ZETASIZER BASIC

ZETASIZER 1000/2000/3000
4000/5000/4700

Manual Number MAN 0179
Issue 1.0
February 1997

Spring Lane South, Malvern, Worcs.WR14 1XZ
Telephone: +44 (0) 1684 892456 Fax: +44 (0) 1684 892789

Malvern Instruments makes every effort to ensure that this document is correct. However, due to Malvern Instruments policy of continual product development we are unable to guarantee the accuracy of this, or any other document after the date of publication. We therefore disclaim all liability for any changes, errors or omissions after the date of publication.

Head office:
Malvern Instruments Ltd.
Spring Lane South,
Malvern.
Worcestershire. WR14 1XZ
U.K.

Tel + [44] (0) 1684-892456
Fax + [44] (0) 1684-892789

Printed in England

CHAPTER 1

CHAPTER 2

CHAPTER 3

CHAPTER 4

CHAPTER 5

CHAPTER 1

Introduction to this manual

1.1 Welcome

Welcome to the Malvern Basic Reference manual. This manual has been designed to give a detailed description of the operation and use of Malvern Basic. It gives details on the structure of a Malvern Basic program and details all commands available.

Warning: You must remember that the Zetasizer or the samples to be measured are potentially dangerous if misused. You must read the Health and safety booklet before operating the system.

1.2 Systems covered by this manual

Zetasizer is a generic name given to a family of systems. Each system within the family uses the same software and only vary in operation in small areas. For this reason this manual has been written to cover more than one instrument.

This manual covers the operation of:

Instrument	**Ref. Number**
Zetasizer 1000	DTS 5100
Zetasizer 2000	DTS 5200
Zetasizer 3000	DTS 5300
Zetasizer 4000	DTS 5400
Zetasizer 5000	DTS 5500
4700	PCS 100SM

1.3 Asumed information

Within this manual the Zetasizer system will be referred to as the “Zetasizer” or the “system” unless the information given is for a particular instrument.

Within this manual “BASIC” will refer to the specific Malvern Zetasizer version of the language.

Any program examples wil be shown in the `Courier font`.

1.4 Windows Terms

It is important that you understand some Windows terms before reading further. (Note that US spelling is used for some terms for compatibility)

Program - The Zetasizer **software -** it can also mean the Zetasizer Basic program used within the main Zetasizer software.

Cursor or **Pointer** - The graphic - usually a pointer that is moved on the screen by operation of the mouse.

Icon - The graphic on the desktop that represents a program.

Click - The mouse button is depressed and released. If this is not qualified with a button description then assume it is the left button. ‘Clicking a button’ means click the left mouse button with the cursor over the button.

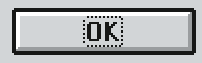
Double-click - Press and release the mouse button twice in quick succession. If this is not qualified with a button description then assume it is the left button. Use the Mouse icon in Program Managers **Control Panel** to change the double-click speed.

Dialogue Box - A window containing **controls**. The **OK** button accepts changes in the dialogue box. The **Cancel** button closes the dialogue without accepting the changes.

Control - This can mean a graphic on a dialogue like a button, listbox, textbox etc.

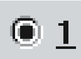
Press or **Select** - This means click the mouse over a control or use the accelerator key (the underlined letter) or use the **Tab** key to move the focus to a **control** then use the Enter key. Menu items can be selected using the cursor keys in the same way.

Button - This acts like a real-life button. Click to carry out an action. A typical button is shown below.



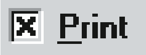
ILL 1992

Option Button or Radio Button - A series of buttons in a group, selecting one button cancels the others in the group. A radio button is shown below.



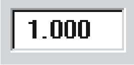
ILL 1993

Check Box - A button that can be toggled on and off. A check box is show below.



ILL 1994

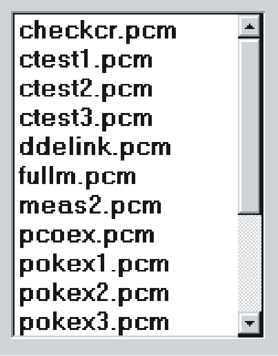
Text Box or Edit Box - A box you can type text or values into. A text box is shown below.



1.000

ILL 1995

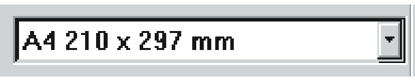
List Box - A box containing a list of options. Some List Boxes allow multiple entries to be selected.



checkcr.pcm
ctest1.pcm
ctest2.pcm
ctest3.pcm
ddelink.pcm
fullm.pcm
meas2.pcm
pcoex.pcm
pokex1.pcm
pokex2.pcm
pokex3.pcm

ILL 3864

Combination List Box or Combo Box- A combination of a list box with a text box. A button beside the text box displays or hides the list part of the control. In some cases you can type new values into the text box part, in others the text box just shows the current selection from the list.



A4 210 x 297 mm

ILL 3863

Drag - An action with the mouse which involves moving the mouse while holding down the left mouse button. This is used for moving icons or making multiple selections in a list box.

1.5 Menu Commands

Menu commands from the Malvern software are referred to in the form main menu-menu item. As an example, the command File-Save Sample refers to selecting the Save Sample item in the File menu. The same rules apply for sub-menus of sub-menus, so that Edit-Copy-Data refers to the Data item in the Copy sub-menu, which itself is a sub-menu of the Edit menu. Menu commands are always shown in bold text.

1.6 Where to find information

The information within this manual is divided into five chapters. The information within these chapters is summarized below.

Chapter 1 (This chapter)
Chapter 1 gives general information on the content and protocols used within this manual.

Chapter 2
Chapter 2 details the structure of a Malvern Basic program, giving details on how to write and run a program.

Chapter 3
Chapter 3 explains how to write page description programs for views and reports. These views and reports can then be allocated to the View menu.

Chapter 4
Chapter 4 is the main section of the manual. This section gives comprehensive details on all the commands and functions available within Malvern Basic.

Chapter 5
Finally chapter 5 gives details of advanced features.

It should be remembered that on-line help can be gained at any point when using the Malvern software.

1.7 Reporting Problems

Before reporting a problem please check the relevant sections of the user and software reference manual, or any accessory manuals, which may have an answer. If the problem persists try to give as much detail as possible.

If there is a problem in the software try to give information that will allow the engineers at Malvern to reproduce the conditions. If the problem is in the measurement or the analysis the Malvern engineers will require a copy of the Fullsize.pcp report.

To print a Data report:

- Change the View menu to Data.
- Select a report print in the File - Print dialogue.

CHAPTER 2

Introduction to Zetasizer basic

2.1 Introduction

The program language used in the Zetasizer software is similar to many other variants of Basic. Some features of other versions are not needed for our application and many extra functions have been added for specific use in particle sizing. Throughout these notes the term Basic will mean the specific Malvern Zetasizer version of the language.

Basic is a line interpreted language. This means that as a program is executed each line is checked for syntax and then executed. Unless controlled to do otherwise, the execution will continue with the next line. This means that, when you test a program, an error some way down the program will not be reported until you reach that line.

Zetasizer Basic program examples in the manual are shown in the `Courier` font.

The fundamental building block of a program is the statement. This is a single instruction to perform some action. A line in the program may contain more than one statement. Any statements on a line must be separated by colons (**:**).

For example :

```
X = 3 : Y = 2 : Z =-5
```

and

```
X = 3
Y = 2
Z =-5
```

are equivalent.

The lines of a program may be labeled. This is not necessary except when you need to refer to a specific line from within the program itself.

The apostrophe character (‘) is used to mark comments. Any line beginning with this character is ignored when the program is executed. Comments may be added at the end of lines using the same character.

For example:

```
‘Area of a rectangle

‘Prompt for length’
10 Input “Input length”, Rectlength
‘Prompt for width’
Input “Input width” Rectwidth


‘ Calculate area
RectArea = RectLength * RectWidth
Print dialogue, Using, “####.##”, RectArea
GoTo 10
```

Notice the following:

- The use of comments.

- The use of line numbers with tests to change the order of execution.

Basic is not case sensitive. That is, you can use upper and lower case characters in any combination. Spaces are not significant. Thus,

```
RectWidth = Val ( X$ )
```

and

```
RectWidth = Val (   X$ )
```

are equivalent.

The exception to this is that names of variables, commands, etc. may **not** contain spaces. You can use

```
InputBox$...
Inputbox$...
inputbox$...
etc.
```

but

```
Input Box$...
```

will fail.

There are various standard ways to make names of variables more readable. The example uses capital letters. You can also use the underscore character. For example **MeanSize** or **mean_size.**

2.2 Building blocks of programs

The building blocks of Basic program statements are:

Commands
The commands cause actions such as “measure”, “print”, etc. They also provide means of controlling the program by repeating loops, branching as the result of tests, etc.

Functions
The functions provide values for testing or printing.

Variables and numerical constants
The variables allow you to store and calculate with values from functions.

Operators
The operators are the means of adding, multiplying, etc. the variables and constants.

Expressions
Expressions are the result of evaluating combinations of variables, functions and operators. They may often be used as options for commands.

2.3 Commands and functions

Commands may perform a variety of actions depending on the choice made by additional parameters. They may also modify those actions with further parameters. Functions may be similarly modified.

The full list of all functions and commands (over 200 items and all their variant parameters) will not be described in detail here. The precise format of each command and function is given in chapter 4.

Within each entry in the reference section there is the name of the command or function followed by 'command' or 'function' as appropriate. Remember commands perform actions and functions give you a value to use.

2.4 Constants and variables

Numbers in Basic are all real numbers (sometimes referred to as floating point numbers). There are no integer variables. That is to say that all numbers are treated as if they contain decimal points. The number 2.0 has the same value as the integer (whole number) 2 for all uses in the language. If it is important to have a whole number in a particular case the language includes a function **Int()** which removes the part after the decimal point. For example, **Int(2.54)** returns the value **2.**

Numerical values are stored to single precision giving 6 significant figures and a range of 10^{-38} to 10^{38}.

There are also text values referred to as strings.

Constants are explicit values. For example the number 3.14159 is an explicit numerical constant and the string "MALVERN" is an explicit text constant.

Variables contain or represent numbers or text. Again, there are no integer variables. The name of a variable can contain up to 20 characters. The name may consist of alphabetic (A...Z, a...z) and numeric (0 ... 9) characters, the underscore (_) or period (.) characters, but must start with an alphabetic character.

Some examples of valid variable names are:

```
My_Variable
P.XPosition
Fi257
```

Some examples of invalid variable names are:

Variable	**Comment**
`5Fix`	Starts with a number.
`P&G`	Contains the & operator.
`A_Very_Long_Variable_Name`	Too long - would be truncated.

The names of variables are not case sensitive, MyVar , MYVAR and myvar would all refer to the same variable.

There is space for 50 user variables in any program. An error is reported if this number is exceeded. If one program calls another the variables in the calling program are not accessible from the called program (but see below for arrays).

It is good practice to choose variable names which describe the values they will contain. This 'self-documenting' approach together with comments will make program maintenance and debugging much easier.

Certain names will be disallowed. You cannot use a command or function name as a variable name.

The names of string (text) variables follows the convention of numeric variables but the names must end with a $ (dollar symbol). String arrays are not allowed. The total number of numeric, array and string variables must not exceed 50 per program. The maximum length of strings must be less than 256 characters.

Unlike some versions of Basic, the Zetasizer Basic does not require you to declare all variables explicitly before use. A numeric or string variable is recognized the first time a value is assigned to it. However arrays must be declared before use.

2.5 Arrays

An array is a group of values referred to by a single name. The individual values in the array are referred to by an index number.

For example MyData could be an array of 6 elements which would be referenced as MyData(0), MyData(1), MyData(2), ... , MyData(5).

Notice that the numbering of the array is assumed to start at zero. If it is convenient for clarity to use only the elements from number 1 onwards you can, of course, ignore element 0.

Unlike simple variables, it is necessary to declare arrays before they are used so that memory can be reserved for them. This is done using the **Dim** command.

Array names follow the same rules as for simple variables. String arrays are not allowed. All arrays are one-dimensional. That is arrays of the type X (3, 5) are not allowed.

For example:

```
Dim Ax ( 10 ), Ay ( 10 ), Big_List ( 1000 )
```

The dimension of an array is limited to 1000 (i.e. 1001 elements).

In a Dim command you may use variables and functions to specify the array sizes.

For example:

```
X = 10
Dim A ( 2 * X + 1 )
Dim B ( StoreChannel )
```

are valid.

Once an array is dimensioned you may not change its dimension.

Array elements may be used in any place in which a simple variable is used.
For example:

```
Dim X ( 5 ), Y ( 5 )
For i = 0 To 5
 A$ = InputBox$ ( “ ”, “ ”, “1.0" )
 X ( i ) = Val ( A$ )
Y ( i ) = Log10 ( X ( i ) )
Next
```

Note that when referencing array elements the index can be any valid expression.

```
Dim X ( 5 ), Y ( 5 )
For i = 0 To 5
 A$ = InputBox$ ( “ ”, “ ”, “0" )
 X ( i ) = Val ( A$ )
Next
XMax = -1000
For i = 0 To 5
 If X ( i ) > XMax Then
XMax = X ( i )
End if
Next
For i = 0 To 5
 Y ( 5 - i ) = X ( i ) * 100 / XMax
Next
```

2.6 Expressions and operators

Expressions consist of:

- Constants and variables
- Functions returning numerical or text values
- Operators

Expressions may be used in every place where a single constant or variable could be used. For example you may write

```
Save RecordAs “myfile”, 5
```

or

```
Save RecordAs “myfile”, i + 3
```

Operators are used to combine variables, constants and function values.

For numeric values the operators are:

Operator	
( )	Expressions in parentheses are evaluated first.
+ -	Unary plus and minus.
^	The exponentiation operator.

* / %	Multiply, divide and modulo.
+ -	Addition and subtraction.
< >=	Relational operators.
>=	Relational operator.
<=	Relational operator.
<>	Relational operator.
|	Bit-wise OR operator.
&	Bit-wise AND operator.

The expressions are evaluated left to right except with exponentiation which evaluates right to left.

Unary plus and minus attach a sign to a number. For example -FirstNumber is -1 if FirstNumber is 1, 7 if FirstNumber is -7, etc.

Exponentiation raises a number to a power. For example 3^4 is 81.

Multiply (*) and divide (/) are the normal arithmetic operations. The modulo operator calculates the remainder after division. For example 27% 5 is 2.

Expressions involving the relational operators evaluate to 1.0 if TRUE or 0.0 if FALSE.

The AND (&) and OR (|) operators convert the decimal numbers to integers before evaluation. Therefore the expressions on either side of these operators must resolve to values in the range 0 to 65535. For example:

```
211 & 112 = 80
(211.53) & 112 = 80
211 | 112 = 243
```

2.7 Control Structures

The simple progress from line to line of a program is very limited. In order to give the language real power we need commands which allow us to change the order of operation according to tests performed on variables.

In addition, the structure of the program can be simplified, the number of variables needed can be reduced and parts of the program can be made more portable by using subroutines.

In this sense, portable means that useful sequences of instructions can be copied and pasted into new programs.

Finally, we need to be able to repeat sequences of operations for whole sets of variables or for a pre-defined number of times.

2.7.1 Tests and Branches

The basic test is the structure:

```
If expression Then command
```

When the program reaches this line it evaluates the expression and if the answer is false (i.e. the expression evaluates to 0.0) the rest of the line is ignored. Otherwise the command is executed.

For example:

```
If CountsPerSec < 30000 Then Print Status, “Count Rate too small”;
Pause 5
```

This will check if the count rate is below 30000 and if it is will display the message in the status bar. It will then pause for 5 seconds.

What happens if the expression is false? The message is not displayed, the program goes to the next line and there is still a 5 second pause. We only want the pause if the message has been displayed so we could write instead:

```
If CountsPerSec < 30000  Then Print Status, “Count Rate too smalll”; :
Pause 5
```

This time the group of commands after Then is executed if true and ignored if false. Note that the commands after the Then must all be on one line. The enter key should be pressed after ...Pause 5 (enter key).

We could go on in this way adding commands separated by colons. The line might then become very long or difficult to read. In this case we could use the block form of the structure. This form is:

```
If expression Then
 command
....command
....
End if
```

Again the expression is tested and if it is true the sequence of commands following the If ... Then line is executed up to the line End if.

For example:

```
If CountsPerSec < 30000 Then
Print Status,“Count Rate too small”;
 Beep
Pause 5
 Print Status, Add more sample and re-run”;
End if
```

There is one more stage to go. Consider:

```
If CountsPerSec < 30000 Then
Print Status,“Count Rate too small”;
 Pause 5
End if
Print Report
```

If the test is false the block is ignored and the Print Report will be executed. But, if it is true we will get the warning and still get the Print Report. We could get round this in two ways. The GoTo command could be used:

```
If CountsPerSec < 30000 Then
Print Status,“Count Rate too small”;
 Pause 5
GoTo 20
End if
Print Report
20 ‘ This is where to continue.
```

The neater and clearer solution is to write:

```
If CountsPerSec < 30000 Then
Print Status,“Count Rate too small”;
 Pause 5
GoTo 20
Else
Print Report
End if
```

This general form:

```
If expression Then
 command
 command
 ...
Else
command
command
 ...
End if
```

executes the first block of commands if the expression is true and the second block if false.

2.7.2 Subroutines

Suppose that you write a test:

```
If CountsPerSec < 30000 Then
Print dialogue, Using, “###.### is too large”, ResultBelow (10 )
Beep
Pause 5
End if
```

and subsequently want to test some other parameters, say ResultAbove (60), etc. in the same way. You could write lots of bits of code copying the above.

The better way would be to use a subroutine.

2.7.3 Loops

It is often useful to be able to perform a group of operations a number of times in a loop. To achieve this Basic has the structure:

```
For variable = expression1 To expression2
...
...
Next
```

The counter variable is set to the value of expression1, the program lines following the For statement are executed until the Next statement is encountered. Then, 1 is added to the counter variable and it is compared with the value of expression2. The loop terminates when the value of variable is greater than or equal to expression2.

If expression1 and expression2 have the same value, the loop executes once

For example:

```
Data_Max = - 1000
For i = 1 To NumChannels
If StoreChannel ( i ) > Data_Max Then Data_Max = StoreChannel ( i )
Next
```

This will check through the data to find the maximum value.

Avoid changing the value of counter within the loop. Changing the loop counter is poor programming practice; it can make the program more difficult to read and debug.

The counter variable may be just that. For example:

```
For i = 1 To 4
Measure Sample
 Calculate
 Print Report
Next
```

Sometimes we may want to step through a set of values using, for example, every other one. To do this we can use the extended form of this command structure

```
For variable = expression1 To expression2 Step expression3
...
...
Next
```

This time instead of increasing the variable by 1 each time the Next is reached it is increased by the value of expression3.

For example:

```
OpenFile “standard”
‘Read the number of records in the file.
LastRecord = Numrecords
‘Process pairs of records to show differences between 1 & 2, 3 & 4,
etc.
```

```
For i = 1 To LastRecord Step 2
 Load Record i
 DifferenceRecord i + 1
View Difference
Pause 10
Next
```

2.7.4 Nesting

Control structures may be nested, that is placed one inside another. If .. Then can be placed inside For ... Next loops as seen above, or vice versa. If ... Then can be placed inside another If ... Then as below:

```
If CountsPerSec < 30000 Then
Print Status, “Count rate too smalll”;
 Pause 5
GoTo 20
Else
If ResultValid Then
  Print Report
 Else
Print dialogue, “Result invalid”
End if
End if
```

Similarly, For ... Next loops can be placed inside other For ... Next loops.

```
For i = 1 To 5
 For j = 1 To 8
 Move 20 * i, 10 * j
 Box 20 * i + 19, 10 * j + 9
  Next
 Next
```

There are limits placed on the number of levels to which you can nest structures. For example, For ... Next loops can be nested to 25 levels. None of the limits should cause you any practical problems !

You can see from the above examples that it would be easy, in complex structures, to lose track of the match between For and its corresponding Next or between the parts of If ... Then ... Else ... End if You can help to avoid this by indenting each block of statements as in the examples.

2.8 Special programs in Zetasizer Basic

The Zetasizer Basic language is used, not only to provide control and calculation routines, but also as the standard method of producing result tables and reports. It is also used for special programs to perform remote reporting and DDE linking.

For a detailed description of the use of Malvern Basic in page design see chapter 3. For DDE linking see ‘Dynamic Data Exchange’ in the Software Reference Manual and for remote operation see ‘Remote’ in the Software Reference Manual.

A program may also be run automatically when the Zetasizer software is first started. This may be used, for example, to automatically start a measurement sequence. See Control-Assign Program in the Software Reference Manual for more information.

2.9 Functions - a review

There are over 220 functions available but they will not all be dealt with individually here. For precise details on every function please consult chapter 4.

Page layout functions allow you to design pages by spacing rows and columns of figures in a way which is best adapted to the specific output device. For more details of page design see chapter 3.

Result information functions allow you to use values which describe the size distribution.

Sample documentation functions record the Sample identifier, Sample Notes and run number for accurate identification of the information on print-outs.

Hardware information records the specific settings of hardware used in the measurement.

Software settings give the parameters used in analysing the data.

2.10 Writing and running programs

2.10.1 Recording a program

If you intend to write a program that controls the Zetasizer measurements or process sample records you may find it convenient to record the basic outline of the program.

There are two modes of recording - if you elect to Fill in dialogue Boxes then the Zetasizer will act normally but all your commands will be recorded. If you do not select Fill in dialogue Boxes then the normal action of the commands will not be carried out - the program will record only the commands that were actioned.

See the Control-Record Program command in the Software Reference Manual for more information.

2.10.2 Editing a program

A recorded program may be modified or a new program may be written using the program editor.

The program editor is a separate window that is also used for entering page and report description programs. To edit a normal program the editor window is produced by selecting the Control-Edit Program command.

The editor window has its own menu bar.

Menu item	Function
File-New	Edit a new program
File-Load	Load a program from disk
File-Save	Save a program to disk
File-Save As	Save a program with a new name

File-Exit	Close the editor window
Edit-Undo	Undo the last edit
Edit-Cut	Cut the selected text to the clipboard
Edit-Copy	Copy the selected text to the clipboard
Edit-Paste	Paste at the cursor the text in the clipboard
Edit-Delete	Delete the selected text

The Run button runs the program and changes to show Stop. Click Stop to terminate the program.

2.10.3 Running a program

The first time you run a program you should use the Run button in the Program Editor. If there are any mistakes or errors then the cursor will move to the line with the error.

When the program is debugged it may be saved on disk (preferable in the PROGRAMS subdirectory). The program may then be run from the Control-Run Program command (See the Software Reference Manual for details).

2.10.4 Assigning Programs to Keys and Menu

The Control-Run Program command allows a single program to be run several times without having to pick it from the list of all the programs. If you have several programs you wish to run you may want to assign them to the end of the Control menu. An assigned program can be run either by selecting the entry from the Control menu or by using the function keys F5 to F8 (or those keys in combination with the control and shift keys).

See the Control-Assign Program command in the Software Reference Manual for more information.

2.11 Program Errors

There are two types of errors involved in using the Zetasizer programming language. Firstly there are errors caused by mistakes in the construction of your program. For example the statements:

```
If X > 0
GoTo 100
End if
```

will produce the error “THEN expected” because the first line is incomplete.

A list of these errors is given below. To correct such an error, consult the details of the particular command or function, if necessary, correct the error and re-run the Program.

The second type of error occurs when you run the Program. For example, if you attempt to use the command:

```
Load Record N
```

and the expected record cannot be loaded for some reason, the system will return an error code which you can use to make decisions about the next action:

```
Load Record N
If Error = 0 Then
Print Report
Else
X = Print Dialogue “Record no found”
End
End if
```

The values of any error codes which might be returned and their meanings are shown in the description of each individual command and function.

2.11.1 Basic Errors

The message displayed at the top of the program editing box gives you information about the type of error which has occurred. If the message is Syntax error in line n then inspect the line to ensure that the commands, functions, etc. are correctly spelled.

Specific error messages are :

Unbalanced parentheses
The number of left and right parentheses do not match. For example

```
X= Int ( ( N - 1 ) / 2
```

Correct to

```
X = Int ( ( N - 1 ) / 2 )
```

No expression present
A value was expected.

Equals sign expected
The line is interpreted as beginning with a variable name but does not have a value assigned. This often occurs when an incomplete line is entered or a command is mis-spelled.

```
For I = 1 to 5
Load Record 5
Print Report
Nxt
```

Correct the last line:

```
Next
```

Label table full
You may use up to 100 labels. Your program contains more than this number. To correct this you will need to simplify the structure of the program to use less labels. Using If... Then ... Else structures to replace GoTo may help.

Duplicate label
You have used a label more than once. To correct this change one of the labels.

Undefined label
There is a Goto or Gosub which uses a label which you have not included in your Program. To correct this check that you have typed the label correctly. Add the required label if necessary.

THEN expected
There is an If statement which is incomplete. For example:

```
If X > 0
GoTo 100
End if
```

Correct to

```
If X > 0 Then
GoTo 100
End if
```

TO expected
There is an error in the For statement. For example:

```
For I = 1, 10
```

Correct to

```
For I=1 To 10
```

Too many nested FOR loops
For ... Next loops may be nested to a maximum of 25. Your program structure is too complex.

NEXT without FOR
Each block of statements beginning For ... must end with a corresponding Next. The error is most easily caused when nesting several levels of For ... Next loops or when code has been cut and pasted incorrectly.

Too many nested GOSUBS
When you call a subroutine you may call another subroutine before returning to the original point in the Program. This process can continue up to 25 levels. This error indicates that you have exceeded the limit. Your Program structure is too complex.

RETURN without GOSUB
The most common cause of this error is not including an End statement. For example

```
For I = 1 to 10
GoSub 100
Next
100
Load Record I
Print Report
Return
```

Correct to:

```
For I = 1 To 10
GoSub 100
```

```
Next
End

100
Load Record I
Print Report
Return
```

Too many variables
You are allowed a maximum of 50 variables in any program. Try reducing the number you use using expressions and not calculating intermediate values as variables. For example:

```
X = I + 1
Y = 10 * X
Z = J
Move Y, Z
```

Change to:

```
Move 10 * ( I + 1 ), J
```

No such variable
A variable name is not recognised. The variable has not been assigned. For example:

```
For I = 1 To 3
X= I * 5 + A
Next
```

Correct to:

```
A = 3
For I = 1 To 3
X= I * 5 + A
Next
```

END IF expected
You have missed the end of a conditional block. For example:

```
If ( X <> 0 ) Then
...
...
Else
...
...
End
```

Correct to:

```
If ( X <> 0 ) Then
...
...
Else
...
...
```

```
End if
End
```

Divide by Zero
An attempt has been made to divide by an expression which evaluates to zero. This is often caused by mis-spelling a variable name or forgetting to assign a value. For example:

```
NewX = I + 3
Y = 1 / NwX
```

Math domain error
You have used a Math function with an illegal expression. For example, the following will give an error when calculating Log(0):

```
For I = 0 To 10
Y = Log ( I )
Next
```

Terminated by User
You have used the Stop button while the program is executing.

CHAPTER 3

Designing pages and reports

3.1 Introduction

The standard tables and reports provided with the Zetasizer system are produced by various page and report programs. A page (or report) program is a "script" written in Malvern Page Description Language (PDL), which allows selection of the display items, text, graphics, and font and format of the displayed text.

Although Zetasizer provides you with a number of standard tables and reports for presenting measurement data and results, you have the option to design your own tables and reports in styles that suit your need.

This chapter describes the PDL language and program, and example of generating customised table and report. See the Software Reference Manual for details on standard Zetasizer tables and report.

3.2 Writing page description programs

A page description program is a special form of Zetasizer Basic program which only allows you to access the Zetasizer data and results and display them in selected style and format. It cannot be used to control certain operations of the Zetasizer as a normal Zetasizer Basic program would do. Therefore, before attempting to write a table or report program you should familiar yourself with the concepts of the Zetasizer Basic language. See chapter 2 for detailed description on Zetasizer Basic variables, expressions and functions.

Like all Basic programs the execution of a page description program starts from the first line in the program and ends when an **End** statement is reached or when the execution reaches the last line of the program. The program stops when a programming error is encountered. Like writing any program it is recommended to use comment lines to give brief description on the program, so that the program is easy to read and debug should any errors occur. A comment line starts with an apostrophe character (‘).

A table or report program should start by defining the area where the contents of the table or report are to be placed. Once the display area is defined, text, lines and boxes can be drawn to form a table, and the values of the Zetasizer data and results accessed and displayed. When there are many items to be displayed, it is a good idea to separate the items into groups and each group is drawn by a section of the program.

A table is drawn in the Table pane on the right hand size of the Malvern Zetasizer main window. The size of a table is limited by the dimension of the computer screen. Although you can define a table size larger than the size of the screen, you will need to use window scroll bars to move the table around to allow you to view the contents. It is therefore recommended that the width of a table does not exceed the width of the screen. All standard Zetasizer tables are half the screen width.

Text, lines and boxes can be drawn at a location specified by (X,Y) co-ordinate on a table or report. The co-ordinate has its origin at the top left corner of the display area. All dimensions in the program are measured in millimetres (mm).

A page description program can be created or edited using the built-in Malvern page program editor or any text editor. It is recommended, however, that the Malvern editor is used so that the program can be tested while being edited.

To create a new program select the **New** command from the **File** menu in the Malvern Editor. **New** automatically enters as a first line the **Table** command to allow you to give a title to the page or report. This name appears in the View menu when the page is assigned and when the **Info** button is pressed in the **Setup - Table** dialogue.

During editing your program you can use the **Run** button in the editor to test the program, the table or report you are editing will be shown on screen. A screen ruler can be switched on using the **Options - Rulers** command to help you position items on the table or report.

The program is saved using the editor’s **File - Save** / **-Save As** commands.

For more information on the editor window see ‘Editing a Program’ in chapter 2.

CHAPTER 4

Zetasizer Basic reference

4.1 List of All commands and functions

A

Abort Command

ABS() Function

Achannel Function

ACOS Function

AnalysisType Function

Analysis Command

Angle Command

ASC Function

Asin Function

Asm Command

B

BackGround Function

BackGroundRatio Function

Bchannel Function

Beep Command

Box Function

C

CalcFarPoint Function

Calculate Command

CalcZimm Command

Capscan Command

ChangeTable Command

ChDir Command

Chr$(character) Function

Clear Command

Comm Command

Common Command

Copy Command

Corr Command

Cos( ) Function

CountsPerSec Function

CountsPerST Function

CurrentTime Function

D

Data Function

DataType Function

DateFormat Command

Dde Command

Delay Function

Delete Command

Dilation Function

Dim Command

Duration Function

E

Edit Function

ElapsedTime Function

ElectroMobility

Else Command

Enable Command

Else Command

End Command

EndAngle

Error Function

Execute Command

Exit Command

EXP Function

F

File Command

FillColour

FirstPoint Function

Fit Function

Fitdata Function

FitNum

FitOrder Function

FitTime Function

FitResiduals Function

Font Command

For Command

Format$

Frequency Function

FrequencyChannels Function

G

Get Function

Gosub command

Goto

Graph

K

Keep command

I

If Command

Inp Function

Input Command

InRange Function

InStr function.

Int Function

Intensity Function

Iscan Mode command

InstrumentType Function

L

LastPoint Function

LCase$ Function

Left$ Function

Len Function

Line Command

LineColour Command

LineTo Command

Ln Function

Load Command

Log Function

LTrim$ Function

M

MAddSelect Command

Mcalculate Command

MDelete

MeasFarPoint Function

Measure Command

Merit Function

Message Function

Mid$ Function

Mie Function

Mload Command

Mobility Function

MobilityChannels Function

MolecularWeight

Move (x,y)

Mprint Command

Msave

MSelect

N

Number function

NumChannels function

NumFreq Function

NumScans Function

NumSizes

O

Open Command

OsmoticMob Function

Out Command

P

Page Command

Pause

Peak Command

Picture Command

Polydispersity Function

Print Command

Print Page Command

Print Report Command

Print ...Using command

R

Range

RawFreq Function

RecordNumber Function

Reference Function

Residuals

Return

Rg

Right$ Function

RTrim$ Function

Run

RunMode

S

Sample Time

Save Command

ScaledCount

ScanAngle

ScanDuration

ScanQ Function

ScanType

Select Command

Set Command

Setup Command

Setup table Command

Setup Report

Setup Userlogtable

Simulate Command

Size Mode Command

Sizes Function

Spacing

StartAngle

Step

StoreChannel

Str$ Function

StrCmp Command

T

TextAlign

TextColour

Then

TimeArray

TotalSamples

To

Transfer Command

U

UCase$

Use Command

V

Val Function

View

ViewNew Command

Virial2 Function

Virial3 Function

Volume

W

Write

WriteAt Command

X

Xchar Function

Xpage Function

Xpos Function

Y

Ychar Function

Ypage Function

Ypos Function

Z

ZAnal1

ZAnal2

ZAveMean Function

ZeroFieldWidth Function

ZetaChannels Function

ZetaPotential Function

ZetaSpectrum

ZetaVar

ZimmError Function

ZimmFitOrder Function

ZimmNumConcs

ZimmNumAngles

ZimmNumPoints

ZimmAngle Function

ZimmQ Function

ZimmResidual

ZimmIntensity Function

ZimmFit

ZimmConc

RadiusofGyration

ZimmFitorder

ZimmError

Zerowidth

Zeta Command/Function

ZetaPotential

ZetaValue Function

ZetaVar Function

Zlower Function

Zupper Function

4.1 Abort Command

Application
Aborts measurement sequence.

Syntax
Abort

4.2 ABS() Function

Application
Return the absolute (constrained positive) value of the argument.

Syntax
Abs(x)

Notes
This function returns the absolute value of a number.

Example
b=-123
a=abs(b)

print dialogue using “####”, a ‘prints 123 in a dialogue box.

4.3 Achannel Function

Application
Data information for Size and Zeta records.

Syntax
AChannel

Notes
AChannel monitor counts of current selected record. ie. the total number of photon detections processed by the correlator.

4.4 ACOS Function

Application
Mathematical Function

Syntax
acos (number)

Notes
This function returns the angle, in radians, of which the cosine equals the number. The value returned is in the range 0.0 to 2Pi.

The value of number must be in the range -1.0 to 1.0.

Examples
acos(0) ‘returns 1.57

4.5 AnalysisType Function

Application
Returns current analysis type of record.

Syntax
AnalysisType

Notes
The return value can be interpreted as follows:-

0 = monomodal(Cumulant method),
1 = multimodal sampling,
2 = Multi Angle,
3 = Contin
4 = NNLS
5 = Automatic

4.6 Analysis Command

Application
Sets current analysis type.

Syntax
Analysis Cumulants [,order,weighting]
Analysis multimodal [,x1,x2]
Analysis Contin [,x1,x2]
Analysis MultiAngle [,x1,x2]
Analysis NNLS [,x1,x2]
Analysis Pref
Analysis Fit
Analysis ZeroFieldWidth

Notes
Analysis Cumulants

The two parameters are optional. The first is the order of the fit, the second the weighting used in the analysis, which must take the values of 0, 2 or 4. If no parameters are supplied, Cumulants is used as the default method, and default values of the parameters are assumed.

Analysis multimodal, Analysis Contin , Analysis MultiAngle, and Analysis NNLS.

For the other analysis types, either of the parameters are optional.If none are given, the analysis range used is automatically chosen. If one is given then it is interpreted as a range. If two are supplied, then they are interpreted as upper and lower limits of the analysis range.

Analysis Pef
Brings up the setup analysis dialog

Analysis Fit
Order of fit for current analysis.

Analysis ZeroFieldWidth
Calculate zerofield and apply to current record

4.7 Angle Command

Application
Instrument control

Syntax
Angle (x)

Notes
Set angle on instrument connected if the instrument has multi-angle properties.

4.8 ASC Function

Application
Get ASCII code of string

Syntax
ASC(Str$)

Notes
This function returns the numerical ASCII value of the first character of Str$.

Example
ASC(“ALPHA”) ‘return 65 - the ASCII value of ‘A’

4.9 Asin Function

Application
Mathematical function

Syntax
Asin(number)

Notes
This function returns the angle, in radians, of which the sine equals the number. The value returned is in the range -Pi to Pi

The value of number must be in the range -1.0 to 1.0.

Example
v = Asin(0) ‘ returns 0.0

4.10 Asm Command

Application
Allows Autosampler control from macro language.

Syntax

```
Asm Ini [,PortId] [,BaudRate]
Asm Wait [,string$] sm  File filename$
Asm Tube                TubeNumber
Asm Dilute              DiluteVolume
Asm Key                 KeyId
Asm Syringe             SyringVolume
```

Asm Rows	NumberRows
Asm Columns	NumberColumns
Asm RackCode	RackCode
Asm Height	Height

Notes
Asm Ini[,port,baud rate] Initialise Autosampler. Optionally set RS232 port number and baud rate. These can also be set under Setup RS232 from the menu.

Wait
Waits for string (if none “END”)

File 10meSends ascii file to autosampler

Ini iPortInitialises gilson on comm port i

Tube Goto tube i
Dilute
Row
Key
Syringe
Rows
Columns
RackCode
Height

Asm Tube number
Select a particular sample tube, inject a sample at the injection port. The details of the inject process can be altered by editing the file “tube.asf”. Normally number is in the range 1-70.

Asm Wait [,text$]
Wait until a particular message is displayed on the autosampler control panel. (defaults to “END” if text omitted). The match is over the number of characters supplied as text. Case is ignored.

Asm File filename$
Download filename$ (must have extension “.asf”, from disk to the autosampler and run it. Examples of useful programs follow:

“wash.asf” Cleans sample system between injections
“flush.asf”Flush the sample system with solvent

These are supplied, but could be edited by the user to alter details of the sample processing. The filename must be in quotes and the extension can be omitted if desired.

Key
Set the sampler to keypad (local) control. use asm ini again to claim remote control.

Dilute ,nn [,]
Dilutes the sample placed in the first row (nearest the sampler unit) over nn columns. ll sets the number of rows of sub samples to generate. If omitted it is set to one less than set by asm row, or 6 by default. The dilution process is carried out by code stored in file “dil.asf” and can be modified by the user.

Syr [,n1][,n2]
Define device to use for dilution. Default n1=0, n2=10000. (use diluter 0, volume 10000 microliters)

RackCode
Set the Autosampler rackcode. Consult your Gilson manual for details.

Rows
Number of rows in the sample tray. This is needed for the Tube command to address the correct row and column correctly.

Cols
Number of columns in the sample tray. This is needed for the Tube command to address the correct row and column correctly.

Height
Sets syringe height. This is necessary with no standard cuvette sizes, for example, the narrow cuvettes used on a 320 sample type rack.

Example
Some autosampler examples.

```
asm ini  ‘ Initilialise sampler
asm fil,wash’ load and run sampler program ‘wash.asf’
asm wai’ wait for display END to signify wash complete.
asm dil,5,4 ‘ take the sample in row 1, dilute it 4 times into successive
‘ rows of the same column. Repeat over the first 5 columns.
size mode ‘select size analysis
for i=1 to 25
asm tube,i’ select tubes from 1 to 25
asm wait for tube inject to finish then wash inject line
asm fil,wash
measure sequence 1’ run a measurement
asm wait for the wash sequence to finish
‘ (unless the measurement time is short
‘ this will already have happened), but it is ‘ ‘
‘ safer to include the step.
```

4.11 BackGround Function

Application
Returns background used in a size analysis.

Syntax
Background()

4.12 BackGroundRatio Function

Application
Data information for Size records.

Syntax
BackgroundRatio

Notes
The background ratio is derived from the ratio of the far point, and the theoretical background or baseline value derived from the monitor channels. This is primarily used as an indication of signal quality for a size measurement. It is further used to derive In Range.

4.13 Bchannel Function

Application
Data information for Size and Zeta records.

Syntax
BChannel

Notes
The contents of the B monitor channel of the correlator. It is used in Autosizer systems for temperature readout, available for User input on 4700/Zetasizer.

4.14 Beep Command

Application
Command to alert the user.

Syntax
Beep

Notes
This command will cause an audible tone to sound. Use this to draw attention to an alert box, dialogue or input.

4.15 Box Function

Application
Page program graphics.

Syntax
Box x,y

Notes
Draws a rectangle from the current anchor position x,y mm in size.

The current line style and colour is used to draw the outline and the box is filled with current fill colour.

Example
Move 10,10
Box 120,50

4.16 CalcFarPoint Function

Application
The background or baseline calculated from the monitor channels.

Syntax
CalcFarPoint

4.17 Calculate Command

Application
Does calculation with current analysis.

Syntax
Calculate

Notes
The calculate command operates on the record currently in memory. Note that the commands Select, Load Rec, Transfer implicitly load a record into memory.

Size mode
Does calculation with analysis set by the Analysis Analysistype command on the record currently in memory.

Zeta mode
Recalculates the zeta potential distribution of the record currently in memory. The use of the secondary command Zero Field Width will recalculate the value of the current zero field correction width.

Intensity mode
Performs a ‘Radius of Gyration’ calculation on the record currently in menu.

4.18 CalcZimm Command

Application
Calculates the Molecular weight, Radius of gyration and virial coefficients for a group of selected records of the appropriate type, ie. an existing Zimm plot.

4.19 Capscan Command

Application
In a Zeta potential context a number of records are fitted with a parabola for mobility against cell position to extract electro-osmotic and electrophoretic results separately.

4.20 ChangeTable Command

Application
Redraw current table display.

4.21 ChDir Command

Application
Changes current directory on the system disk. Parameter is new directory eg. ChDir “c:\pcs”

4.22 Chr$(character) Function

Application
Returns a string of one character which has the ASCII value of Character.

4.23 Clear Command

Application
Clears the live buffer of records.

Close Command

Application
Closes specified device.

Syntax
Close File
Close Comm

Notes
Close File
Closes the ASCII file opened by the File command without deleting it, permitting input/output to start from the start of the file.

Close Comm
Closes currently open comm port.

4.24 Comm Command

Application
Utility command for aspects of the BASIC serial communications.

Syntax

Comm	delimiter	delimeter
Comm	timeout	time_out
Comm	clear	

Delimiter delimeter
Specifies end of line character for comms input. The number delimeter specifies the ASCII value of the delimiter text.

Timeout time_value
Specifies the maximum time the comms input will wait for incoming data.

4.25 Common Command

Application
BASIC variable definition

Syntax
Common Dim a(100)
Common b

Notes
Declares the variables listed as common, so that they can be shared between programs, including between page and macro programs.

Examples
common dim a(100) ‘ No separate declaration is needed.
common b,c

4.26 Copy Command

Application
Copy information via the DDE to the Clipboard or another Application.

Syntax
Copy Graph
Copy Table
Copy Link
Copy String

Notes
Copy Graph
Copy Table

Place the current display from the table area as a metafile onto the clipboard, so that it can then be placed in another application. Typically this would involve executing an Edit Paste in that application.

Copy string
Copies the contents of the system string buffer to the clipboard. See Print String for details.

Copy Link
Runs a Macro program that places selected information in a DDE link to another application. See Chapter 4 for more details.

4.27 Corr Command

Application
Correlator control commands.

Syntax

```
Corr    Prescale dialog
Corr    Prescale n1, n2, ...
Corr    Wait
Corr    Clear
Corr    start
Corr    halt
Corr    config
Corr    options
Corr    serial
Corr    parallel
Corr    background
Corr    progress
Corr    sampletime [,sampletime]
Corr    countrate
Corr    size
Corr    dilation d
Corr    delay
Corr    duration
Corr    divide   [,n     ]
Corr     live
Corr    rate
Corr    select
Corr     test     TestClock [,Test]
Corr CountRate
```

Notes

Prescale

This is applicable to the 7032 only. It sets the prescale values of the various sub-groups of the correlator. If no arguments are supplied, auto prescale is implied.

Wait

Wait for experiment to finish (use after Start)

Clear

Clear correlator. This is only applicable to the 7032

Start

Start correlator.

Halt

Stop correlator.

SampleTime

Set correlator sample time (in microseconds). If the parameter value is omitted an experiment is run to set the sample time automatically as can be requested by checking ‘auto’ in a measurement sequence.

Size

Set correlator size in channels (64,128,256).

Dilation

Set dilation factor 1 - 15. (7032 only)

Delay

Set delay to far point, 0 - 4096 in steps of 16. This is applicable to the 7032 only.

Duration

Set experiment duration (in seconds)

Divide [,n]

For the 7032, n may take the value of 1,2,4 or 8 divides the correlator into ,n Set parallel mode (2,4,8). If the parameter is omitted the selection is made in accordance with the automatic selection made in a measurement sequence.

For the 7132 and 70128 correlators n 1 will use linear mode, n=1 will put the correlator into log mode.

Live

0 Hide live display, 1 show live display.

Options

Bring up options dialogue from Correlator control window.

Config
Bring up Config dialog from Correlator control window.

Serial
Brings up dialogue box for user to enter settings like sample time, duration etc.

Parallel
Brings up dialogue box for user to enter settings like sample time, duration etc.

CountRate
Returns the current countrate measured by the correlator.

Test
This sets the correlator into test mode. The parameters depend on the correlator type.

7032 Correlator.
TestClock 0 turns on a 1 MHz test clock

70128 Correlator

TestClock	0	Turns off the test cloc
	1	1 count per sampletime test clock
	2	1 count per Test sample times
	3	100 KHz clock

7132 Correlator

Test Clock	0	Turns off the test clock
	1	25 KHz
	2	200 KHz
	3	5 MHz

Example
Example of correlator control using correlator commands:

```
corr divide 1
corr duration 200
corr sampletime 23
corr live 1
corr clear
corr start
```

Set the correlator to serial mode (all store channels disposed in a single section, with common sample time), set a duration of 200 seconds, a correlator sample time of 23 microseconds, enable the live display window, clear the correlator contents to zero, then start it accumulating data.

4.28 Cos( ) Function

Application
Mathematical Function

Systax
Cos(x)

Notes
Returns the cosine of the argument. The argument is in radians, and should take values between 0 and 2Pi.

Example
a=3.14159
print dialog using “##.###”,cos(a) ‘will display -1.000

4.29 CountsPerSec Function

Application
Returns the value of the Counts per second stored in the data of the current record.

Syntax
CountsPerSec ()

Notes
By definition this is equal to the contents of the A monitor channel divided by the time in seconds that the correlator ran to gather that data. Note that this is the value of the currently loaded record, NOT the actual count rate.

4.30 CountsPerST Function

Application
Data information for Size and Zeta records.

Syntax
CountsPerST

Notes
The Contents of the A monitor divided by the total number of correlator samples taken to get that count. This is the value of the value stored in the current record, and so does NOT reflect the current value monitored by the system.

4.31 CurrentTime Function

Application
Determines the elapsed time since system startup.

Syntax
CurrentTime

Notes
Returns Current time (in seconds) since system startup

4.32 Data Function

Application
Returns data values stored in the current record.

Syntax
data corr i
data qualityfactor
data (parameter_keyword)

Notes
Data corr i
Returns channel i of the first order correlation function.

Data QualityFactor
Returns the quality factor (one or zero). The quality is satisfactory if the function returns one.

Data Parameter_Function
Returns the value of the particular Parameter_Function in the record.

Valid values of Parameter_Function commands are:-

data CellType Instrument CellType

Valid values are -1 unknown (4700 etc)

0	ZEM5104	
1	ZETA5104	
2	ZETA5103	
3	ZETA5110	
4	ZETA5126	
10	ZEM010	Capillary cell
11	ZEM011	Non-aqueous cell
14	ZEM014	Aqueous parallel
data	ModulatorFrequency	Modular frequency
data	RefIndex	Refractive index of dispersant
data	Viscosity	Viscosty of the dispeserant
data	Angle	Scattering angle
data	Temperature	Read back temperature
data	Wavelength	Laser wavelength
data	Conductivity	Conductivity
data	CalibrationConst	CalibrationConstant for conductivity calibration
data	FringeSpacing	Fringe spacing for Zeta analysis
data	Dielectric	Dielectric constant
data	CellSpacing	Distance between electrodes (mm)
data	RFReal	Real Refractive index of sample
data	RFImag	Imaginary Refractive index or absorbance of sample
data	MarkhowinkA	Mark Houwinck A parameter
data	MarkhowinkK	Mark Houwinck K parameter
data	Cellposition	Cell position
data	Ph	pH measurement
data	CellCUrrent	Cell current
data	CoreRatio	Ratio of inner to outer diameter for shell model
data	RFRealCore	Real refractive index of core
data	RFImagCore	Imaginary refractive index of core
data	Fka	F(ka) of sample
data	Concentration	Concentration of sample (g/l)
data	RFStandard	Refractive index of standard (Zimm Scan)
data	dndc	Refractive index increment (ml/g)
data	RayleighRatio	Rayleigh ratio

Example
e = Data CellVoltage / Data CellSpacing

A calculation of electric field strength for the current record.

4.33 DataType Function

Application
Sets data type of the function

Syntax
DataType n

Notes
In Size mode this flag determines whether the data is a first or second order correlation function, depending on whether the value is 1 or 2 respectively.

In Zeta mode this flag determines whether the raw zerofield width is used or a voltage corrected zerofield width is used, depending on whether the value is 2 or 1 respectively.

4.34 DateFormat Command

Set date format n= 1, = long (default) eg. 22nd June 1992, 0 = short eg. 22/6/92

4.35 Dde Command

Application
Communication with other Windows programs.

Syntax

DDE Connect	,conversation_number, service_name$, topic$
DDE Poke	,conversation_number, item$ [,text$]
DDE Execute	,conversation_number
DDE Terminate	,conversation_number
DDE Dialog	see diffraction manual entry
DDE Request	,conversation_number, item$, information$
DDE Update	
DDE Timeout	, time

Notes
The DDE command allows information to be exchanged with other Windows programs through Dynamic Data Exchange (DDE). conversation_number is the number of the conversation and takes integer values between 1 and 5. The PCS software allows 5 DDE conversations at any one time. See chapter 11 “Linking with other applications” in the Software Reference Manual for more information.

DDE Connect, conversation_number, service_name$, topic$
This command is used to connect to a DDE server. service_name$ is the service name of the other appliciation you wish to link to which is normally the same as the .EXE file name. topic$ is the topic of conversation which is application dependent.

DDE Terminate, conversation_number
The command terminates the DDE conversation with number conversation_number

DDE Timeout, time
This sets the timeout time for the DDE exhange. The number is the time in seconds.

DDE Poke, conversation_number, item$ [,string$]
This command allows the user to send information to another Application which is linked by

conversation_number. The string item$ is the item name assigned to the data and is application dependent. The data sent will be the information held in the Basic system string if the parameter string$ is not supplied (see Print String) otherwise the data in string$ will be sent.

DDE Execute converation_number, command$
The command allows the user to send a command to another application known as the server linked by conversation_number. The format of the text in command$ is application dependent.

DDE Update
This refreshes the information which is sent to the partner program. Normally this is updated for an established hot link at the end of a calculation, but this command enables you to force the update measurement on request.

Example
This example transfers values to a column of a Microsoft Excel spreadsheet.

```
DDE Connect ,3,"Excel", “sheet1"
for i=1 to numchannels
 a$ = str$(StoreChannel(i))
 F$ = “R” + Str$ (i) + “C1"
 DDE Poke, 3, F$ ,a$
next
DDE Terminate 3
```

4.36 Delay Function

Application
Result and data information from Size records.

Syntax
Delay

Notes
The delay period (in sample times) set in the post computational card before the Far point channels. This function is only available on the 7032 correlator.

4.37 Delete Command

Application
Dynamic Data exchange and result logging.

Syntax
Delete File
Delete Log
Delete String

Notes
Delete File deletes the file last named in the File command. It does not close the file, but simply empties it.

Delete String clears the BASIC system string which has been created with the Print String command. This is necessary before adding new items to a DDE link or copying information to the clipboard. Missing the deletion will mean that the items sent or copied will include all previous items. Delete Log clears the current log file.

Example
File “tempdata.txt”

Delete File

4.38 Dilation Function

Application
Result and data information from Size records.

Syntax
Dilation

Notes
The dilation setting of the correlator, ie. the sample time multiplier between the different store blocks. For the 7132 and 70128 correlators this is always two.

4.39 Dim Command

Application
Basic variable definition

Syntax
Dim array_name(n)

Notes
The Dim command reserves space for arrays. array_name is the name for the array being defined and n the size required.

The array names follow the convention for other variables. Only the first 30 characters are significant and the name must start with an alphabetic character.

Arrays may not be redimensioned after they have been declared. All array must be dimensioned before use.

Example
Dim Ax(5), Another(20), LastOne(1000)

4.40 Duration Function

Application
Result and data information from Size or Zeta records.

Syntax
Duration

Notes
The length (in seconds) of the experimental ‘run’ that gave rise to that data component of the current record.

4.41 Edit Function

Application
Modification of Size or Zeta record.

Syntax

```
Edit    Note1
Edit    Note2
Edit    Note3
Edit    Program
Edit    TimeArray i ,           v
Edit    StoreChannel i,         v
Edit    Parameter_keyword,      v
Edit    AChannel                v
Edit    BChannel                v
Edit    SampleTime              v
Edit    TotalSamples            v
```

Notes

Alter the value of a parameter of the current record in memory. For a permanent change the record should be Saved.

Edit Note1 “This is the first line”
Changes the first line of the title in the relevent record.

Edit Program

Edit TimeArray i, delay_time
Sets the value of the ith channel of the time array to delay_time.

Edit StoreChannel i, value
Sets the value of the ith channel of the store channel to value.

Edit Parameter_keyword, value
Changes the value rameter_keyword to the value of v

For a list of valid values of parameter_keyword see the Data command.

Example
Edit Angle 45‘ alter the angle parameter of the current record to 45 degrees.

Edit Storechannel,15,1007 ‘sets the value of the correlation channel 15 to 1007’

An example from a program using Edit is shown below.

```
edit sampletime samt
edit note1 “Generated Data”
edit MeasFarPoint 10000
edit temperature tem
edit viscosity vis
edit dielectric die
edit CellVoltage volt
edit electrodespacing spac
edit fringespacing fri
edit modulatorfreq modf*sign
edit measfarpoint baseline
edit calcfarpoint baseline
edit zerofieldwidth newwidth
```

4.42 ElapsedTime Function

Application
System command

Syntax
Elapsedtime

Notes
Returns the time elapsed since the start of the current session.

4.43 ElectroMobility

Application
Result information from Zeta record

Syntax
ElectroMobility

Notes
Returns the mobility calculated from a Capillary scan calculation (assuming a ZET5104 cell).

4.44 Else Command

Part of the If .. Then... Else ...End If construct. See If for more details.

4.45 Enable Command

Application
System configuration command, generally switching some option on or off.

It is intended for the advanced user only.

Syntax
Enable RefBeam on_off
Enable InstMenu1,0
Enable AccessLevel access_level
Enable Smoothing on_off, cutoff_value

Notes
Enable RefBeam on_off
Turns reference beam on or off, according to whether the value of on_off is 1 or 0 respectively. This is applicable to the Zeta5000 only.

Enable InstMenu1,0
Enables the sub menu access under the Setup. Some menu items here are engineering functions and not normally available to the normal user.

Enable AccessLevel level
Change the access level, where level takes values 1,2 or 3. 1 is the lowest level (most restricted) 3 is the highest level(totally unrestricted).

Enable Smoothing on_off, cutoff
Configures the Zeta potential analysis. If on_off is 0 or 1 then smoothing is turned off or on. The

value of cutoff defines the noise threshold as percentage of the maximum peak height. This command is intended for diagnostic purposes only.

4.46 Else Command

Part of If ... Then ... Else... End If construct. See If for more information.

4.47 End Command

Application
Program control.

Syntax
End
End If

Notes
Ends the program and returns to menu control. Note that this does not terminate the PCS program or Windows. The End is essential if the natural sequence of program flow does not finish at the last line of code. When code in subroutines is placed at the end of the file End prevents the program running on into the subroutine.

For the use of End If see If... Then ...Else...End If

Example

```
....
Measure Sequence
GoSub 1000
....
....
End
1000 Calculate
Print Status “Finished measurement”
Return
```

4.48 EndAngle

Application
Angle limits used in angle scan.

Syntax
EndAngle

Notes
An angle scan is a sequence of intensity measurements starting from a lower angle and ending at an upper angle. EndAngle is the upper limit used.

4.49 Error Function

Application
Fitting error in data analysis.

Syntax
Error

Notes
The square root of the average of the squared residuals, ie SUM(FitData - Fit)2 /FitNum from the calculation of size distribution for the current record.

4.50 Execute Command

Application
Staring other windows applications.

Syntax

Execute Maximised	program_name$, max_number_instances
Execute Minimised	program_name$, max_number_instances
Execute Normal	program_name$, max_number_instances
Execute Iconic	program_name$, max_number_instances

Notes
This command will run an instance of another Windows application. Use Minimised, Maximised, Normal or Iconic to indicate what state the application will run in. The program_name$ is the full path name and file name of the application.

max_number_instances is the maximum number of allowed instances of the program that may be run. If there are more currently running when this command is run then the command will have no effect.

Example
Execute Normal “c:\excel\excel.exe”, 1

4.51 Exit Command

Application
Close PCS application.

Syntax
Exit n

Notes
Exit to Windows from the PCS application. If save_config = 1 then the current PCS application configuration is saved. If save_config is zero or omitted, it is not.

4.52 EXP Function

Application
Mathematical function

Syntax
exp(x)

Notes
Returns the exponential of the argument.

4.53 File Command

Application
Writing to and from ASCII files.

Syntax
File filename$

Notes
This command names the file to be used with the Print, Delete, Input and Close File commands. filename$ is the full file name but without a pathname. The file is opened at the origin.

Example

```
NL$ = chr$(13)
F$ = “Please enter a filename”
Input F$, “Input”, X$
X$ = X$ + “.txt”
File X$
```

4.54 FillColour

Application
For use in page programs.

Syntax
FillColour n

Notes
This command changes the fill colour for the Box command. n can range from 0 to 33 and is an index into the PCS colour palette. Use the Setup Table dialog to see the list of colours. Note that for the standard VGA screen, colours numbered 16 and onwards are created by dithering and may give undesirable effects when the text is draw onto a filled box.

Examples

```
FillColour 10
Move 10,10
Box 100, 60
```

4.55 FirstPoint Function

Application
Results information for sizing records.

Syntax
FirstPoint

Application
Reports the first channel in the correlator used in the fitting procedure for size analysis.

4.56 Fit Function

Application
Results information for sizing records.

Syntax
Fit(iChannel)

Notes
The value of channel ichannel of the fit. The fit is that distribution which exactly corresponds to a given size distribution.

4.57 Fitdata Function

Application
Results information for sizing records.

Syntax
FitData(iChannel)

Notes
The value of channel iChannel of the normalised correlation function from the set actually fitted, the delay time being returned using FitTime.

4.58 FitNum

Application
Results information for sizing records.

Syntax
FitNum

Notes

The number of points included in the data analysis (fitting) procedure: ie. The maximum value of the index i in the three former functions.

4.59 FitOrder Function

Application
Results information for sizing records.

Syntax
Fitorder

Notes
The order of fit used in an electrophoresis record for the polynomial beam stripping process. This process removes any exponentially decaying component of the correlation function leaving the fringe transit data to be processed by the Fourier transform.

4.60 FitTime Function

Application
Results information for sizing records.

Syntax
FitTime(i)

Notes
The delay time of the ith point actually included in the fitting process.

4.61 FitResiduals Function

Application
Results information for sizing records.

Syntax
FitResiduals i

Notes
The residual associated with the ith point actually included in the fitting process.

4.62 Font Command

Application
For use in page programs.

Syntax

Font	iFont changes font i
Font	iFont, facename$, font_height, BOLD,
Font	iFont, facename$, font_height, ITALIC,
Font	iFont, facename$, font_height, UNDERLINE,
Font	iFont, facename$, font_height, BOLD,

Notes
iFont ranges from 1 to 4. If no arguments are set apart from iFont, then the font is set to that setup by the Setup Table Dialog. facename$ is a standard windows font facename.

4.63 For Command

Application
Program control

Syntax

```
For variable = expression1 To expression2 [Step expression3]
....
Next
```

Notes
The counter variable is set to the value of expression1. The program lines following the For statement are executed until the Next statement is encountered.

The amount specified by the Step statement expression3 is then added to the counter variable and is compared with the value of expression2. The loop terminates when the value of the varible is greater than or equal to expression2.

If expression1 and expression2 have the same value, the loop executes once.

If Step expression3 is ommitted the value of expression3 is assumed to be 1.0.

Loops can be nested, but ensure that the counter variables of the nested loops are unique. Counter variables should never be modified.

Example

```
count = 0
```

```
For i = 1 to 10
 For j = 1 to 20
 print status count
 count = count + 1
 Next
Next
```

4.64 Format$

Application
String formatting function.

Syntax
Format$(format_string$, variable1, variable2, ...)

Notes
variable1, variable2 and any other variables (represented by ...) in the list are converted according to format specifiers in format_string$.

For details of formating variables see the Print Dialog using command.

Example
A$ = Format$ ("Number of channels is ##", numchannels)

4.65 Frequency Function

Application
Results information for Zeta records.

Syntax
Frequency(f)

Notes
Returns the amplitude of the Fourier transform of the current electrophoresis record at frequency f.

4.66 FrequencyChannels Function

Application
Results information for Zeta record.

Syntax
FrequencyChannels i

Notes
Returns ith channel of spectrum in frequency space

4.67 Get Function

Application
Access system parameter values.

Syntax
Get Zerofieldwidth last measurement

Get	ModulatorFrequency
Get	RefIndex
Get	Viscosity
Get	Angle
Get	Temperature
Get	Wavelength
Get	Conductivity
Get	CalibrationConst
Get	FringeSpacing
Get	Dielectric
Get	CellType
Get	Density
Get	RFReal
Get	RFImag
Get	MarkhowinkA
Get	MarkhowinkK
Get	Cellposition
Get	Width
Get	Ph
Get	LaserPower
Get	CellCUrrent
Get	VoltageDiff
Get	CurrentDiff
Get	CoreRatio
Get	RFRealCore
Get	RFImagCore
Get	Fka
Get	Concentration
Get	RFStandard
Get	DnDc
Get	RayleighRatio

Notes

Get	Zerofieldwidth	returns the value of the ZeroFieldWidth	
Get	ModulatorFrequency	returns system modulator frequency	
Get	RefIndex	returns system dispersant refractive index	
Get	Viscosity	returns system viscosity	value
Get	Angle	returns system angle	value
Get	Temperature	returns system temperature	value
Get	Wavelength	returns system Wavelength	value
Get	Conductivity	returns system Conductivity	value
Get	CalibrationConst	returns system CalibrationConst	value
Get	FringeSpacing	returns system FringeSpacing	value
Get	Dielectric	returns system Dielectric	value
Get	CellType	returns system CellType	value
Get	CellSpacing	returns system CellSpacing	value
Get	CellSpacing	returns system CellSpacing	value
Get	RFReal	returns system RFReal	value
Get	RFImag	returns system RFImag	value
Get	MarkhowinkA	returns system MarkhowinkA	value
Get	MarkhowinkK	returns system MarkhowinkK	value
Get	Cellposition	returns system Cellposition	value
Get	Width	returns system Width	value
Get	Ph	returns system Ph	value
Get	LaserPower	returns system LaserPower	value

Get	CellCurrent	returns system CellCurrent	value
Get	VoltageDiff	returns system VoltageDiff	value
Get	CurrentDiff	returns system CurrentDiff	value
Get	CoreRatio	returns system CoreRatio	value
Get	RFRealCore	returns system RFRealCore	value
Get	RFImagCore	returns system RFImagCore	value
Get	Fka	returns system Fka	value
Get	Concentration	returns system Concentration	value
Get	RFStandardcase	returns system RFStandard	value
Get	dndc	returns system DnDc	value
Get	RayleighRatio	returns system RayleighRatio	value

The list is the same as under the Set command, which can be used to alter them.

4.68 Gosub command

Application
BASIC gosub statement.

Syntax
Gosub label

Notes
The program operation temporarily jumps to the subroutine at the line starting with the label label. The use of subroutines allows you to repeat program lines which are needed in more than one place.

Labels are numerical values.

At the next Return the program continues at the instruction after the GoSub

Example

```
Measure Sample
GoSub 1000
End
1000 Calculate
Print File using “#.#\r\n”, ZAvemean
Return
```

4.69 Goto

Application
Program Control

Syntax
GoTo label

Notes
The program operation jumps to the line starting with “label”. Do not use goto’s without good reason as their use is generally considered bad programming practice in all but exceptional circumstances.

Labels are numerical values.

4.70 Graph

Application
Configuration of graphs.

Syntax

```
Graph   Pref      graph_type
Graph   Limits    graph_type      x_min ,x_max
Graph   Scaling   graph_type      x_min, x_max
Graph   Xlimits   graph_type      x_min,x_max
Graph   Ylimits   graph_type      y_min,y_max
Graph   Styles    graph_type      graticules, fill_type
```

Notes
In the following graph_type is a keyword that can be one of the following:-

Size mode
Intensity
Number
Volume
Residuals
Correlogram
Fit
MolecularWeight
RelaxationTime
Diffusion
Zeta Mode
Mobility
Zeta
Frequency
RawFreq
ZetaData
FringeModel

Graph Pref graph_type.
Brings up graph preferences dialog for the graph specified by graph_type.

Graph Limits graph_typex_scale , y_scale Sets auto/manual limits for the graph. If x_scale 0 then the x axis scales automatically for the graph specified by graph_type. If y_scale 0 then the y axis scales automatically for the graph specified by graph_type.

Graph Scaling graph_typex_log, y_log if x_log 0 then the x axis uses logarithmic scaling for the graph specified by graph_type. if y_log 0 then the y axis uses logarithmic scaling for the graph specified by graph_type.

Graph Xlimits graph_typex_min,x_max
Sets the upper and lower limits for the x axis of the graph specified by graph_type.

Graph Ylimits graph_typey_min,y_max
Sets the upper and lower limits for the y axis of the graph specified by graph_type.

Graph Styles graph_typegraticules, style_type
Specifies the style of the graph specified by graph_type.

Valid values of style_type are 0 No colour fill

1 Colour fill
2 Histogram
3 Colour filled histogram

4.71 Keep command

Application
Record management for the live records buffer.

Syntax
Keep n

Notes
Keep n

Saves the current record as record n of the live buffer. The companion function is Select n. Be aware that selecting a record and then modifying it by say analysing it will not permanently save the changes in the buffer.

The Keep command is needed to commit the changes to the buffer.

4.72 If Command

Application
Program control

Syntax

```
If expression Then command_expression
If expression Then
...
Else
...
End If
```

Notes
The If construct takes two forms. If the single line form is used, the command_expression is executed if expression is true.

In the block form only comments are allowed after the Then.

If the expression is false execution continues after the corresponding Else or End If. The Else and block between Else and End If may be omitted.

If Then Else End If can be nested to any level.

```
If ZaveMean  5000 Then
 Print dialog “Size out of PCS size range”
else
 Print dialog “Result OK”
End if
```

4.73 Inp Function

Application
Computer port function.

Syntax
Inp address

Notes

The Inp function returns the value from an input port on a PC ISA bus.

4.74 Input Command

Application
Data input from external device.

Syntax
Input prompt_string$, input_value
Input Prompt_string$, input_string$
Input File var1 [, var2 ...]
Input Comm string$

Notes
Input prompt_string$, input_value

Prompts the user to input the numeric value input_value from a dialog. The string prompt_string$ is the message displayed to the user. Only a single variable can be input at one time.

Input prompt_string$, input_string$
Prompts the user to input the string input_string$ from a dialog.
Input Filevar1 [,var2]

Reads var1, var2 ... from the file opened by the File command. Note that the variables can be numeric or string type. Strings are comma delimited.

Input Comm string$

Reads string$ from RS232 communications. The end of line delimiter is by default a carriage return but can be reset to any ASCII character by using Comm Delimater.

4.75 InRange Function

Application
Results information in Size record.

Syntax
InRange

Notes
The InRange value calculated from the ratio of the far points. A high value (85 -100 %) indicates that the correlation function has nearly decayed to 0 by the measured far point, and hence the sample time is set to a suitable value, and the experiment a well founded one.

4.76 InStr function.

Application
String Function

Syntax
InStr(start, String$,SearchString$)

Notes
This function returns the position of the string search$ in string$ starting at position start.

If start is omitted then the search starts at the first character.

If search$ does not appear in string$ the function returns 0.

Example
a$ = “This is a string”

p = InStr(1,a$, “is”)

4.77 Int Function

Application
Mathematical function

Syntax
Int(x)

Notes

Returns the whole number part of the number n. This is useful for the counter in For...Next loops.

4.78 Intensity Function

Application
Results information in Size record.

Syntax
Intensity(i)

Notes
Intensity distribution amplitude of ith Size class.

4.79 Iscan Mode command

Application
Changing mode of instrument

Syntax
iscan mode

Notes
Set system to intensity mode.

4.80 InstrumentType Function

Application
Instrument function.

Syntax
InstrumentType

Notes
The numbers currently defined are as follows:-

AutoSizer 2C	1
Lo-C	2
Hi-C	3
ZetaSizer 4	4
ZetaSizer 4S	5
4700	6
4700 Manual	7
Zetamaster	11
Zetamaster S	12
ZetaSizer 1000	13
ZetaSizer 2000	14
ZetaSizer 3000	15
ZetaSizer 4000	16
ZetaSizer 5000	17
4700 with PCS78	18

4.81 LastPoint Function

Application
Results information from Size record.

Syntax
LastPoint

Notes
Returns the last data channel used in a Size analysis.

4.82 LCase$ Function

Application
String function.

Syntax
LCase(string$)

Notes
This function returns a string with all upper case characters in string$ translated to lower case.

Lower case characters are not affected.

4.83 Left$ Function

Application
String function.

Syntax
Left$(string$,n)

Notes
This function returns a string made up of the first ncharacters of string$.

If n is greater than the number of characters in string$ then the whole string is returned.

Example
After the call A$ contains “Ab”

A$ = Left$(“AbcDe”,2)

4.84 Len Function

Application
String Function

Syntax
Len(String$)

Notes
This function returns the number of characters in the string string$

Examples
A = Len(A$) + 2

4.85 Line Command

Application
Page graphics.

For use in pages only.

Syntax
Line x1,y1, x2,y2

Notes

Draws line from (x1,y1) to (x2,y2) and sets current position to x2,y2.

The current line colour and style is used. Positions are defined in mm.

4.86 LineColour Command

Application
Page graphics.

For use inpages and reports only.

Syntax
LineColour n

Notes
This command changes the line colour of the lines draw on tables. n is the line colour as an index into the PCS colour palette. Use the Setup Table dialog to see the list of colours. Only values of n between 1 and 15 can be used.

Example

```
Line 10,10 ,100,10
LineColour 2
Line 100,10, 100,100
LineColour 3
Line 100,100, 10,100
LineColour 4
Line 10,100, 10,10
```

4.87 LineTo Command

Application
Page graphics.

For use in pages only.

Syntax
LineTo x,y

Notes
Draw line from current position to x,y, and sets current position at x,y.

4.88 Ln Function

Application
Mathematical function.

Syntax
Ln(x)

Notes
Returns the natural logarithm of the argument, which should be greater than zero.

4.89 Load Command

Application
File system command

Syntax
Load Record i
Load Contin i
Load Configuration FileName$

Notes
Load Record i

Loads record i from the current file into memory. If changes are made to this record that you wish to save, remember to use the Save Record command.

Load Contin i
Loads record i from the set of records produced by the last Contin analysis into memory.

Load Configuration FileName$
Loads a PCS configuration file. FileName$ is the filename.

4.90 Log Function

Application
Mathematical function.

Syntax
Log(x)

Notes
Returns the Log to the base 10 of the argument, which should be always greater than zero.

4.91 LTrim$ Function

Application
String command

Syntax
LTrim$(String$)

Notes
Returns a string with all leading space, tab and other non-printing characters of String$ removed.

4.92 MAddSelect Command

Application
File system command

Syntax
MAddSelect i

Notes
This adds to the current set of records selected. Where the records will be selected is dependent on the file context. This selection will be used in any subsequent MCalculate command for example. Whether the records are selected in the current file or the live buffer can be set by the Use command.

4.93 Mcalculate Command

Application
File system/analysis command

Syntax
MCalculate

Notes
Recalculate the answer on the currently selected records using the current analysis, replacing the changed records in their original places.

4.94 MDelete Command

Application
File system command.

Syntax
MDelete

Notes
Deletes all records in file.

See clear.

4.95 Measure Command

Application
Configure measurement sequence.

Syntax

Measure Delay	hours,minutes,seconds
Measure Sequence	Number_of_measures, Duration, sample_size
Measure ZimmPlot	Number_of_measures
Measure AngleScan	Number_of_measures, Lower_Angle, Upper_Angle
Measure Document	Document dialog
Measure Duration	Measurement duration
Measure ZLimits	LowerZeta ,UpperZeta
Measure Temperature	LowerTemp, UpperTemp
Measure Angle	LowerAngle, UpperAngle
Measure Multiple	Number_measures
Measure Capscan	n
Measure Inspect	b
Measure Setup	

Notes
Measure Delay hours,minutes,seconds

Set delay time between measurements (defined in hours, minutes and seconds).

Measure Sequence [,Number_of_measures, Duration, sample_size]
Starts measurement sequence. Note that for a Zeta measurement sequence the sample_size argument is not necessary. If any parameters are missing then automatic defaults are supplied.

Measure ZimmPlot
Starts Zimm scan sequence. (Intensity mode).

Measure AngleScanNumber_of_measures, Lower_Angle, Upper_Angle
Starts Angle scan sequence (intensity mode) with Number_of_measures measurements.

Lower_Angle and Upper_Angle are optional and set the angular range over which the measurement is carried out.

Measure DocumentDocument dialog
Initiates the Measure Document dialog

Measure DurationMeasurement duration
Sets experiment duration. Execution of this command will set the measurement duration of the measurement relevent to the current mode. For example in Intensity mode the durations of the TimeTrace/AngleScan measurements will be set.

Measure ZLimitsZLower , ZUpper
Set Zeta potential range. This defines the modulator frequency in a measurement sequence. The modulator frequency is set as follows:

ZLower + ZUpper	0	-	250 Hz
ZLower + ZUpper	0	+	250 Hz
ZLower + ZUpper =	0	+	1000Hz

MeasureTemperatureLowerTemp, UpperTemp
Set lower and upper limits for a temperature scan in size or zeta mode.

Measure AngleLowerAngle, UpperAngle
Set lower and upper limits for an angle scan in size mode.

Measure Multiple[,Number_measures]
Do ACCF measurement with Number_measures sub-measures. If Number_measures is missing then 10 is used by default.

Measure Capscan
Performs capillary scan of n measurements.

MeasureInspect b
Set Inspect Live Data on/off depending on whether b is 1 or 0.

Measure Setup
Initiates the Setup Measurement dialog

4.96 Merit Function

Application
Results information about Size record.

Syntax
Merit

Notes
The percentage of (correlation_intercept - baseline)/baseline. Typically 10 - 60 %.

4.97 Message Function

Application
Notification messages to user.

Syntax
Message caption$, message$, style

Notes
Caption$
Caption displayed by message box.

Message$
Message string displayed by message box.

Style
The following values can be added together.

0	OK button, No Icon
1	Ok & Cancel
2	Abort, Retry & Ignore
3	Yes, No & Cancel
4	Yes & No
5	Retry & Cancel
16	Stop Icon
32	? Icon
48	! Icon
64	* Icon
256	Button 2 is default
512	Button 3 is default

Return value
1 = left, 2 = middle, 3 = right button

4.98 Mid$ Function

Application
String Command

Syntax
Mid$(string$,start,n)

Notes
This function returns a string with n characters of string$ starting at character start.

An error is issued if start is beyond the end of string$. If n is greater that the number of characters remaining in the string then the remaining characters are returned.

Example
After the call A$ contains “cdefg”
a$ = Mid$(“abcdefghijk”,3,5)

4.99 Mie Function

Application
Mathematical function

Syntax
Mie diameter, angle, RefIndex, RealRI, ImageRI, polarization [, core_ratio, CoreRealRI, CoreRealRI]

Notes
Returns scattering Intensity calculated by Lorentz Mie theory for a layered spherical particle. For a homogeneous particle CoreRatio=0.

The wavelength of the illuminating light is set in the current record, and could be altered by set wavelength.

The parameters are as follows:-

diameter
particle diameter in nm.

Angle
Scattering angle in degrees.

RefIndex
Refractive index of dispersant

RealRI
Real refractive index of sample

ImageRI
Imaginary refractive index of sample

Polarization
Polarization of incident light.

0	unpolarised
1	horizontal
2	vertical

4.100 Mload Command

Application
File and record management

Syntax
MLoad

Notes
Loads selected records from disk to the live buffer. Use Select and MSelect to select records.

4.101 Mobility Function

Application
Result information from Zeta record

Syntax
Mobility mobility_value

Notes
Returns amplitude corresponding to mobility mobility_value.

4.102 MobilityChannels Function

Application
Result information from Zeta record

Syntax
MobilityChannels i

Notes
Returns ith channel of spectrum in mobility space. The channels are defined by the Fourier transform underlying the analysis.

4.103 MolecularWeight

Application
Result information from intensity record.

Syntax
MolecularWeight

Notes
Returns the MolecularWeight calculated from a Zimm Plot or Debye plot. This is only valid if the record has been obtained from a Zimm scan or Debye scan.

The result is in AMU.

4.104 Move (x,y)

Application
Page graphics and text.

Syntax
Move x,y

Notes
Moves the current anchor position for lines, boxes or text to position x,y on the table page or report. No drawing is performed x and y are in mm and if either is omitted the current value is used. x,y are measured from the top left hand margin of the page. y increases positively down the page.

Example

```
for k =1 to 10
FillColour k
LineColour 10 -k
X1 = k*10
Y1 = k*10
Move X1,Y1
Box X1+10, Y1+10
next
```

4.105 Mprint Command

Application
Record printing

Syntax
MPrint

Notes
Prints current selection of records. These have been selected using Select and MSelect. They are selected from the live records buffer or current file according to whether Use Buffer or Use Disk have been used.

4.106 Msave

Application
File and record management

Syntax
MSave

Notes
Saves selected records from Buffer to Disk. Records are selected using Select and MSelect.

4.107 MSelect

Application
File and record management

Syntax
MSelect

Notes
Selects a record in the current file or live record buffer according to the last Use File or Use Buffer command. The selection can be added to using the MAddSelect function. Once selected the records can then be analysed using MCalculate, or printed using MPrint, saved or loaded using MSave/MLoad. If no parameters are supplied then the relevant records list dialogue box is opened.

Mtransfer
Transfers data from the correlator and performs analysis and appends it to the live buffer. The appended record is made the selected record in the buffer. It has the same effect as using the Measure Transfer menu item.

Next
For full details please consult the For command. This is part of the For...To...Next structure.

4.108 Number function

Application
Results information from Size record

Syntax
Number size_band

Notes
Returns the number disribution amplitude of size class size_band. The number of classes is equal to NumSizes (see later).

4.109 NumChannels function

Application
Results information from Size record.

Syntax
NumChannels

Notes
Returns the number of correlator channels used in current record.

4.110 NumFreq Function

Application
Results information from Size record.

Syntax
NumFreq

Notes
Number of frequency channels in Fourier transform result.

4.111 NumScans Function

Application
Data information from Intensity record.

Syntax
NumScans

Notes
Returns number of scans in current intensity record.

4.112 NumSizes

Application
Results information from Size record.

Syntax
NumSizes

Notes
Number of size classes in size result.

4.113 Open Command

Application
Open File or comms port.

Syntax
Open File file_name$
Open Comm port_id, baud_rate, Parity, data_bits, stop_bits, handshaking

Notes
Open File file_name$

A file is opened in the current directory. If the file extension is omitted one of the same type as the current mode (ie .sz2, .zet etc.) will be included. Opens new data file for the current mode. If an extension is not given then one will be supplied.

Open Comm port_id, baud_rate, Parity, data_bits, stop_bits, handshaking
Opens and configure a comms port for serial communiciation.

Port_id
Id of port to be used. This is limited by the number of physical ports in the computer.

baud_rate
Baud rate at which communication will operate. Valid values are 2400,4800,9600 and 19200.

Parity
One of the following keywords must be used:- EVEN, ODD, NONE.

data_bits
Number of data bits. Valid values are 6,7 or 8.

stop_bits
Number of stop bits. Valid values are 0 ,1 or 2.

handshaking
Flow control. Valid values are 0 (none), 1 (XOn/XOff) or 2 (Hardware).

If any parameter is omitted, then a default will be supplied from the Setup configuration

Secondary command
Comm Comm Port , Baudrate, databits. StopBits, Parity,XonXoff

Parity may be EvenParity,OddParity or NoParity. If any parameters are missed out, defaults will be supplied set up under the Setup Configuration.

4.114 OsmoticMob Function

Application
Result information from Zeta record.

Syntax
OsmoticMob

Notes
The equivalent value of mobility calculated for the cell wall by the last capillary scan calculation applied to the current record.

4.115 Out Command

Application
Computer port function

Syntax
Out address, value

Notes
Outputs a 16 bit digital value to the port address on the computer bus.

4.116 Page Command

Application
Page program design.

Syntax
page page_name$, x_extent, y_extent

Notes
Specifies page size (in mm). Program name is used as a label which is longer and therefore potentially more descriptive than the filename containing the page program. This command should be the first line of a page program. Pressing Info on the page setup dialogue displays the program name field for the selected program file.

4.117 Pause

Application
Program control

Syntax
Pause time

Notes
The operation of the program is suspended for time in seconds. Other Windows programs can run during the pause unless the program is in Exclusive mode.

Examples
Print status "About to open valve"
Pause 5

4.118 Peak Command

Application
Results information for Zeta and Size records.

Syntax

Peak	Number	Area i	number distribution
Peak	Volume	Means i	volume distribution
Peak	Intensity Width i		intensity distribution
Peak	Zeta Number		zeta distribution
Peak	Mobility Area		mobility distribution
Peak	Frequency Width		frequency distribution

Notes
The peak command returns information about the modes in distributions.

The following distributions are valid:-

Intensity
Number
Volume

Zeta
Mobility
Frequency

The following keywords are valid:-

Area
Means
Width
Number
Peak
Area i

Return the relative area of the ith mode of the distribution as a percentage of the complete distribution. The value of i must be greater than zero and less than or equal to the total number of peaks in the distribution.

If i is absent then 100 is returned.

Peak Means i
Return the mean of the ith mode of the distribution. The value of i must be greater than zero and less than or equal to the total number of peaks in the distribution. If i is absent then the mean of the complete distribution is returned.

Peak Width i
Return the width of the ith mode of the distribution. The value of i i must be greater than zero and less than or equal to the total number of peaks in the distribution. If i is absent then the standard deviation of the complete distribution is returned.

Peak Number
Returns the total number of modes found in the distribution.

Examples
To find the number of peaks in the number distribution of a size result we could use the following

```
n = Peaks number number
if n then goto 1000
write “Peak Mean (nm) Width Area”
for j=1 to n
m = Peaks number means j
w = Peaks number width j
a = Peaks number area i
write “## ####.# ####.# ####.#”, j,m,w,a
next
1000
end
```

4.119 Picture Command

Application
Drawing bitmaps, metafiles or graphs in page programs.

Syntax
Picture
Picture   intensity   x,y

Picture	number	x,y
Picture	volume	x,y
Picture	residuals	x,y
Picture	zetadata	x,y
Picture	sizedata	x,y
Picture	correlogram	x,y
Picture	fit	x,y
Picture	molecularweight	x,y
Picture	relaxationtime	x,y
Picture	diffusion	x,y
Picture	mobility	x,y
Picture	zeta	x,y
Picture	frequency	x,y
Picture	rawfreq	x,y
Picture	fringemodel	x,y
Picture	inverseQScan	x,y
Picture	Qscan	x,y
Picture	anglescan	x,y
Picture	zimmplot	x,y
Picture	timetrace	x,y
Picture	bitmap	x,y, filename$
Picture	metafile	x,y, filename$

Notes
All picture types except the bitmap and metafile type correspond to graph types produced by the system. For the bitmap and metafile options, the filename$ is the file containing the associated bitmap or metafile.

The availability of the particular graph will depend on the operating mode.

Intensity, Volume, Number, Correlogram, Fit, MolecularWeight, RelaxationTime and Diffusion are only available in Size mode.

Zeta, Frequency, Rawfreq, Fringemodel and Correlogram are available in Zeta mode.

InverseQScan, Qscan, Anglescan, Zimmplot and Timetrace are available in Intensity mode.

4.120 Polydispersity Function

Application
Results information from Size record.

Syntax
Polydispersity

Notes
The Polydispersity calculated using the initial cumulants fit to the current size result.

4.121 Print Command

Application
Command to output to various destinations

Print	Status,	expression_list
Print	File,	expression_list

Print	String,	expression_list
Print	Dialog	expression_list
Print	Comm	expression_list
Print	Remote	expression_list
Print	Printer	expression_list
Print	a$ using	expression_list

Notes
These commands print the result of a list of expressions to different destinations. expression_list is a list of expressions containing variables, constants, string variables or quoted strings. Each expression is separated by a comma or by a semi-colon.

Print Dialog , expression_list
The printing appears in a message box with OK and Cancel buttons.

Print File
Printing will output to the file last named by the File command. Use delete file to prepare a new file for saving information to . Each subsequent use of print file append information to the file.

Print Comm
Printing will output to the comm port opened by the last call to Open Comm.

Print Status
The printing is sent to the status line at the bottom of the graph window.

Print string
The printing is sent to a global buffer which can be used for DDE operations. Use Delete string to empty the string before use. Each call appends information to Copy string transfers the contents to the clipboard.

the string.
The string produced by evaluating the list of expressions must not exceed 255 characters. A line feed and carriage return are appened to the end of the printed line unless the list i

Print Printer
Prints directly to the LPT port. This is only useful if a printed log is required.

4.122 Print Page Command

Application
Printing to paper

Syntax
Print Page type1 [,type2], graphsize

Notes
This command prints items to the printer. type1, type2 are options from the list Report, Graph, Table, Key, Title.

Any of these items may be used in any combination. If a graph is required the size of the graph may be specified by the graph_size, which can be:-

FullFull page graph
HalfHalf page graph

FitFit the graph to the page (when another option has been selected).

If no parameters are supplied use print dialog

4.123 Print Report Command

Application
Prints report to paper

Syntax
Print Report [,PageFile1$] [, PageFile2$]

Notes
If no parameters are supplied, then the currently selected options are used. Otherwise a report is printed using the page files Page1$, and Page2$ as page files.This function returns the angle, in radians, of which the cosine equals the number. The value returned is in the range 0.0 to 2Pi

The value of number must be in the range -1.0 to 1.0.

4.124 Print ...Using command

Application
Command for formatted output to various destinations.

Syntax
Print dialog using, format$, expression_list
Print dialog using format$, expression_list
Print status using format$, expression_list
Print string using format$, expression_list

Notes
This is a formatted version of the Print command, See Print for information on the destinations for the print.

expression_list is a list of expressions containing variables, constants, string variables or quoted strings. Each expression is separated by a comma. The format$ contains information on how to format each item in the expression list and must not exceed 255 characters in length. For each item in the expression_list there must be a corresponding format item in format$. Details on what may appear in format are given below.

Formatting numbers
To format numbers use the # character to represent each digit and the period to represent the decimal point. The number of digits before and after the decimal point can be varied by adding # characters. An example for format$ is "##.###". Negative numbers are prefixed with a minus sign. The number may also be represented in scientific format using a format such as "#.###^^^^",The ^ characters represent the position of the exponent termEsxx where E denotes the exponent , s is the sign (+ or -) and xx is the power of 10 to be used.

Note: The scientific format must have 4^ characters and must have only one # charater before the decimal point. For example, we show the result of displaying the number 123.4567 with various formats.

FormatResult

###.###	123.456
###.##	123.46
####	1234
#.####^^^^	1.2345ED2

String Characters
To format a string variable, string functions or quoted strings use the $ character to represent each character of the string. The character string is left justified in the format string if the format string is longer than required., For long strings you do not need to use a large number of $ characters. If the format is “$n”, where n is a number between 1 and 256, then it acts as if n $ characters were used. The format"$*" acts as if the same number of $ were used in the format as appear in the string.

For example, below is the result of displaying the string “PCS Software” with various formats.

Format	Result
$$$$$$$$$$$$$	PCS Software#
$$$$$$	PCS Sof
$8	PCS Softw
$*	PCS Software

Special characters
The character \ is taken to be a special character in format strings. In combination with another character they form escape sequences to produce non-printable characters.

\r the carriage return character - Chr$(13)
\n the new line character Chr$)10)
\" the double quotation mark Chr$(4)
\\ the back slash Chr$(92)
\t the tab character

These are meant for special formatting of items. The \r or \n sequences may be used for creating a new line in a print dialog message box. If you prefer you may produce the same effect using the Chr$() function.

Other characters
All other characters apart from those printed above printed without alteration.

4.125 Range

Application
Range for size analysis.

Syntax
Range

Notes
If the analysis limits have been set using upper and lower sizes the Range will return 0.

4.126 RadiusofGyration

Application
Data information for intensity record.

Syntax
RadiusOfGyration

Notes
Results of Zimm Calculation (Rg from all points fitted simultaneously, so not ,in general, equal to average Rg values for different records)

4.127 RawFreq Function

Application
Results information for Zeta record

Syntax
RawFreq (index)

Notes
Intensity of Fourier transform (for electrophoresis data) at given channel, the number of channels in this case being equal to that of the correlator.

4.128 RecordNumber Function

Application
File and record management

Syntax
RecordNumber

Notes
The record number of the current record from the Live buffer, or the current file for disk based data.

4.129 Reference Function

Application
Instrument control

Syntax
Reference

Notes
References the stepper motor on the 4700 spectrometer.

4.130 Residuals

Application
Result information from size record

Syntax
Residuals (i)

Notes
The difference between fit and data in the ith channel of the current record.

4.131 Return

Application
Returns from GoSub.

4.132 Rg

Application
Result information for intensity record

Syntax
Rg

Notes
Returns the radius of gyration from a Guinier plot. This is different from the Radius of gyration as calculated from a Zimm plot.

4.133 Right$ Function

Application
String function.

Syntax
Right$(string$,n)

Notes
This function returns a string made up of the last n characters of string$. If n is greater than the number of characters in string$ then the whole string is returned.

4.134 RTrim$ Function

Application
String command

Syntax
RTrim$(String$)

Notes
Returns a string with all trailing space, tab and other non-printing characters of String$ removed.

4.135 Run

Application
Program control

Syntax
Run filename$

Notes
The program stored in the specified file is loaded and run. Control is passed back to the calling program when the subsidiary program terminates. Further nesting and recursion (calling of a program by itself) is permitted - be careful ! - a method of terminating such a sequence must be provided.

4.136 RunMode

Application
Program control

Syntax
RunMode warning, exclusive

Notes
This command controls the current running mode of the program. If the expression warning is greater than zero then any command that fails will cause a message box to appear on the screen. Normally between each command in the BASIC program any pending Windows messages are processed. If the expression exclusive is greater than zero then no messages are processed and other programs will not run concurrently with the PCS program. In this mode the program cannot be terminated by selecting Stop from the main menu but will respond to the cancel button in Print Dialog.

Example
Runmode 1,0

4.137 SampleTime

Application
Data information for Size and Zeta record

Syntax
SampleTime

Notes
Correlator sample time in microseconds, from current record.

4.138 Save Command

Application
File and record processing

Syntax
Save Record i

Save Config

Notes
Save Record i
Save current record as record i of the currently open file. If the parameter i is missing then the record is appended to the file.

Save Config fileName$
Saves current configuration in the file filename$. If no file is specified then the configuration is saved in the current configuration file.

4.139 ScaledCount

Application
Data information for Size record. Applicable to the 7032 correlator only.

Syntax
ScaledCount i

Notes
Returns the ith scaled count monitor channel.

4.140 ScanAngle

Application
Data information from an Intensity record.

Syntax
ScanAngle i

Notes

Returns the angle at which the ith measurement of an angle scan was performed.

4.141 ScanDuration

Application
Data information from an Intensity record.

Syntax
ScanDuration

Notes
Returns duration of each measurement carried out at each angle during an angle scan.

4.142 ScanQ Function

Application
Data information from an Intensity record.

Syntax
ScanQ i

Notes
ScanQ i returns the square of the wavevector associated with the ith measurement of an angle scan. The wavevector is related to the angle by the following:-

Wavevector := 4*Pi*RefIndex*sin(0.5*Angle)/WaveLength where RefIndex is the refractive index of the dispersant and Wavelength is the wavelength of the incident light.

4.143 ScanType

Application
Data information from an Intensity record.

Syntax
ScanType

Notes
ScanType returns the type of measurement that produced the record. The following

values are valid:-

0	Time trace scan
1	Angle scan
2	Zimm scan
3	Standard scan
4	Background scan
5	Zeroconcentration scan
6	Zero concentration standard scan. (Zero concentration measurement copied from standard scan)
8	Debye scan

4.144 Select Command

Application
Record management

Syntax
Select irecord

Notes
Selects record n in the live buffer as the current record and loads it into memory. Note that any modifications to the record must be saved by using the Keep command.

4.145 Set Command

Application
Set data variables in current record.

Syntax
Setnote1string$

Set note2 string$
Set note3 string$
Set Timearray iChannel ,value
Set StoreChannel iChannel, value
Set TotalSamples value
Set SampleTime value
Set AChannel value
Set BChannel value
Set MeasFarPoint value
Set CalcFarPoint value
Set Zerofieldwidth v
Set ModulatorFrequency v
Set RefIndex v
Set Viscosity v
Set Angle v
Set Temperature v
Set Wavelength v
Set Conductivity v
Set CalibrationConst v
Set FringeSpacing v
Set Dielectric v
Set CellType v
Set CellSpacing v

Set CellSpacing	v
Set RFReal	v
Set RFImag	v
Set MarkhowinkA	v
Set MarkhowinkK	v
Set Cellposition	v
Set Ph	v
Set LaserPower	v
Set CellCurrent	v
Set CoreRatio	v
Set RFRealCore	v
Set RFImagCore	v
Set Fka	v
Set Concentration	v
Set RFStandard	v
Set DnDc	v
Set RayleighRatio	v

Notes
See Get command for details on the various commands.

4.146 Setup Command

Application
Setup system configuration.

Syntax

Setup	document	sample details dialog
Setup	measure	measure details dialog
Setup	timetrace	timetrace dialog
Setup	buttons	buttons dialog
Setup	program	assign program dialog
Setup	printer	printer setup dialog
Setup	key	setup key dialog
Setup	page	setup page options

4.147 Setuptable Command

Application
Configure user defined tables

Syntax

Setup Table	
Setup Table Intensity	page_file$
Setup Table Number	page_file$
Setup Table Volume	page_file$
Setup Table Data	page_file$
Setup Table Fit	page_file$
Setup Table Residuals	page_file$
Setup Table Correlogram	page_file$
Setup Table Diffusion	page_file$
Setup Table Relaxationtime	page_file$
Setup Table Molecularweight	page_file$

Notes
Setup Table
Displays the Setup Table dialog.

Setup Table name
Set table for table type Table_name to a user defined page program , page_file$.

4.148 Setup Report

Application
Configure printed report setup.

Syntax
Setup Report [,page1$,page2$]

Notes
Set report pages to page1$ and page2$. If page2$ is missing then only one page is used If page1$ and page2$ are missing then the Setup Report dialog is shown.

4.149 SetupUserlogplot Command

Syntax
Setup userlogplot x_axis_parameter , y_axis_parameter

Notes
The following values are valid for x_axis_parameter and y_axis_parameter.

Rec	record number
Data	data type
DateDate	date of measurement
DateTime	time of measurement
ResultDate	date of analysis
ResultTime	time of analysis
Duration	duration of measurement
countrate	countrate of measurement
ZAve	Z average mean of measurement
Poly	Polydispersity
Zeta	Mean Zeta potential
Width	Width of Zeta distribution
CellPosition	Cell position of measurement
Error	Error of fit
Merit	Merit figure for size measurement
Inrange	Inrange figure for size measurement
Temperature	Sensed temperature of measurement
Viscosity	Viscosity
RFIndex	Refractive index of dispersant
Wavelength	Wavelength of laser
Angle	Angle of measurement
RFReal	Real refractive index of sample
RFImag	Imaginary refractive index of sample
Spacing	Analytical dilation
Range	Range
Mobility	Average mobility
Rg	Radius of gyration
AngleStart	Starting angle

AngleEnd	End angle of measurement
ElapsedTime	Time at which measurment was started
Ph	pH
Conductivity	Conductivity
NumPoints	Number of points used in analysis
Title	Title

4.150 Setup Userlogtable

Application
Configure user log table.

Syntax
Setup Userlogtable x1, x2, ...

Notes
This command configures the user log table so that the table displays the options x1, x2 ... Valid option keywords are as for the Setup Userlogplot command. If an empty list is supplied then the Setup UserLogTable dialog is shown.

4.151 Simulate Command

Application
Put software into simulation mode.

Syntax

Simulate	Start	simulate_zeta
Simulate	Stop	
Simulate	Zeta	peak_position
Simulate	Size	peak_position
Simulate	Countrate	count_rate
Simulate	Signaltonoise	signal_to_noise
Simulate	Noiselevel	noise_level

Notes
Simulate start simulate_zeta.
Puts the software into simulation mode. If simulate_zeta is greater than zero then the input functions are simulated too - that is artifical values of voltage, temperature etc are supplied.

Simulate Stop
Returns from simulate mode to the standard mode of operation.

Simulate Size peak_position
When in size mode simulate a signal arising from a monomodal distribution at peak_position nm.

Simulate Zeta peak_position
When in zeta mode simulate a signal arising from a monomodal zeta potential distribution centred at peak_position mV.

Simulate Countrate count_rate
Sets count rate of simulated signal.
Simulate SignalToNoise signal_to_noise
Sets signal to noise ratio of simulated signal
Simulate NoiseLevel noise
Set noise level of simulated signal.

4.152 Size Mode Command

Application
System command

Syntax
Size Mode

Notes
Put system into size mode.

4.153 Sizes Function

Application
Result information from a Size record.

Syntax
Sizes i

Notes
Size Class(i) in nanometres. i ranges from 1 to NumSizes.

4.154 Spacing

Application
Result information from a Size record.

Syntax
Spacing

Notes
The nominal multiplicative factor (analysis dilation) specifying the minimum separation between successive correlator channels chosen to form the set FitTime. Since the hardware of the correlator can only generate channels at certain integer delay times the actual channel chosen will be the nearest integer greater than the previous channel multiplied by this spacing.

4.155 StartAngle

Application
Result information from an intensity record.

Syntax
StartAngle

Notes
StartAngle returns the starting angle of the current record if the data is an angle or zimm scan. Otherwise it is the angle at which the timetrace or Debye scan was carried out. See EndAngle.

4.156 Step

This is part of the control structure For... = ...To...Step...Next

For full details see the For command

4.157 StoreChannel

Application
Result information from an Size or Zeta record.

Syntax
StoreChannel i

Notes
Contents of the ith store channel of the correlator data stored in the current record

4.158 Str$ Function

Application
String command

Syntax
Str$(value)

Notes
Returns a string representaion for the number value (or result of numeric expression)

4.159 StrCmp Command

Application
String Command

Syntax
StrCmp(String1$,String2$)

Notes
Compares the strings String1 and String2 lexicographically and returns a value indicating their relationship. The comparison made is based on the current language selected by the user at setup or the Control Panel. The comparison is case-sensitive. The value return is

1 = String1 String2
0 = String1 = String2
-1= String1 String2

4.160 TextAlign

Application
Table and graphics output.

For use in pages and reports only.

Notes
This command controls text alignment to the current page position. If n= 0 the text is left aligned. If n =1 then the text is centre aligned and if n=2 the text is right aligned. In all cases the current anchor position is at the top of the text.

Examples
TextAlign 1

WriteAt Xpage/2, 5, “Label centred on the page”

4.161 TextColour

Application
Table and report text output
For use in pages and reports only.

Syntax
TextColour text [, background]

Notes
This command changes the colour of subsequent text output by the write command. Text and background are indices in the Zetasizer colour palette.

Use Setup Table dialog to inspect the list of colours. Only values of text and background between 0 and 15 are valid. Text is the value of the text colour and background is the value for the background character cell. If background is omitted the n printing mode is set to transparent and any colours or patterns behind the text will show through.

Examples
TextColour 0 ‘Black text
TextColour 15,0 ‘white on black
Text colour i,(j) = textcolour, text background. The colour numbers should be in the range 0 to 32.

4.162 Then

Part of the If… Then… Else… End… If construct .
See If for more details.

4.163 TimeArray

Application
Data information from a Size or Zeta record.

Syntax
Timearray ichannel

Notes
Delay time associated with correlator channel number i of the current record.

4.164 TotalSamples

Application
Data information from a Size or Zeta record.

Syntax
TotalSamples

Notes
The contents of the total samples monitor channel. In effect the length of experiment used to collect the current contents of the correlator, measured in sample times.

4.165 To

This is part of the For...To...Next construct
See the For command for details

4.166 Transfer Command

Application
File and record management

Syntax
Transfer i

Notes
Transfers correlator data and associated parameters to ith position in buffer. If i is omitted the data block is appended to the existing contents of the buffer. This record remains selected as the default. No analysis is performed on that record.

4.167 UCase$

Application
String Command

Syntax
UCase(String$)

Notes
Returns a string with all lower case characters in String$ translated to upper case.

4.168 Use Command

Application
File and record management

Syntax
Use Buffer
Use Size

Notes
Specifies whether source of records is from the buffer or user specified data file for commands such as MCalculate, etc. All measurements performed from the menu are stored in the buffer, and may also be stored in a specific disk file, if such action is enabled. Before loading one or more records the source may need to be specified.

Examples
Use Buffer
Use Disk

4.169 Val Function

Application
Numerical conversion function

Syntax

Val(String$)

Notes

This function returns the number represented by string$. If the string does not represent a number, 0, is returned.

4.170 View

Application

Displaying graphs and tables.

Syntax

Size mode

```
View        Details
View        log
View        current
View        intensity
View        volume
View        number
View        sizedata
View        fit
View        residuals
View        correlogram
View        relaxationtime
View        molecularweight
View        diffusion
View        userlogtable
View        userlogplot
```

Zeta mode

```
View        details
View        log
View        current
View        fringemodel
View        mobility
View        frequency
View        zeta
View        correlogram
View        zetadata
View        rawfreq
View        userlogtable
View        userlogplot
```

Intensity mode

```
View        anglescan
View        qscan
View        inverseqscan
View        zimmplot
View        timetrace
```

Notes
Views selected records by adding them to the plot area in the selected type and style.

4.171 ViewNew Command

As for the View command described above but clears the existing display first

4.172 Virial2 Function

Application
Virial coefficients from Zimm plot calculation

Syntax
Virial2

Notes
Returns second Virial coefficent from a Zimm plot calcualation.

4.173 Virial3 Function

Application
Virial coefficients from calculation

Syntax
Virial3

Notes
This returns the third virial coefficient of the current record, providing it was analysed as a Zimm plot.

4.174 Volume

Application
Results information for Size record

Syntax
Volume i

Notes
Volume i

Returns the contents of the ith class of the volume distribution from the particle size analysis of the current record.

4.175 Write

Application
Table text output.

Syntax
Write format$,expression_list

Notes
This command writes text to the table window at the current anchor position, using the colour

set by the last TextColour command and the alignment set by the last TextAlign command. expression_list is a list of expressions containing variables, constants, string variables or quoted strings. Each expresion is separated by a comma. The format$ contains information on how to format each item in the expression list and must not exceed 255 characters in length. For each item in the expression_list there must be corresponding format item in format$.

Details on what may appear in format are given below.

Formatting numbers
To format numbers use the # character to represent each digit and the period to represent the decimal point. The number of digits before and after the decimal point can be varied by adding # characters. An example for format$ is “##.###”. Negative number are prefixed with a minus sign. The number may also be represented in scientific format using a format such as “#.###^^^^”,The ^ characters represent the position of the exponent termEsxx where E denotes the exponent , s is the sign (+ or -) and xx is the power of 10 to be used.

Note The scientific format must have 4^ characters and must have only one # charater before the decimal point. For example, we show the result of displaying hte number 123.4567 with variaous formats.

Format	Result
###.###	123.456
###.##	123.46
####	1234
#.####^^^^	1.2345ED2

String Characters
To format a string variable, string functions or quoted strings use the $ character to represent each character of the string. The character string is left justified in the format string if the format string is longer than required. For long strings you do not need to use a large number of $ characters. If the format is “$n”, where n is a number between 1 and 256, then it acts as if n $ characters were used. The format"$*" acts as if the same number of $ were used in the format as appear in the string.

For example, below is the result of displaying the string “PCS Software” with various formats.

Format	Result
$$$$$$$$$$$$$	PCS Software
$$$$$$	PCS Sof
$8	PCS Softw
$*	PCS Software

Other characters
All other characters in the format string are printed without alteration.

4.176 WriteAt Command

Application
Table text output.

Write format$,expression_list

Syntax
WriteAt x,y,"Format",expression_list

Notes
This command is similar to the write command but allows the starting point for the text to position as well. The anchor point is moved to x mm from the left margin and y mm from the top margin.

4.177 Xchar Function

Application
Page programming.

Syntax
XChar

Notes
Returns average character width of current font.

4.178 Xpage Function

Application
Report page dimensions. For use with page programs only.

Syntax
XPage

Notes
This function returns the width of the current printer page in mm.

4.179 Xpos Function

Application
Report current position in page program. For use with page program only.

Syntax
XPos

Notes
Returns the current x (horizontal) position in mm of the anchor point on the table page from the left hand margin.

4.180 Ychar Function

Application
Character height of current font. For use in page programs only.

Syntax
YChar

Notes
Returns the current character height in mm of the font currently in use.

4.181 Ypage Function

Application
Report page dimensions.

Synax
YPage

Notes
Returns length of the page in mm.

4.182 Ypos Function

Application
Report table position.

Syntax
YPos

Notes
Returns the current y (vertical) position in mm of the anchor point from the top margin of the page.

4.183 ZAnal1

Application
Results information for Zeta record

Syntax
ZAnal1

Notes
Returns the lower zeta potential limit used in the data analysis. The use of an upper and lower zeta potential to window the result spectrum allows the exclusion of spectra with no physical significance.

4.184 ZAnal2

Application
Results information for Zeta record

Syntax
ZAnal2

Notes
Returns the upper zeta potential limit used in the data analysis. The use of an upper and lower zeta potential to window the result spectrum allows the exclusion of spectra with no physical significance.

4.185 ZAveMean Function

Application
Results information for Size record

Syntax
ZAveMean

Notes
The Z Average mean Size result of current record.

4.186 ZeroFieldWidth Function

Application
Data information for Zeta records.

Syntax
ZeroFieldWidth

Notes
Returns the width of the zerofield measurement performed when the data was collected. If no measurement was performed then the value will be zero.

4.187 ZetaChannels Function

Application
Result information from current Zeta record.

Syntax
ZetaChannels (z)

Notes
Returns the intensity corresponding to the zeta potential z.

4.188 ZetaPotential Function

Application
Result information for Zeta records.

Syntax
ZetaPotential

Notes
This function the mean, or average, Zeta Potential of the Zeta potential distribution.

4.189 ZetaSpectrum

Application
Result information from current Zeta record.

Syntax
ZetaSpectrum(z)

Notes
ZetaSpectrum (z)
Raw intensity at zeta potential z. Normally the standard analysis fits gaussian peaks to the frequency spectrum, so that interpolated results are effectively obtained. This function enables access to the direct frequency spectrum, that is the frequency bin which the zeta potential z is mapped into.

4.190 ZetaVar

Application
Result information from current Zeta record.

Syntax
ZetaVar

Notes
Width of Zeta potential distribution.

4.191 ZimmError Function

Application
Returns the fit error of a Zimm plot

Syntax
ZimmError

Notes
Returns the global fit error of the Zimm plot.

4.192 ZimmFitOrder Function

Application
Returns the order of fit of a Zimm plot.

Syntax
ZimmFitOrder

Notes
The function returns 1 or 2 depending on whether the fit is first or second order respectively.

4.193 ZimmNumConcs

Application
Data information for intensity record

Syntax
ZimmNumConcs

Notes
Returns the number of concentrations used in the Zimm Scan

4.194 ZimmNumAngles

Application
Data information for intensity record.

Syntax
ZimmNumAngles

Notes
Returns the number of angles in the Zimm plot

4.195 ZimmNumPoints

Application
Data information for intensity record.

Syntax
ZimmNumPoints

Notes
Total number of calculated points in Zimm plot

4.196 ZimmAngle Function

Application
Data information for intensity record.

Syntax
ZimmAngle (concentration, angle)

Notes
Return angle of a particular point on a Zimm plot

4.197 ZimmQ Function

Application
Returns wavevector of a particular point on a Zimm plot.

Syntax
ZimmQ (concentration, angle)

Notes
Returns the wavevector of a particular point on a Zimm plot.

4.198 ZimmResidual

Application
Returns residual of a particular point on a Zimm plot.

Syntax
ZimmResidual

Notes
Returns residual of a particular point on a Zimm plot. The residual is defined to be (ZimmIntensity - ZimmFit) of that point.

4.199 ZimmIntensity Function

Application
Data information for intensity record.

Syntax
ZimmIntensity

Notes
ZimmIntensity (iconc, iangle) Zimm plot y coordinate of a point (Kc/R - inverse Mw)

4.200 ZimmFit

Application
Data information for intensity record.

4.201 ZimmConc

Application
ZimmFit(iConc, iAngle)
Concentration of a particular point (g/l)

4.202 ZimmFitorder

Application
Result information from an intensity record

Syntax
ZimmFitOrder

Notes
Order of fit in concentration.

4.203 ZimmError

Application
Result information from an intensity record.

Syntax
ZimmError

Notes
Square root of sum of squares of residuals.

4.204 Zerowidth

Application
System information for Zeta mode.

Syntax
ZeroWidth

Notes
Zero field width from last Zeta potential analysis. Expressed in Hz.

4.205 Zeta Command/Function

Application
Sends commands to the Zetasizer or Zetamaster optics. This will only operate if the RS232 communications with instrument are valid.

Syntax
Zeta Voltage (voltage, supply_mode)
Zeta Current (voltage, supply_mode)

Zeta Start
Zeta Stop
Zeta Angle (number)
Zeta ReadTemperature
Zeta Temperature (number)
Zeta Inject (number)
Zeta Flush (number)
Zeta Shu (number)
Zeta ModFreq (number)
Zeta OnTime (number)
Zeta OffTime (number)
Zeta Strobe (number)
Zeta OnTime (number)
Zeta Shutter(number)
Zeta Valve (number, number)
Zeta Mode
Zeta CellPosition (number)
Zeta GetSystemArray (number)
Zeta SetSystemArray(number,number)
Zeta Valve (number, number)
Zeta Cellposition (number)
Zeta CellDrive
Zeta ModulatorFreq
Zeta Setup
Zeta Motors
Zeta Connect
Zeta calibrate

Notes
Zeta Voltage
Voltage level for next electrophoresis measurement. Institute voltage regulation mode. Voltage must be between 0 and 399.99, typically 150. i = 1 DC-, 2 = DC+, 3 = AC

Zeta Current
Current level for next electrophoresis measurement. Institute current regulation mode. Current (in mA) must be between 0 and 19.99, typically 5. i = 1 DC-, 2 = DC+, 3 = AC

Zeta Start
Start Zetsizer optics measuring in selected mode. In size mode this command has little effect, in Zeta Mode the Cell voltage and modulator are turned on.

Zeta Stop
Stop Zetasizer optics. Cell etc. turned off, and readings of voltage and current in each DC sense, at the beginning and end of the measurement are recorded. These and other relevant parameters are loaded up to the software running on the host computer (cell type, position pH etc.)

Zeta Angle
Change scattering angle for size measurements. Range 0 - 135.

Zeta ReadTemperature
Read back current temperature of cell.

Zeta Temperature x
Output requested temperature to optics unit. 5 - 75. Range for Zeta 1000 - 3000

Zeta Inject x
Send an Inject command to the optics. For a unit fitted with a sample handling unit, and that unit enabled an Inject sequence will start. The command will then terminate when the sample injection is complete. The parameter is the number of ml to inject. If omitted the last value sent or default is used. If no sample handling unit is fitted, the inject valve is opened, and the system waits for the operator to confirm manual injection is complete, by pressing the ‘inject’ button on the optics unit.

Zeta Flush x
A flush sequence identical logically to the ‘inject’ described above, is invoked.

Zeta Shu 1 returns the sample handling unit On or Off to enable manual inject/flush to intervene. Naturally a ZET5115 must be mounted in the optics for a Shu 1 to be effective. This flag is stored in the optics unit memory when a Save Config action is performed.

Zeta ModFreq
Modulator frequency and phase. 125,250,500,1000 are all valid numbers for the frequency. A positive sign indicates that the phase will be such that negatively charged particles will produce doppler frequencies less than the modulator, positively charged ones greater: ModFreqA negative sign reverses this condition.

Zeta OnTime
The electrophoresis cell is ON for the period specified in modulator cycles: ie the period is OnTime * 1 /Modfreq.

OffTime
Zeta The electrophoresis cell is OFF for the period specified in a similar fashion to the OnTime.

Zeta Strobe
A delay before enabling the correlator to collect data in each electrophoresis cell cycle. Strobe -1 is the normal setting, which effectively means 0 delay with data collection during the OnTime. Larger negative values delay in modulator cycles from the start of the. OnTime. The only positive value that should be used is 1 which collects data for the entire OffTime.

Zeta Mode
Puts experiment into zeta mode, so that commands select the appropriate main Window to operate.

Zeta CellPosition
Set cell position (0 - 100, are the extreme)

Zeta Shutter (number)
Zeta 1000 - 3000. Only. Block reference beam if number is 0, opens ref. beam if 70.

Zeta Valve iValve, boolean.
Opens or closes valve on Zetasizer.

4.206 ZetaPotential

Application
Result information from a zeta record.

Syntax
ZetaPotential

Notes
Zeta potential result of current record. ie. the mean value.

4.207 ZetaValue Function

Application
Value of intensity at given zeta potential of current record.

Syntax
ZetaValue (z)

Notes
Returns the value of the intensity corresponding to the zeta value of number.

4.208 ZetaVar Function

Application
Zeta potential width of current record.

Syntax
ZetaVar

Notes
Returns the width of the current zeta potential distribution.

4.209 Zlower Function

Application
Data information from Zeta record

Syntax
Zlower

Notes
Lower Zeta limit used for data aquisition. See Measure command.

4.210 Zupper Function

Application
Data information from Zeta record.

Syntax
Zupper

Notes
Upper Zeta limit used for data aquisition. See Measure command.

CHAPTER 5

Advanced Features

5.1 Exchange of Data with other programs

Two basic methods are available to the user. DDE (dynamic data exchange) is a message passing system which can be used to set up a link to other applications This is essentially a Macro program in the PCS program which is run to transmit the necessary data. This is either done explicitly by running a program containing DDE POKE'S addressed to the partner application (in this mode the PCS program is referred to as the 'client', the partner as the 'server'). Alternatively the client-server relationship can be reversed by 'posting' data as being available from the PCS program. This is done by selecting COPY LINK from the EDIT menu.

The 'client' application that wants to receive the data must then provide a PASTE LINK function which will receive the information. The actual format of the content of the data is governed by the macro program DDELINK. This program is run, after COPY LINK is pressed, whenever the LIVE BUFFER is updated by an analysis or file transfer operation.

Also under EDIT are COPY GRAPH and COPY TABLE which are used to place 'pictorial' data on the clipboard allowing data to be 'pasted' into the target application. This method is referred to as a 'cold link'

The other method, possibly simpler in conception, is to write a variant of a page description or other Macro program containing statements that write data to a file in an essentially text format. Each of these methods are explored further below. The DDE and file methods are obviously useful when 'live data' needs to be transferred, as it is available, to a spreadsheet or process control application, for example. The clipboard method is suitable for building up a report incorporating output from the PCS application.

5.1.1 Transferring data by DDE with the PCS program as Client

In this method of interprocess communication a Macro program run from the PCS application is 'in charge' of the transfer. It can run the 'server' application and uses special transfer statements to send string data to that application. Since Print ... using can be used to format even numerical data into a string form, a very flexible type of communication is possible. It is quite possible for the user to write their own applications to communicate by DDE with the PCS program, but the examples we shall discuss here are restricted to the PCS program sending data to 'standard' applications, in particular Microsoft's Word and Excel packages. (© Microsoft Ltd.). The example shown below will copy a single piece of data from the PCS program to Word.

The first line of the program 'execute normal "d:\winword\winword" will run Word for Windows. (Details of the pathname may be different on your system). It is shown as 'commented out' as Word was already running when the example was being tested. With Word running the DDE Connect statement opened DDE channel 1 to Document1, the default Word filename. A print ... using statement is used to assign to a$ a simple zeta potential result, which is then printed to the special DDE 'string buffer' with print string. The DDE poke statement sends data on DDE channel 1, the parameter being the 'item'. In the case of 'Word for Windows' this item is a so called 'Bookmark', a user inserted label delineating the destination for the transferred data. The status of the 5 DDE links allowed can be examined under View DDE Links, shown for the present example below.

To put in the bookmark, go to Word's Insert Bookmark menu item, and put (for example) label1 as a bookmark. When the POKEX1 is run the string (for a particular example) Zeta Potential = 22.2 will appear in document1. Note that the topic and item can be referred to via string variables that can be set up using a 'print string' construct.

We show such an example below for communication with the spreadsheet program Excel. Valid items for Excel include strings of the form “R1C1", which addresses row 1 column 1 in a spreadsheet document. In the example we make use of the Macro language string handling capability to randomly access the first two columns of the target spreadsheet.

```
‘ Program to transmit Zeta potential values
 ‘ and class contents to Excel
 ‘_____________________________________________________-
 ‘
‘
 ‘______________________________________________________
 ‘
 ‘ First connect to the default spreadsheet
‘ sheet1
‘
 ‘
dde connect 1,"excel","sheet1"
 k=0 :’           initialise row counter
for z=0 to -40 step -2: ‘ Range of zeta values
 ‘
 ‘ initialise a$ then append zeta spectrum value
 ‘ to it
 ‘
a$=""
print a$ using “ ####.#”,zetavalue(z)
 k=k+1
b$=""
‘
 ‘ Clear b$ and append the row and column
‘ string as the ‘item’ for the poke
‘
 print b$ using “R#C2",k
‘
 ‘ clear the special buffer for the poke data
‘ and append the zeta data in a$ to it
‘
 delete string
 print string a$
 dde poke,1,b$
 ‘
‘ Now do the same in column 1 for the zeta potential
 ‘ value itself
 ‘
a$=""
 print a$ using “ ####.#”,z
b$=""
print b$ using “R#C1",k
 delete string
 print string a$
 dde poke,1,b$
 next
 end
```

5.1.2 Transferring data by File

In the example listed below (PCOEX.PCM) a program has been written to dump the major parameters of a PCS correlation experiment to disk. This program is named PCOEX by analogy to a command (PCO) in previous DOS based Malvern software which 'put the correlogram' to a specified file or device, for processing by other programs. The data supplied would be sufficient for another application to analyse the correlogram by some other method than that built in, a force fit to one or more exponentials, for example.

```
' This program saves the correlator contents
 ' and main operating parameters
' in a text file "corrfun.txt" .
 ' This would typically be sufficient information
' for a user written analysis routine to function.
' Other output parameters would be easy to supply.
' ______________________________________________________________-
 '______________________________________________________________
 '
 file "corrfun.txt"
print file using "$40 \n",note1$
 print file using "Sample time ####.# Total Samples ########## \n",SampleTime(),TotalSamples()
 print file using "A = ######### B = ########## \n", Achannel(),Bchannel()
 for i=1 to numchannels
 print file using "########### ########### \n",TimeArray(i),StoreChannel(i)
 next
 print file using "Far Point (M) ########### (C) ############ \n",MeasFarPoint(),CalcfarPoint()
 end
```

The program begins, typically, with comments describing its function. A file 'corrfun.txt' is opened, then 'print file using' statements are executed to output ...

(a) The first 40 characters of the sample title.

(b) Correlator sample time and total samples monitor channel.

(c) The A and B monitor channels

(d) A tabulation of all correlator channel times and contents, terminated by the measured and calculated far point values. A typical output from running this macro program is listed below.

Note that the 'file' command opens a file without clearing it. The print file statements then append their output to the file. The '\n' included in the formatting text forces a new line character. To shorten the list a few channels from the middle of the function have been omitted from the listing.

```
Duke "54nm" latex
Sample time                 4.9 Total Samples   229596
A = 2409110 B =     0
1           554775
2           549476
3           546255
4           541389
 5          535989
6           532879
7           528519
```

```
 8          524296
9           520346
10          516587
 11         512896
12          509013
13          505311
 14         503627
15          499345
16          495914
 4          541193
8           524698
12          509311
 16         496582
20          485082
24          474354
 28         464848
32          457085
36          450052
 40         444154
44          438557
48          433481
 52         428824
________________________
 208        395565
 224        395493
 240        395372
 256        394850
 64         427010
 128        400173
 192        396249
 256        395156
 320        394904
 384        394946
 448        394994
 512        395353
 576        395009
 640        395060
 704        394969
 768        394878
 832        394742
 896        395195
 960        395048
 1024       395113
Far Point (M)       394938 (C)          394978
```

Improvements that could be considered could include allowing the user to enter the filename, by including an ‘input’ statement eg.

```
input “Filename (no extension) ”, fn$
fn$=fn$+".txt"
file fn$
```

The user might prefer to start with a clear file by executing a *file “corrfun.txt”*:*delete file* before opening it again, now empty, to receive the correlogram.

A simple variant of this program could readily be used to output results rather than data. Please refer to one of the supplied ‘page programs’ (sizerep1, zetarep1) that display results. The functions *sizes* and *intensity* that return size classes and contents could easily be used with *print file using* rather than *write* for this purpose.

5.1.3 Transferring data by Macro program via the clipboard

First check out the data to send by loading and examining the program DDELINK under the program editor. (Control -Edit). An example for sending size data is shown below.

```
‘ Program to Create Data for
‘ copy link
‘ (this program must be saved as DDELINK.PCM)
 ‘
‘____________________________________
‘
 delete string
for i = 1 to numsizes
print string using “####.#\t###.#\r\n”,sizes(i),intensity(i)
 next
 end
```

The program sends two columns of numbers, separated by tabs, the second number being terminated by carriage return/line feed. The first column is the size class, the second the intensity fraction at that size. It would be very simple to alter, for example, the program to send a ‘number’ distribution, rather than intensity, (by altering *intensity(i)* to *number(i) .* To send correlator channel data, rather than size results would simply involve incorporating parts of the PCOEX program used for File generation above, so this method is very flexible.

Having got the program to send the required data, simply press EDIT COPY LINK in the PCS program. Go to EXCEL (for example) select a range of cells of suitable extent to contain the data generated by the program above, (24 rows of 2 columns, for the example given), and press Paste Link. The selected space in the spreadsheet should be filled with the appropriate information. Whenever another calculation is performed in the PCS program, so that the current record in the LIVE BUFFER is updated, the DDELINK will rerun automatically and the spreadsheet will update. It should not be necessary to press Paste Link again: This type of link is referred to as a HOT LINK, since data is transmitted automatically as it is updated.

5.1.4 Remote control by DDE

It is possible to send commands to the PCS program over the DDE. One must establish what is known as an Execute link between the client and server, the client in this case being the controlling application, the PCS program the server. An example written in Microsoft Visual Basic (©Microsoft Ltd). is shown which sends commands typed in by the user. The command is a recognisable Macro language string. An example would be *corr clear:corr start*, which would invoke correlator operation.

5.2 Automatic startup

This process is a conceptually very simple one. A Macro program is written and debugged in the normal way, using Control Edit. When the program is correct it may be assigned as the autorun program. When the application starts, the program will automatically execute. A typical startup program might be used to run a validation on the correlator, then relinquish control to the keyboard/mouse again. Alternately a startup program might be in control for the entire session, possibly responding to remotely signalled DDE events (running with the Autosampler Scheduler programme is a case in point). A program might run in a completely stand alone way, performing a series of processing steps, then terminate using the Exit command.

5.3 Remote Control

The PCS program can be put into a remote control mode by which commands are received over an RS232 link. Such commands are text based and in fact consist of statements in the Macro control language. Often they will consist of requests to a run a predefined program, since this minimises the comm's overhead. At the start and as each line of commands are executed, a prompt is transmitted from the PCS program, so that the remote controller can recognise that the application is ready to receive more input.