
1708	RPC_S_NO_ENDPOINT_FOUND
1709	RPC_S_INVALID_TIMEOUT
1710	RPC_S_OBJECT_NOT_FOUND
1711	RPC_S_ALREADY_REGISTERED
1712	RPC_S_TYPE_ALREADY_REGISTERED
1713	RPC_S_ALREADY_LISTENING
1714	RPC_S_NO_PROTSEQS_REGISTERED
1715	RPC_S_NOT_LISTENING
1716	RPC_S_UNKNOWN_MGR_TYPE
1717	RPC_S_UNKNOWN_IF
1718	RPC_S_NO_BINDINGS

Error Code	Description
1719	RPC_S_NO_PROTSEQS
1720	RPC_S_CANT_CREATE_ENDPOINT
1721	RPC_S_OUT_OF_RESOURCES
1722	RPC_S_SERVER_UNAVAILABLE
1723	RPC_S_SERVER_TOO_BUSY
1724	RPC_S_INVALID_NETWORK_OPTIONS
1725	RPC_S_NO_CALL_ACTIVE
1726	RPC_S_CALL_FAILED
1727	RPC_S_CALL_FAILED_DNE
1728	RPC_S_PROTOCOL_ERROR
1730	RPC_S_UNSUPPORTED_TRANS_SYN
1731	RPC_S_SERVER_OUT_OF_MEMORY
1732	RPC_S_UNSUPPORTED_TYPE
1733	RPC_S_INVALID_TAG
1734	RPC_S_INVALID_BOUND
1735	RPC_S_NO_ENTRY_NAME
1736	RPC_S_INVALID_NAME_SYNTAX
1737	RPC_S_UNSUPPORTED_NAME_SYNTAX
1739	RPC_S_UUID_NO_ADDRESS
1740	RPC_S_DUPLICATE_ENDPOINT
1741	RPC_S_UNKNOWN_AUTHN_TYPE

Error Code	Description
1742	RPC_S_MAX_CALLS_TOO_SMALL
1743	RPC_S_STRING_TOO_LONG
1744	RPC_S_PROTSEQ_NOT_FOUND
1745	RPC_S_PROCNUM_OUT_OF_RANGE
1746	RPC_S_BINDING_HAS_NO_AUTH
1747	RPC_S_UNKNOWN_AUTHN_SERVICE
1748	RPC_S_UNKNOWN_AUTHN_LEVEL
1749	RPC_S_INVALID_AUTH_IDENTITY
1750	RPC_S_UNKNOWN_AUTHZ_SERVICE
1751	EPT_S_INVALID_ENTRY
1752	EPT_S_CANT_PERFORM_OP
1753	EPT_S_NOT_REGISTERED
1755	RPC_S_INCOMPLETE_NAME
1756	RPC_S_INVALID_VERS_OPTION
1757	RPC_S_NO_MORE_MEMBERS
1758	RPC_S_NOT_ALL_OBJS_UNEXPORTED
1759	RPC_S_INTERFACE_NOT_FOUND
1760	RPC_S_ENTRY_ALREADY_EXISTS
1761	RPC_S_ENTRY_NOT_FOUND
1762	RPC_S_NAME_SERVICE_UNAVAILABLE
1764	RPC_S_CANNOT_SUPPORT

Error Code	Description
1765	RPC_S_NO_CONTEXT_AVAILABLE
1766	RPC_S_INTERNAL_ERROR
1767	RPC_S_ZERO_DIVIDE
1768	RPC_S_ADDRESS_ERROR
1769	RPC_S_FP_DIV_ZERO
1770	RPC_S_FP_UNDERFLOW
1771	RPC_S_FP_OVERFLOW
1772	RPC_X_NO_MORE_ENTRIES
1773	RPC_X_SS_CHAR_TRANS_OPEN_FAIL
1774	RPC_X_SS_CHAR_TRANS_SHORT_FILE
1775	RPC_X_SS_IN_NULL_CONTEXT
1776	RPC_X_SS_CONTEXT_MISMATCH
1777	RPC_X_SS_CONTEXT_DAMAGED
1778	RPC_X_SS_HANDLES_MISMATCH
1779	RPC_X_SS_CANNOT_GET_CALL_HANDLE
1780	RPC_X_NULL_REF_POINTER
1781	RPC_X_ENUM_VALUE_OUT_OF_RANGE
1782	RPC_X_BYTE_COUNT_TOO_SMALL
1783	RPC_X_BAD_STUB_DATA
1784	ERROR_INVALID_USER_BUFFER
1785	ERROR_UNRECOGNIZED_MEDIA

Error Code	Description
1786	ERROR_NO_TRUST_LSA_SECRET
1787	ERROR_NO_TRUST_SAM_ACCOUNT
1788	ERROR_TRUSTED_DOMAIN_FAILURE
1789	ERROR_TRUSTED_RELATIONSHIP_FAILURE
1790	ERROR_TRUST_FAILURE
1791	RPC_S_CALL_IN_PROGRESS
1792	ERROR_NETLOGON_NOT_STARTED
1793	ERROR_ACCOUNT_EXPIRED
1794	ERROR_REDIRECTOR_HAS_OPEN_HANDLES
1795	ERROR_PRINTER_DRIVER_ALREADY_INSTALLED
1796	ERROR_UNKNOWN_PORT
1797	ERROR_UNKNOWN_PRINTER_DRIVER
1798	ERROR_UNKNOWN_PRINTPROCESSOR
1799	ERROR_INVALID_SEPARATOR_FILE
1800	ERROR_INVALID_PRIORITY
1801	ERROR_INVALID_PRINTER_NAME
1802	ERROR_PRINTER_ALREADY_EXISTS
1803	ERROR_INVALID_PRINTER_COMMAND
1804	ERROR_INVALID_DATATYPE
1805	ERROR_INVALID_ENVIRONMENT
1806	RPC_S_NO_MORE_BINDINGS

Error Code	Description
1807	ERROR_NOLOGON_INTERDOMAIN_TRUST_ACCOUNT
1808	ERROR_NOLOGON_WORKSTATION_TRUST_ACCOUNT
1809	ERROR_NOLOGON_SERVER_TRUST_ACCOUNT
1810	ERROR_DOMAIN_TRUST_INCONSISTENT
1811	ERROR_SERVER_HAS_OPEN_HANDLES
1812	ERROR_RESOURCE_DATA_NOT_FOUND
1813	ERROR_RESOURCE_TYPE_NOT_FOUND
1814	ERROR_RESOURCE_NAME_NOT_FOUND
1815	ERROR_RESOURCE_LANG_NOT_FOUND
1816	ERROR_NOT_ENOUGH_QUOTA
1817	RPC_S_NO_INTERFACES
1818	RPC_S_CALL_CANCELLED
1819	RPC_S_BINDING_INCOMPLETE
1820	RPC_S_COMM_FAILURE
1821	RPC_S_UNSUPPORTED_AUTHN_LEVEL
1822	RPC_S_NO_PRINC_NAME
1823	RPC_S_NOT_RPC_ERROR
1824	RPC_S_UUID_LOCAL_ONLY
1825	RPC_S_SEC_PKG_ERROR
1826	RPC_S_NOT_CANCELLED
1827	RPC_X_INVALID_ES_ACTION

Error Code	Description
1828	RPC_X_WRONG_ES_VERSION
1829	RPC_X_WRONG_STUB_VERSION
1898	RPC_S_GROUP_MEMBER_NOT_FOUND
1899	EPT_S_CANT_CREATE
1900	RPC_S_INVALID_OBJECT
1901	ERROR_INVALID_TIME
1902	ERROR_INVALID_FORM_NAME
1903	ERROR_INVALID_FORM_SIZE
1904	ERROR_ALREADY_WAITING
1905	ERROR_PRINTER_DELETED
1906	ERROR_INVALID_PRINTER_STATE
1907	ERROR_PASSWORD_MUST_CHANGE
1908	ERROR_DOMAIN_CONTROLLER_NOT_FOUND
1909	ERROR_ACCOUNT_LOCKED_OUT
2000	ERROR_INVALID_PIXEL_FORMAT
2001	ERROR_BAD_DRIVER
2002	ERROR_INVALID_WINDOW_STYLE
2003	ERROR_METAFILE_NOT_SUPPORTED
2004	ERROR_TRANSFORM_NOT_SUPPORTED
2005	ERROR_CLIPPING_NOT_SUPPORTED
2138	ERROR_NO_NETWORK

Error Code	Description
2202	ERROR_BAD_USERNAME
2250	ERROR_NOT_CONNECTED
2401	ERROR_OPEN_FILES
2404	ERROR_DEVICE_IN_USE
3000	ERROR_UNKNOWN_PRINT_MONITOR
3001	ERROR_PRINTER_DRIVER_IN_USE
3002	ERROR_SPOOL_FILE_NOT_FOUND
3003	ERROR_SPL_NO_STARTDOC
3004	ERROR_SPL_NO_ADDJOB
3005	ERROR_PRINT_PROCESSOR_ALREADY_INSTALLED
3006	ERROR_PRINT_MONITOR_ALREADY_INSTALLED
4000	ERROR_WINS_INTERNAL
4001	ERROR_CAN_NOT_DEL_LOCAL_WINS
4002	ERROR_STATIC_INIT
4003	ERROR_INC_BACKUP
4004	ERROR_FULL_BACKUP
4005	ERROR_REC_NON_EXISTENT
4006	ERROR_RPL_NOT_ALLOWED
6118	ERROR_NO_BROWSER_SERVERS_FOUND

This page was left blank intentionally.

Index

A

B

C

D

E

F

G

H

I

J

K

T

U

V

Z

Technology by QVI