

ADDENDUM TO

N W A Q U A L I T Y A N A L Y S T

**User's Manual:
Chapters 9 and 12**

This addendum contains Chapters 9 and 12 of the NWA Quality Analyst User's Manual in their entirety. It is intended to accompany a downloadable update to Quality Analyst and describes new Run File and Database Connectivity features of that update.

Table of Contents

CHAPTER 9

Automating Procedures with Run Files	1
Creating Run Files.....	2
How Run Files Work.....	2
Run File Commands and Parameters.....	3
Replaceable Parameters.....	92
Creating Complete Task Sequences	97
Executing Run Files	102
Run File Errors.....	110
DOS-based Run Files.....	111

CHAPTER 12

Database Connectivity.....	113
Overview.....	113
Before Using Connectivity.....	114
Defining Connectivity Data Sets	115
Using Replaceable Parameters with Filter Criteria in SQL Statements	137
Using Connectivity Data Sets.....	143
Modifying Connectivity Data Sets.....	144
Using Database Connectivity in Run Files.....	147
Configuring and Troubleshooting	151
Getting Technical Support for Connectivity	163

INDEX

Index.....	165
-------------------	------------

CHAPTER 9

Automating Procedures with Run Files

Most Quality Analyst functions can be automated with “Run Files”—scripts created by you that instruct Quality Analyst to automatically perform one or more of its functions. Quality Analyst Run Files can be used to automate repetitive tasks and to enable less experienced users to perform complex sequences of tasks. Run Files can be especially useful if you want to:

- Perform standard sequences of tasks where a large number of charts are routinely created from the same data sources.
- Create time-consuming printer or plotter graphics output unattended.
- Automate the creation of multiple chart layouts.
- Set up an automated charting session for workers less familiar with the use of computers and Quality Analyst.
- Create a control chart automatically from another software package by setting up a Run File to perform the required operations.
- Use Quality Analyst as the SQC charting component of an integrated manufacturing-information system.

Run Files are easy-to-create text files, and can be as simple as a series of commands to print control charts. The largest section of this chapter—***Run File Commands and Parameters*** beginning on page 3—discusses each Run File command in detail and provides examples of the required command syntax.

You can execute Run Files from within Quality Analyst (selecting the desired Run File from a list), from icons on the desktop, or from other applications. This is explained in the section ***Executing Run Files*** beginning on page 102.

More advanced Run Files can prompt the user for information, and perform complex work flows including page layout, data filters, and calculations. The

section *Creating Complete Task Sequences*, beginning on page 97, explains the creation of complex work flows and includes examples of complete Run Files. Run Files have allowed many Quality Analyst users to create SPC/SQC charts as part of an integrated process-information system. These systems often include combinations of data systems and other computers sharing process information by transferring data files. Charting is automated by passing a data file to the system that is running Quality Analyst and initiating Quality Analyst with a Run File to generate the desired charts. Contact NWA if you are interested in this type of application.

Creating Run Files

Run Files are ASCII text files, and may be created and edited using any text editor (such as Windows' Notepad) that can handle pure ASCII files. Each line in a Run File contains a single Run File command with arguments and parameters as explained under *Run File Commands and Parameters* below. Run File names must have the extension "RUN."

Note: Most full-featured word processors such as Microsoft Word and WordPerfect normally create files using their own proprietary format. Files created in these formats *cannot* be used as Quality Analyst Run Files. These programs *can* load and save ASCII files, but only when specifically instructed to do so. Consult the documentation for your word processor for instruction in using it to create ASCII files.

Quality Analyst includes a Run File Wizard that guides you, step-by-step, through the process of creating a Run File. A friendly user-interface lets you select and insert commands, parameters, file names, and variable names with point-and-click ease. To launch the **Run File Wizard**, select Run File Wizard from the **Run** menu.

How Run Files Work

Quality Analyst executes Run Files using a special program (called the Run File Interpreter) that reads each line of a Run File, interprets each Run File command, and manages the various other Quality Analyst routines needed to deliver the requested output. Once started, a Run File will continue to execute

automatically, and will only pause when instructed to do so by a Run File command.

Each Run File consists of one or more lines, with each line containing the instructions for performing a Quality Analyst function. There is no limit on the number of lines in a Run File, and some users have created Run Files with over 500 lines.

Run File Commands and Parameters

In this section, we describe all commands and command parameters available for use within Run Files. The table below lists the Run File commands and parameters in alphabetical order, as they are presented in the rest of this section.

Alphabetical List of Run File Commands and Parameters

ANOVA	AskTimeMatchFilter	MULTREG
ANOVAR	AskTimeRange	NEWVAR
APPEND	AskTimeRangeFilter	NORMPLOT
AskAlpha	AskTimeSelectFilter	PARETO
AskAlphaMatchFilter	AskVariable	PAUSE
AskAlphaRange	ATTRIBUT	PCAP
AskAlphaRangeFilter	BOXPLOT	POLREG
AskAlphaSelectFilter	CALC	PRINT
AskDate	CHART	PROMPT
AskDateMatchFilter	CHITEST	REDUCE
AskDateRange	CLEANUP	REGRESS
AskDateRangeFilter	CLEAR	REM
AskDateSelectFilter	\$COMMENT	REPORT
AskDATFile	CONNECT †	RUN
AskInteger	CORVAR	RUNCHART
AskIntegerMatchFilter	CUSUM	SCATTER
AskIntegerRange	EWMA	SET
AskIntegerRangeFilter	EXTRACT	SHELL

AskIntegerSelectFilter	\$FILTER	SINGCOL
AskList	FLIP	SUMMARY
AskNumber	GROUP	SUMSTAT
AskNumberMatchFilter	IR	SYSTEM
AskNumberRange	KRUSKAL	TITLE
AskNumberRangeFilter	MCAPREP	TTESTS
AskNumberSelectFilter	MERGE	USETEMPDIR
AskOpenFile	MPCAP	WALDRUN
AskSaveFile	MR	WEIBULL
AskTime	MSORT	XRS

The “Ask” commands and APPEND, CLEAR, SET, and USETEMPDIR are available only in the 32-bit version of Quality Analyst.

† The CONNECT command is available only in the Database Connectivity version of Quality Analyst. See *Using Database Connectivity in Run Files* on page 147.

Note: Run File commands are **not** case-sensitive. As long as they are spelled correctly they will be interpreted correctly by the Run File Interpreter. To enhance readability, some commands are shown in this manual using mixed case.

See Also: *Replaceable Parameters*, p. 92
Command Line Options, p. 108

Use the following table to find the appropriate command for the function you want to perform.

Run File Commands and Parameters

Function Type	Function	Command or Parameter
Charting	c-chart	ATTRIBUT
	Cumulative Sum	CUSUM
	Exponentially Weighted Moving Average	EWMA
	Individuals	IR

Function Type	Function	Command or Parameter
	Individuals & Range	IR
	Median	MR
	Median & Individuals	MR
	Median & Range	MR
	Normal Probability	NORMPLOT
	np-chart	ATTRIBUT
	Pareto	PARETO
	p-chart	ATTRIBUT
	Process Capability Histogram	PCAP
	Process Capability Report (display)	MCAPREP
	Process Capability Report (prepare)	MPCAP
	Quantile plot	NORMPLOT
	Range	XRS, IR
	Run chart	RUNCHART
	Scatter diagram	SCATTER
	Standard Deviation	XRS
	u-chart	ATTRIBUT
	Weibull analysis	WEIBULL
	X-bar	XRS
	X-bar & Range	XRS
	X-bar & Standard Deviation	XRS

Function Type	Function	Command or Parameter
General Statistics	Analysis of variance	ANOVA, ANOVAR
	Bartlett's test	CHITEST
	Box Plots (display)	BOXPLOT
	Box Plots (prepare)	MPCAP
	Chi-square statistics	CHITEST
	Descriptive statistics	SUMSTAT
	Formatted reporting	REPORT
	Kruskal-Wallis statistic	KRUSKAL
	Linear correlation	CORVAR
	Pearson Product Moment Correlation	TTESTS
	Regression (multivariable)	MULTREG
	Regression (polynomial)	POLREG
	Regression (single-variable)	REGRESS
	t-statistics	TTESTS
Wald-Wolfowitz Runs test	WALDRUN	
Output	Comment a chart	\$COMMENT
	Exception reporting	SUMMARY
	Formatted reporting	REPORT
	Group layout	GROUP
	Printed output	PRINT
	Screen display	CHART
General	Add new data	APPEND
	Calculations	NEWVAR
	Comment a Run File	REM
	Convert to/from single column	SINGCOL
	File conversion	CLEANUP
	File subset extraction	EXTRACT
	Filter data for charting	\$FILTER
	Manipulate rows, columns, and blocks	MERGE
	Reduce data	REDUCE
Rotate rows and columns	FLIP	

Function Type	Function	Command or Parameter
	Sort a data file	MSORT
	Title a Run File	TITLE
Flow Control	Assign values to parameters	SET
	Call (nest) another Run File	RUN
	Execute a program	SHELL
	Pause execution	PAUSE
	Prompt user for input	All commands beginning with "Ask" PROMPT
	Quit Run File	SYSTEM

When executing the Quality Analyst Run File Interpreter from a Windows command line or program icon, you can include any of several parameters in the command line to control the way the Run File behaves. These are explained in the section *Command Line Options* on page 108.

Note: In the examples in the following sections, some Run File command lines are too long to fit on a single line in this manual so they wrap to an indented second line. In your actual Run File, however, *you must enter each command line on a single line*. For example, two sample Run File command lines in this manual might look like this:

```
PARETO NOZZLE.DAT M SEAL THREADS WELD G
      NOZZLE.PLT
PRINT NOZZLE.PLT
```

But in your Run File they should look like this:

```
PARETO NOZZLE.DAT M SEAL THREADS WELD G NOZZLE.PLT
PRINT NOZZLE.PLT
```

The indented second line is actually the end of the first line. The text editor you use to create your Run Files will be capable of creating lines much longer (usually up to 255 characters) than we can display on a single line in this manual.

General Form of Run File Commands

Each line in a Run File begins with a Run File command. Each command performs one or more Quality Analyst functions. The rest of the command line includes information required for the command to be interpreted and executed correctly. This information includes the names of input files and data variables, as well as command options and parameters used to create and control output. Often, more than one command line is needed to produce the desired output.

For most commands, the syntax of a command line in a Run File is:

```
command infile vars instructions output outfile
```

where the elements of the command are as follows:

Command Element	Description
command	the Quality Analyst command to be executed
infile	the name of the file containing the data to analyze or chart
vars	the name of the variable(s) in the input file to use
instructions	one or more specific instructions to the command
output	an output type or device
outfile	the name of the file where the output of the command will be stored

Some command elements may not be required for some commands.

Note: Notable exceptions to this general form are the “Ask” commands (starting on page 13) which prompt the user for input.

Examples

To set up an X-bar and Range chart for the variable GAP in the data file HOUSING.DAT, the command line would be:

```
XRS HOUSING.DAT GAP X R G
```

XRS is the name of the Run file command. **HOUSING.DAT** is the name of the input file. **GAP** is the name of the variable. **X** and **R** are command parameters (instructions) that tell **XRS** to produce both an X-bar and a Range chart. **G** tells Quality Analyst to create a graphics (PLT) file containing the chart information.

This command line alone would produce no “visible” output. To display the chart on the screen, you would follow it with a **CHART** command. To print it, you would follow it with a **PRINT** command. A Run File that produces a process capability histogram in addition to the X-bar and Range chart might look like:

```
XRS HOUSING.DAT GAP X R G
CHART
PCAP HOUSING.DAT GAP G
CHART
```

Requirements

All file names used in a command line in a Run File must include the file-name extension. Quality Analyst does *not* automatically add standard or default extensions to file names in Run File command lines. This applies to input files, output files, and definition files.

Quality Analyst Run Files can produce three types of **output**. The available output options vary with each command and are discussed in the section on each command.

Run File “output” Options

Output Option	Description
G	Graphics output: creates a graphics (PLT) file to be displayed by a subsequent CHART or PRINT command.
W	Windowed text output: Creates a text (PTX) file, formatted with tab characters, to be displayed by a subsequent CHART or PRINT command.
D	Data file: either a summary (SMY) file (containing out-of-control points and rule violations) for subsequent use by a SUMMARY command, or a data (DAT) file, depending on the function.

If a graphics output file is created by a Run File command (by including the “G” parameter), and the graphics file name is not specified, the default name GRAPHICS.PLT will be used.

Also note that the “G” and “W” options respectively create a graphics (PLT) file and formatted text (PTX) file but do *not* display or print them. This is accomplished by an additional command line containing a “CHART” or a “PRINT” command or both.

ANOVA

The ANOVA command performs an analysis of variance for one-way experimental designs with equal or unequal subclass numbers (uneven Data Set). The syntax of the ANOVA command line in a Run File is:

```
ANOVA inpfile vars output outfile
```

Command Element	Description
<code>inpfile</code>	the name of the input file to use
<code>vars</code>	the names of two or more variables (separated by spaces) in the input file that you want to analyze
<code>output</code>	“W” to create a formatted text (PTX) file.
<code>outfile</code>	the name of the output file

Example: The command line to analyze the variance of each individual measurement in the subgroup variable GAP in the Data Set BENDCLIP.DAT is:

```
ANOVA BENDCLIP.DAT GAP:1 GAP:2 GAP:3 GAP:4 GAP:5 W
      DEFAULT.PTX
```

The “W” option creates a formatted text (PTX) file but does *not* display or print it. This is accomplished by an additional command line containing either a “CHART” or a “PRINT” command.

ANOVAR

The ANOVAR command performs an analysis of variance for one-way randomized block designs: a cross-classification of data by blocks (or subjects) and treatments. This is also referred to as “repeated measures” or “subjects-by-treatments” design. The syntax of the ANOVAR command line in a Run File is:

```
ANOVAR infile vars output outfile
```

Command Element	Description
<code>infile</code>	the name of the input file to use
<code>vars</code>	the names of two or more variables (separated by spaces) in the input file that you want to analyze
<code>output</code>	“ W ” to create a formatted text (PTX) file.
<code>outfile</code>	the name of the output file

Example: The command line to analyze the variance of the variables LINE1, LINE2, LINE3, and LINE4 in the file BAGGER.DAT is:

```
ANOVAR BAGGER.DAT LINE1 LINE2 LINE3 LINE4 W DEFAULT.PTX
```

The “**W**” option creates a formatted text (PTX) file but does *not* display or print it. This is accomplished by an additional command line containing either a “**CHART**” or a “**PRINT**” command.

APPEND

The APPEND command displays an abbreviated version of the Quality Analyst Editor and lets you add data to a Data Set. This command is available only in the 32-bit version of Quality Analyst. The syntax of the APPEND command is:

```
APPEND filename numrecords
```

Command Element	Description
<code>filename</code>	the name of the data (.DAT) file

numrecords

the number of records (lines) to add to the file

The window for the Append editor will display five data rows and is not resizable. If **numrecords** is greater than five, a vertical scroll bar provides access to the additional rows.

In the Append editor, you **can**:

- see the structure (column headings) of the data,
- add data,
- edit or delete the added data,
- see the results of calculated variables, if any.

In the Append editor, you can **not**:

- see or edit data in the original Data Set,
- change the structure of the Data Set,
- add or delete rows of data.

Click **Continue** to append the data records you entered, or click **Quit** to proceed using the original Data Set.

Example: The following Run File statement displays the editor as shown below.

APPEND DEFECTS.DAT 7

Appending To DEFECTS.DAT								
Row	DATE 1---(D)	FOREIGN 2---(P)	BUBBLES 3---(P)	SINKS 4---(P)	MOUNT 5---(P)	MARKS 6---(P)	TRIM 7---(P)	EDGE 8---(P)
1								
2								
3								
4								
5								

Date of Production Autocopy Mode: ROW

Notice that the vertical scroll bar indicates the presence of additional rows past the fifth.

AskAlpha

AskDate

AskInteger

AskNumber

AskTime

Note: There are several Run File commands that prompt the Run File user for one or more inputs of a specific data type and then assign those inputs to Replaceable Parameters (see page 92). Run File authors can specify the on-screen text used to prompt the user for input, or they can use default text. The user's input is checked to make sure that it complies with the requirements of the data type.

Several variants of the Ask commands prompt the user for a lower and upper limit for a range of values, and several other variants prompt the user for inputs that are to be used as arguments for the \$FILTER parameter (see page 49) of a charting command.

The basic Ask commands ("Ask" followed by a data type) display a dialog box that prompts you for an input of a specific data type and assigns that input to a Replaceable Parameter after making sure that it is a valid input. These commands are available only in the 32-bit version of Quality Analyst. The syntax of the basic Ask command is:

```
command param "prompt" "default"
```

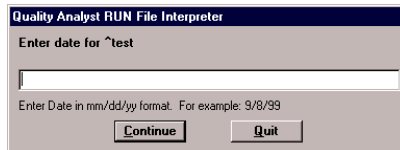
Command Element	Description
command	the command for the data type being requested: AskAlpha (text string) AskDate AskInteger AskNumber (real number) AskTime
param	the name of the Replaceable Parameter that the input will be assigned to (See <i>Replaceable Parameters</i> starting on page 92 for naming requirements.)

prompt	the text to display in the dialog box to prompt the user for input
default	the value to display in the input box when the dialog box first appears

Note: To prompt the user for input to be used with the \$FILTER parameter of a charting Run File command, use the MatchFilter or SelectFilter forms of the Ask command instead.

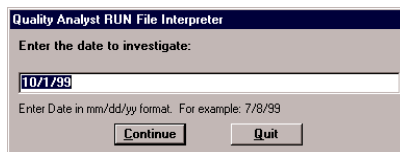
Use double quotation marks as shown in the syntax statement. Both **prompt** and **default** are optional, but if you omit **prompt**, you must also omit **default**. You can use empty quotation marks for **prompt**, in which case standard text will be used. To suppress the **prompt**, use a quoted space. The **default** value, if specified, will be assigned to the Replaceable Parameter when the user clicks **Continue** unless it is first deleted or replaced.

If you omit the optional quoted parameters, this command displays the following dialog box (varying slightly for the different data types):



Example: The following Run File statement displays the dialog box shown below.

```
AskDate ^MatchDate "Enter the date to investigate:" "10/1/99"
```



In this example, Quality Analyst will make sure that the input meets the requirements for the DATE data type.

AskAlphaMatchFilter

AskDateMatchFilter

AskIntegerMatchFilter

AskNumberMatchFilter

AskTimeMatchFilter

Note: See note at *AskAlpha AskDate AskInteger AskNumber AskTime* on page 13.

The MatchFilter forms of the Ask command are intended to be used in conjunction with a \$FILTER parameter (see page 49) on a Run File command line that requests a chart function. They are available only in the 32-bit version of Quality Analyst. (For the 16-bit version, use the PROMPT command, which is explained on page 70, with the \$MATCHFILTER parameter.)

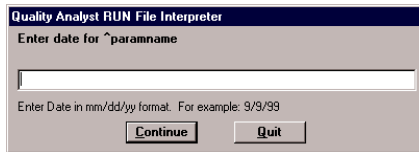
This command displays a dialog box that prompts you for an input of a specific data type, translates your entry into a valid “matching-filter” argument for the \$FILTER parameter, and assigns the resulting string to a Replaceable Parameter. The syntax of the MatchFilter form of the Ask command is:

```
command param "prompt" "default"
```

Command Element	Description
<code>command</code>	the command for the data type being requested: AskAlphaMatchFilter (text string) AskDateMatchFilter AskIntegerMatchFilter AskNumberMatchFilter (real number) AskTimeMatchFilter
<code>param</code>	the name of the Replaceable Parameter that the input will be assigned to (See <i>Replaceable Parameters</i> starting on page 92 for naming requirements.)
<code>prompt</code>	the text to display in the dialog box to prompt the user for input
<code>default</code>	the value to display in the input box when the dialog box first appears

Use double quotation marks as shown in the syntax statement. Both **prompt** and **default** are optional, but if you omit **prompt**, you must also omit **default**. You can use empty quotation marks for **prompt**, in which case standard text will be used. To suppress the **prompt**, use a quoted space. The **default** value, if specified, will be assigned to the Replaceable Parameter when the user clicks **Continue** unless it is first deleted or replaced.

If you omit the optional quoted parameters, this command displays the following dialog box (varying slightly for the different data types):



Read more about *Replaceable Parameters* starting on page 92 and about *Data Filters* starting in Chapter 3.

Example: The first of the following two Run File statements displays the dialog box shown below.

```
AskAlphaMatchFilter ^MatchInsp "Enter the Inspector's name:"
    "STEVE"
ATTRIBUT BOARDS.DAT PART_MIS N G MISSING.PLT $FILTER =
    "INSPECT ^MatchInsp"
```



If the user accepts the default value “STEVE,” then the string “M STEVE” is assigned to the Replaceable Parameter ^MatchInsp and the second Run File command is executed as:

```
ATTRIBUT BOARDS.DAT PART_MIS N G MISSING.PLT $FILTER =
    "INSPECT M STEVE"
```

This command creates an *n*-chart (number defective) of the variable PART_MIS in the file BOARDS.DAT for all records where the value of the variable INSPECT is "STEVE."

AskAlphaRange

AskDateRange

AskIntegerRange

AskNumberRange

AskTimeRange

Note: See note at *AskAlpha AskDate AskInteger AskNumber AskTime* on page 13.

The Range forms of the Ask command display a dialog box that prompts you for two (beginning and ending) inputs of a specific data type and assigns those inputs to two Replaceable Parameters after making sure that they are valid inputs. These commands are available only in the 32-bit version of Quality Analyst. The syntax of the Range form of the Ask command is:

```
command param1 param2 "mainprompt" "prompt1" "prompt2"
      "default1" "default2"
```

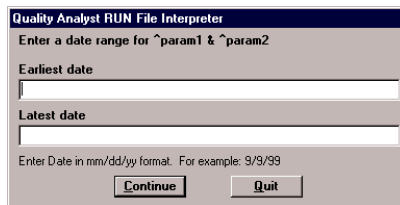
Command Element	Description
<code>command</code>	the command for the data type being requested: AskAlphaRange (text string) AskDateRange AskIntegerRange AskNumberRange (real number) AskTimeRange
<code>param1</code>	the name of the first (beginning) Replaceable Parameter (See <i>Replaceable Parameters</i> starting on page 92 for naming requirements.)
<code>param2</code>	the name of the second (ending) Replaceable Parameter
<code>mainprompt</code>	the text to display in the dialog box to prompt the user for both inputs

prompt1	the text to display in the dialog box to prompt the user for the first input
prompt2	the text to display in the dialog box to prompt the user for the second input
default1	the value to display in the first input box when the dialog box first appears
default2	the value to display in the second input box when the dialog box first appears

Note: To prompt the user for input to be used with the \$FILTER parameter of a charting Run File command, use the RangeFilter or SelectFilter forms of the Ask command instead.

Use double quotation marks as shown in the syntax statement. All quoted elements are optional *from last to first*. That is, you may omit a quoted element *only if you also omit all* elements after it. You may, however, use empty quotation marks for any of the quoted elements. In that case, standard text will be used for **prompts**, and **defaults** will be left blank. To suppress any **prompt**, use a quoted space. The **default** values, if specified, will be assigned to the Replaceable Parameters when the user clicks **Continue** unless they are first deleted or replaced.

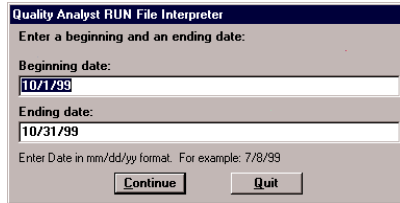
If you omit the optional quoted parameters, this command displays the following dialog box (varying slightly for the different data types):



Read more about *Replaceable Parameters* starting on page 92.

Example: The following Run File statement displays the dialog box shown below.

```
AskDateRange ^BegDate ^EndDate "Enter a beginning and an
ending date:" "Beginning date:" "Ending date:" "10/1/99"
"10/31/99"
```



Quality Analyst will make sure that the inputs meet the requirements for the DATE data type and that the first date is before (or the same as) the second.

AskAlphaRangeFilter

AskDateRangeFilter

AskIntegerRangeFilter

AskNumberRangeFilter

AskTimeRangeFilter

Note: See note at *AskAlpha AskDate AskInteger AskNumber AskTime* on page 13.

The RangeFilter forms of the Ask command are intended to be used in conjunction with a \$FILTER parameter (see page 49) on a Run File command line that requests a chart function. They are available only in the 32-bit version of Quality Analyst. (For the 16-bit version, use the PROMPT command, which is explained on page 70, with the \$RANGEFILTER parameter.)

This command displays a dialog box that prompts you for two inputs of a specific data type, then translates your entries into a valid “range-filter” argument for the \$FILTER parameter, and assigns the resulting string to a Replaceable Parameter. The syntax of the RangeFilter form of the Ask command is:

```
command param "mainprompt" "prompt1" "prompt2" "default1"
"default2"
```

Command Element	Description
command	the command for the data type being requested: AskAlphaRangeFilter (text string) AskDateRangeFilter AskIntegerRangeFilter AskNumberRangeFilter (real number) AskTimeRangeFilter
param	the name of the Replaceable Parameter (See <i>Replaceable Parameters</i> starting on page 92 for naming requirements.)
mainprompt	the text to display in the dialog box to prompt the user for both inputs
prompt1	the text to display in the dialog box to prompt the user for the first input
prompt2	the text to display in the dialog box to prompt the user for the second input
default1	the value to display in the first input box when the dialog box first appears
default2	the value to display in the second input box when the dialog box first appears

Use double quotation marks as shown in the syntax statement. All quoted elements are optional *from last to first*. That is, you may omit a quoted element *only if* you also omit *all* elements after it. You may, however, use empty quotation marks for any of the quoted elements. In that case, standard text will be used for **prompts**, and **defaults** will be left blank. To suppress any **prompt**, use a quoted space. The **default** values, if specified, will be included in the string assigned to the Replaceable Parameters when the user clicks **Continue** unless they are first deleted or replaced.

If you omit the optional quoted parameters, this command displays the following dialog box (varying slightly for the different data types):

Read more about *Replaceable Parameters* starting on page 92 and about *Data Filters* starting in Chapter 3.

Example: The first of the following two Run File statements displays the dialog box shown below.

```
AskDateRangeFilter ^DateFilter "Enter the date range to
investigate:" "Start of investigation period:" "End of
investigation period:" "10/1/1999" "10/31/1999"
XRS FLUID.DAT VISC X S G $FILTER="DATE ^DateFilter"
```

If the user accepts the default values “10/1/1999” and “10/31/1999” as shown, then the string “L 10/1/1999 U 10/31/1999” is assigned to the Replaceable Parameter ^DateFilter and the second Run File command is executed as:

```
XRS FLUID.DAT VISC X S G $FILTER="DATE L 10/1/1999 U
10/31/1999"
```

This command creates X-bar and Standard Deviation charts of the variable VISC in the file FLUID.DAT for all records where the value of the variable DATE is on or between “10/1/1999” and “10/31/1999.”

Quality Analyst will make sure that the inputs meet the requirements for the DATE data type and that the first date is before (or the same as) the second.

AskAlphaSelectFilter

AskDateSelectFilter

AskIntegerSelectFilter

AskNumberSelectFilter

AskTimeSelectFilter

Note: See note at *AskAlpha AskDate AskInteger AskNumber AskTime* on page 13.

The SelectFilter forms of the Ask command are intended to be used in conjunction with a \$FILTER parameter (see page 49) on a Run File command line that requests a chart function. They are available only in the 32-bit version of Quality Analyst. (For the 16-bit version, use the PROMPT command, which is explained on page 70, with the \$ASKFILTER parameter.)

This command displays a dialog box that prompts you for either one input (to use as the criterion for a match filter) or two inputs (to use as the criteria for a range filter), translates your entries into a valid argument for the \$FILTER parameter, and assigns the resulting string to a Replaceable Parameter. The syntax of the SelectFilter form of the Ask command is:

```
command param "mainprompt" "prompt1" "prompt2" "prompt3"
             "default1" "default2" "default3"
```

Command Element	Description
<code>command</code>	the command for the data type being requested: AskAlphaSelectFilter (text string) AskDateSelectFilter AskIntegerSelectFilter AskNumberSelectFilter (real number) AskTimeSelectFilter
<code>param</code>	the name of the Replaceable Parameter (See <i>Replaceable Parameters</i> starting on page 92 for naming requirements.)
<code>mainprompt</code>	the text to display in the dialog box to prompt the user for all inputs
<code>prompt1</code>	the text to display in the dialog box to prompt the user for the match input

prompt2	the text to display in the dialog box to prompt the user for the first range input
prompt3	the text to display in the dialog box to prompt the user for the second range input
default1	the value to display in the match input box when the dialog box first appears
default2	the value to display in the first range input box when the dialog box first appears
default3	the value to display in the second range input box when the dialog box first appears

Use double quotation marks as shown in the syntax statement. All quoted elements are optional *from last to first*. That is, you may omit a quoted element *only if* you also omit *all* elements after it. You may, however, use empty quotation marks for any of the quoted elements. In that case, standard text will be used for **prompts**, and **defaults** will be left blank. To suppress any **prompt**, use a quoted space. The **default** values, if specified, will be included in the string assigned to the Replaceable Parameters when the user clicks **Continue** unless they are first deleted or replaced.

If you omit the optional quoted parameters, this command displays the following dialog box (varying slightly for the different data types):

Read more about **Replaceable Parameters** starting on page 92 and about **Data Filters** starting in Chapter 3.

Example: The first of the following two Run File statements displays the dialog box shown below.

```

AskDateSelectFilter ^DateFilter "Specify the Date(s) to
investigate:" "Day to investigate:" "Start of
investigation period:" "End of investigation period:"
"10/15/1999" "10/1/1999" "10/31/1999"
XRS FLUID.DAT VISC X S G $FILTER="DATE ^DateFilter"

```

If the user selects **From** and accepts the default values “10/1/1999” and “10/31/1999” as shown, then the string “L 10/1/1999 U 10/31/1999” is assigned to the Replaceable Parameter ^DateFilter and the second Run File command is executed as:

```

XRS FLUID.DAT VISC X S G $FILTER="DATE L 10/1/1999 U
10/31/1999"

```

This command creates X-bar and Standard Deviation charts of the variable VISC in the file FLUID.DAT for all records where the value of the variable DATE is on or between “10/1/1999” and “10/31/1999.”

If the user selects **Match** and accepts the default value “10/15/1999” as shown, then the string “M 10/15/1999” is assigned to the Replaceable Parameter ^DateFilter and the second Run File command is executed as:

```

XRS FLUID.DAT VISC X S G $FILTER="DATE M 10/15/1999"

```

This command creates X-bar and Standard Deviation charts of the variable VISC in the file FLUID.DAT for all records where the value of the variable DATE is “10/15/1999.”

Quality Analyst will make sure that the inputs meet the requirements for the DATE data type and that the first date in a range is before (or the same as) the second.

AskDATFile

The AskDATFile command displays the Quality Analyst file-selection dialog box and assigns the selected file name to a Replaceable Parameter. This command is available only in the 32-bit version of Quality Analyst. The syntax of the AskDATFile command is:

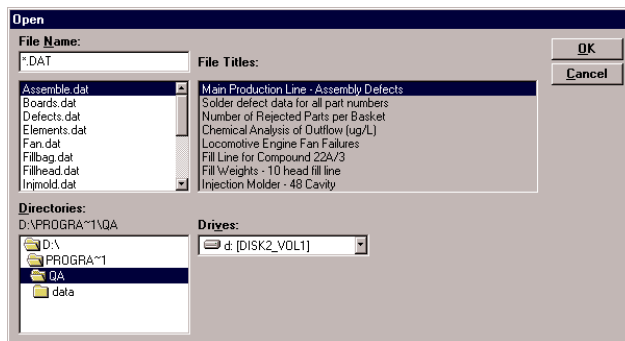
```
AskDATFile param "path" "title"
```

Command Element	Description
param	the name of the Replaceable Parameter that the file name will be assigned to (See <i>Replaceable Parameters</i> starting on page 92 for naming requirements.)
path	the folder (directory) whose files will be listed when the dialog box initially appears
title	the text to appear in the title bar of the dialog box

path and **title** must be enclosed in double quotes.

Use double quotation marks as shown in the syntax statement. Both **path** and **title** are optional, but if you omit **path**, you must also omit **title**. You can use empty quotation marks for **path**, in which case the default Quality Analyst folder will be used.

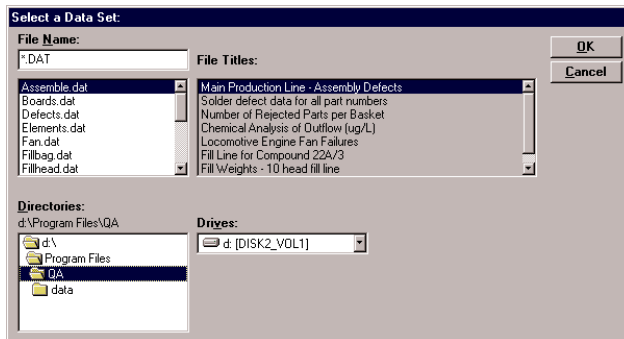
If you omit the optional quoted parameters, this command displays the following dialog box:



Read more about *Replaceable Parameters* starting on page 92.

Example: The following Run File statement displays the dialog box shown below.

```
AskDATFile ^DataFile "d:\program files\qa\" "Select a Data Set:"
```



See Also: AskOpenFile on page 29 and AskSaveFile on page 31.

AskList

The AskList command displays a dialog box that prompts you for a choice from a drop down list of options and assigns that choice to a Replaceable Parameter. This command is available only in the 32-bit version of Quality Analyst. The syntax of the AskList command is:

```
AskList param optlist "prompt" "default"
```

Command Element	Description
param	the name of the Replaceable Parameter that the choice will be assigned to (See <i>Replaceable Parameters</i> starting on page 92 for naming requirements.)
optlist	a list of possible values to assign to the Replaceable Parameter (See instructions below)

prompt	the text to display in the dialog box to prompt the user for input
default	the value from optlist to use as a default value

Use double quotation marks as shown in the syntax statement. Both **prompt** and **default** are optional, but if you omit **prompt**, you must also omit **default**. You can use empty quotation marks for **prompt**, in which case standard text will be used. To suppress the **prompt**, use a quoted space. The **default** value, if specified, will be assigned to the Replaceable Parameter when the user clicks **Continue** unless it is first deleted or replaced.

An **optlist** is a set of possible values for the Replaceable Parameter that will appear in the drop-down list. The list is enclosed in square brackets (“[]”) and the options are enclosed in quotes and separated by fences (“|”). For example, this **optlist**:

```
["ZN" | "PB" | "AS"]
```

would produce this drop-down list:



If the possible values of the Replaceable Parameter are not sufficiently descriptive to be useful to the user, you can include a “friendlier” string to display for each option by adding a colon and a second quoted string to each option. For example, this **optlist**:

```
["ZN": "Zinc" | "PB": "Lead" | "AS": "Arsenic"]
```

would produce this drop-down list:



For any option in the list, the leftmost item will be assigned to the Replaceable Parameter if that option is selected. In the example above, if the user selects “Lead,” “PB” will be assigned to the Replaceable Parameter.

An `optlist` can contain both one-item and two-item options. For example, this `optlist`:

```
[ "ZN" | "PB": "Lead" | "AS" ]
```

would produce this drop-down list:

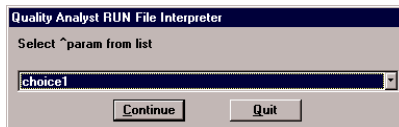


Note that `default`, if included, must be the **rightmost** item of the desired default option. For example, in this `optlist`:

```
[ "ZN" | "PB": "Lead" | "AS" ]
```

“PB” is the only **illegal** item for `default`.

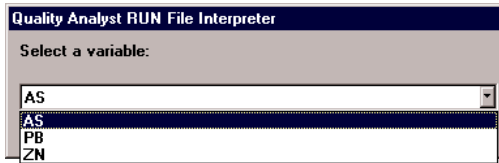
If you omit the optional quoted parameters, this command displays the following dialog box:



Read more about *Replaceable Parameters* starting on page 92.

Example 1: The following Run File statement displays the dialog box shown below.

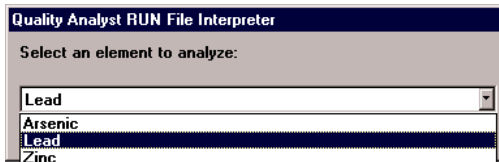
```
AskList ^Var1 ["ZN"|"PB"|"AS"] "Select a variable:" "AS"
```



Notice that because “**AS**” was specified as the **default**, “**AS**” is selected when the dialog box appears.

Example 2: The following Run File command displays the dialog box shown below.

```
AskList ^Element ["AS":"Arsenic"|"PB":"Lead"|"ZN":"Zinc"]
           "Select an element to analyze:" "Lead"
```



Notice that “**Lead**” (the rightmost item of the option), not “**PB**” (the leftmost), is correctly specified as the **default**.

AskOpenFile

The AskOpenFile command displays the Windows File Open dialog box and assigns the selected file name to a Replaceable Parameter. This command is available only in the 32-bit version of Quality Analyst. The syntax of the AskOpenFile command is:

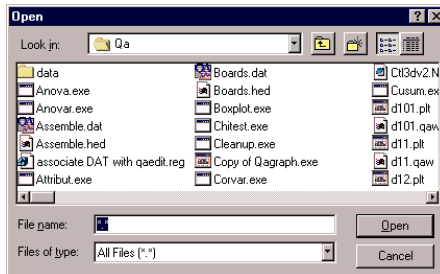
```
AskOpenFile param "type" "path" "title"
```

Command Element	Description
<code>param</code>	the name of the Replaceable Parameter that the file name will be assigned to (See <i>Replaceable Parameters</i> starting on page 92 for naming requirements.)

type	the extension of the file type you want listed when the dialog box initially appears
path	the folder (directory) whose files will be listed when the dialog box initially appears
title	the text to appear in the title bar of the dialog box

Use double quotation marks as shown in the syntax statement. All quoted elements are optional *from last to first*. That is, you may omit a quoted element *only if* you also omit *all* elements after it. You may, however, use empty quotation marks for any of the quoted elements. Empty quotation marks for **type** will cause all file types to be listed. Empty quotation marks for **path** will cause the contents of the default Quality Analyst folder to be listed.

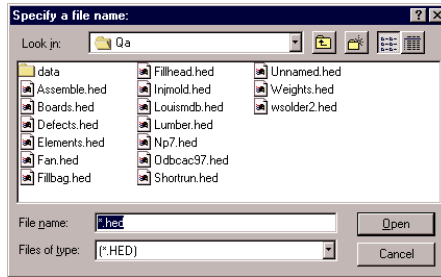
If you omit the optional quoted parameters, this command displays the following dialog box:



Read more about *Replaceable Parameters* starting on page 92.

Example: The following Run File statement displays the dialog box shown below.

```
AskOpenFile ^FileName "HED" "d:\program files\qa\" "Specify a
file name:"
```



See Also: AskDATFile on page 25 and AskSaveFile below.

AskSaveFile

The AskSaveFile command displays the Windows File Save dialog box and assigns the selected file name to a Replaceable Parameter. This command is available only in the 32-bit version of Quality Analyst. The syntax of the AskSaveFile command is:

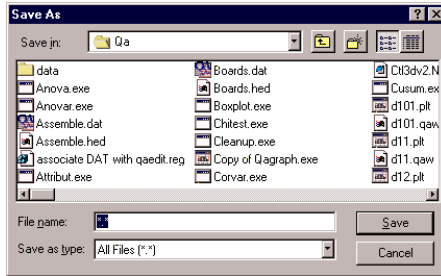
AskSaveFile param "type" "path" "title"

Command Element	Description
param	the name of the Replaceable Parameter that the file name will be assigned to (See <i>Replaceable Parameters</i> starting on page 92 for naming requirements.)
type	the extension of the file type you want listed when the dialog box initially appears
path	the folder (directory) whose files will be listed when the dialog box initially appears
title	the text to appear in the title bar of the dialog box

Use double quotation marks as shown in the syntax statement. All quoted elements are optional *from last to first*. That is, you may omit a quoted element *only if* you also omit *all* elements after it. You may, however, use empty quotation marks for any of the quoted elements. Empty quotation marks for

type will cause all file types to be listed. Empty quotation marks for **path** will cause the contents of the default Quality Analyst folder to be listed.

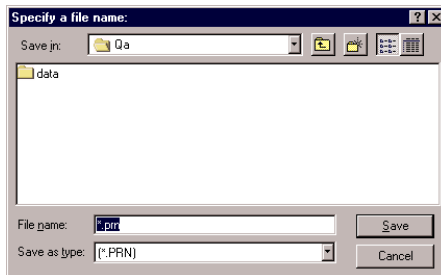
If you omit the optional quoted parameters, this command displays the following dialog box:



Read more about *Replaceable Parameters* starting on page 92.

Example: The following Run File statement displays the dialog box shown below.

```
AskSaveFile ^FileName "PRN" "d:\program files\qa\" "Specify a
file name:"
```



See also: AskDATFile on page 25 and AskOpenFile on page 29.

AskVariable

The AskVariable command displays a dialog box that prompts you to select a variable name from a list and assigns that variable name to a Replaceable Parameter. This command is available only in the 32-bit version of Quality Analyst. The syntax of the AskVariable command is:

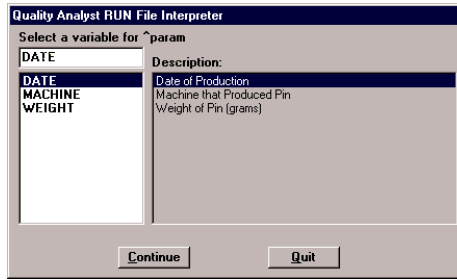
```
AskVariable param "filename" "type" "prompt" "COLUMN"
```

Command Element	Description
param	the name of the Replaceable Parameter that the choice will be assigned to (See <i>Replaceable Parameters</i> starting on page 92 for naming requirements.)
filename	the name of the data file (.DAT) whose variables should be listed in the list
type	a string of characters, each of which represents a type of data that you want to include in the drop-down list. Include a space in this string to include subgroup variables.
prompt	the text to display in the dialog box to prompt the user for input
COLUMN	include "COLUMN" if you want the individual columns of subgroup variables listed separately.

Use double quotation marks as shown in the syntax statement. Except for **filename**, all quoted elements are optional *from last to first*. That is, you may omit a quoted element *only if* you also omit *all* elements after it. You may, however, use empty quotation marks for **type** or **prompt**. Empty quotation marks for **type** will cause all variable types to be listed. Empty quotation marks for **prompt** will cause a standard prompt to be displayed. To suppress the **prompt**, use a quoted space.

The **type** string may include any character in the **Type** column of the table of Data Types in Chapter 3, except that instead of a number (for subgroup data), use a space. **"COLUMN"** has no effect unless **type** contains a space.

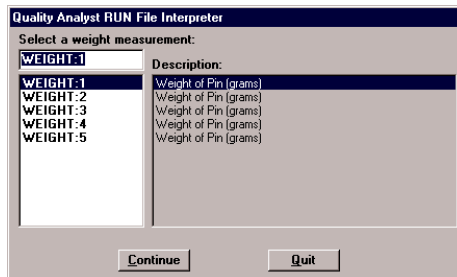
If you omit the optional quoted parameters and specify WEIGHTS for **filename**, this command displays the following dialog box:



Read more about *Replaceable Parameters* starting on page 92.

Example 1: The following Run File statement displays the dialog box shown below.

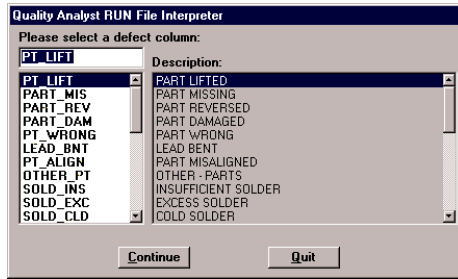
```
AskVariable ^Var1 "WEIGHTS.DAT" " " "Select a weight
measurement:" "COLUMN"
```



Notice that only subgroup variables are listed because **type** is specified as " " (a quoted space) in the statement.

Example 2: The following Run File statement displays the dialog box shown below.

```
AskVariable ^Var1 "BOARDS.DAT" "P" "Please select a defect
column:"
```



Notice that only variables of data type “P” appear in the list.

ATTRIBUT

The ATTRIBUT command creates a graphics (PLT) file containing an attribute control chart. The syntax of the ATTRIBUT command line in a Run File is:

```
ATTRIBUT ininfile var chart output [outfile]
```

Command Element	Description
<code>infile</code>	the name of the input file to use
<code>var</code>	the name of the variable in the input file to use
<code>chart</code>	the type of chart to create: “P” for p-chart (percent defective) “N” for np-chart (number defective) “U” for u-chart (nonconformities per unit) “C” for c-chart (nonconformities)
<code>output</code>	“D” to create a summary (SMY) file, “G” to create a graphics (PLT) file.
<code>outfile</code>	the name of the output file, if applicable

Example: The command line to create a p-chart for the variable THREADS in the data file NOZZLE.DAT and produce the graphics file THREADS.PLT is:

```
ATTRIBUT NOZZLE.DAT THREADS P G THREADS.PLT
```

The “**G**” option creates a graphics (PLT) file but does *not* display or print the graphic. This is accomplished by an additional command line containing either a “**CHART**” or a “**PRINT**” command.

BOXPLOT

The BOXPLOT command creates a graphics (PLT) file containing a box plot. The input file for this command is an intermediate file created by an MPCAP command.

Note: You *must* precede a BOXPLOT command with an MPCAP command (page 58) in your Run File.

The syntax of the BOXPLOT command in a Run File is:

```
BOXPLOT inpfile statlist type units specs output [outfile]
```

Command Element	Description
inpfile	the name of an intermediate file created by the MPCAP command
statlist	optional list of statistics, separated by spaces, to display on the plot box (for example, “4 3 6 11”) See table of statistic codes below.
type	the type of box plot to create: “ B ” for box & whiskers “ C ” for capability box plot “ S ” for ± 3 sigma box plot
units	the units for scaling plotted values: “ M ” for measurement units “ S ” for standardized units
specs	“ S ” to display specifications and target “ N ” to suppress them
output	“ G ” to create a graphics (PLT) file.
outfile	the name of the output file, if applicable

Example 1: To create a graphics file named BX1.PLT containing a box-and-whiskers plot without specifications using measurement units based on the intermediate file DXSHIFT.TMP, the command line is:

```
BOXPLOT DXSHIFT.TMP B M N G BX1.PLT
```

Example 2: To create a graphics file named BX2.PLT containing a capability box plot with specifications using standardized units and showing the Cpk statistic based on the intermediate file DXSHIFT.TMP, the command line is:

```
BOXPLOT DXSHIFT.TMP 38 C S S G BX2.PLT
```

The “G” option creates a graphics (PLT) file but does *not* display or print the graphic. This is accomplished by an additional command line containing either a “CHART” or a “PRINT” command.

Statistics Codes for “statlist” Parameter of BOXPLOT and MCAPREP Commands in Alphabetical Order

Code	Statistic	Code	Statistic	Code	Statistic
35	Actual Above USL	53	Cpm Confidence Limit	17	Minimum
34	Actual Below LSL	43	CPU	8	Number of Samples
36	Actual Outside Specifications	41	CR%	0	Page Break
6	Breakdown	7	Distribution	50	Pp
5	Breakdown Type	32	Est. Above USL	56	Pp Confidence Limit
4	Breakdown Variable	31	Est. Below LSL	49	Ppk
14	Capability Distribution	33	Est. Outside Specifications	57	Ppk Confidence Limit
13	Capability Sigma	1	File Title	11	Skewness
61	CAMcp	20	First Quartile	10	Standard Deviation
62	CAMcpk	26	Geary's	28	Target
58	Confidence Level	27	Geary's significance	21	Third Quartile
40	Cp	59	In The Box	60	Tz
55	Cp (aiag) Confidence Limit	39	k	23	Upper Adjacent
48	Cp aiag	12	Kurtosis	16	Upper Capability
52	Cp Confidence Limit	22	Lower Adjacent	25	Upper Outliers
38	Cpk	15	Lower Capability	30	USL
54	Cpk (aiag) Confidence Limit	24	Lower Outliers	3	Variable Description
47	Cpk aiag	29	LSL	2	Variable Name
51	Cpk Confidence Limit	18	Maximum	46	Zlsl
42	CPL	9	Mean	44	Zmin
37	Cpm	19	Median	45	Zusl

Note: A numerically ordered version of this table is on page 56.

CALC

The CALC command calculates calculated variables, if any, in a Data Set. The syntax for the CALC command is:

CALC filename.dat

If your Data Set contains calculated variables *and* contains data that was added to the Data Set outside the Quality Analyst Editor (for example, by other data-collection software), you should include a CALC command in your Run File *above* any charting or analysis commands.

Example: To create an Individual and Range chart for a variable in a Data Set that contains a calculated variable, the Run File command lines would be:

```
CALC VENEER.DAT
IR VENEER.DAT THICK I R G
```

CHART

The CHART command displays an existing graphics (PLT) file on the screen. To use it, place a CHART command line after one or more other command lines containing Quality Analyst functions that create PLT files. When used with the GROUP command, CHART produces a multichart page layout for display. The CHART command can also be used to create simultaneous on-screen and printed output.

CHART Command Option	Description
GROUP <i>style</i>	Instructs CHART to form a multichart layout in the specified style . Style is followed by a list of PLT files to include in the Group Layout. These PLT files must all have been produced before this command is executed. See GROUP on page 51, for examples and a list of layout styles.
NONE	Used in place of a PLT file name following a GROUP command to leave part of a Group Layout blank.
/CLOSE	Closes the graphics display immediately.
/DELAY <i>seconds</i>	Displays the chart for the specified number of seconds before continuing the execution of the Run File. Adding “HIDE” after [seconds] will suppress the display of the DELAY message box, as discussed under <i>Executing Run Files</i> on page 102.
/PRINT [LANDSCAPE]	Causes the displayed graphics to be printed. The printed graphics will be in portrait orientation unless “LANDSCAPE” is included.
/CONTINUE	Continues execution of the Run File without displaying the Continue/Quit prompt.

Example 1: Display a graph generated by the XRS command—in this case, an X-bar and Range chart from the variable WEIGHT in the file TEST.DAT:

```
XRS TEST.DAT WEIGHT X R G
CHART
```

Note that the default graphics file name (GRAPHICS.PLT) need not be specified if the CHART command follows immediately and only one chart is to be displayed.

Example 2: Create the graphics file CAPX.PLT, which contains a Process Capability histogram of the variable WEIGHT in the file TEST.DAT, and display it for 15 seconds:

```
PCAP TEST.DAT WEIGHT G CAPX.PLT
CHART CAPX.PLT /DELAY 15
```

Example 3: Create a 2×2 group layout of four p-charts from the variables BUBBLES, MARKS, CRACKS, and TRIM in the file REJECTS.DAT. Note the use of the four individual graphics (PLT) files later combined in the CHART command:

```
ATTRIBUT REJECTS.DAT BUBBLES P G REJ1.PLT
ATTRIBUT REJECTS.DAT MARKS P G REJ2.PLT
ATTRIBUT REJECTS.DAT CRACKS P G REJ3.PLT
ATTRIBUT REJECTS.DAT TRIM P G REJ4.PLT
CHART GROUP 2V2H REJ1.PLT REJ2.PLT REJ3.PLT REJ4.PLT
```

Example 4: To also print the group layout from Example 3, modify the last line:

```
CHART GROUP 2V2H REJ1.PLT ... REJ4.PLT /PRINT
```

To print the group layout in landscape orientation, modify the last line:

```
CHART GROUP 2V2H REJ1.PLT ... REJ4.PLT /PRINT LANDSCAPE
```

CHITEST

The CHITEST command calculates one of four types of Chi-square statistics, each of which is appropriate for a particular type of data. Refer to the section titled *Chi-square* in Chapter 8 for more details about Chi-square statistics.

The syntax of the CHITEST command line to calculate **Bartlett's** Chi-square statistic in a Run File is:

```
CHITEST infile B sampvars degfree output outfile
```

The syntax of the CHITEST command line to calculate the **Difference Among Proportions** Chi-square statistic in a Run File is:

```
CHITEST infile D posresp sampsize output outfile
```

The syntax of the CHITEST command line to calculate the **Equal Expected Frequencies** Chi-square statistic in a Run File is:

```
CHITEST infile E obsfreq output outfile
```

The syntax of the CHITEST command line to calculate the **Unequal Expected Frequencies** Chi-square statistic in a Run File is:

```
CHITEST infile U obsfreq expfreq output outfile
```

Command Element	Description
<code>infile</code>	the name of the input file to use
<code>sampvars</code>	the name of the variable to use for the sample variances
<code>degfree</code>	the name of the variable to use for the degrees of freedom
<code>posresp</code>	the name of the variable to use for the positive responses
<code>sampsize</code>	the name of the variable to use for the sample size
<code>obsfreq</code>	the name of the variable to use for observed frequencies
<code>expfreq</code>	the name of the variable to use for expected frequencies
<code>output</code>	“W” to create a formatted text (PTX) file.
<code>outfile</code>	the name of the output file

Example 1: The command line to calculate Bartlett’s Chi-square statistic where, in the file CHECK.DAT, the variable VARS contains sample variances and the variable DFX contains degrees of freedom, is:

```
CHITEST CHECK.DAT B VARS DFX W DEFAULT.PTX
```

Example 2: The command line to calculate the Difference Among Proportions Chi-square statistic, where, in the file DMAIL.DAT, the variable RETURN contains the number of positive responses and the variable SENT contains the sample size, is:

```
CHITEST DMAIL.DAT D RETURN SENT W DEFAULT.PTX
```

Example 3: The command line to calculate the Equal Expected Frequencies Chi-square statistic where, in the file BINCHK.DAT, the variable COUNT contains observed frequencies, is:

```
CHITEST BINCHK.DAT E COUNT W DEFAULT.PTX
```

Example 4: The command line to calculate the Unequal Expected Frequencies Chi-square statistic where, in the file DISTX.DAT, the variable MEAS contains observed frequencies and the variable THEORY contains expected frequencies, is:

```
CHITEST DISTX.DAT U MEAS THEORY W DEFAULT.PTX
```

The “w” option creates a formatted text (PTX) file but does *not* display or print it. This is accomplished by an additional command line containing either a “CHART” or a “PRINT” command.

CLEANUP

With the CLEANUP Run File command, you can convert files from external sources (other software or systems) to Quality Analyst format or vice versa.

External to Quality Analyst—E

To convert a comma-delimited file (or any of several other file formats) into the space-delimited format required by Quality Analyst, the syntax of the Run File command line is:

```
CLEANUP E infile charlist numcols outfile
```

Command Parameter	Description
<code>infile</code>	the name of the input file
<code>charlist</code>	list the characters (without spaces) to be removed, or use “N” for none
<code>numcols</code>	the number of columns to check for, or “U” to use the number of columns in the first row of the data file, or

	“N” to perform no column check
outfile	the name of the output (DAT) file

The parameters **charlist** and **numcols** are explained in more detail under *Importing a Comma-Separated-Value File* in Chapter 10.

Example: To convert the external file COMM6.DTX into the Quality Analyst file CTEST.DAT, removing no characters, using the first row to determine the number of columns, the Run File command line would be:

```
CLEANUP E COMM6.DTX N U CTEST.DAT
```

Fixed-Field to Quality Analyst—F

To convert a fixed-field file into the space-delimited format required by Quality-Analyst, the syntax of the Run File command line is:

```
CLEANUP F inpfile ffdfile outfile
```

Command Parameter	Description
inpfile	the name of the input file
ffdfile	the name of the fixed-field-definition file
outfile	the name of the output file

Before performing a fixed-field conversion from within a Run File, you must create a fixed-field definition file, as explained under *Importing a Fixed-Field File* in Chapter 10.

Quality Analyst to Comma-Delimited—S

To convert a Quality Analyst data file into a comma-delimited file for use by other software, the syntax of the Run File command line is:

```
CLEANUP S inpfile outfile
```

Command Parameter	Description
<code>infile</code>	the name of the input file
<code>outfile</code>	the name of the output file

Quality Analyst to Comma-Delimited with Quotes—Q

To convert a Quality Analyst data file into a comma-delimited file with quotation marks around each text field, the syntax of the Run File command line is:

```
CLEANUP Q infile outfile
```

Command Parameter	Description
<code>infile</code>	the name of the input file
<code>outfile</code>	the name of the output file

Change Missing-Data Symbol—C

To replace the missing-data symbol in a Quality Analyst data file with another character or string of text, the syntax of the Run File command line is:

```
CLEANUP C infile old new
```

Command Parameter	Description
<code>infile</code>	the name of the input file
<code>old</code>	the old missing-data symbol
<code>new</code>	the new missing-data symbol

CLEAR

The CLEAR command clears, or “unassigns,” a Replaceable Parameter. This command is available only in the 32-bit version of Quality Analyst. The syntax of the CLEAR command is:

```
CLEAR param
```

Command Element	Description
param	the name of the Replaceable Parameter to clear

Example: To clear the Replaceable Parameter `^FileName`, the Run File statement would be:

```
CLEAR ^FileName
```

Read more about *Replaceable Parameters* starting on page 92.

\$COMMENT

You can add a line of text to the header lines of a chart by including a \$COMMENT command element and specifying the text as part of the chart command. This is particularly useful in describing any filters that have been applied, since Quality Analyst does not do this automatically in Run Files. The syntax is:

```
$COMMENT= "comment text"
```

The \$COMMENT and \$FILTER commands must come *after* all other parameters of the main command, but they can occur in any order relative to each other.

Example: To add the comment “Lot Number 2250 Only” to a chart command that includes a filter specification, include this line in your Run File:

```
IR FLUID.DAT VISC I R G $FILTER="LOT M 2250" $COMMENT="Lot
Number 2250 Only"
```

CONNECT

The CONNECT command is used by the Database Connectivity version of Quality Analyst to establish a connection to an external database. See *Using Database Connectivity in Run Files* on page 147 for details.

The syntax for the CONNECT command is:

```
CONNECT filename.dat
```

CORVAR

The CORVAR command computes the linear correlation coefficients for two or more variables and produces a correlation matrix for all possible combinations of the selected variables. The syntax of the CORVAR command line in a Run File is:

```
CORVAR infile vars missing output outfile
```

Command Element	Description
<code>infile</code>	the name of the input file to use
<code>vars</code>	the names of two or more variables (separated by spaces) in the input file that you want to analyze
<code>missing</code>	“C” to treat missing data casewise, or “P” to treat missing data pairwise
<code>output</code>	“D” to create a data (DAT) file, or “W” to create a formatted text (PTX) file.
<code>outfile</code>	the name of the output file

Refer to the section titled *Correlation* in Chapter 8 for a detailed explanation of casewise and pairwise treatment of missing data.

Example: The command line to compute the correlation coefficients for the variables PB, CD, ZN, HG, CR, MO, MN, and AS in the file ELEMENTS.DAT, treating missing data casewise, and storing the results in a formatted text file is:

```
CORVAR ELEMENTS.DAT PB CD ZN HG CR MO MN AS C W DEFAULT.PTX
```

The “W” option creates a formatted text (PTX) file but does *not* display or print it. This is accomplished by an additional command line containing either a “CHART” or a “PRINT” command.

CUSUM

The CUSUM command creates a graphics (PLT) file that contains a Cumulative Sum chart. The syntax of the CUSUM command line in a Run File is:

```
CUSUM inpfile var vgroup output [outfile]
```

Command Element	Description
<code>inpfile</code>	the name of the input file to use
<code>var</code>	the name of the variable in the input file to use
<code>vgroup</code>	subgroup for V-mask: “1, 2, 3,…” for ordinal subgroup number “L” for last “N” for none
<code>output</code>	“D” to create a summary (SMY) file, “G” to create a graphics (PLT) file.
<code>outfile</code>	the name of the output file, if applicable

Example: The command line to chart the variable THICK from the data file VENEER.DAT and produce the graphics file VENEER.PLT with the V-mask located on the 22nd subgroup is:

```
CUSUM VENEER.DAT THICK 22 G VENEER.PLT
```

With Quality Analyst, you can produce a series of CUSUM charts with the V-mask at different positions by creating a Run File containing multiple CUSUM commands with the same variable.

The “G” option creates a graphics (PLT) file but does *not* display or print the graphic. This is accomplished by an additional command line containing either a “CHART” or a “PRINT” command.

EWMA

The EWMA command creates a graphics (PLT) file that contains an Exponentially Weighted Moving Average chart (optionally in combination

with a Range or Standard Deviation chart). The syntax of the EWMA command line in a Run File is:

```
EWMA infile var chart1 chart2 output [outfile]
```

Command Element	Description
<code>infile</code>	the name of the input file
<code>var</code>	the name of the variable in the input file to use
<code>chart1</code>	the type of the chart: “E” for EWMA chart “R” for Range chart “S” for Standard Deviation chart
<code>chart2</code>	the type of the second chart of a pair: “E” for EWMA chart “R” for Range chart “S” for Standard Deviation chart “N” for none
<code>output</code>	“D” to create a summary (SMY) file, “G” to create a graphics (PLT) file.
<code>outfile</code>	the name of the output file

Note: `chart1` and `chart2` must be different.

Example 1: To create an EWMA chart for the variable THICK in the file VENEER.DAT, and produce a graphics file named VENEER.PLT for later use by the PRINT or CHART commands, the Run File command line would be:

```
EWMA VENEER.DAT THICK E N G VENEER.PLT
```

Note the mandatory “N” (for “none”) for the nonexistent second chart type.

Example 2: To create an EWMA and Range chart for the same variable, and produce a graphics file for later use, the Run File command line would be:

```
EWMA VENEER.DAT THICK E R G
```

In this example, the absence of the `outfile` parameter tells Quality Analyst to create a graphics file with the default name GRAPHICS.PLT.

The “G” option creates a graphics (PLT) file but does *not* display or print the graphic. This is accomplished by an additional command line containing either a “CHART” or a “PRINT” command.

EXTRACT

With the EXTRACT Run File command, you can extract subsets from data files and create a new file containing the selected data. The syntax for the Run File command line is:

```
EXTRACT infile selfile outfile
```

Command Parameter	Description
infile	the name of the input file
selfile	the name of the extract definition file (see below)
outfile	the name of the output file

Before executing the EXTRACT Run File command, you must create an extract definition (SEL) file containing the information needed by the EXTRACT command. Creating this file is explained under *Extracting Subsets from a Data File* in Chapter 11.

Example: To generate a file named LTNEW.DAT from the data file LTEX.DAT, using the extract definition file LTEX.SEL, the Run File command line would be:

```
EXTRACT LTEX.DAT LTEX.SEL LTNEW.DAT
```

\$FILTER

The Data Filter feature of Quality Analyst can be used in Run Files to select subsets of data. To do this, place the \$FILTER command parameter at the end of a line in the Run File that requests a chart function. You can include up to three filters in the same chart command. The syntax is:

```
$FILTER="var type value [type value]" [$FILTER="..."]  
[$FILTER="..."]
```

Command Element	Description
var	the name of the variable in the input file to use
type	the type of filter criterion: “L” means lower limit “M” means match “U” means upper limit
value	the value to be matched or compared with

With the filter criterion, you can select data that either match a value (with “M”) or fall within a range (with “L” and/or “U”). Only one boundary of a range need be declared. Normally, the boundary values are included in the range. To exclude the boundary values from the range, put a “>” in front of the lower limit and a “<” in front of the upper limit.

In a single \$FILTER command, the second **type** and **value** are allowable only when you are specifying a *second* limit for values of the variable—that is, an upper limit in addition to a lower limit, or vice versa.

The \$FILTER and \$COMMENT commands must come *after* all other parameters of the main command, but they can occur in any order relative to each other.

Example 1: To request an Individual-and-Range chart for the variable VISC in the file FLUID.DAT for Lot Number 2250 only (variable name LOT), include this line in your Run File:

```
IR FLUID.DAT VISC I R G $FILTER="LOT M 2250"
```

Example 2: To request an X-bar and Standard Deviation chart for the variable VISC in the file FLUID.DAT using only data from July 1995 (using the DATE variable) where the value of the variable PH is greater than 7, include this line in your Run File:

```
XRS FLUID.DAT VISC X S G $FILTER="DATE L 7/1/95 U 7/31/95"  
$FILTER="PH L >7.0"
```

Without the “>” in the example above, the analysis would include data where the value of PH is greater than *or equal to* 7.0.

Note: To prompt users for filter criteria, use any of the Run File commands that begin with “Ask” and end with “Filter” such as AskDateRangeFilter.

FLIP

With the FLIP Run File command, you can rotate data files 90 degrees, turning rows into columns and columns into rows. To rotate rows, the syntax of the Run File command line is:

```
FLIP infile R rows D outfile
```

To rotate columns, the syntax of the Run File command line is:

```
FLIP infile C cols D outfile
```

Command Parameter	Description
infile	the name of the Quality Analyst input file
rows	a list of row numbers, separated by spaces, or “ A ” to rotate all rows
cols	a list of column names, separated by spaces, or “ A ” to rotate all columns
outfile	the name of the output file

Example: To rotate the entire file TEST1.DAT by rows, producing TEST2.DAT, the Run File command line would be:

```
FLIP TEST1.DAT R A D TEST2.DAT
```

GROUP














The GROUP command is really an optional parameter to the CHART and PRINT commands that instructs the Run File Interpreter to form a multichart Group Layout in a specified arrangement or “style.” The syntax of the GROUP parameter in a Run File command line is:

```
GROUP style filelist
```

Command Parameter	Description
style	the style code for the desired layout (see table below)
filelist	a list of graphics (PLT) files (separated by spaces), one for each pane in the group layout. To leave a pane blank, use “ NONE ” in place of the file name.

The following table lists the available layout styles and their command-line codes.

GROUP Command Option Style Codes

				
2V	3V	4V	5V	
				
2H	2V2H	3V2H	3V3H	4V3H
				
2VST	2VSB	2HSL	2HSR	

The section **Group Layout** in Chapter 6 discusses creating Group Layouts interactively with the Graphics Viewer.

Example: To print two smaller charts (SAMPLE1.PLT and SAMPLE2.PLT) over a third full-width chart (SAMPLE3.PLT), use style code “2VST”:

```
PRINT GROUP 2VST SAMPLE1.PLT SAMPLE2.PLT SAMPLE3.PLT
```

Note that the GROUP command option and the style code follow the CHART or PRINT command immediately and come before the file names. The charts from the specified files are placed in the specified layout left-to-right then top-to-bottom.

IR

The IR command creates a graphics (PLT) file containing an Individuals (and/or a Range) chart. The syntax of the IR command line in a Run File is:

```
IR infile var chart1 chart2 output [outfile]
```

Command Element	Description
<code>infile</code>	the name of the input file
<code>var</code>	the name of the variable in the input file to use
<code>chart1</code>	the type of the chart: “ I ” for Individual chart “ R ” for Range chart
<code>chart2</code>	the type of the second chart of a pair: “ I ” for Individual chart “ R ” for Range chart “ N ” for none
<code>output</code>	“ D ” to create a summary (SMY) file, “ G ” to create a graphics (PLT) file.
<code>[outfile]</code>	the name of the output file

Note: `chart1` and `chart2` must be different.

Example 1: To create an Individual chart for the variable THICK in the file VENEER.DAT, and produce a graphics file named VENEER.PLT for later use by the PRINT or CHART commands, the Run File command line would be:

```
IR VENEER.DAT THICK I N G VENEER.PLT
```

Note the mandatory “**N**” (for “none”) for the nonexistent second chart type.

Example 2: To create an Individual and Range chart for the same variable, and produce a graphics file for later use, the Run File command line would be:

```
IR VENEER.DAT THICK I R G
```

In this example, the absence of the `outfile` parameter tells Quality Analyst to create a graphics file with the default name GRAPHICS.PLT.

The “**G**” option creates a graphics (PLT) file but does *not* display or print the graphic. This is accomplished by an additional command line containing either a “**CHART**” or a “**PRINT**” command.

KRUSKAL

The KRUSKAL command performs a Kruskal-Wallis test, which is a nonparametric analysis of variance for populations where the assumptions required by one-way ANOVA are not met. The syntax of the KRUSKAL command line in a Run File is:

```
KRUSKAL inpfile vars output outfile
```

Command Element	Description
inpfile	the name of the input file to use
vars	the names of two or more variables (separated by spaces) in the input file that you want to analyze
output	“ W ” to create a formatted text (PTX) file
outfile	the name of the output file

Example: The command line to perform the Kruskal-Wallis test on the variables SINKS, MOUNT, MARKS, TRIM, and EDGE in the file DEFECTS.DAT is:

```
KRUSKAL DEFECTS.DAT SINKS MOUNT MARKS TRIM EDGE W DEFAULT.PTX
```

The “**w**” option creates a formatted text (PTX) file but does *not* display or print it. This is accomplished by an additional command line containing either a “**CHART**” or a “**PRINT**” command.

MCAPREP

The MCAPREP command creates a formatted text (PTX) file containing a process capability report. The input file for this command is an intermediate file created by an MPCAP command.

Note: You *must* precede a MCAPREP command with an MPCAP command (page 58) in your Run File.

The syntax of the MCAPREP command in a Quality Analyst Run File is:

```
MCAPREP inpfile statlist output outfile
```

Command Element	Description
<code>inpfile</code>	the name of an intermediate file created by the MPCAP command
<code>statlist</code>	list of statistics, separated by spaces, to display on the report (for example, "4 3 6 11") See table of statistic codes below.
<code>output</code>	"W" to create a formatted text (PTX) file.
<code>outfile</code>	the name of the output file

Example: To create a formatted text file named CAPREP1.PTX containing a capability report from the intermediate file DXSHIFT.TMP and that includes the mean, estimated above and below specifications, Cpk, and Ppk, the command line is:

```
MCAPREP DXSHIFT.TMP 9 31 32 38 49 W CAPREP1.PTX
```

The "W" option creates a formatted text (PTX) file but does *not* display or print it. This is accomplished by an additional command line containing either a "CHART" or a "PRINT" command.

Statistics Codes for “statlist” Parameter of MCAPREP and BOXPLOT Commands in Numerical Order

Code	Statistic	Code	Statistic	Code	Statistic
0	Page Break	21	Third Quartile	42	CPL
1	File Title	22	Lower Adjacent	43	CPU
2	Variable Name	23	Upper Adjacent	44	Zmin
3	Variable Description	24	Lower Outliers	45	Zusl
4	Breakdown Variable	25	Upper Outliers	46	Zlsl
5	Breakdown Type	26	Geary’s	47	Cpk aiag
6	Breakdown	27	Geary’s significance	48	Cp aiag
7	Distribution	28	Target	49	Ppk
8	Number of Samples	29	LSL	50	Pp
9	Mean	30	USL	51	Cpk Confidence Limit
10	Standard Deviation	31	Est. Below LSL	52	Cp Confidence Limit
11	Skewness	32	Est. Above USL	53	Cpm Confidence Limit
12	Kurtosis	33	Est. Outside Specifications	54	Cpk (aiag) Confidence Limit
13	Capability Sigma	34	Actual Below LSL	55	Cp (aiag) Confidence Limit
14	Capability Distribution	35	Actual Above USL	56	Pp Confidence Limit
15	Lower Capability	36	Actual Outside Specifications	57	Ppk Confidence Limit
16	Upper Capability	37	Cpm	58	Confidence Level
17	Minimum	38	Cpk	59	In The Box
18	Maximum	39	k	60	Tz
19	Median	40	Cp	61	CAMcp
20	First Quartile	41	CR%	62	CAMcpk

Note: An alphabetically ordered version of this table is on page 38.

MERGE

With the MERGE Run File command, you can insert, delete, or extract rows or columns from a file. You can also extract or replace an entire block from within a file and merge files in any orientation.

You specify the MERGE operation by including an operation number in the command line with other parameters appropriate for the operation. The following table lists the operation numbers, the operation, and the command syntax for that operation.

Syntax for MERGE Operations in Run Files

Operation Number	Operation	Command Syntax
1	Extract Rows	<code>MERGE infile 1 begrow endrow outfile</code>
2	Extract Columns	<code>MERGE infile 2 begcol endcol outfile</code>
3	Extract a Block	<code>MERGE infile 3 begrow endrow begcol endcol outfile</code>
4	Delete Rows	<code>MERGE infile 4 begrow endrow output [outfile]</code>
5	Delete Columns	<code>MERGE infile 5 begcol endcol output [outfile]</code>
6	Insert Rows	<code>MERGE infile 6 insfile begrow output [outfile]</code>
7	Insert Columns	<code>MERGE infile 7 insfile begcol output [outfile]</code>
8	Replace a Block	<code>MERGE infile 8 replfile begrow begcol output [outfile]</code>

The command parameters and meanings are as follows:

Command Parameter	Description
<code>infile</code>	the name of the input file
<code>begrow</code>	when extracting or deleting rows: the number of the beginning row when inserting rows: the number of the beginning row, or “ B ” for beginning, or “ E ” for end
<code>begcol</code>	when extracting or deleting columns: the number of the beginning column when inserting columns: the number of the beginning column, or “ B ” for beginning, or “ E ” for end

endrow	when extracting or deleting rows: the number of the ending row, or “ L ” for last. when inserting rows: the number of the ending row, or “ B ” for beginning, or “ E ” for end.
endcol	when extracting or deleting columns: the number of the ending column, or “ L ” for last. when inserting columns: the number of the ending column, or “ B ” for beginning, or “ E ” for end.
infile	the name of the insertion file
replfile	the name of the replacement file
output	“ C ” to create a new file (outfile required), or “ O ” to overwrite the original input file.
outfile	the name of the output file, if applicable

Example: To insert the file PART2.DAT at the right side of PART1.DAT (operation 7, “insert columns”), and create a new file named OUT.DAT, the Run File command would be:

```
MERGE PART1.DAT 7 PART2.DAT E C OUT.DAT
```

MPCAP

The MPCAP command prepares an intermediate file for use by the BOXPLOT (page 36) and MCAPREP (page 54) Run File commands. The syntax of the MPCAP command line in a Run File is:

```
MPCAP infile vars bkdntype [bkdncol] [bkdnsort] [bkdnint]
      output outfile
```

Command Element	Description
infile	the name of the input (DAT) file
vars	the names of one or more variables in the input file to use

bkdntype	the type of breakdown to apply to the data: “B” to break down by a data column “D” to break down by a date column “N” for no breakdown
bkdncol	the name of the data column by which to break down the data (omit parameter if no break down)
bkdnsort	“S” to sort by breakdown column “U” to not sort by breakdown column (omit parameter unless bkdntype is B)
bkdnint	interval of a breakdown by date: “DAY” for daily breakdown “WEEKS” for weekly, beginning on Sundays “WEEKM” for weekly, beginning on Mondays “MONTH” for monthly (omit parameter unless bkdntype is D)
output	“A” to append information to the output file “C” to create a new output file
outfile	the name of the output file

The “A” option for **output** lets you execute several MPCAP commands on different data files and include them all in a single report. Be sure to use the “C” option on the first MPCAP command in the series.

Example 1: The following command line creates an intermediate file named DXSHIFT.TMP containing process capability information on the variable DIAMETER in the file DIAM.DAT. The information is broken down and sorted by the variable SHIFT.

```
MPCAP DIAM.DAT DIAMETER B SHIFT S C DXSHIFT.TMP
```

Because of the “C” parameter, if a file named DXSHIFT.TMP existed, it would be overwritten.

Example 2: The following command line creates an intermediate file named DXMONTH.TMP containing process capability information on the variable DIAMETER in the file DIAM.DAT. The information is broken down by the variable DATE and grouped by month..

```
MPCAP DIAM.DAT DIAMETER D DATE MONTH C DXMONTH.TMP
```

MR

The MR command creates a graphics (PLT) file containing a Median or Median/Individual chart (and/or a Range chart). The syntax of the MR command line in a Run File is:

```
MR inpfile var chart1 chart2 output [outfile]
```

Command Element	Description
inpfile	the name of the input file
var	the name of the variable in the input file to use
chart1	the type of the chart: “M” for Median chart “R” for Range chart “MI” for Median/Individual chart
chart2	the type of the second chart of a pair: “M” for Median chart “R” for Range chart “MI” for Median/Individual chart “N” for none
output	“D” to create a summary (SMY) file, “G” to create a graphics (PLT) file.
outfile	the name of the output file, if applicable

Note: **chart1** and **chart2** must be different.

Example 1: To create a Median chart for the variable THICK in the file VENEER.DAT, and produce a graphics file named VENEER.PLT for later use by the PRINT or CHART commands, the Run File command line would be:

```
MR VENEER.DAT THICK M N G VENEER.PLT
```

Note the mandatory “N” (for “none”) for the nonexistent second chart type.

Example 2: To create a Median/Individuals and a Range chart for the same variable, and produce a graphics file for later use, the Run File command line would be:

```
MR VENEER.DAT THICK MI R G
```

In this example, the absence of the `outfile` parameter tells Quality Analyst to create a graphics file with the default name GRAPHICS.PLT.

The “**G**” option creates a graphics (PLT) file but does *not* display or print the graphic. This is accomplished by an additional command line containing either a “**CHART**” or a “**PRINT**” command.

MSORT

With the MSORT (for multikey sort) Run File command, you can sort a data file based on one or more columns of alpha, numeric, time, or date data. You can sort in either ascending or descending order and you can either replace the original data with the sorted data or send the sorted data to a new file. The syntax of the MSORT command line in a Run File is:

```
MSORT infile sortvars sortorder output [outfile]
```

Command Parameter	Description
<code>infile</code>	the name of the input file
<code>sortvars</code>	the names of one or more variables in the input file (separated by spaces) by which to sort the file: primary key first, secondary key second, and so on.
<code>sortorder</code>	“ A ” for ascending, or “ D ” for descending
<code>output</code>	“ C ” to create a new file “ S ” to sort in place (overwriting original input file).
<code>outfile</code>	the name of the output file (required if <code>output</code> is “ C ”)

Example: To generate a new file named LTSORT.DAT from the data file LTEX.DAT, sorted in ascending order by the variable “DATE,” the Run File command line would be:

```
MSORT LTEX.DAT DATE A C LTSORT.DAT
```

MULTREG

With the MULTREG Run File command, you can perform a multiple linear regression. The syntax of the MULTREG command line in a Run File is:

```
MULTREG inpfile depvar indvars confint resid output [outfile]
```

Command Parameter	Description
inpfile	the name of the input file
depvar	the name of the dependent variable
indvars	the names of one or more independent variables
confint	Confidence intervals for graphics output (see below)
resid	“R” to include the residuals, or “N” to skip them.
output	“D” to create a data (DAT) file, “G” to create a graphics (PLT) file, or “W” to create a formatted text (PTX) file.
outfile	the name of the output file, if applicable

For **confint**, specify “C” to include the confidence intervals at 95% based on the individual measurements, “N” for no confidence intervals, or a number from 1 to 99 to specify the probability level percent. If you specify a number, follow it with either “I” to base confidence intervals on the individual measurements, or “E” to base confidence intervals on the equation. You must specify confidence intervals even if you do not select graphics output.

Example: To perform a multiple linear regression on the dependent variable TENSILE and the independent variables MIX, BINDER, MOISTURE, and DWELL in the file REGTEST.DAT, and send output that includes residuals to a graphics file, the Run File command line would be:

```
MULTREG REGTEST.DAT TENSILE MIX BINDER MOISTURE DWELL C R G
```

The “G” and “W” options respectively create a graphics (PLT) file and a formatted text (PTX) file but do *not* display or print them. This is accomplished by an additional command line containing a “CHART” or a “PRINT” command or both.

NEWVAR

With the NEWVAR Run File command, you can create new variables in a data file from calculations on existing data in the file. The syntax of the NEWVAR command line in a Run File is:

```
NEWVAR inpfile vdf file output [outfile]
```

Command Parameter	Description
<code>inpfile</code>	the name of the input file
<code>vdf file</code>	the name of the variable-definition file (see below)
<code>output</code>	“ A ” to add the variable(s) to the existing file, or “ C ” to create a new file containing the new variable(s) ((<code>outfile</code>) required)
<code>(outfile)</code>	the name of the output file, if applicable

Before executing the NEWVAR Run File command, you must create a variable-definition file (VDF) containing the information needed by the NEWVAR command. Creating this file is explained under *Defining Calculations for Run Files* in Chapter 11.

Example: To generate a new file named LTNEW.DAT from the data file LTEX.DAT, using the variable definition file LTEX.VDF, the Run File command line would be:

```
NEWVAR LTEX.DAT LTEX.VDF C LTNEW.DAT
```

NORMPLOT

With the NORMPLOT Run File command, you can create a normal probability plot for a single variable or a quantile plot for two variables.

To create a normal probability plot, the syntax of the Run File command line is:

```
NORMPLOT inpfile N var output [outfile]
```

To create a quantile plot, the syntax of the Run File command line is:

```
NORMPLOT infile Q xvar yvar output [outfile]
```

Command Parameter	Description
<code>infile</code>	the name of the input file
<code>var</code>	the name of the variable in the input file to plot
<code>xvar</code>	the name of an X-axis variable in the input file
<code>yvar</code>	the name of a Y-axis variable in the input file
<code>output</code>	“G” to create a graphics (PLT) file
<code>outfile</code>	the name of the output file, if applicable

Example: To produce a normal probability plot using the variable THICK from the file VENEER.DAT, and create a graphics output file with the default file name, the Run File command line is:

```
NORMPLOT VENEER.DAT N THICK G
```

The “G” option creates a graphics (PLT) file but does *not* display or print the graphic. This is accomplished by an additional command line containing either a “CHART” or a “PRINT” command.

PARETO

The PARETO command creates a graphics (PLT) file containing a Pareto diagram. The PARETO command can accept either a multi-variable file or a single-variable file and allows charts with multiple variables (multiple defects). The syntax of the PARETO command line in a Run File is:

```
PARETO infile format vars output [outfile]
```

Command Parameter	Description
<code>infile</code>	the name of the input file
<code>format</code>	“M” for multiple variables, or “S” for single variable
<code>vars</code>	the name(s) of the variable(s) to plot
<code>output</code>	“D” to create a data (DAT) file,

	“G” to create a graphics (PLT) file.
outfile	the name of the output file, if applicable

Example: To create a Pareto chart from the file NOZZLE.DAT (which contains defect data in multiple variables) for the variables SEAL, THREADS, WELD, and SURFACE, and send output to the graphics file NOZZLE.PLT, the Run File command line would be:

```
PARETO NOZZLE.DAT M SEAL THREADS WELD SURFACE G NOZZLE.PLT
```

The “G” option creates a graphics (PLT) file but does *not* display or print the graphic. This is accomplished by an additional command line containing either a “CHART” or a “PRINT” command.

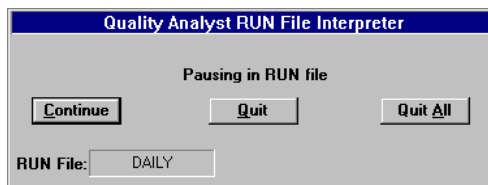
PAUSE

The PAUSE command pauses the Run File. The syntax is:

```
PAUSE [seconds] [message]
```

The PAUSE command may contain either, both, or no parameters.

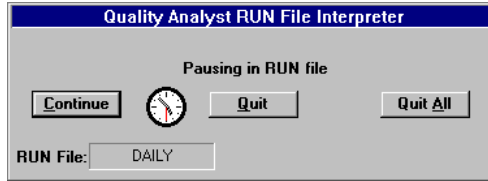
If the PAUSE command contains no parameters, this dialog box will appear when the PAUSE command is executed.



PAUSE dialog box

On this (and all other PAUSE dialog boxes), select **Continue** button to resume the Run File, **Quit** to terminate the execution of the innermost Run File, if it is nested, or **Quit All** to terminate the execution of all Run Files including any that are nested.

If the PAUSE command includes a **seconds** parameter, the Run File will pause for the specified number of seconds and then continue automatically. The PAUSE dialog box will include a “timer.”



PAUSE dialog box with a “seconds” parameter.

One of the hands on the timer will move each second. Select **Continue** to resume the Run File *before* the programmed pause expires.

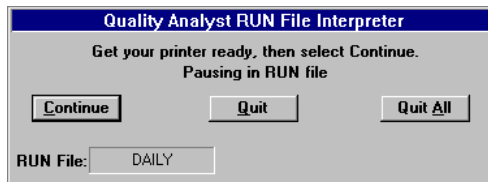
For example, to pause for 20 seconds and then continue automatically, the Run File command line would be:

```
PAUSE 20
```

If the PAUSE command line includes a **message** parameter, the PAUSE dialog box will display the message you specify. For example, this Run File command:

```
PAUSE Get your printer ready, then select Continue.
```

will display this dialog box:

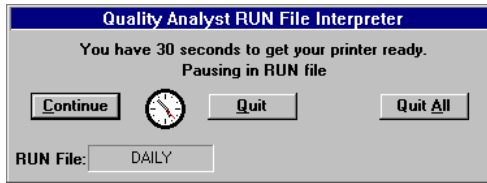


PAUSE dialog box with a “message” parameter

If your PAUSE command includes both a **seconds** and a **message** parameter, the PAUSE dialog box will include a timer and your message. For example, this Run File command:

`PAUSE 30 You have 30 seconds to get your printer ready.`

will display this dialog box:



PAUSE dialog box with “seconds” and “message” parameters

PCAP

The PCAP command creates a graphics (PLT) file containing a process capability histogram. The syntax of the PCAP command line in a Run File is:

```
PCAP infile var output [outfile]
```

Command Element	Description
<code>infile</code>	the name of the input file to use
<code>var</code>	the name of the variable in the input file to use
<code>output</code>	“D” to create a data (DAT) file containing bin counts (the number of data points represented by each bar) and distribution estimates (the value of the distribution curve at the midpoint of each bar), “G” to create a graphics (PLT) file.
<code>outfile</code>	the name of the output file, if applicable

Example: The command line to produce a process-capability chart on the variable PH in the file LTEX.DAT and send graphics output to the default file (GRAPHICS.PLT) is:

```
PCAP LTEX.DAT PH G
```

The “**G**” option creates a graphics (PLT) file but does *not* display or print the graphic. This is accomplished by an additional command line containing either a “**CHART**” or a “**PRINT**” command.

POLREG

With the POLREG Run File command, you can fit your data, using the least-squares method, to a polynomial equation of the form:

$$y = B_0 + B_1x + B_2x^2 + B_3x^3 + \dots B_nx^n$$

where y is the dependent variable and x is the independent variable. To perform a polynomial regression, the syntax for the Run File command line is:

```
POLREG inpfile indvar depvar order confint resid output
      [outfile]
```

Command Parameter	Description
inpfile	the name of the input file
indvar	the name of the independent variable
depvar	the name of the dependent variable
order	the order of the polynomial (n), an integer between 1 and 20
confint	Confidence intervals for graphics output (see below)
resid	“ R ” to include the residuals, or “ N ” to skip them.
output	“ D ” to create a data (DAT) file “ G ” to create a graphics (PLT) file “ W ” to create a formatted text (PTX) file
outfile	the name of the output file, if applicable

For **confint**, specify “**C**” to include the confidence intervals at 95% based on the individual measurements, “**N**” for no confidence intervals, or a number from 1 to 99 to specify the probability level percent. If you specify a number, follow it with either “**I**” to base confidence intervals on the individual measurements, or “**E**” to base confidence intervals on the equation. You must specify confidence intervals even if you do not select graphics output.

If you specify graphics output *and* residuals output, the resulting graphics (PLT) file will contain the residuals plot, *not* the regression plot. If you want the graphics file to contain the regression plot, *do not* specify residuals output. If you want to generate graphics files for both the residuals and the regression, include separate POLREG commands for each in your Run File.

Example: To perform a third-order polynomial regression with the independent variable LOAD and dependent variable RESPONSE in the file POLYTEST.DAT, with the regression plot in the graphics output file, the Run File command line would be:

```
POLREG POLYTEST.DAT LOAD RESPONSE 3 C N G
```

The “G” and “W” options respectively create a graphics (PLT) file and a formatted text (PTX) file but *do not* display or print them. This is accomplished by an additional command line containing a “CHART” or a “PRINT” command or both.

PRINT

The PRINT command sends output from graphics (PLT) files to the default Windows printer without displaying the graph on the screen. To use it, place a PRINT command line after one or more other command lines containing Quality Analyst functions that create PLT files. When used with the GROUP command, PRINT prints a multichart page layout.

PRINT Command Options	Description
GROUP <i>style</i>	Instructs PRINT to form a multichart layout in the specified style . Style is followed by a list of PLT files to include in the Group Layout. These PLT files must all have been produced before this command is executed. See GROUP on page 51, for examples and a list of layout styles.
NONE	Used in place of a PLT file name following a GROUP command to leave part of a Group Layout blank.
/LANDSCAPE	Prints the graphics in landscape orientation. (The default is portrait.)

Example 1: Print a graph generated by the XRS command—in this case, an X-bar and Range chart from the variable WEIGHT in the file TEST.DAT using the default (portrait) orientation:

```
XRS TEST.DAT WEIGHT X R G
PRINT
```

Note that the default graphics file name (GRAPHICS.PLT) need not be specified if the PRINT command follows immediately and only one chart is to be displayed.

Example 2: Create the graphics file CAPX.PLT, which contains a Process Capability histogram of the variable WEIGHT in the file TEST.DAT, and print it in landscape orientation:

```
PCAP TEST.DAT WEIGHT G CAPX.PLT
PRINT CAPX.PLT /LANDSCAPE
```

Example 3: Print a 2×2 group layout of four p-charts from the variables BUBBLES, MARKS, CRACKS, and TRIM in the file REJECTS.DAT.

```
ATTRIBUT REJECTS.DAT BUBBLES P G REJ1.PLT
ATTRIBUT REJECTS.DAT MARKS P G REJ2.PLT
ATTRIBUT REJECTS.DAT CRACKS P G REJ3.PLT
ATTRIBUT REJECTS.DAT TRIM P G REJ4.PLT
PRINT GROUP 2V2H REJ1.PLT REJ2.PLT REJ3.PLT REJ4.PLT
```

PROMPT

Note: In the 32-bit version of Quality Analyst, the PROMPT command has been superseded by the newer and more flexible “Ask” commands which are described beginning on page 13.

With the PROMPT Run File command, you can have your Run File pause and request input (such as a file name, variable name, text string, or filter criterion) to assign to a Replaceable Parameter (see *Using the “\$” Replaceable Parameters* on page 95). You can include a message to give the user of the Run File specific information about the item to be specified. To prompt for user input, the syntax for the Run File command line is:

```
PROMPT parameter [message]
```

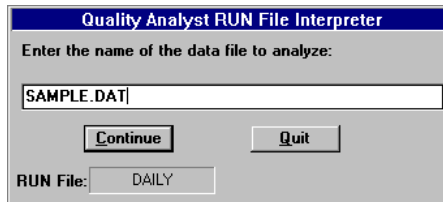
Command Parameter	Description
<code>parameter</code>	The Replaceable Parameter to which the user's input will be assigned. Choices are <code>\$FILE$n$</code> , <code>\$VARn</code> , <code>\$STRING$n$</code> , <code>\$MATCHFILTERn</code> , <code>\$RANGEFILTER$n$</code> , or <code>\$ASKFILTERn</code> where n is a digit from "1" to "9" or a letter from "a" to "z."
<code>message</code>	The message to display in the dialog box while prompting for user input

The PROMPT command will display a dialog box whose appearance depends on the type of parameter requested. Here are a few examples.

This command line:

```
PROMPT $FILE1 Enter the name of the data file to analyze:
```

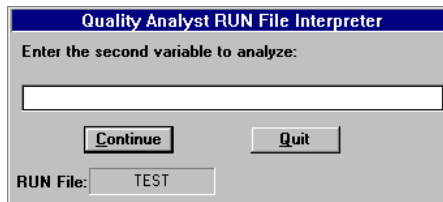
will display this dialog box:



This command line:

```
PROMPT $VAR2 Enter the second variable to analyze:
```

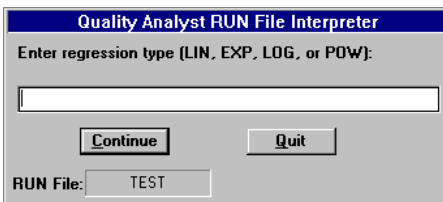
will display this dialog box:



This command line:

```
PROMPT $STRINGr Enter regression type (LIN, EXP, LOG, or POW) :
```

will display this dialog box:

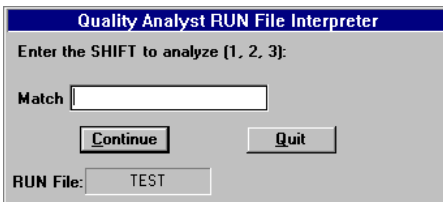


The dialog box is titled "Quality Analyst RUN File Interpreter". It contains the text "Enter regression type (LIN, EXP, LOG, or POW):" above a text input field. Below the input field are two buttons: "Continue" and "Quit". At the bottom, there is a "RUN File:" label followed by a text input field containing the word "TEST".

This command line:

```
PROMPT $MATCHFILTERd Enter the SHIFT to analyze (1, 2, 3) :
```

will display this dialog box:

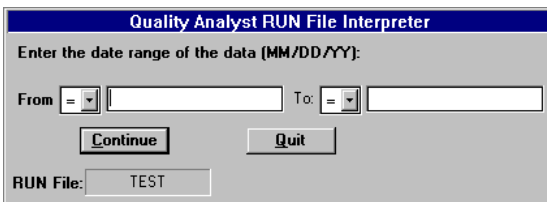


The dialog box is titled "Quality Analyst RUN File Interpreter". It contains the text "Enter the SHIFT to analyze (1, 2, 3):" above a text input field. Below the input field are two buttons: "Continue" and "Quit". At the bottom, there is a "RUN File:" label followed by a text input field containing the word "TEST".

This command line:

```
PROMPT $RANGEFILTER7 Enter the date range of the data (MM/DD/YY) :
```

will display this dialog box:



The dialog box is titled "Quality Analyst RUN File Interpreter". It contains the text "Enter the date range of the data (MM/DD/YY):" above two date input fields. Each input field is preceded by a dropdown menu with "=" selected. Below the input fields are two buttons: "Continue" and "Quit". At the bottom, there is a "RUN File:" label followed by a text input field containing the word "TEST".

This command line:

```
PROMPT $ASKFILTER4 Enter the date(s) of the data (MM/DD/YY) :
```

will display this dialog box:

During the execution of the Run File, you simply enter, when prompted, the requested value(s) and click **Continue** to proceed to the next Run File command. If you select **Quit**, the execution of the Run File (whose name is displayed in the recessed box labeled **Run File:**) will stop. If the PROMPT command is in a nested Run File, only the innermost Run File will stop.

For an explanation of what the Run File Interpreter does with the values you enter, see *Replaceable Parameters* beginning on page 92.

REDUCE

With the REDUCE Run File command, you convert a data file containing measurement data in subgroups into a data file contain the average (mean or X-bar), range, and standard deviation of each subgroup. The syntax for the Run File command line is:

```
REDUCE infile var output [outfile]
```

Command Parameter	Description
<code>infile</code>	the name of the input file
<code>var</code>	the name of a variable in the input file
<code>output</code>	“A” to append the reduced data to an existing data file, “C” to create a data file containing the reduced data, “O” to add the reduced-data variable to the original data file.

outfile	the name of the output data (DAT) file, if applicable
----------------	---

Quality Analyst will automatically include description variables from the input file in the output file.

Example: To reduce the subgrouped variable WEIGHT in the file TOMATO.DAT, and send the output to a new data file named TOMATXRS.DAT, the Run File command line would be:

```
REDUCE TOMATO.DAT WEIGHT C TOMATXRS.DAT
```

REGRESS

With the REGRESS Run File command, you can compute the least-squares coefficients of a single variable regression for the following forms of equations:

Linear: $y = A + Bx$

Exponential: $y = A + e^{Bx}$

Logarithmic: $y = A + B \times \text{Ln}(x)$

Power Law: $y = A \times x^B$

To perform a regression analysis, the syntax for the Run File command line is:

```
REGRESS infile indvar depvar types detail resid [confint]
      output [outfile]
```

Command Parameter	Description
infile	the name of the input file
indvar	the name of the independent variable
depvar	the name of the dependent variable
types	one or more of: “LIN” for linear, “EXP” for exponential, “LOG” for logarithmic, “POW” for power law, or “ALL” for all four

detail	“ D ” to include the detail data, or “ N ” to skip it
resid	“ R ” to include the residuals, or “ N ” to skip them.
confint	Confidence intervals for graphics output (see below)
output	“ D ” to create a data (DAT) file “ G ” to create a graphics (PLT) file “ W ” to create a formatted text (PTX) file
outfile	the name of the output file, if applicable

For **confint**, specify “**N**” for no confidence intervals, or a number from 1 to 99 to specify the probability level percent. If you specify a number, follow it with either “**I**” to base confidence intervals on the individual measurements, or “**E**” to base confidence intervals on the equation. If you omit **confint**, the graph will include the confidence intervals at 95% based on the individual measurements. Confidence intervals are not displayed if more than one regression type is selected.

If you specify graphics output *and* residuals output, the resulting graphics (PLT) file will contain the residuals plot, *not* the regression plot. If you want the graphics file to contain the regression plot, *do not* specify residuals output. If you want to generate graphics files for both the residuals and the regression, include separate REGRESS commands for each in your Run File.

Example 1: To perform linear regression on the variables LEVEL and OUTPUT in the file LOGDATA.DAT, with the regression plot in the graphics output file, the Run File command line would be:

```
REGRESS LOGDATA.DAT LEVEL OUTPUT LIN N N G
```

Example 2: To perform the same analysis as in Example 1, but with the residuals plot instead of the regression plot in the graphics output file, the Run File command line would be:

```
REGRESS LOGDATA.DAT LEVEL OUTPUT LIN N R G
```

Example 3: To perform the same analysis as in Example 1, but with text output (including details) in the file REG1.PTX, the Run File command line would be:

```
REGRESS LOGDATA.DAT LEVEL OUTPUT LIN D N W REG1.PTX
```

Example 4: To perform all available regressions on the variables LEVEL and OUTPUT in the file LOGDATA.DAT, with the regression plot in the graphics output file, the Run File command line would be:

```
REGRESS LOGDATA.DAT LEVEL OUTPUT ALL N N G
```

The “G” and “W” options respectively create a graphics (PLT) file and a formatted text (PTX) file but do *not* display or print them. This is accomplished by an additional command line containing a “CHART” or a “PRINT” command or both.

REM

With the REM Run File command, you can include comments in a Run File. This lets you annotate the Run File and describe or explain other command lines. Any line beginning with “REM” will be ignored during the execution of the Run File. For example:

```
REM Create X-bar chart and print it:
XRS VENEER.DAT THICK X N G
PRINT
```

REPORT

With the REPORT Run File command, you can print all or selected columns of Quality Analyst data files in a format suitable for reports, summaries, and hard-copy records. You define the format for the report in a format definition file.

The syntax for a REPORT Run File command line is:

```
REPORT infile rptfile output outfile [$PAGEWIDTH=n]
```

Command Parameter	Description
infile	the name of the input file
rptfile	the name of the report format definition file (see below)
output	“W” to create a formatted text (PTX) file

<code>outfile</code>	the name of the output file
<code>\$PAGEWIDTH=n</code>	the number of characters per line of output

Before executing the REPORT Run File command, you must create a report-format file (RPT) that contains the information needed by the REPORT command. Creating this file is explained under *Printing a Formatted Report* in Chapter 3.

You can control the width of a printed report by including a `$PAGEWIDTH=n` parameter on the **REPORT** command line. Substitute the number of characters per line of output (up to 240) for `n` (the default is 120).

Example: To generate a report from the data file LTEX.DAT, using the format definition file LTEX.RPT, and send the output to a formatted text file named LTEX.PTX, the Run File command line would be:

```
REPORT LTEX.DAT LTEX.RPT W LTEX.PTX
```

The “**W**” option creates a formatted text (PTX) file but does *not* display or print it. This is accomplished by an additional command line containing either a “**CHART**” or a “**PRINT**” command.

RUN

With the RUN Run File command, you can “nest” Run Files. The syntax of the command line is:

```
RUN filename
```

where **filename** is the name of a Run File. For example, the following Run File command line will execute the Run File named SECOND.RUN and resume with the next command line in the original Run File.

```
RUN SECOND.RUN
```

You can use the PROMPT command to request the name of the nested Run File from the user. For example:

```
PROMPT $FILE3 Enter the name of the secondary run file:
RUN $FILE3
```

One useful application of “nested” Run Files is error recovery. If an error occurs in a Run File, the Run File operation terminates. If an error occurs in a nested Run File, the execution of the “parent” Run File resumes at the next command. (This is the default behavior, which the user can set in Quality Analyst Settings.) This is very handy if you anticipate errors (such as missing data files or insufficient data after filtering) and want the Run File to continue when those errors occur. In this case, the portion of the Run File with the error potential should be “nested.”

RUNCHART

The RUNCHART command creates a graphics (PLT) file containing a run chart. The syntax of the RUNCHART command line in a Run File is:

```
RUNCHART inpfiler xvar yvars output [outfile]
```

Command Parameter	Description
inpfiler	the name of the input file
xvar	the name of a variable in the input file, or “ A ” for automatic (use row number)
yvars	from one to four variable names
output	“ G ” to create a one-chart graphics (PLT) file, or “ M ” to create a multi-chart graphics (PLT) file (see below).
outfile	the name of the output file, if applicable

Both the “**G**” and “**M**” options create a graphics (PLT) file. If you specify only one Y variable, **output** should be “**G**”. If you specify more than one Y variable, output type “**G**” produces a chart on which all Y variables are scaled to a single Y axis (see example in Chapter 5), and output type “**M**” produces charts on which each Y variable is scaled to its own Y axis (see example in Chapter 5).

The “**G**” and “**M**” options both create a graphics (PLT) file but do *not* display or print the graphic. This is accomplished by an additional command line containing either a “**CHART**” or a “**PRINT**” command.

Example 1: To plot the variable MOISTURE against the row number from the data file SAMPLE.DAT, and send output to the graphics file SAMPLE.PLT, the Run File command line would be:

```
RUNCHART SAMPLE.DAT A MOISTURE G SAMPLE.PLT
```

Example 2: To plot the variables MOISTURE, PROTEIN, and WEIGHT in the file SAMPLE.DAT on a single Y axis, the Run File command line would be:

```
RUNCHART SAMPLE.DAT A MOISTURE PROTEIN WEIGHT G
```

To plot them on separate Y axes, the Run File command line would be:

```
RUNCHART SAMPLE.DAT A MOISTURE PROTEIN WEIGHT M
```

SCATTER

With the SCATTER Run File command, you can create an automatically-scaled scatter diagram of one X-axis variable and up to four Y-axis variables in a data file. The syntax for the SCATTER command line in a Run File is:

```
SCATTER infile xvar yvars output [outfile]
```

Command Parameter	Description
<code>infile</code>	the name of the input file
<code>xvar</code>	the name of an X-axis variable in the input file
<code>yvar</code>	the names of up to four Y-axis variables
<code>output</code>	“G” to create a graphics (PLT) file
<code>outfile</code>	the name of the output file, if applicable

Example: To plot the variables PH and MOISTURE against the variable CONC from the data file SAMPLE.DAT, and send output to the graphics file SAMPLE.PLT, the Run File command line would be:

```
SCATTER SAMPLE.DAT CONC PH MOISTURE G SAMPLE.PLT
```

The “G” option creates a graphics (PLT) file but does *not* display or print the graphic. This is accomplished by an additional command line containing either a “CHART” or a “PRINT” command.

SET

The SET command assigns a value to a Replaceable Parameter. This command is available only in the 32-bit version of Quality Analyst. The syntax for the SET command is:

```
SET param = "value"
```

Command Element	Description
param	the name of the Replaceable Parameter that the value will be assigned to (See <i>Replaceable Parameters</i> starting on page 92 for naming requirements.)
value	a legal value for the Replaceable Parameter

Read more about *Replaceable Parameters* starting on page 92.

Example 1: In the following Run File statements:

```
SET ^File1 = "WEIGHT.DAT"
SET ^Var1 = "WT"
XRS ^File1 ^Var1 X R G
```

the first two assign “WEIGHT.DAT” and “WT” to the Replaceable Parameters “^File1” and “^Var1” respectively so that the third executes as:

```
XRS WEIGHT.DAT WT X R G
```

The following examples illustrate situations where using the SET command (rather than “hard coding” the parameters) is particularly useful.

Example 2: Creating and maintaining a Run File with repeated commands can be easier using the SET command. In this example the ATTRIBUT and CHART commands are all identical but they produce different results because of the SET commands.

```
SET ^InFile = "DEFECTS.DAT"
```

```

SET ^Var = "FOREIGN"
ATTRIBUT ^InFile ^Var U G
CHART
SET ^Var = "BUBBLES"
ATTRIBUT ^InFile ^Var U G
CHART
SET ^Var = "SINKS"
ATTRIBUT ^InFile ^Var U G
CHART
SET ^Var = "MOUNT"
ATTRIBUT ^InFile ^Var U G
CHART
SET ^Var = "MARKS"
ATTRIBUT ^InFile ^Var U G
CHART

```

Example 3: In the following Run File statements, a single, multipurpose, nested Run File is called using different parameters each time:

```

SET ^File1 = "CANDY.DAT"
SET ^Var1 = "WEIGHT"
RUN XRS.RUN
SET ^Var1 = "LENGTH"
RUN XRS.RUN
SET ^Var1 = "WIDTH"
RUN XRS.RUN

```

To clear, or “unassign,” a Replaceable Parameter, use the CLEAR command (page 44).

SHELL

With the SHELL Run File command, you can execute a DOS or Windows command from within a Run File as if from a Windows command line or a DOS system prompt. After the DOS command executes, the Run File resumes with the next command line. The syntax is:

```
SHELL command [parameters]
```

Command Parameter	Description
command	the DOS command you want to execute from within your Run File.

parameters	command-line parameters, if any, to pass to the command
-------------------	---

The SHELL command can be particularly useful if the purpose of your Run File is to integrate Quality Analyst with other applications. The command you specify on the SHELL command line can, among other things, copy or rename files, call a DOS batch file, or execute another DOS application.

Example: To run a third-party program named DATAMSTR.EXE which creates a file named OUTPUT.DBF, and rename that file INPUT.DBF for use by subsequent Run File commands, the command lines would be:

```
SHELL DATAMSTR.EXE
SHELL RENAME OUTPUT.DBF INPUT.DBF
```

To backup files created by preceding commands in a Run File, the command line might be:

```
SHELL COPY *.PLT C:\QA\BACKUP
```

SHELL should only be used by users familiar with its operation, and will only work on systems using DOS version 3.0 or higher.

SINGCOL

With the SINGCOL Run File command, you can convert a single column of data into a data file containing multiple columns of data, and vice versa.

To convert several data columns *to* a single column, the syntax for the command line is:

```
SINGCOL infile T vars desc [descvars] D outfile
```

To convert *from* a single column of data into multiple columns, the syntax for the command line is:

```
SINGCOL infile F var size desc [descvars] D outfile
```

Command Parameter	Description
infile	the name of the input file

var, vars	the name(s) of the variable(s) in the input file to convert
size	the number of observations per sample (that is, the number of columns you are converting the single data column into)
desc	“ D ” to include description variables, or “ N ” not to include description variables
descvars	the name(s) of the description variable(s)
outfile	the name of the output file

Example: To convert the column WEIGHT in the input file TEST1.DAT to four columns in the output file TEST2.DAT, and include the description variables DATE and LOTNUM, the Run File command line would be:

```
SINGCOL TEST1.DAT F WEIGHT 4 D DATE LOTNUM D TEST2.DAT
```

SUMMARY

With the SUMMARY Run File command, you can summarize the output of a Quality Analyst chart command. You can generate either a list of out-of-control points in the file, or a summary or tabulation of out-of-control points based on user-selected description fields. To generate a summary report, the syntax of the Run File command line is:

```
SUMMARY infile type output outfile
```

Command Parameter	Description
infile	the name of the summary (SMY) input file
type	“ L ” for list, “ S ” for summary, or “ B ” for both
output	“ W ” to create a formatted text (PTX) file
outfile	the name of the output file

The **infile** for a SUMMARY command is the *output* file from a control chart command with the “**D**” output option. You must therefore precede a

SUMMARY command with a chart command that includes the “D” output option and an SMY file name.

Example: To generate both list output and summary output from the summary file VENEER.SMY, and send the output to a formatted text file named VENEER.PTX, the Run File command line would be:

```
SUMMARY VENEER.SMY B W VENEER.PTX
```

The “W” option creates a formatted text (PTX) file but does *not* display or print it. This is accomplished by an additional command line containing either a “CHART” or a “PRINT” command.

SUMSTAT

With the SUMSTAT Run File command, you can compute “sample” or “population” descriptive statistics for one or more variables in a data file. To compute descriptive statistics, the syntax for the Run File command line is:

```
SUMSTAT infile vars type output outfile
```

Command Parameter	Description
<code>infile</code>	the name of the input file
<code>vars</code>	the names of the variables to analyze
<code>type</code>	“S” for sample data, or “P” for population data.
<code>output</code>	“D” to create a data (DAT) file “W” to create a formatted text (PTX) file
<code>outfile</code>	the name of the output file

Example: To calculate “sample” statistics for the variable WEIGHT in the file TOMATO.DAT, and send output to a formatted text file named TOMATO.PTX, the Run File command line would be:

```
SUMSTAT TOMATO.DAT WEIGHT S W TOMATO.PTX
```

The “**w**” option creates a formatted text (PTX) file but does *not* display or print it. This is accomplished by an additional command line containing either a “**CHART**” or a “**PRINT**” command.

SYSTEM

The SYSTEM command stops the execution of the Run File and the Run File Interpreter, returning you to Windows. The syntax is:

```
SYSTEM
```

TITLE

With the TITLE Run File command, you define a title for the Run File. Any text following this command will be used as the title. This command must be the first line in the Run File. For example:

```
TITLE Northwest Manufacturing Co.
```

Run File titles are displayed next to the Run File names in the **Execute Run File** dialog box.

TTESTS

The TTESTS command calculates one of four types of t-statistics. Refer to the section titled *t-Statistics* in Chapter 8 for a detailed explanation of t-statistics.

The syntax of the TTESTS command line to calculate an **ordinary unpaired sample** t-statistic in a Run File is:

```
TTESTS inpfile U var1 var2 samevari hypmeandiff output  
      outfile
```

The syntax of the TTESTS command line to calculate a **paired sample** t-statistic in a Run File is:

```
TTESTS inpfile P var1 var2 hypmeandiff output outfile
```

The syntax of the TTESTS command line to calculate a **single sample** t-statistic in a Run File is:

```
TTESTS infile S var hyppopmean popvari output outfile
```

The syntax of the TTESTS command line to calculate the **Pearson Product Moment Correlation** in a Run File is:

```
TTESTS infile M var1 var2 output outfile
```

Command Element	Description
<code>infile</code>	the name of the input file to use
<code>var1</code>	the name of the variable to use for the first sample set
<code>var2</code>	the name of the variable to use for the second sample set
<code>samevari</code>	“Y” if the variables have the same variance, or “N” if not.
<code>hypmeandiff</code>	the hypothesized mean difference between the two sample sets
<code>var</code>	the name of the variable with a known population mean
<code>hyppopmean</code>	the hypothesized population mean of the variable
<code>popvari</code>	the population variance: enter a number, if known, or “U” if unknown
<code>output</code>	“W” to create a formatted text (PTX) file.
<code>outfile</code>	the name of the output file

Example 1: The command line to calculate an ordinary unpaired sample t-statistic on the variables TORQUE1 and TORQUE2 in the file CAPS.DAT, where both variables have the same variance and the hypothesized mean difference is 0.0, is:

```
TTESTS CAPS.DAT U TORQUE1 TORQUE2 Y 0 W DEFAULT.PTX
```

Example 2: The command line to calculate a paired sample t-statistic on the variables INIT and DUP in the file CALCIUM.DAT where the hypothesized mean difference is 0.0, is:

```
TTESTS CALCIUM.DAT P INIT DUP 0 W DEFAULT.PTX
```

Example 3: The command line to calculate a single sample t-statistic on the variable VISCOS in the file NP7.DAT where the hypothesized population mean is 1300 and the population variance is unknown, is:

```
TTESTS NP7.DAT S VISCOS 1300 U W DEFAULT.PTX
```

Example 4: The command line to calculate the Pearson Product Moment Correlation on the variables SIEVE1 and SIEVE2 in the file SHAKER.DAT is:

```
TTESTS SHAKER.DAT M SIEVE1 SIEVE2 W DEFAULT.PTX
```

The “w” option creates a formatted text (PTX) file but does *not* display or print it. This is accomplished by an additional command line containing either a “CHART” or a “PRINT” command.

USETEMPDIR

The USETEMPDIR command creates a temporary directory (folder) on your local PC and makes that directory the current Quality Analyst directory. This command is available only in the 32-bit version of Quality Analyst. The syntax for the USETEMPDIR command line in a Run File is:

```
USETEMPDIR [ON] | OFF
```

USETEMPDIR ON (or just **USETEMPDIR**) creates and enables the temporary directory. **USETEMPDIR OFF** restores the current Quality Analyst directory to its previous value.

USETEMPDIR is intended to be used when more than one Quality Analyst user might access the same Connectivity Data Set stored in a shared location. This command makes sure that all such users are working with their own local copies of the retrieved data.

When USETEMPDIR is ON, the CONNECT command copies the .HED file from the **old** current directory into the **new** temporary directory, then retrieves data from the database into a .DAT file in the temporary directory. All charting commands in the Run File look in the current directory, which has been changed to the new temporary directory.

Note If the file name in the CONNECT command or any charting command includes the **full path** to the data file, Quality Analyst will use the full path, overriding USETEMPDIR.

Example: The Run File listed below and the Data Set EXTDATA.DAT and EXTDATA.HED are located in a shared directory, G:\QA\DATA\.

```
USETEMPDIR ON
CONNECT EXTDATA.DAT
PCAP EXTDATA.DAT VAR G 1.PLT
CHART 1.PLT
```

Here's what happens as each command line is executed:

Command Line	Action
USETEMPDIR ON	<ol style="list-style-type: none"> 1. Create the directory C:\WINDOWS\TEMP\QA\.[†] 2. Make C:\WINDOWS\TEMP\QA\ the current directory.
CONNECT EXTDATA.DAT	<ol style="list-style-type: none"> 3. Copy EXTDATA.HED from G:\QA\DATA\ to C:\WINDOWS\TEMP\QA\. 4. Open C:\WINDOWS\TEMP\QA\EXTDATA.HED and retrieve external data into C:\WINDOWS\TEMP\QA\EXTDATA.DAT.
PCAP EXTDATA.DAT VAR G 1.PLT	<ol style="list-style-type: none"> 5. Create a graphics file (C:\WINDOWS\TEMP\1.PLT[‡]) containing a process-capability histogram of the variable VAR from C:\WINDOWS\TEMP\QA\EXTDATA.DAT.
CHART 1.PLT	<ol style="list-style-type: none"> 6. Display the chart in the file C:\WINDOWS\TEMP\1.PLT.
[Run File ends]	<ol style="list-style-type: none"> 7. Delete all files in C:\WINDOWS\TEMP\QA\ then delete the directory.

[†] Assuming the Quality Analyst Graphics and Text File Path is C:\WINDOWS\TEMP\.

[‡] Assuming the Quality Analyst Temporary File Path is C:\WINDOWS\TEMP\.

See *File Paths* on page 227 (in the main User's Manual) for details.

WALDRUN

The WALDRUN command performs a Wald-Wolfowitz test to determine the degree of randomness in a data series. Refer to the section titled *Wald-Wolfowitz Runs Test* in Chapter 8 for a detailed explanation of the Wald-Wolfowitz test.

The syntax of the WALDRUN command line in a Run File is:

```
WALDRUN infile var conditions output outfile
```

Command Parameter	Description
<code>infile</code>	the name of the input file
<code>var</code>	the name of the variable to test for randomness
<code>conditions</code>	two criteria for dividing your data (see below)
<code>output</code>	“W” to create a formatted text (PTX) file
<code>outfile</code>	the name of the output file

For `conditions`, enter two conditions for dividing your data into two mutually exclusive sets. Each condition consists of a comparison operator (from the table below) and a value. For example, to divide your data into sets containing values that are less than 134 and greater than 134, `condition` would be:

```
.LT. 134 .GT. 134
```

Comparison Operators

Operator	Meaning
.LT.	less than
.GT.	greater than
.LE.	less than or equal to
.GE.	greater than or equal to

Example: The command line to perform the Wald-Wolfowitz test on the variable VISCOS in the file NP7.DAT where values of VISCOS are divided into those less than 1300 and those greater than or equal to 1300, is:

```
WALDRUN NP7.DAT VISCOS .LT. 1300 .GE. 1300 W DEFAULT.PTX
```

WEIBULL

With the WEIBULL Run File command, you can fit the Weibull or the exponential probability distribution to data and produce a probability plot. The syntax for the Run File command line is:

```
WEIBULL infile var output [outfile]
```

Command Parameter	Description
<code>infile</code>	the name of the input file
<code>var</code>	the name of the variable to fit
<code>output</code>	“G” to create a graphics (PLT) file
<code>outfile</code>	the name of the output file, if applicable

Example: To use the variable LIFE from the file FAN.DAT, and create a graphics output file, the Run File command would be:

```
WEIBULL FAN.DAT LIFE G
```

The “G” option creates a graphics (PLT) file but does *not* display or print the graphic. This is accomplished by an additional command line containing either a “CHART” or a “PRINT” command.

XRS

The XRS command creates a graphics (PLT) file containing an X-bar, X-bar and Range, or X-bar and Standard Deviation chart. The syntax of the XRS command line in a Run File is:

```
XRS infile var chart1 chart2 output [outfile]
```

Command Element	Description
<code>inpf</code>	the name of the input file
<code>var</code>	the name of the variable in the input file to use
<code>chart1</code>	the type of the chart: “X” for X-bar chart “R” for Range chart “S” for Standard Deviation chart
<code>chart2</code>	the type of the second chart of a pair: “X” for X-bar chart “R” for Range chart “S” for Standard Deviation chart “N” for none
<code>output</code>	“D” to create a summary (SMY) file, “G” to create a graphics (PLT) file.
<code>outfile</code>	the name of the output file

Note: `chart1` and `chart2` must be different.

Example 1: To create an X-bar chart for the variable THICK in the file VENEER.DAT, and produce a graphics file named VENEER.PLT for later use by the PRINT or CHART commands, the Run File command line would be:

```
XRS VENEER.DAT THICK X N G VENEER.PLT
```

Note the mandatory “N” (for “none”) for the nonexistent second chart type.

Example 2: To create an X-bar and Range chart for the same variable, and produce a graphics file for later use, the Run File command line would be:

```
XRS VENEER.DAT THICK X R G
```

In this example, the absence of the (`outfile`) parameter tells Quality Analyst to create a graphics file with the default name GRAPHICS.PLT.

The “G” option creates a graphics (PLT) file but does *not* display or print the graphic. This is accomplished by an additional command line containing either a “CHART” or a “PRINT” command.

Replaceable Parameters

Introduction

A Quality Analyst Run File may contain Replaceable Parameters wherever certain parameters (such as file names, variable names, or filter criteria) are normally expected.

Replaceable Parameters have several uses:

- You can create generic Run Files that perform a specific set of tasks, but on different files or variables, and using different filter criteria on each run.
- Users can be prompted for values to be assigned to Replaceable Parameters at run time, providing flexible SQC charting and reporting capabilities for those who are unfamiliar with Quality Analyst.
- A set of Run File command lines that create charts can be repeated throughout a Run File, each preceded by a different set of command lines that assign different values to the Replaceable Parameters before the charting commands are executed. This makes it easier to create Run Files that perform repetitive operations.
- You can “call” a Run File from within other programs using different command lines that assign different values to Replaceable Parameters. This provides more flexibility in designing integrated SQC systems.
- Users can be prompted at run time for Replaceable Parameters that will be used to modify queries to databases. (See *Using Replaceable Parameters with Filter Criteria in SQL Statements* on page 137 for more information.)

Once a user has assigned a value to a Replaceable Parameter, the Run File Interpreter will substitute that value for each occurrence of the parameter in the Run File.

There are two types of Replaceable Parameters:

- Replaceable Parameters beginning with a caret (“^”) are more flexible and user-friendly. When used in a Run File with an appropriate “Ask” command, Quality Analyst prompts the user for input using labeling and data-checking appropriate for the type of data being requested.

Several forms of this type of Replaceable Parameter can also be used to modify database queries before they are performed. (See *Using Replaceable Parameters with Filter Criteria in SQL Statements* on page 137 for more information.) The “^” Replaceable Parameters are a more recent addition to Quality Analyst and are available only in the 32-bit version.¹

- Replaceable Parameters beginning with a dollar sign (“\$”) are an older form and do not provide as much flexibility as “^” Replaceable Parameters. They have been retained to provide backward compatibility. “\$” Replaceable Parameters prompt automatically for user input if the Run File Interpreter encounters one that has not been assigned a value.

The following discussion mainly addresses the “^” Replaceable Parameters. The “\$” Replaceable Parameters are discussed under *Using the “\$” Replaceable Parameters* beginning on page 95.

Naming Replaceable Parameters

Replaceable Parameters begin with the caret character (“^”) and may contain alphabetic characters (“A”–“Z” or “a”–“z”), digits (“0”–“9”), and the underscore (“_”). (The characters “.” and “!” are also used in Replaceable Parameters used for querying Connectivity Data Sets.) Example legal names for Replaceable Parameters are:

```
^ProductCode
^VarName
^Prompt_Text
^Date!MatchFilter
^WT:3!ToFilter
```

Replaceable Parameter names are not case-sensitive. That is, Quality Analyst will treat `^ProductCode` and `^PRODUCTCODE` as the same Replaceable Parameter. Spaces are not allowed in Replaceable Parameter names; use the underscore instead. (Note that Quality Analyst will treat `^ProductCode` and `^Product_Code` as **different** Replaceable Parameters.)

¹ To see if you have the 32-bit version, open the **Help** menu, select **About**, and look for “32-bit” in the resulting message box.

Assigning Values to Replaceable Parameters

There are three ways you can assign a value to a Replaceable Parameter in a Run File:

- by having Quality Analyst prompt you for it at run time,
- by assigning it within a Run File, or
- by assigning it on the command line for the Run File Interpreter.

Prompting for Replaceable Parameters at Run Time

To have Quality Analyst prompt you for values of Replaceable Parameters when you execute a Run File, you should use any of the “**Ask**” Run File commands (such as **AskDateRange** or **AskInteger**) which are described starting on page 13.

Setting Replaceable Parameters within Run Files

To assign values to Replaceable Parameters within a Run File, use the **SET** Run File command. See **SET** on page 80 for details.

Passing Replaceable Parameters from the Command Line

Assigning values to Replaceable Parameters on the command line can be useful when running the Quality Analyst Run File Interpreter:

- from a program shortcut, or
- from within another program.

See Also: *Executing Run Files from a Program Shortcut*, p. 105
Executing Run Files from Other Programs, p. 107

To assign a value to a Replaceable Parameter on the command line for the Run File Interpreter you must first include a Replaceable Parameter in the Run File. For example, let’s say the Run File WTTASK.RUN contains the following statement.

```
XRS ^File1 WT X R G
```

“**^File1**” in this statement is the Replaceable Parameter. When the command line:

```
QARFI.EXE $RUN=WTTASK.RUN ^File1="WEIGHT.DAT"
```

is executed, “**^File1**” in the Run File statement will be replaced with “**WEIGHT.DAT**” from the command line and the statement will be executed as:

```
XRS WEIGHT.DAT WT X R G
```

You can include more than one Replaceable-Parameter assignment in a command line. For example, let’s say WTTASK.RUN contains the following statement.

```
XRS ^DatFile ^XRvar X R G
```

When the command line:

```
QARFI.EXE $RUN=WTTASK.RUN ^DatFile="WEIGHT.DAT" ^XRvar="WT"
```

is executed, “**^DatFile**” in the Run File statement will be replaced with “**WEIGHT.DAT**,” “**^XRvar**” will be replaced with “**WT**,” and the statement will be executed as:

```
XRS WEIGHT.DAT WT X R G
```

See Also: *Executing Run Files Using a Command Line*, p. 104

Using the “\$” Replaceable Parameters

Quality Analyst recognizes six types of “\$” Replaceable Parameters as shown in this table.

“\$” Replaceable Parameters

Parameter Name	Description
\$FILEn	File name
\$VARn	Variable name

\$STRINGn	Text string used for labels or miscellaneous parameters
\$RANGEFILTERn	Range filter criteria
\$MATCHFILTERn	Match filter criteria
\$ASKFILTERn	User-selected (match or range) filter type and criteria

The FILTER parameters let you specify the **[type]** and **[value]** elements of the \$FILTER command option (described on page 49) at run time. (See the example below.)

Quality Analyst lets a single Run File contains up to 35 “\$” Replaceable Parameters of *each* type by allowing the “n” in each parameter name to be a digit from “1” to “9” or a letter from “A” to “Z.” Legal Replaceable Parameters include \$FILE1, \$VAR9, \$STRINGA, and \$RANGEFILTERZ.

Note: The \$STRINGn parameter accepts all text entered by the Run File user *as is*, and is therefore potentially troublesome. The Run-File author is well-advised to include explicit instructions in the message portion of the PROMPT command when including a \$STRINGn Replaceable Parameter.

You can assign values to “\$” Replaceable Parameters in one of two ways:

- Include a PROMPT command for each Replaceable Parameter in the Run File. When the PROMPT command is executed, the user enters the “value” of the Replaceable Parameter (like a file name) in the Prompt dialog box. (See **PROMPT** on page 70). You will probably find this method most useful.
- Specify the values of the Replaceable Parameters in the Windows command line when you execute the Run File.

Example: During the execution of a Run File, if the user is asked for a data file name with the command:

```
PROMPT $FILE3 Enter a file name to analyze:
```

and the user enters **COOKIE.DAT** when prompted, then the Run File command:

```
XRS $FILE3 WEIGHT X R G
```

will be executed as:

```
XRS COOKIE.DAT WEIGHT X R G
```

In the same manner, users can be prompted for variable names, filter criteria, or text strings (for use as labels or special command parameters). For example, if the user makes the following assignments at run time (either on the command line or in response to PROMPT commands):

Replaceable Parameter	Assigned Value
\$FILE1	ELEMENTS.DAT
\$VAR1	PB
\$VAR2	CD
\$VARB	DATE
\$STRINGa	LIN
\$RANGEFILTER7	L 8/1/96 U 8/31/96

then the Run file line:

```
REGRESS $FILE1 $VAR1 $VAR2 $STRINGa N N G $FILTER="$VARb
$RANGEFILTER7 "
```

will be executed as:

```
REGRESS ELEMENTS.DAT PB CD LIN N N G $FILTER="DATE L 8/1/96 U
8/31/96 "
```

Note: If the Run File Interpreter encounters a "\$" Replaceable Parameter for which a value has not been provided, the user will be prompted for the value of the parameter as if a PROMPT command *without* a message parameter had been executed.

Creating Complete Task Sequences

One of the advantages of Run Files is that they allow you to set up complete sequences of tasks and operations. The following examples illustrate this capability.

Example 1

In this example, the Run File will create an X-bar and Range chart on the screen for the variable THICK in the data file VENEER.DAT. It will then create a process-capability distribution on the screen. The Run File will pause when each chart is displayed and wait for the user to resume execution. The default graphics file, GRAPHICS.PLT, will be used since no file name is specified. The Run File contains the following lines:

```
TITLE XR and PCAP of veneer thickness
XRS VENEER.DAT THICK X R G
CHART
PCAP VENEER.DAT THICK G
CHART
```

Example 2

In this example, we expand the first example to add an Individuals chart and a Normal Probability plot and then assemble the charts into a group layout for printing.

```
TITLE Print XR-PCAP-I group of veneer thickness
XRS VENEER.DAT THICK X R G NW.PLT
PCAP VENEER.DAT THICK G NE.PLT
IR VENEER.DAT THICK I N G SW.PLT
NORMPLOT VENEER.DAT THICK G SE.PLT
PRINT GROUP 2V2H NW.PLT NE.PLT SW.PLT SE.PLT
```

Example 3

In this example, the Run File creates separate graphics (PLT) files containing p-charts for each defect variable in the data file NOZZLE.DAT. The Run File then creates a graphics file containing a Pareto chart for four variables in the same data file. In the last portion of the Run File, the newly created graphics files are used to print charts with the PRINT command.

```
TITLE Print p-charts of nozzle defects
ATTRIBUT NOZZLE.DAT TDEFECT P G TDEFECT.PLT
ATTRIBUT NOZZLE.DAT SEAL P G SEAL.PLT
ATTRIBUT NOZZLE.DAT THREADS P G THREADS.PLT
ATTRIBUT NOZZLE.DAT WELD P G WELD.PLT
ATTRIBUT NOZZLE.DAT SURFACE P G SURFACE.PLT
```

```
PARETO NOZZLE.DAT M SEAL THREADS WELD SURFACE G NOZZLE.PLT
PRINT TDEFECT.PLT
PRINT SEAL.PLT
PRINT THREADS.PLT
PRINT WELD.PLT
PRINT SURFACE.PLT
PRINT NOZZLE.PLT
```

Note that this Run File could be broken into two Run Files (between the **PARETO** command and the first **PRINT** command), each of which could be executed at a different time, perhaps by different users. This could be useful if, for example, your data and your printer are not always available at the same time.

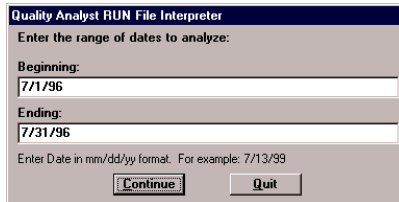
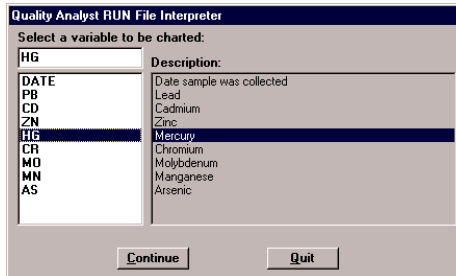
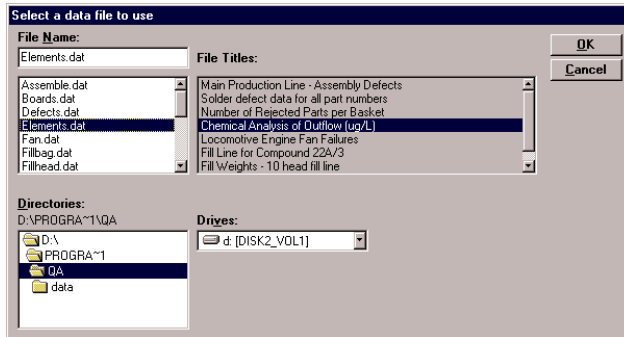
Example 4a

In this example, the Run File prompts the user for input using commands that assign values to Replaceable Parameters. The user is prompted for the name of a data file, the name of a variable, and a date range. The Run File then creates a file containing an Individual and a Range chart for the variable specified by the user and displays it on the screen.

```
TITLE Generic IR with date filter
AskDATFile ^File1 "" "Select a data file to use"
AskVariable ^Var1 "^File1" "" "Select a variable to be
  charted:"
AskDateRangeFilter ^DateFilter "Enter the range of dates to
  analyze:" "Beginning:" "Ending:"
IR ^File1 ^Var1 I R G $FILTER="DATE ^DateFilter"
CHART
```

When this Run File executes, you will be prompted first for the name of the data file to use, then for the name of the variable to use, and then for the range of dates to include in the analysis. If you specify ELEMENTS.DAT for the file name, HG for the variable name, and 7/1/96 and 7/31/96, as shown below, the effect of the example Run File will be the same as that of the following Run File:

```
TITLE Generic IR with date filter
IR ELEMENTS.DAT HG I R G $FILTER="DATE L 7/1/96 U 7/31/96"
CHART
```



Example “Ask” dialog boxes

Example 4b

This example illustrates the same task as the previous example. In this case, however, the Run File prompts the user for input using commands that assign values to “\$” Replaceable Parameters.

```
TITLE Generic IR with date filter
PROMPT $FILE1 Enter the data file to use
PROMPT $VAR1 Enter the variable to be charted
PROMPT $RANGEFILTER1 Enter the range of dates to analyze:
IR $FILE1 $VAR1 I R G $FILTER="DATE $RANGEFILTER1"
CHART
```

When this Run File executes, you will be prompted first for the name of the data file to use, then for the name of the variable to use, and then for the range of dates to include in the analysis. If you specify ELEMENTS.DAT for the file name, HG for the variable name, and 7/1/96 and 7/31/96, as shown below, the effect of the example Run File will be the same as that of the following Run File:

```
TITLE Generic IR with date filter
IR ELEMENTS.DAT HG I R G $FILTER="DATE L 7/1/96 U 7/31/96"
CHART
```

The image shows three sequential dialog boxes from the 'Quality Analyst RUN File Interpreter' software. Each dialog has a blue title bar and a grey background.

- Dialog 1:** Titled 'Quality Analyst RUN File Interpreter'. It prompts 'Enter the data file to use'. A text box contains 'ELEMENTS.DAT'. Below are 'Continue' and 'Quit' buttons. At the bottom, 'RUN File:' is followed by a text box containing 'DAILY'.
- Dialog 2:** Titled 'Quality Analyst RUN File Interpreter'. It prompts 'Enter the variable to be charted'. A text box contains 'HG'. Below are 'Continue' and 'Quit' buttons. At the bottom, 'RUN File:' is followed by a text box containing 'EXAMPLE3'.
- Dialog 3:** Titled 'Quality Analyst RUN File Interpreter'. It prompts 'Enter the range of dates to analyze:'. It has two date pickers: 'From' with a dropdown arrow and 'To' with a dropdown arrow. The 'From' date is '7/1/96' and the 'To' date is '7/31/96'. Below are 'Continue' and 'Quit' buttons. At the bottom, 'RUN File:' is followed by a text box containing 'TEST'.

Example PROMPT dialog boxes

Example 5

In this example, another software package runs the Quality Analyst Run File Interpreter, invoking a Run File, and providing the file and variable names. A second Run File is “nested” within the first, and its name is provided at run time. The name of the nested Run File (and which additional charts are to be created) is determined at run time by the originating software.

The following Run File setup will create an X-bar and Range chart and a Process Capability chart for the variable ACID from the file TEST.DAT, start up a second Run File to create a Run Chart for variables DSOLID and SSOLID, and return to the originating Run File to create a report using all three variables in TEST.DAT. Control returns to Windows (and hence, to the originating program) at the end of Run File execution:

- 1 The originating program executes the following Windows command which runs the Quality Analyst Run File Interpreter:

```
QARFI $RUN=BASIC.RUN ^File1="TEST.DAT" ^File2="BASIC2.RUN"
      ^Var1="ACID" ^Var2="DSOLID" ^Var3="SSOLID"
```

- 2 The parent (unnested) Run File, BASIC.RUN, contains the following commands:

```
XRS ^File1 ^Var1 X R G
CHART
PCAP ^File1 ^Var1 G
CHART
RUN ^File2
REPORT ^File1 BASIC.RPT W
CHART
SYSTEM
```

- 3 The nested Run File, BASIC2.RUN, which is specified in the command line by the originating program, contains the following lines:

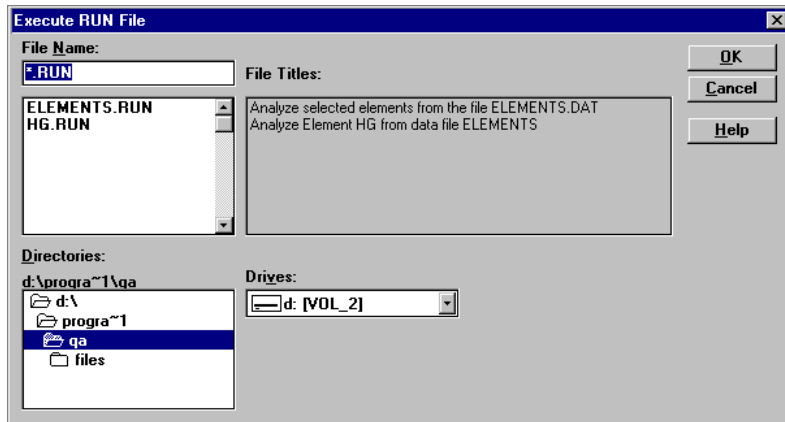
```
RUNCHART ^File1 A ^Var2 $Var3 G
CHART
```

Executing Run Files

Executing Run Files From within Quality Analyst



After you have created a Run File, you initiate Run File processing by clicking the Run File button or by opening the **Run** menu of the Editor and selecting **Run File Processing**. The **Execute Run File** dialog box will appear.



Execute Run File dialog box

Select the Run File you want to process by clicking its name in the list, then clicking **OK**.

The Run File Interpreter will act on the commands in the Run File. While the Run File is being processed, the Run File Interpreter may display any of the **Quality Analyst Run File Interpreter** dialog boxes, shown below.

The dialog box shown below will appear at the completion of a CHART command that does *not* include the /DELAY option. Clicking the **Continue** button will clear the dialog box and resume the Run File. Clicking the **Quit** button will terminate the execution of the Run File.



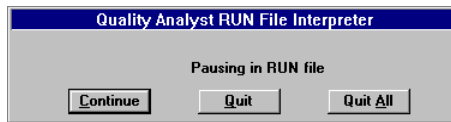
After a CHART command without the /DELAY option

The dialog box shown below will appear at the completion of a CHART command that *does* include the /DELAY option. One of the hands on the timer will move each second. Click the **Continue** button to resume the Run File before the programmed delay period expires. Click the **Quit** button to terminate the execution of the Run File.



After a CHART command with the /DELAY option

The dialog box shown below will appear when a PAUSE command is executed. Click the **Continue** button to resume the Run File. Clicking **Quit** will terminate the execution of the innermost Run File, if it is nested. Clicking **Quit All** will terminate the execution of all Run Files including any that are nested.



The result of a PAUSE command

Executing Run Files Using a Command Line

You can also execute Run Files from a Windows command line, such as:

- the one that appears when you click the **Start** button and select **Run**,
- the one in the definition of a program shortcut,
- the one in the definition of a Windows 98 Task Scheduler event, or
- the argument of a “shell” command in another program.

To invoke a charting function, enter a command line that includes the name of the Quality Analyst Run File Interpreter (QARFI.EXE), a \$RUN command, and the name of a Run File.

For example, to execute the Run File CHART.RUN from another application or as part of an icon’s properties, the command line would be:

```
C:\QA\QARFI.EXE $RUN=CHART.RUN
```

(This assumes QARFI.EXE is located in C:\QA\ and that CHART.RUN is in a subdirectory where QARFI.EXE can find it.)

See *Passing Replaceable Parameters from the Command Line* on page 94 for more information.

Executing Run Files from a Program Shortcut

You can set up Quality Analyst Run Files to execute from their own program shortcuts (called “program items” in Windows 3.x).

In Windows 95/98:

- 1 Click the **Start** button, point to **Settings**, and click **Taskbar (Taskbar & Start Menu** in Windows 98).
- 2 Select the **Start Menu Programs** tab.
- 3 Click the **Add** button to display the **Create Shortcut** dialog box.
- 4 In the **Command Line** box, enter the name of the Quality Analyst Run File Interpreter (“QARFLEXE”), the command “\$RUN=”, and the name of the Run File. For example, to execute the Run File “CHART.RUN,” the command line might be:

```
"C:\Program Files\QA\QARFI.EXE" $RUN=C:\QA\CHART.RUN
```

(You can include Replaceable Parameters in this command line. See *Passing Replaceable Parameters from the Command Line* on page 94 for more information. Also, if either file path includes spaces, you may need the quotations marks as shown, depending on your operating system.)

Click **Next**.

- 5 In the **Select Program Folder** dialog box, select the Start Menu folder where you want to save the program shortcut. Click **Next**.
- 6 In the **Select a Title for the Program** dialog box, enter the text you want to appear on the Start Menu to represent this shortcut. For example:

```
Create QA Chart
```

Click **Finish**.

In Windows 3.x:

- 1 In Program Manager, open the **File** menu and select **New**.
- 2 Choose **Program Item** and fill out the **Program Item Properties** dialog.

- The **Description** can be any text, and will appear below the icon. For example:

Create QA Chart

- The **Command Line** should contain the name of the Quality Analyst Run File Interpreter (“QARFI.EXE”), the command “\$RUN=”, and the name of the Run File. For example, to execute the Run File “CHART.RUN,” the command line might be:

C : \QA \QARFI . EXE \$RUN=C : \QA \CHART . RUN

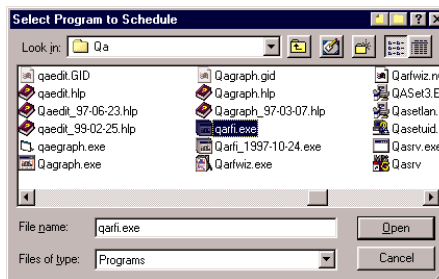
(You can include Replaceable Parameters in this command line. See *Passing Replaceable Parameters from the Command Line* on page 94 for more information.)

- 3 Click **OK**.

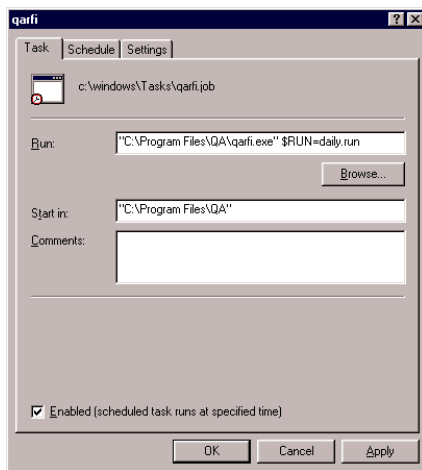
Executing Run Files Using Task Scheduler

Windows 98 includes the Task Scheduler (which appears as “Scheduled Tasks” in Windows Explorer). This feature lets you schedule programs to run periodically. You can use this feature to execute a Run File at an interval you specify.

- 1 In Windows Explorer, select **Scheduled Tasks** in the folder tree on the left.
- 2 In the list of tasks on the right, double-click **Add Scheduled Task** to launch the Scheduled Task Wizard.
- 3 When the list of applications appears, click **Browse**, locate QARFI.EXE in the Quality Analyst folder, and click **OK**.



- 4 Follow the instructions provided by the wizard to specify the frequency at which you want to execute the run file.
- 5 On the last wizard screen, select the **Open advanced properties for this task when I click Finish** check box, then click **Finish**.
- 6 On the next wizard screen, in the **Run** box, add “**\$RUN=**” and the name of the Run File to the end of the command line. Be sure to include the entire file path if the Run File is not in a folder where Quality Analyst expects to find it.



- 7 Click **OK**.

Executing Run Files from Other Programs

Many of the programs that might be a part of your quality-control system (such as Microsoft Access or Microsoft Excel) can run other programs. (For example, Microsoft Access uses the SHELL command for this purpose, but other programs might use commands named something like RUN or EXEC or EXECUTE.)

You may find it useful to run the Quality Analyst Run File Interpreter from within those programs. To do so, create a command line (as explained under *Executing Run Files Using a Command Line* on page 104) and use it as an argument to the “shell” command of the other program. For example:

```
SHELL C:\QA\QARFI.EXE $RUN=CHART.RUN ^InFile="WEIGHT.DAT"
```

Command Line Options

The command parameters listed in the following table are available on a Run File Interpreter command line.

Command Parameters for Run Files

Syntax	Explanation
<code>\$RUN=filename.run</code>	Name of Run File to execute.
<code>\$TITLE="text"</code>	System title text (replaces default title).
<code>\$REPORTDATE="text"</code>	Text to use for Report Date
<code>\$DATA=folder</code>	Folder (directory) containing data files †
<code>\$TEMP=folder</code>	Folder (directory) to store temporary files †
<code>\$PLOT=folder</code>	Folder (directory) to store graphics files †
<code>\$QANWA=folder</code>	Folder (directory) containing QA.NWA configuration file †
<code><Replaceable Parameters></code>	One or more Replaceable Parameters. See examples below.
<code>@<command_file_name></code>	Name of a file containing command line options. (Used if options are too long for a command line.)

Note: Include and exclude quotation marks as indicated in this table.

† Under Windows 95/98/NT, folder names must be the MS-DOS (that is, short) names of the folders; long folder names are not allowed.

For example, the following Windows command line will start the Quality Analyst Run File Interpreter and execute the Run File ANALYZE.RUN. Every time the Run File Interpreter encounters “`^DatFileA`,” “`^VariableA`,” or “`^VariableB`” in ANALYZE.RUN, it will use the strings “ELEMENTS.DAT,” “PB,” and “CD” respectively, instead.

```
QARFI $RUN=ANALYZE.RUN ^DatFileA="ELEMENTS.DAT"
      ^VariableA="PB" ^VariableB="CD"
```

Similarly, the following Windows command line will start the Quality Analyst Run File Interpreter and execute the Run File ANALYZE.RUN. Every time the Run File Interpreter encounters “`^InFile`,” “`^Var1`,” or “`^Var2`” in ANALYZE.RUN, it will use the strings “ELEMENTS.DAT,” “PB,” and “CD” respectively, instead.

```
QARFI $RUN=ANALYZE.RUN ^InFile="ELEMENTS.DAT" ^Var1="PB"
      ^Var2="CD"
```

See *Replaceable Parameters* on page 92 and *Passing Replaceable Parameters from the Command Line* on page 94 for more information.

To use the @ Run File command parameter, create a text file with a single command parameter on each line, and include the name of the file in the Run File command following an "@" character. For example, suppose you want to execute the following Run File command.

```
QARFI $RUN=BASIC.RUN $TITLE="SEMIANNUAL REPORT: JANUARY-JUNE
      1996" ^DatFile1="JAN96.DAT" ^DatFile2="FEB96.DAT"
      ^DatFile3="MAR96.DAT" ^DatFile4="APR96.DAT"
      ^DatFile5="MAY96.DAT" ^DatFile6="JUN96.DAT"
      ^DateVar="DATE" ^ElemVar2="PB" ^ElemVar3="CD"
      ^ElemVar4="ZN" ^ElemVar5="HG" ^ElemVar6="CR"
      ^ElemVar7="MO" ^ElemVar8="MN" ^ElemVar9="AS"
```

Instead, you could create a file named SEMIANN.TXT with the following contents:

```
$RUN=BASIC.RUN
$TITLE="SEMIANNUAL REPORT: JANUARY-JUNE 1996"
^DatFile1="JAN96.DAT"
^DatFile2="FEB96.DAT"
^DatFile3="MAR96.DAT"
^DatFile4="APR96.DAT"
^DatFile5="MAY96.DAT"
^DatFile6="JUN96.DAT"
^DateVar="DATE"
^ElemVar2="PB"
^ElemVar3="CD"
^ElemVar4="ZN"
^ElemVar5="HG"
^ElemVar6="CR"
^ElemVar7="MO"
^ElemVar8="MN"
^ElemVar9="AS"
```

and then execute the following Run File command:

```
QARFI @SEMIANN.TXT
```

Run File Errors

Occasionally, the Run File Interpreter will encounter a command or command parameter that it does not expect or cannot execute properly.

The Run File Interpreter can handle most serious errors encountered in a Run File. When the interpreter encounters an error, a dialog box will appear containing an error message that will indicate the error and the location of the error in the Run File. The processing of the Run File will terminate whenever an error is encountered depending on how Quality Analyst is configured (see *Run File Error Handling* in Chapter 7).

The following table contains example error messages, the error(s) they represent, and explanations of the error and possible causes.

Run File Error Messages

Example Message:	*** FILE ('VENER.DAT') NOT FOUND IN RUN FILE 'VENEER.RUN' AT LINE 5
Error:	Input file not found
Explanation:	The line number of the Run File will be identified, as "LINE n", to help you correct the problem. You may have mistyped the file name or deleted, moved, or renamed the file.
Example Message:	*** INVALID VARIABLE NAME ('PART99') IN RUN FILE 'TEST.RUN' AT LINE 3
Error:	Variable name not found in the data file.
Explanation:	You may have mistyped the variable name or you may have typed the variable name from a different file.
Example Message:	*** INVALID FIRST CHART TYPE ('T') IN RUN FILE 'ANALYZE.RUN' AT LINE 1
Error:	Mistyped or missing parameter
Explanation:	For example, an erroneous chart type, such as "T" instead of "R" in the XRS command.
Example Message:	*** 'PRINTT' NOT FOUND
Error:	Wrong or missing Quality Analyst Run File command
Explanation:	Typographical error.

Tip: You can mitigate the effect of errors on the execution of lengthy Run Files by nesting Run Files. See the discussion under *RUN* on page 77.

DOS-based Run Files

Using Existing DOS-based Run Files Under Windows

There are several things you should be aware of if you are using Run Files written for an earlier DOS-based version of Quality Analyst. The Windows Run File Interpreter will interpret many DOS Run File commands correctly as follows:

- The DOS CSCREEN command will be interpreted as the Windows CHART command, displaying the graphics on the screen under Windows.
- The DOS CPRINT and CPLOT commands will be interpreted as the Windows PRINT command, sending output to the Windows printer.
- “Named” graphics drivers (user-supplied names like “laser1”) will be ignored. (No output will be produced.)
- The DOS I (Interactive) command for screen graphics *is not recognized*. (No output will be produced.)
- The ATTRIB command will be executed as ATTRIBUT which has been renamed as such to avoid a name conflict with a system-support command.

Converting DOS Run Files to Windows

Although most DOS-based Run Files will execute under Windows, you should eventually convert those Run Files.

To convert DOS Run Files to Windows:

- Change the screen-output command CSCREEN to CHART.
- Change the printer-output commands CPRINT and CPLOT to PRINT.
- Change ATTRIB to ATTRIBUT.

- Change named drivers to PRINT (if they are used to assemble a page layout, you should also rewrite the command statement to include a GROUP option).
- Change S, P, and T output types to W, add the name of a PTX file to the command line, and insert a CHART or PRINT command for the PTX file.
- Change the I command for screen-graphics output to G and add a new line containing CHART. For example, change:

```
XRS COOKIE.DAT WEIGHT X R I
```

to

```
XRS COOKIE.DAT WEIGHT X R G  
CHART
```

CHAPTER 12

Database Connectivity

Overview

Note: The features described in this chapter do not apply to **NWA Quality Analyst *without Connectivity***.

NWA Quality Analyst lets you define Quality Analyst Data Sets that use external databases as the source of data. This lets you connect to external data—either directly or via ODBC (*Open DataBase Connectivity*)—while retaining the complete functionality of Quality Analyst. This version of Quality Analyst provides seamless connections to any ODBC-compliant database and direct connections to many PC-based databases.

Using Quality Analyst, you define a connection between a Quality Analyst Data Set and an external data source. Once the link is established, using the external data is transparent to you. Quality Analyst stores the information needed to establish the connection to the external data source as part of the Data Set definition. Whenever you use a Connectivity Data Set, Quality Analyst simply uses the definition to read the data from the external data source instead of from a Quality Analyst data file. Quality Analyst Connectivity Data Sets are listed alongside the standard Quality Analyst Data Sets in the File Open dialog box.

Note: Refer to the *Setup and Tutorial* manual for important information about installing and configuring the connectivity components of NWA Quality Analyst.

Before Using Connectivity

Before you can link a Quality Analyst Data Set to an external database, you must comply with the following requirements.

Direct Connection

Quality Analyst can connect directly to “desktop” databases created by dBase, FoxPro, Microsoft Access, and Paradox. It can also connect directly to data stored in Microsoft Excel spreadsheets.

The Quality Analyst installation program installs the Microsoft Data Access Components (MDAC) which handles all direct connections to desktop databases. Installing additional drivers is not necessary.

ODBC Connection

Quality Analyst should connect to virtually any data source that supports ODBC. To connect to ODBC-compliant databases, you must:

- Have the appropriate ODBC drivers installed on your system
- Define an ODBC data source for each database you want to read (as discussed below)

This manual assumes that you comply with these requirements.

Appropriate Windows ODBC drivers are usually installed along with the database software whose data you want to connect to via ODBC. If not, you may need to run the setup program for your database software and add the ODBC drivers, or obtain drivers from one of many third-party sources.

Establishing an ODBC data source based on a particular ODBC driver is done using the ODBC setup normally found in the Windows Control Panel. Refer to your Windows documentation or contact your information systems support staff for information about setting up ODBC data sources.

Quality Analyst allows connections to Oracle and SQL Server databases via 32-bit ODBC using drivers installed as part of MDAC.

Defining Connectivity Data Sets

Quality Analyst provides a complete user interface for defining a Connectivity Data Set. You can define a Connectivity Data Set in one of two ways:

- Use Quality Analyst to create the definition. A convenient user interface lets you select each component from lists provided by querying the data source. You may select fields only from a single table in the data source selected. You can define any number of filters to limit the amount of data retrieved from the data source or to sort it on any field as it is retrieved.
- Provide a complete SQL statement that performs all the functions necessary to retrieve the desired data from the database. You can enter the SQL statement by composing it at the keyboard or by copying and pasting it from other applications. You should use this method if you need to perform multitable operations or complex manipulation to retrieve the correct data. (The Quality Analyst connectivity filters can not be used with the SQL statement.) Although Quality Analyst performs some syntax checking, you are responsible for the SQL statement and the consequences of its use.

Defining a Connectivity Data Set with Quality Analyst involves these steps:

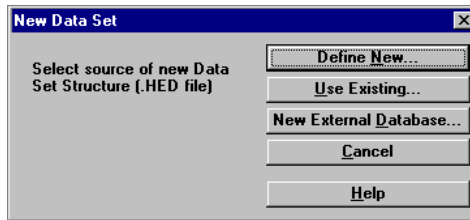
- 1 Select New External Database when creating a new Data Set.
- 2 Specify a connection type (either ODBC, or a specific database for direct connection).
- 3 Specify an external data source.
- 4 Do *one* of the following:
 - Select a table from the external data source
 - or**
 - Enter SQL commands to define the data within the data source to use.
- 5 Assign fields from the data source to columns in the Quality Analyst Data Set.
- 6 Specify data filters to limit the amount of data retrieved. (This step is available only if you select a table in step 4, and is optional.)

- 7 Specify how the data should be sorted as it is retrieved. (This step is available only if you select a table in step 4, and is optional.)

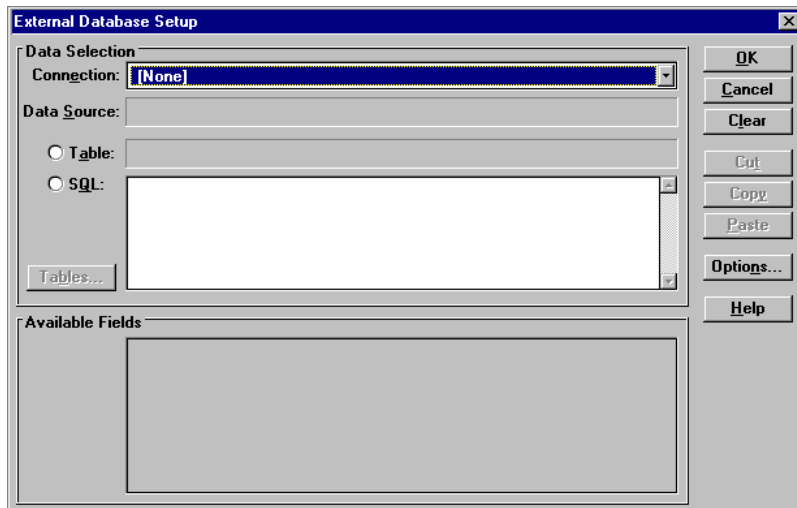
The rest of this section describes this process in detail.

Select New External Database

Begin by running Quality Analyst and select **Create a New Data Set** from the Quality Analyst startup screen. The **New Data Set** dialog box will appear.



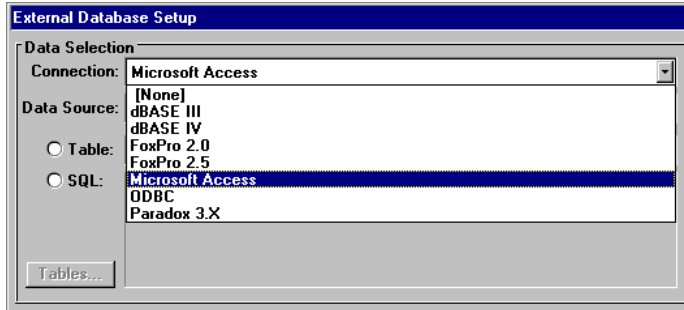
Select **New External Database**. The Quality Analyst **External Database Setup** dialog box will appear.



External Database Setup dialog box

Selecting a Connection Type

In the drop-down list labeled **Connection**, select the type of connection you want to make. Select **ODBC** to connect to an ODBC-compliant database, or select any of the database formats listed to connect to that type of database directly.

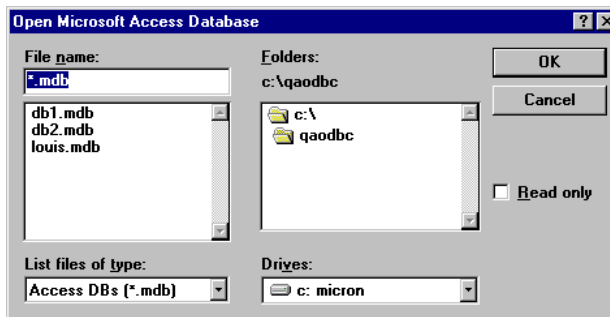


External Database Connection list

Selecting a Data Source

Direct-Connection Sources

If you selected one of the direct-connection (non-ODBC) database types, the Windows Open File dialog box will appear.

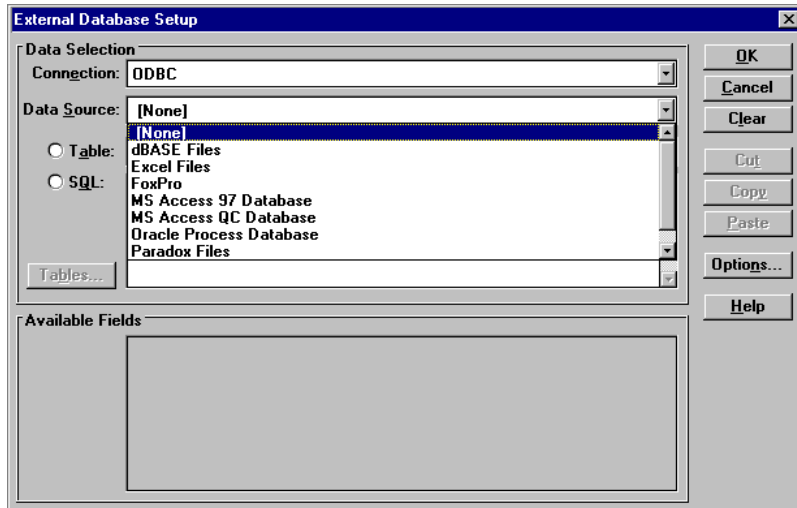


Selecting a direct-connection data source

The dialog box lists files of the selected database type (*.MDB for Microsoft Access, for example). Select the external data file you want to connect to. Make sure it is a file created by the database application you selected in the previous step. After you click **OK**, the name of the file you select will be displayed in the box labeled **Data Source**.

ODBC Sources

If you selected an ODBC connection, select the source of the data you want to use for the Data Set from the drop-down list. This list will contain all ODBC data sources available on your system. If no sources are listed, or if the desired source is not listed, you need to create an ODBC source definition using the ODBC module of Windows Control Panel. See *ODBC Connection* on page 114.



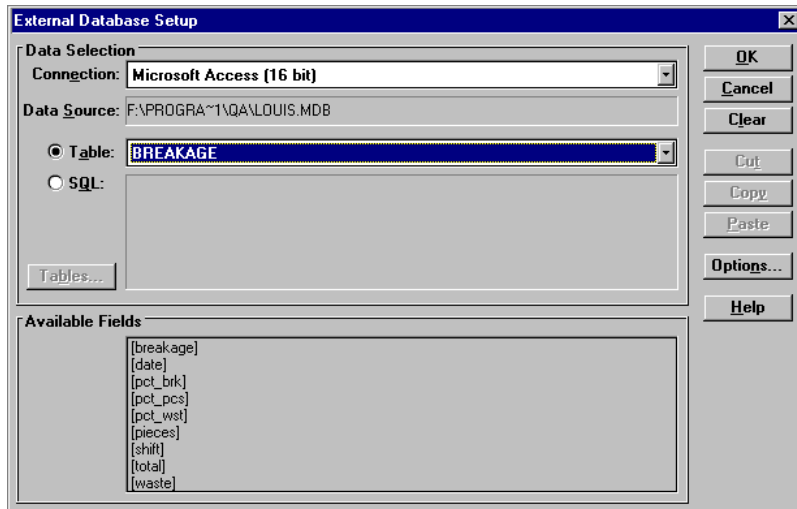
ODBC Sources drop-down list

Specifying a Data Table

With Quality Analyst, you can select an existing single table in your data source or specify SQL commands to manipulate one or more tables in the data source and create the table you need for your Quality Analyst Data Set. The next two sections discuss each option.

Using an Existing Table in the Data Source

To use an existing table in the external data source, select the **Table** option in the **External Database Setup** dialog box. From the adjacent drop-down list, select the table you want to use. This list contains the names of all tables in the data source you selected. In this example, we selected the **BREAKAGE** table:



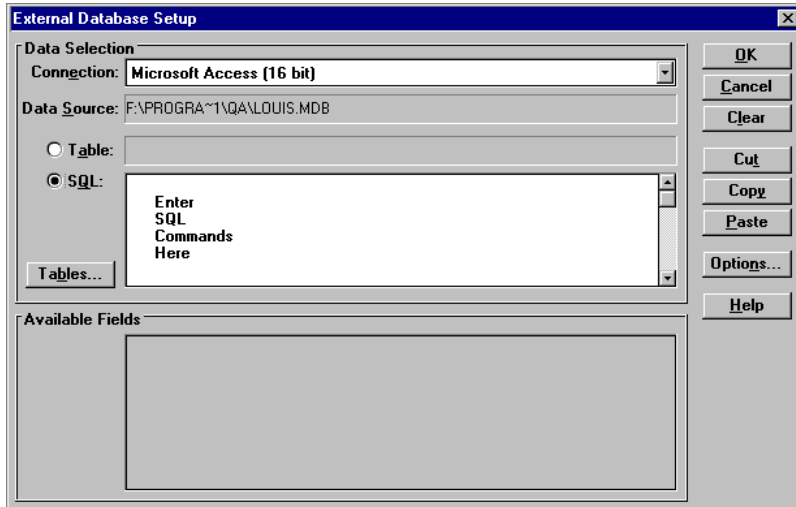
External Database Setup dialog box with Data Source and Table selected

When you select a table, the names of the fields in that table will be displayed in the area of the dialog box labeled **Available Fields**. You can use this list of fields to confirm that you have selected the correct table.

When you have selected the table, click **OK** and skip to the section below on *Assigning Database Fields to Quality Analyst Variables*.

Using SQL to Create a Data Table

To use SQL commands to create a data table, select **SQL** in the **External Database Setup** dialog box. The large adjacent box will become active, as shown below.



External Database Setup dialog box with SQL option selected

Enter the SQL commands that perform the operations required to define the data to which you will link your Data Set. (You can use the **Paste** button on the right to insert SQL statements copied to the Windows clipboard from other applications.)

Tip: If you type the statements into the dialog box directly, begin a new line by pressing <CTRL+ENTER>.

Preparing SQL Statements

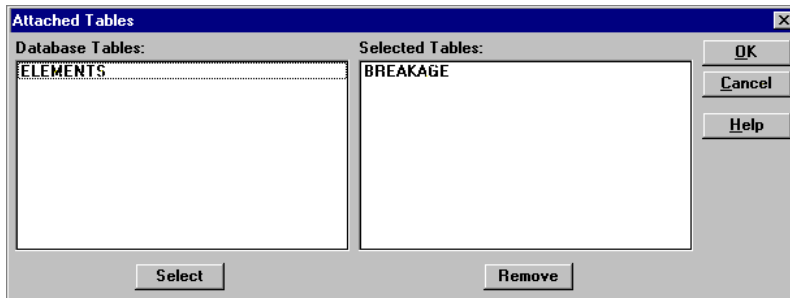
SQL, with its not-quite-standard syntax and cryptic error messages, can be difficult to work with. Quality Analyst passes the commands you enter directly to the data source with little or no error checking. We suggest building your SQL statements in Microsoft Access using its QBE (query by example) feature, copying them to the clipboard, and pasting them into Quality Analyst. The section *Configuring and Troubleshooting*, beginning on page 151, contains important instructions for configuring Quality Analyst to connect to external databases.

Note: You can use Quality Analyst's **Replaceable Parameters** in your SQL statements for added flexibility in querying the source data. See *Using*

Replaceable Parameters with Filter Criteria in SQL Statements
starting on page 137 for details.

Defining Attached Tables

If you use SQL to connect to a database such as Oracle using “attached” Microsoft Access files, you must tell Quality Analyst which tables in the target database the SQL statements will refer to. To do this, click the **Tables** button to display the **Attached Tables** dialog box.



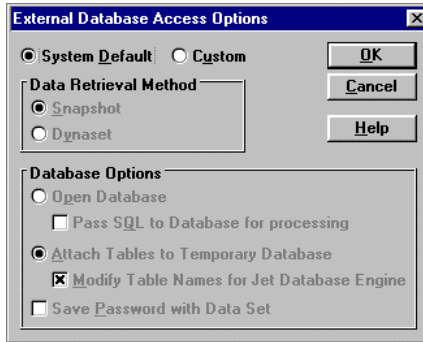
To select a table, highlight it in the list of **Database Tables** on the left, and click **Select**. The name of the table will move to the list of **Selected Tables** on the right. Refer to the section *Configuring and Troubleshooting*, beginning on page 151, for more information.

Completing the SQL Query

When you have entered the SQL query, you can press <ENTER> to refresh the list of **Available Fields**, or click **OK** to execute the SQL statements.

Setting External Database Access Options

In the External Database Setup dialog box, clicking **Options** displays the External Database Access Options dialog box which lets you override, at the file level, several global connectivity options. This makes it easier for you to connect to multiple database types.



Select **System Default** to use the global settings. (Configuring these global settings is explained under *Configuring Connectivity* on page 152.)

Select **Custom** to override the global parameters with the parameters you specify in this dialog box. The parameters will be saved for the current Data Set in its header (HED) file.

For explanations of these settings, see *Data Retrieval* on page 154 and *Database Options* on page 152.

Assigning Database Fields to Quality Analyst Variables

Once the external database definition is completed, the **Database Connectivity Definition** dialog box will appear. The upper half of this dialog box contains the Database Connectivity Definition form where each column in the Quality Analyst Data Set is defined. Each row in this grid defines a column in the Data Set you are creating. The lower half of the dialog box contains two tabs labeled **External Database Fields** (discussed in this section) and **External Database Filters** (discussed in the next section—*Defining Connectivity Data Filters*).

	Name	Type	External Database Field
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

External Database Fields | External Database Filters

Connection: ODBC
 Data Source: MS Access 97 Database
 Table: BREAKAGE
 Fields: [breakage], [date], [pct_brk], [pct_pcs], [pct_wst], [pieces], [shift], [total]

Database Connectivity Definition dialog box

You define your new Data Set by specifying a name and data type for each column in the Data Set and then assigning a field from the external data source to the data column using the **External Database Fields** tab.

Select the **External Database Fields** tab at the bottom of the dialog box. At the top of the tab, the **Connection** type, **Data Source**, and **Table** are displayed. Below that, all **Fields** requested from the data source are listed in the format [table].[field_name]. You need not include all of these fields in your Data Set.

You can fill out the Database Connectivity Definition form in any order you find convenient. For example, you can:

- select all fields at once (using the **Assign All** button), modify the automatically assigned variable names and data types, if desired, and add descriptions; or
- enter all the column names, data types, and descriptions, then assign database fields; or
- select all applicable database fields first, then enter a name, data type, and description for each; or

- fill in one row at a time, entering a name and data type, assigning a database field, and entering a description for each as you go.

Selecting Fields Individually

In the following example, we will start by selecting the desired fields from the database data source and then enter a name and data type for each.

To select a database field:

- 1 In the Database Connectivity Definition form, select an empty cell under **External Database Field**.
- 2 In the **External Database Fields** tab:
 - select a field in the **Fields** list and click **Assign**,
or
 - double-click a field in the **Fields** list.

The field name will appear as the **External Database Field** in the Database Connectivity Definition form.

In the following figure, the fields “date” and “shift” have been selected for the first two data columns and the field “pet_brk” is about to be selected for the third.

Database Connectivity Definition

	Name	Type	External Database Field
1	DATE	D	[date]
2	SHIFT	A	[shift]
3			
4			
5			
6			
7			
8			
9			
10			

External Database Fields External Database Filters

Connection: ODBC

Data Source: MS Access 97 Database

Table: BREAKAGE

Fields:

- [breakage]
- [date]
- [pct_brk]
- [pct_pcs]
- [pct_wst]
- [pieces]
- [shift]
- [total]

Buttons: OK, Cancel, Insert, Delete, Cut, Paste, Remove, External..., Assign, Assign All, Help

Selecting External Database Fields for a Data Set

After you have assigned External Database fields to Data Set columns, enter a variable name and data type for each variable.

	Name	Type	External Database Field	
1	DATE	D	[date]	
2	SHIFT	A	[shift]	
3	BREAKAGE	I	[breakage]	
4	PCT_BRK	I	[pct_brk]	
5	PCT_PCS	I	[pct_pcs]	
6	PCT_WST	I	[pct_wst]	
7	PIECES	I	[pieces]	
8	WASTE	I	[waste]	
9	TOTAL	I	[total]	
10				

External Database Fields | External Database Filters

Connection: ODBC
 Data Source: MS Access 97 Database
 Table: BREAKAGE
 Fields: [breakage], [date], [pct_brk], [pct_pcs], [pct_wst], [pieces], [shift], [total]

A full definition of a Quality Analyst External Data Set

The variable name you enter must conform to Quality Analyst rules for variable names:

- They must be two to eight characters long.
- They may contain letters (“A”–“Z”), numerals (“0”–“9”), or the underscore (“_”).
- They must begin with a letter.
- Each must be unique within a data file.

The data type must be a legal Quality Analyst data type. (See *Data Type* on page 36.) External database field names, on the other hand, need not conform to Quality Analyst’s rules for naming variables.

Selecting All Fields

If you want to use all fields in your external data source, click **Assign All** in the **External Database Fields** tab. Quality Analyst will copy all field names to the **External Database Field** column and the first eight characters of all field names to the **Name** column.

Database Connectivity Definition			
	Name	Type	External Database Field
1	BREAKAGE	I	[breakage]
2	DATE	D	[date]
3	PCT_BRK	I	[pct_brk]
4	PCT_PCS	I	[pct_pcs]
5	PCT_WST	I	[pct_wst]
6	PIECES	I	[pieces]
7	SHIFT	A	[shift]
8	TOTAL	I	[total]
9	WASTE	I	[waste]
10			

External Database Fields		External Database Filters	
Connection:	ODBC	External...	
Data Source:	MS Access 97 Database	Assign	
Table:	BREAKAGE	Assign All	
Fields:	[date] [pct_brk] [pct_pcs] [pct_wst] [pieces] [shift] [total] [waste]	Help	

Effect of the Assign All button

You can then edit the column names (if you want) and specify data types for each data column. (Assign All will overwrite any existing contents in the column definition form.)

Note: Using Assign All can result in duplicate field names if, for example, the first eight characters of two external database fields with longer names are the same. When you click OK, Quality Analyst will detect this and ask you to correct it.

Using Date-and-Time Fields

Many database programs store date and time as a single field. Quality Analyst, however, requires that date and time data be stored as separate variables. Quality Analyst handles this conversion automatically based on the data type that you assign to the variable associated with the External Database field. To convert a date-time field to separate Quality Analyst variables:

- 1 Add the date-time field to the list of **External Database Fields**.
- 2 Enter "DATE" in the **Name** column in the same row.

- 3 Enter “D” in the **Type** column in the same row.
- 4 Add the date-time field again to the list of **External Database Fields**.
- 5 Enter “TIME” in the **Name** column in the same row.
- 6 Enter “T” in the **Type** column in the same row.

For example, if you had performed the preceding steps on a date-time field named [breakage].[timestamp], your data definition would look something like this:

Sample Data Definition for converting a date-time field

	Name	Type	External Database Field	Description
1	DATE	D	[breakage].[timestamp]	Date Produced
2	TIME	T	[breakage].[timestamp]	Time Produced

Because of the data types specified, Quality Analyst will extract the date part of [breakage].[timestamp] and assign it to DATE and the time part of [breakage].[timestamp] and assign it to TIME.

Defining Connectivity Data Filters

On the **External Database Filter** tab in the Database Connectivity Definition dialog box you can define filters to limit the amount of data retrieved from the data source.

The screenshot shows the 'Database Connectivity Definition' dialog box with the 'External Database Filters' tab selected. The dialog is divided into several sections:

- Table:** A table with columns 'Name', 'Type', and 'External Database Field'. It lists 10 items: DATE (D, [date]), SHIFT (A, [shift]), BREAKAGE (I, [breakage]), PCT_ALL (I, Calculated), PCT_BRK (I, [pct_brk]), PCT_PCS (I, [pct_pcs]), PCT_WST (I, [pct_wst]), PIECES (I, [pieces]), WASTE (I, [waste]), and TOTAL (I, [total]).
- External Database Fields:** A tab that is currently not selected.
- External Database Filters:** The active tab, containing:
 - Filters:** A list box for defining filters, with radio buttons for 'Match:', 'Like:', and 'From:'. The 'From:' option includes 'To:' fields.
 - Time Filters:** Radio buttons for 'During the last [] month(s)' and 'During the last [] day(s)'. The 'month(s)' option is selected.
 - Order by:** Three dropdown menus, all currently set to '[None]'. A 'Reset' button is located to the right.
 - Sort Order:** Radio buttons for 'Ascending' (selected) and 'Descending'.
 - Help:** A button located at the bottom right of the tab.
- Buttons:** A vertical column of buttons on the right side of the dialog: OK, Cancel, Insert, Delete, Cut, Paste, and Remove.

External Database Filters tab of the Database Connectivity Definitions dialog box

(By default, Quality Analyst will retrieve the entire table from the data source when you open the Data Set.) The filter criteria you specify here become part of the data definition stored in the header file of the Data Set. When you open an external Data Set that includes filter criteria, Quality Analyst translates the criteria into SQL commands and sends them to the external source so the data are filtered by the external source before being sent back to Quality Analyst.

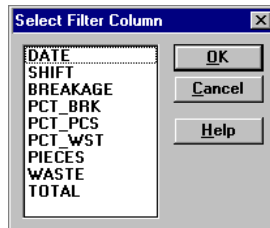
Note: The **External Database Filter** tab is only accessible if your Data Set is based on the single table you selected from the drop-down list in the **External Database Setup** dialog box. If you chose instead to base your Data Set on an SQL query, the External Database Filters tab will not be available and the SQL commands you enter must perform all necessary filtering.

When using connectivity filters, please note:

- You can define any number of filter criteria. Only data records meeting *all* criteria will be returned to Quality Analyst. (That is, “AND” logic is used.)
- Any External Database field assigned to the Quality Analyst Data Set may be used in a filter.
- You can specify filters that return records with fields that match specific values or that fall within a range of values. (See explanations below.)
- Filters based on a date field can be set to return records from a specified number of months or days ending today.
- If you change the definition of the Connectivity Data Set, the External Database filter(s) may be cleared. Quality Analyst will warn you that, unless you redefine the filter before you click **OK**, the entire (unfiltered) table will be retrieved.
- The **Reset** button clears (erases) all current filter criteria and reestablishes the lists of available fields (Columns). (You may need to click **Reset** to initialize the filters when you first create the Data Set.)

Defining Filters to Retrieve Matching Data**To define a connectivity filter to retrieve matching data:**

- 1 Click **New** and select the database field on which you want to base the filter.



The field name will appear in the list of **Filters** on the left and in the **Column** box.

- 2 Select **Match**.
- 3 Do one of the following:

- Enter a value to match in the adjacent box. This must be a legal value for that variable type.
- Select **(Ask)** from the drop-down list if you want to specify this filter criterion each time you use the database. (See *Using the (Ask) Settings* on page 134.)

Here is an example of a match filter that will retrieve only data produced on June 1, 1997.

The screenshot shows a dialog box titled "Filters". On the left, a list box contains "DATE". To the right, the "Match:" radio button is selected, and its dropdown menu shows "6/1/97". Below it, the "Like:", "From:", and "To:" options are unselected. At the bottom, the "During the last" options for "month(s)" and "day(s)" are also unselected. There are "New..." and "Remove" buttons at the bottom left.

Example of a matching-data filter

In this example, the user will be prompted for a match date when the database is opened.

The screenshot shows the same "Filters" dialog box. The "Match:" radio button is selected, and its dropdown menu shows "(Ask)". All other options and buttons are the same as in the previous screenshot.

Example of a prompted matching-data filter

Defining Filters to Retrieve Similar Data

To define a connectivity filter to retrieve similar data:

- 1 Click **New** and select the database field on which you want to base the filter. The field name will appear in the list of **Filters** on the left and in the **Column** box.
- 2 Select **Like**.
- 3 In the adjacent box, do one of the following:

- Enter the string you want to search for, using question marks and asterisks as “wild cards.” Use a question mark (“?”) to represent any single character and an asterisk (“*”) to represent one or more characters.
- Select **(Ask)** from the drop-down list if you want to specify this filter criterion each time you use the database. (See *Using the (Ask) Settings* on page 134.)

Here is an example of a filter that will retrieve only data records in which the BATCH field begins with 1178-, ends with -A, and contains any three characters in between.

The screenshot shows a 'Filters' dialog box. On the left, a list of filters contains 'BATCH'. Below this list are 'New...' and 'Remove' buttons. On the right, there are radio buttons for 'Match:', 'Like:', 'From:', and 'To:'. The 'Like:' radio button is selected, and its corresponding text box contains '1178-???-A'. Below these are two rows of radio buttons for 'During the last' with input boxes for 'month(s)' and 'day(s)'.

Example of a similar-data filter

Defining Filters to Retrieve Ranges of Data

To define a connectivity range filter:

- 1 Click **New** and select the database field on which you want to base the filter. The field name will appear in the list of **Filters** on the left and in the **Column** box.
- 2 Select **From**.
- 3 Select a lower-limit condition from the adjacent drop-down list:
 - Select “=” to include records where the field value is exactly equal to the lower limit.
 - Select “>” to exclude records where the field value is exactly equal to the lower limit. Values “just greater than” the lower limit will be included.
- 4 In the adjacent box, do one of the following:
 - Enter the lower limit of the desired range.

- Select **(Ask)** from the drop-down list if you want to specify the lower limit each time you use the database. (See *Using the (Ask) Settings* on page 134.)
- 5 Next to **To:**, select an upper-limit condition from drop-down list:
- Select “=” to include records where the field value is exactly equal to the upper limit.
 - Select “<” to exclude records where the field value is exactly equal to the upper limit. Values “just less than” the upper limit will be included.
- 6 In the adjacent box, do one of the following:
- Enter the upper limit of the desired range.
 - Select **(Ask)** from the drop-down list if you want to specify the upper limit each time you use the database. (See *Using the (Ask) Settings* on page 134.)

Here is an example of a range filter that will retrieve only data collected during June 1997.

The screenshot shows a dialog box titled "Filters". On the left, there is a list box containing "DATE". To the right, there are several options:

- Match: [dropdown]
- Like: [text input]
- From: [=] [dropdown] 6/1/1997 [dropdown]
- To: [=] [dropdown] 6/30/1997 [dropdown]
- During the last [dropdown] month(s)
- During the last [dropdown] day(s)

 At the bottom left, there are two buttons: "New..." and "Remove".

Example of a data-range filter

You can define an “open-ended” range filter by leaving either the **To** or the **From** entry blank. Here is an example of a range filter that will retrieve only data collected before June 30, 1997 (“**To** <”).

The screenshot shows a dialog box titled "Filters". On the left, there is a list box containing "DATE". To the right, there are several options:

- Match: [dropdown]
- Like: [text input]
- From: [=] [dropdown] [blank] [dropdown]
- To: [<] [dropdown] 6/30/1997 [dropdown]
- During the last [dropdown] month(s)
- During the last [dropdown] day(s)

 At the bottom left, there are two buttons: "New..." and "Remove".

Example of an “up to” data-range filter

Here is an example of a range filter that will retrieve only data collected on or after June 1, 1997 (“From =”).

The screenshot shows a dialog box titled "Filters". On the left, there is a list box containing "DATE". Below the list box are two buttons: "New..." and "Remove". On the right side of the dialog, there are several options:

- Match: [dropdown menu]
- Like: [text input field]
- From: [dropdown menu] 6/1/1997 [dropdown menu]
- To: [dropdown menu] [text input field]
- During the last [dropdown menu] month(s)
- During the last [dropdown menu] day(s)

Example of an “above” data-range filter

To define a “preceding period” connectivity filter:

- 1 Click **New** and select the database field on which you want to base the filter. The field name will appear in the list of **Filters** on the left and in the **Column** box.
- 2 Select **During the last** [] month(s) or **During the last** [] day(s).
- 3 In the corresponding box, do one of the following:
 - Enter a number of months or days (*including* today).
 - Select **(Ask)** from the drop-down list if you want to specify the number of months or days each time you use the database. (See *Using the (Ask) Settings* below.)

Here is an example of a “preceding period” filter that, if executed on June 30, 1997, would retrieve only data collected from June 24 through June 30, 1997.

The screenshot shows the same "Filters" dialog box. The list box on the left still contains "DATE". The "During the last" options are now:

- During the last [] month(s)
- During the last 7 [dropdown menu] day(s)

 The dropdown menu for the "7" days is open, showing the word "(Ask)" selected.

Example of a “preceding period” filter

Using the (Ask) Settings

When you select the **(Ask)** option for any filter criterion, Quality Analyst inserts an appropriate SQL statement in the header (.HED) file of the Data Set and

sends that SQL query to the database when you connect to it. These SQL statements use the Replaceable Parameter syntax described under *Prompting for Filter Criteria at Run Time* on page 138. For example, if you selected (Ask) for a Match filter for the variable DATE in the database DB then Quality Analyst would include the following SQL statement in the header file for the Data Set:

```
SELECT * FROM DB WHERE DATE=#^DATE!MatchFilter#
```

When you connect to the database, either by opening the Data Set with the Quality Analyst Editor or by using a CONNECT command in a Run File, you will be prompted for the appropriate filter criterion in an appropriately labeled dialog box. (The examples under *Prompting for Filter Criteria at Run Time* on page 138 show samples of these dialog boxes.)

If you do not want to be prompted for this filter criterion, but would rather pass values from the command line or set values within a Run File, you **must** use this same syntax in the command line or the SET command. For example:

```
YES: SET ^DATE!MatchFilter = "10/1/99"
```

```
NO: SET ^DATE = "10/1/99"
```

```
NO: SET ^BeginDate="10/1/99"
```

```
YES: QARFI.EXE $RUN=WTTASK.RUN ^DATE!MatchFilter="10/1/99"
```

```
NO: QARFI.EXE $RUN=WTTASK.RUN ^DATE="10/1/99"
```

```
NO: QARFI.EXE $RUN=WTTASK.RUN ^BeginDate="10/1/99"
```

Removing Filters

To remove a connectivity filter:

- 1 Select the filter in the list of **Filters** at the far left of the dialog box.
- 2 Click **Remove** below the list.

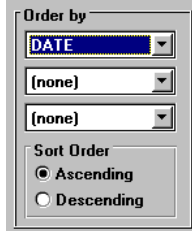
Sorting Filtered Data

In the **Order By** area of the External Database Filters tab, you can specify how you want the retrieved table to be sorted when it is brought in to Quality

Analyst. You can sort the data on up to three keys in either ascending or descending order.

To sort filtered data:

- 1 Select from one to three sort keys from the drop-down lists.
- 2 In the **Sort Order** area, select **Ascending** or **Descending** order. The specified order will be applied to all three keys.



Completing Data Definition

When the Data Set definition is complete, select **OK**. Quality Analyst will process the resulting connectivity request, write the retrieved data table to the data file of the Quality Analyst Data Set, and display the data in the Quality Analyst Editor.

Row	DATE 1---(D)	SHIFT 2---(A)	PCTBRK 3---(I)	PCTPCS 4---(I)	PCTWST 5---(I)
1	1/3/89	1	6.23	2.91	3.33
2	1/3/89	2	2.48	0	2.48
3	1/3/89	3	9.98	5.17	4.81
4	1/4/89	1	4.25	1.35	2.9
5	1/4/89	2	3.67	2	1.67
6	1/4/89	3	7.24	2.57	4.66
7	1/5/89	1	17.49	15.32	2.16
8	1/5/89	2	3.5	.32	3.17
9	1/5/89	3	7.43	3.34	4.09
10	1/6/89	1	4.27	0	4.27
11	1/6/89	2	1.67	0	1.67
12	1/6/89	3	5.77	1.62	4.15
13	1/9/89	1	2.93	1.19	1.74
14	1/9/89	2	6.23	2.65	3.57
15	1/9/89	3	3.28	0	3.28
16	1/10/89	1	6.98	3.25	3.73
17	1/10/89	2	5.06	2.67	2.39
18	1/10/89	3	4.35	.88	3.47
19	1/11/89	1	5.4	3.24	2.16
20	1/11/89	2	6.99	2.63	4.36

Date produced: Mode: ROW

New Connectivity Data Set displayed in the Quality Analyst Editor

Quality Analyst will save the connectivity information in the header file of the new Connectivity Data Set. When you close the Data Set, the data file will be retained on disk. Each time Quality Analyst uses the Data Set, it will try to reread the data from the external source. If it is unable to do so, it will offer to use the data file retained from the last successful retrieval.

Using Replaceable Parameters with Filter Criteria in SQL Statements

If your Connectivity Data Set uses an SQL query to filter the database, you can use Replaceable Parameters within your SQL statements to have Quality Analyst:

- prompt you for a filter criterion as it connects to the database,
- use a filter criterion that is set within a Run File, or
- use a filter criterion that is passed from the command line.

For example, let's look at this typical SQL statement:

```
SELECT PRODUCT, WT FROM DB WHERE DATE>=#10/1/99#
```

This statement retrieves the fields PRODUCT and WT from a database named DB for records where the value of the DATE field is on or later than 10/1/99.

But rather than “hard-wire” the date 10/1/99 into the query, Quality Analyst lets you supply the date each time the database is queried. There are three ways you can supply this value:

- by having Quality Analyst prompt for it at run time,
- by setting it within a Run File, or
- by setting it on the command line for the Run File Interpreter.

Prompting for Filter Criteria at Run Time

To have Quality Analyst prompt you at run time for values of Replaceable Parameters to be used as filter criteria, use the following syntax:

```
^[column name] (prompt type)
```

Command Parameter	Description
[column name]	the column name as it appears in the External Database Filters tab of the Database Connectivity Definition dialog box †
(prompt type)	the type of prompt to display, when applicable: <ul style="list-style-type: none"> “!FromFilter” prompts for the beginning value of a range. ‡ “!LastDaysFilter” prompts for the number of recent days to include. “!LastMonthsFilter” prompts for the number of recent months to include. “!LikeFilter” prompts for a “mask” or “template” with wildcards. “!MatchFilter” prompts for an exact value for the variable. ‡ “!ToFilter” prompts for the ending value of a range. ‡

† If the column is one of a multicolumn subgroup, add a colon and the column number to the variable name (for example, WEIGHT:3).

‡ Wording in the prompt will reflect the data type of the variable and whether the value entered will be included or excluded.

All (**prompt type**)s cause Quality Analyst to display an appropriately labeled dialog box which prompts for user input and checks it for validity. Additionally, the **!FromFilter**, **!MatchFilter**, and **!ToFilter** prompt types are labeled differently depending on the data type of the [**column name**] specified.

Though (**prompt type**) is optional, we highly recommend using it because it reduces the likelihood of user errors. If you omit it, Quality Analyst displays a generically labeled dialog box and does not validate input.

Example: Let's say you have a Run File named LUMBTASK.RUN that connects to a Connectivity Data Set named LUMBER.DAT. If the pertinent section of the Run File contains the line:

```
CONNECT LUMBER.DAT
```

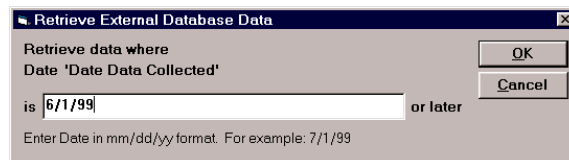
and your SQL contains the statement:

```
SELECT * FROM lumber WHERE Date>=#^Date!FromFilter#
```

and you start the Run File Interpreter with the command line:

```
QARFI.EXE $RUN=LUMBTASK.RUN
```

then Quality Analyst will display this dialog box:



Note that the dialog box displays the variable name (**DATE**), the description of the variable ("**Date Data Collected**") taken from the header (.HED) file of the Data Set, and appropriate labeling for the data-entry box based on the data type (Date) and the ">=" preceding the first "#" in the SQL statement. If the user enters 6/1/99 and clicks **OK**, Quality Analyst replaces

“**^Date!FromFilter**” in the SQL statement with “6/1/99” and passes the SQL statement to the database as:

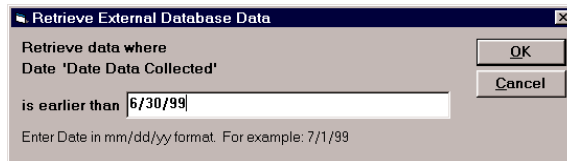
```
SELECT * FROM lumber WHERE Date>=#6/1/99#
```

Here are some examples of other prompting dialog boxes and the SQL statements that produce them.

This SQL statement:

```
SELECT * FROM Lumber WHERE Date<#^Date!ToFilter#
```

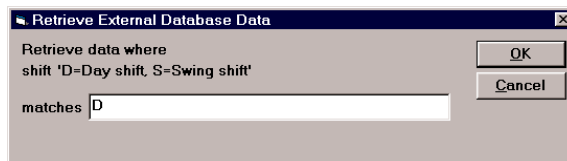
produces the following dialog box.



This SQL statement:

```
SELECT * FROM lumber WHERE shift=#^shift!MatchFilter"
```

produces the following dialog box.



This SQL statement:

```
SELECT * FROM products WHERE ProductName LIKE  
" ^PRODNAME!LikeFilter "
```

produces the following dialog box.

Retrieve External Database Data

Retrieve data where
 PRODNAME 'Name of Product'
 is like *lager

Enter ? to match any single character, * to match any string of characters

OK
 Cancel

This SQL statement:

```
SELECT * FROM Lumber WHERE Date > DATEADD("d", -
    ^Date!LastDaysFilter, NOW())
```

produces the following dialog box.

Retrieve External Database Data

Retrieve data where
 Date 'Date Data Collected'
 is within the last 7 day(s)

OK
 Cancel

This SQL statement:

```
SELECT * FROM Lumber WHERE Date > DATEADD("m", -
    ^Date!LastMonthsFilter, NOW())
```

produces the following dialog box.

Retrieve External Database Data

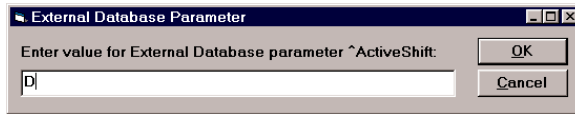
Retrieve data where
 Date 'Date Data Collected'
 is within the last 2 month(s)

OK
 Cancel

This SQL statement:

```
SELECT * FROM Lumber WHERE shift="^ActiveShift"
```

produces the following dialog box.



Setting Filter Criteria within Run Files

Let's say you have a Run File named WTTASK.RUN that connects to a Connectivity Data Set named WEIGHT.DAT. If the pertinent section of the Run File contains the lines:

```
SET ^BegDate = "10/1/99"
CONNECT WEIGHT.DAT
```

and your SQL contains the statement:

```
SELECT PRODUCT, WT FROM DB WHERE DATE>=#^BegDate#
```

and you start the Run File Interpreter with the command line:

```
QARFI.EXE $RUN=WTTASK.RUN
```

then Quality Analyst will take "10/1/99" from the SET command, substitute it for "^BegDate" in the SQL statement, and pass the SQL statement to the database as:

```
SELECT PRODUCT, WT FROM DB WHERE DATE>=#10/1/99#
```

Quality Analyst recognizes the caret ("^") as the beginning of a Replaceable Parameter and processes the following characters accordingly.

If the definition of your Connectivity Data Set includes an (Ask) setting, you can specify a value for that filter criterion within a Run File using a SET command and avoid being prompted for the criterion. If you do so, be sure to use the syntax described under *Prompting for Filter Criteria at Run Time* on page 138. See *Using the (Ask) Settings* on page 134 for more information.

Passing Filter Criteria from the Command Line

Let's say you have a Run File named WTTASK.RUN that connects to a Connectivity Data Set named WEIGHT.DAT. If the pertinent section of the Run File contains the line:

```
CONNECT WEIGHT.DAT
```

and your SQL contains the statement:

```
SELECT PRODUCT, WT FROM DB WHERE DATE>=#^FromDate#
```

and you start the Run File Interpreter with the command line:

```
QARFI.EXE $RUN=WTTASK.RUN ^FromDate ="10/1/99"
```

then Quality Analyst will take "10/1/99" from the command line and substitute it for the Replaceable Parameter "^FromDate" in the SQL statement and pass the SQL statement to the database as:

```
SELECT PRODUCT, WT FROM DB WHERE DATE>=#10/1/99#
```

Quality Analyst recognizes the caret ("^") as the beginning of a Replaceable Parameter and processes the following characters accordingly.

If the definition of your Connectivity Data Set includes an **(Ask)** setting, you can pass a value for that filter criterion to the Run File Interpreter on the command line and avoid being prompted for it. If you do so, be sure to use the syntax described under *Prompting for Filter Criteria at Run Time* on page 138. See *Using the (Ask) Settings* on page 134 for more information.

Using Connectivity Data Sets

Using a Connectivity Data Set is as easy as using a standard Quality Analyst Data Set. In the Quality Analyst Editor, when you select **Open** from the **File** menu, Connectivity Data Sets are listed among the standard Data Sets.

When you select a Connectivity Data Set to open, there is a short delay while Quality Analyst connects to the data source and retrieves the data. If you defined the Data Set to prompt you for any filter criteria (by specifying **(Ask)** as a filter value, for example), one (or more) of several possible dialog boxes may

appear. (See *Prompting for Filter Criteria at Run Time* on page 138 for examples of possible dialog boxes.)

After the data is retrieved from the source, Quality Analyst stores it as a standard Quality Analyst Data Set, so you will experience the same performance level, after the file is open, as when you work with a standard Quality Analyst Data Set.

Also, when Quality Analyst opens a Connectivity Data Set, it will use the connection definition stored in the header file of the Data Set. If the structure or location of the database has changed since the Data Set was created, Quality Analyst may have problems retrieving the data. The error messages produced may not be useful in identifying the source of the error since the problem may be the database application, a network problem, configuration changes, and so on. If you encounter such problems, refer to *Configuring and Troubleshooting* beginning on page 151, or contact your Information Systems administrative or support staff before calling NWA.

Modifying Connectivity Data Sets

After you have created a Connectivity Data Set, you can change the definition of the Data Set by:

- adding or removing fields,
- changing field assignments,
- adding or changing an External Database filter, or
- changing the External Database source file or table.

As you change the data definition, you may experience delays or pop-up messages. This is because Quality Analyst may connect and reconnect to the data source, sending commands and receiving results, as you redefine the Data Set. The nature and length of these delays will depend on the type and location of the data source. Remember, though, that once your Data Set is defined and opened, you can expect the same performance from a Connectivity Data Set as from a standard Quality Analyst Data Set.

To change the External Database definition, select **Database Connectivity Definition** from the **Edit** menu. The **Database Connectivity Definition** dialog box will appear.

Database Connectivity Definition			
	Name	Type	External Database Field
1	DATE	D	[date]
2	SHIFT	A	[shift]
3	BREAKAGE	I	[breakage]
4	PCT_BRK	I	[pct_brk]
5	PCT_PCS	I	[pct_pcs]
6	PCT_WST	I	[pct_wst]
7	PIECES	I	[pieces]
8	WASTE	I	[waste]
9	TOTAL	I	[total]
10			

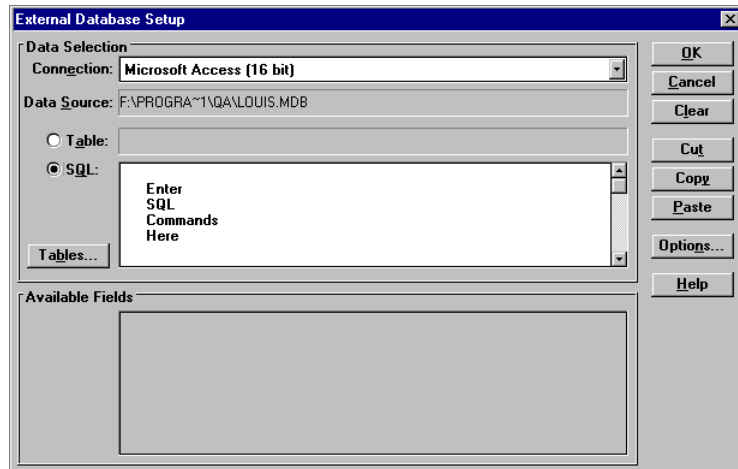
External Database Fields		External Database Filters	
Connection:	ODBC	External...	
Data Source:	MS Access 97 Database	Assign	
Table:	BREAKAGE	Assign All	
Fields:	[breakage] [date] [pct_brk] [pct_pcs] [pct_wst] [pieces] [shift] [total]	Help	

Database Connectivity Definition dialog box

It should look just as it did when you finished defining the External Database columns when you created the Data Set. Remember, the rows in this form define columns in the Data Set.

To change the Connection, Data Source, or Table for the Data Set:

Click the **External** button. The **External Database Setup** dialog box will appear.



- 1 Select a new **Connection** from the drop-down list.
- 2 Click **Select**, and select a new **Data Source** either from the drop-down list or the Windows Open File dialog, as described under *Selecting a Data Source* on page 117.
- 3 Select a new **Table** from the drop-down list,
or
- 4 Select **SQL** and, in the adjacent box, enter SQL commands that will create the table you want to use for the Data Set. (Specify target tables, if necessary, using the **Tables** button.)

Note: If you change the **Data Source** or **Table**, all field references and data filters will be deleted. The Quality Analyst column names will be retained but you may need to modify them to match the fields you choose from the new source or table for the new data definition.

To add a new column to the data definition:

- 1 In the Database Connectivity Definition form, select an empty cell under **External Database Field**.
- 2 In the **External Database Fields** tab:

- select a field in the **Fields** list and click **Assign**,
or
- double-click a field in the **Fields** list.

The field name will appear in the cell in the form.

- 3 Enter a variable name, data type, and description (optional) for each variable.

To remove a database field assignment from the data definition:

- 1 In the Database Connectivity Definition form, select the External Database Field you want to remove from the data definition.
- 2 Click **Remove**.

To change the connectivity filter(s):

- Follow the steps under *Defining Connectivity Data Filters* on page 128.

Note: Be sure to redefine any filters that were based on fields you removed.

When you are done redefining the Data Set, select **OK**. The data from the previous definition will be discarded and the new connectivity definition will be used to create a new data file.

Using Database Connectivity in Run Files

Overview

You can use data from external sources in Quality Analyst Run Files by establishing a connectivity definition for the Data Sets referenced in the Run File. A command placed in the Run File causes Quality Analyst to make the defined connection and download the data. This command must reference each Connectivity Data Set used in the Run File before the data is used for charting.

The steps are:

- 1 Use the Quality Analyst Editor to define the necessary Connectivity parameters for each Data Set used in the Run File and make sure the desired data is downloaded.
- 2 Include a USETEMPDIR command (see below) if the Connectivity Data Set is located in a shared directory for use by more than one user at a time.
- 3 Include a CONNECT command (see below) in the Run File for each Data Set before the Data Set is referenced for charting.
- 4 If your data definition includes calculated variables, include a CALC command (see below) in the Run file after the CONNECT command.

Note: If the CONNECT command is not used, the data present in each Data Set at the time the Run File is executed will be used. This will likely be “left over” from the last time the Data Set was used.

The USETEMPDIR Command

If a Connectivity Data Set is located in a shared directory for use by more than one user at a time, you should use the USETEMPDIR command in your Run Files to avoid potential file conflicts.

The syntax for the USETEMPDIR command is:

```
USETEMPDIR [ON] | OFF
```

See *USETEMPDIR* on page 87 for details.

The CONNECT command

The CONNECT command establishes a connection to an external database. The syntax for the CONNECT command is:

```
CONNECT filename.dat
```

The CONNECT command for a Data Set may occur anywhere in the Run File as long as it is before the Data Set is first referenced for processing.

Tip: Put all CONNECT commands at the top of a Run File where they are easily located and sure to be processed before the Data Sets are used.

The CALC Command

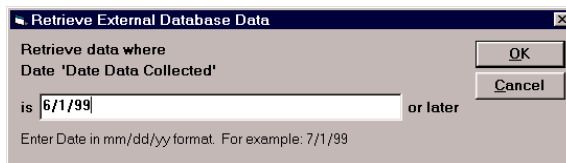
The CALC command calculates calculated variables, if any, in a Data Set. If the data definition for your connectivity Data Set includes calculated variables, you should include a CALC command *after* the CONNECT command to ensure that the calculated variables are updated to reflect any new or changed data in the external database.

The syntax for the CALC command is:

```
CALC filename.dat
```

Replaceable Parameters and Connectivity

If the definition of your Connectivity Data Set includes any of the “(Ask)” settings for filter criteria (see page 134), or if it uses custom SQL statements that include Replaceable Parameters (as explained under *Using Replaceable Parameters with Filter Criteria in SQL Statements* on page 137), then you will be prompted for filter criteria each time you CONNECT to that database (just as you would if you were to open the Data Set in the Quality Analyst Editor).



Example prompt resulting from an “Ask” setting in the definition of a Connectivity Data Set

To avoid being prompted for input at run time, you can include SET commands in your Run File to “pre-assign” values to the Replaceable Parameters in the SQL. Pre-assigning filter criteria is illustrated under *Setting*

Filter Criteria within Run Files on page 142. But you can also pre-assign other Replaceable Parameters, as illustrated in the last example below.

Read more about *Replaceable Parameters* starting on page 92.

Examples

In the following Run File, the data for the Data Set named DIAMETER is downloaded before it is used to create an IR chart:

```
CONNECT DIAMETER.DAT
IR DIAMETER.DAT DIAM I R G
CHART
```

In the following Run File, the data for the Data Set named PLATES, which contains calculated variables, is downloaded and calculated before it is used to create an IR chart:

```
CONNECT PLATES.DAT
CALC PLATES.DAT
IR PLATES.DAT THICK I R G
CHART
```

In the following Run File, four different Data Sets are used. Three of them (LINE1.DAT, LINE2.DAT, and LAB.DAT) are Connectivity Data Sets and one (TEST.DAT) is a standard Quality Analyst Data Set. Note that the CONNECT command for LAB.DAT could have appeared at the top of the file:

```
CONNECT LINE1.DAT
CONNECT LINE2.DAT
XRS LINE1.DAT WEIGHT X R G
CHART
XRS LINE2.DAT WEIGHT X R G
CHART
IR TEST.DAT PH I R G ONE.PLT
PCAP TEST.DAT PH G TWO.PLT
CONNECT LAB.DAT
IR LAB.DAT VISC I R G THREE.PLT
PCAP LAB.DAT VISC G FOUR.PLT
CHART GROUP 2V2H ONE.PLT TWO.PLT THREE.PLT FOUR.PLT
```

Let's say you have a Run File named WTTASK.RUN that connects to a Connectivity Data Set named WEIGHT.DAT. If the pertinent section of the Run File contains the lines:

```
SET ^Prod1 = "PRODUCT"  
SET ^Prod2 = "WT"  
SET ^Date = "10/1/99"  
CONNECT WEIGHT.DAT
```

and the SQL statement in the definition of the Connectivity Data Set contains the statement:

```
SELECT ^Prod1, ^Prod2 FROM DB WHERE DATE>=#^Date#
```

and you start the Run File Interpreter with the command line:

```
QARFI.EXE $RUN=WTTASK.RUN
```

then Quality Analyst will take "PRODUCT" and "WT" and "10/1/99" from the SET commands, substitute them for "^Prod1" and "^Prod2" and "^Date" in the SQL statement, and pass the SQL statement to the database as:

```
SELECT PRODUCT, WT FROM DB WHERE DATE>=#10/1/99#
```

Configuring and Troubleshooting

Quality Analyst with Connectivity was designed to connect to a wide variety of data sources under many circumstances. Setting up the initial connection can be a complex task that may require some trial and error. Quality Analyst with Connectivity is highly configurable and lets you meet the requirements of many different data sources and connection techniques.

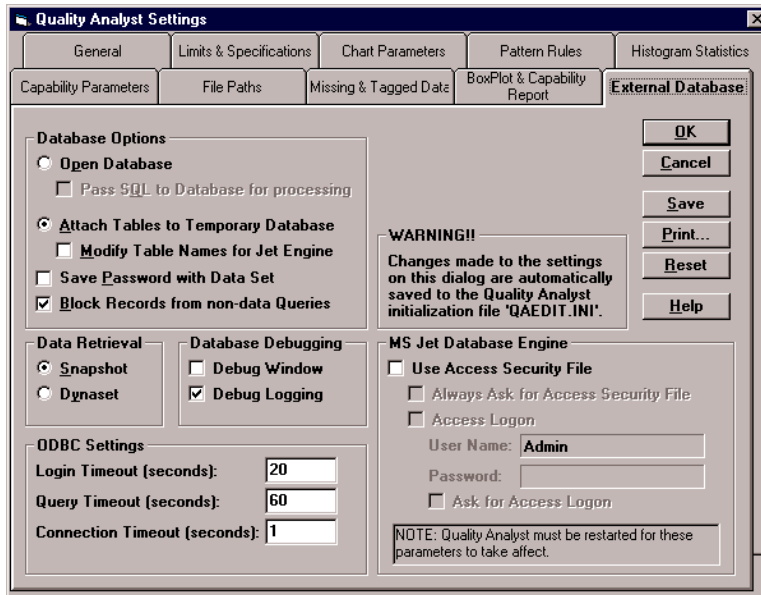
Quality Analyst with Connectivity uses the same database "engine" (named "Jet" by Microsoft) as Microsoft Access. This provides a number of distinct advantages in flexibility and performance, but also affects configuration (see *Microsoft Access / Jet Database Engine* on page 156).

You can configure most of the parameters that control and help debug external database connections on the External Database tab of the Quality Analyst Settings dialog box. These are explained in the next section.

Configuring Connectivity



To configure Connectivity, click **Settings** on the main Quality Analyst start-up screen. In the Quality Analyst Settings dialog box, select the **External Database** tab. This tab contains areas labeled Database Options, Data Retrieval, Database Debugging, ODBC Settings, and MS Jet Database Engine.



Database Options

When connecting to an external database, Quality Analyst can either open the database, or attach the external database tables to a temporary database created by Quality Analyst. It may seem roundabout for Quality Analyst to create a temporary database and attach tables, but there can be advantages depending on the particular database and type of connection used.

Opening Databases

To open the external database select **Open Database**. If you choose this option, select **Pass SQL to Database for Processing** if you want to pass Structured Query

Language (SQL) queries to the external database for evaluation. Clear this check box if you want the Jet Database Engine to evaluate SQL queries.

Despite (or perhaps because of) many SQL “standards,” each database has its own unique SQL syntax. Jet Database Engine SQL matches Microsoft Access SQL. If the Jet Database Engine interprets the query, Microsoft Access can be used to create the SQL statement. Also, with the 16-bit version of Quality Analyst, if the Microsoft Jet 2.0 / Visual Basic 3.0 Compatibility Layer is installed, the query can be optimized using the Rushmore data-access technology.

If the query is passed on to the external database for evaluation, functions unique to its SQL can be used. You can also use the passthrough setting when the table or field names of the external database don’t match the Jet Database Engine naming restrictions.

Attaching Databases

To attach external tables to a temporary database, select **Attach Tables to Temporary Databases**.

Table and field names in remote databases don’t always match the Microsoft Jet Database Engine syntax. In some databases, the table name has a prefix consisting of an “owner” name and a period. Whereas the Jet Database Engine expects database field names in the form “table.field”, these databases return “owner.table.field.” If you select **Modify Table Names for Jet Engine**, Quality Analyst will try to modify the internal table names so they are acceptable to the Jet Database Engine (“owner.table” is changed to “owner_table”). If an SQL query is used, it is also modified.

Saving the Password with a Data Set

If your external database is password-protected, select **Save Password with Data Set** if you want to save your user name and password in the header (.hed) file of the Data Set. When you subsequently open the Data Set, Quality Analyst will retrieve the user name and password from the header file rather than ask you to enter it each time. This applies to normal use and in Run Files containing commands that access the external database. This setting can be a time-saving convenience but can also pose a security risk since the user name and password are stored in an ASCII file and are not encrypted.

Blocking Records From Non-Data Queries

Note: This setting applies only to certain specialized databases, for example, those created by Wonderware's IndustrialSQL Server.

Select the **Block Records from non-data Queries** checkbox to speed up the process of defining the connection to a specialized database. This prevents Quality Analyst from retrieving data records when all it needs is field names. This checkbox is selected by default; you should leave it checked unless you encounter problems while defining the connection.

Data Retrieval

In the area labeled **Data Retrieval**, select **Snapshot** to return actual data records from an external database to Quality Analyst. Select **Dynaset** to return keys to database records instead of the actual records.

Either setting will work with any access method. Which technique is fastest depends on factors outside the scope of Quality Analyst. Only by experimenting with both can you determine the fastest technique. The results may, in fact, be nearly identical.

Database Debugging

In the area labeled **Database Debugging**, select **Debug Window** if you want Quality Analyst to display a dialog box containing details of each database command before the command is issued. This method is tedious to use, but provides immediate feedback on each step of database connectivity.

Select **Debug Logging** if you want Quality Analyst to write details of each database command to a log file.

See *Debug and Message Logging* on page 161.

Note: To view the log file, select **About Quality Analyst** from the **Help** menu of the Quality Analyst Editor, and click the **Show Debug Log** button. You can print the log or copy it to the Clipboard. Northwest Analytical technical support will often ask that you fax or e-mail them a copy of the log. The log file is cleared when you exit Quality Analyst.

ODBC Settings

In the area labeled **ODBC Settings**, enter timing parameters for ODBC (Open DataBase Connectivity) connections.

In the **Login Timeout** box, enter the number of seconds Quality Analyst should wait for the ODBC database to respond before canceling the attempt to connect to the database.

In the **Query Timeout** box, enter the number of seconds Quality Analyst should wait for the ODBC database to respond to a query before canceling the query.

In the **Connection Timeout** box, enter the number of seconds Windows should keep the ODBC connection open after Quality Analyst requests it to close. The default setting is 1 second. If you enter 0, the ODBC connection will remain open until you close Quality Analyst. Windows does this because it's more "efficient" if the connection is reopened a second time. During the time Quality Analyst has an open ODBC connection, applications that require exclusive use of the external database cannot run. Setting this parameter to one second reduces the potential for conflict.

MS Jet Database Engine

The area labeled **MS Jet Database Engine** contains several settings that manage how Microsoft Access security affects Quality Analyst connectivity.

If you select **Use Access Security File**, Quality Analyst will instruct the Jet Database Engine to use the Access System Database (often named SYSTEM.MDW). This file contains the permissions for all user identifications and passwords. The user's identification and password can only be verified if this file is available.

The Access System Database file may be in the NWA Quality Analyst directory where the Jet Database Engine can find it, but generally it is in a different directory. Select **Always Ask for Access Security File** if you want Quality Analyst to prompt you for the location of the Access System Database each time you connect to the database. If you leave this setting cleared, Quality Analyst will ask you once for its location the next time you connect to the database. After you enter its location, Quality Analyst stores it for subsequent use. (To change the location of this file later, return to this tab, select **Always Ask for Access Security File**, connect to the database, enter the location of the file when

prompted, close the connection to the database, return to this tab, and clear the **Always Ask for Access Security File** check box again.)

If you clear the **Use Access Security File** check box, user identifications and passwords cannot be verified, so you should clear the **Access Logon** check box (see below). No Access security will be used.

It might seem that, by clearing the **Use Access Security File** and **Access Logon** check boxes and hiding SYSTEM.MDW, the entire security system can be circumvented. While this will allow the Microsoft Jet Database Engine to be used without a user identification and password, Microsoft Access databases that were created with password security will not be directly accessible.

Select **Access Logon** if you want Quality Analyst to send a user name and password to the Jet Database Engine. Enter the **User Name** and **Password** in the adjacent boxes. If you select **Ask for Access Logon**, Quality Analyst will prompt you for a username and password each time you connect to an external database. The current settings for **User Name** and **Password**, if any, will appear as defaults in that prompt.

Before entering a **User Name** and **Password** here on the External Database tab, you should consider the security risk. Anyone using the computer can view these settings. If strict security is vital, leave **Password** (and maybe even **User Name**) blank and select **Ask for Access Logon**.

The user name and password are used whenever the Jet Database Engine is started. Jet uses them with all external databases. In addition, with password-protected Microsoft Access databases, the user identification and password must have “read” permission for the database.

If you clear the **Access Logon** check box, the Jet Database Engine will use “Admin” for **User Name** and nothing for **Password**.

Microsoft Access / Jet Database Engine

NWA Quality Analyst uses the Microsoft Access / Jet Database Engine to connect to all external databases. Whether Quality Analyst is connected directly to desktop databases or remote corporate databases, the Jet Database Engine handles all database communications. Most of the Database Connectivity parameters affect how the Jet Database Engine is initialized and run. Microsoft Access need not be installed on the user’s system to use the Jet

Database Engine; all necessary Jet Database Engine files are installed with Quality Analyst with Connectivity.

The Jet Database Engine is also used in Microsoft Access, so settings and procedures used in Access usually apply to Quality Analyst Connectivity. For example:

- Structured Query Language (SQL) queries created in Microsoft Access can be pasted directly into Quality Analyst. Since Microsoft Access uses sophisticated Query-By-Example (QBE) technology to easily generate SQL statements, even users with little SQL knowledge can use this technique.
- The security system used in Microsoft Access is also applied to the Jet Database Engine. This can be an advantage if Access databases are used; one security system handles everything. However, if non-Access databases also have security systems, the user may have to enter passwords for both systems. Jet Database Engine security can be disabled; see the following section.

Microsoft Access / Jet Database Engine Security

The Microsoft Jet Database Engine is used by Quality Analyst whenever an external database is accessed. As its name implies, the Microsoft Access / Jet Database Engine is also used by Microsoft's Access Database. If Microsoft Access security is enabled, a legitimate user identification and password are required to connect to any database. All external databases, whether they are Access, ODBC, Paradox, Oracle, SQL Server, or any other data source, will trigger Jet Database Engine security.

Microsoft Access must be used to assign user identification and passwords; this cannot be done from Quality Analyst. Once Quality Analyst has initialized the Jet Database Engine with the user identification and password, they stay in effect through the entire Quality Analyst session. To change the Jet Database Engine user identification and password, you must exit and restart Quality Analyst.

External databases accessed through ODBC may have their own security systems. Generally ODBC drivers will query you for a user identification and password. If Microsoft Access security is used and an external database also has security, you may be queried for several passwords before a database is

available. In this situation, the Jet engine can be run without any security so only the external database security must be satisfied.

You can configure several security parameters to manage how Microsoft Access security affects Quality Analyst connectivity. (See the section on *MS Jet Database Engine* on page 155.) The tradeoffs between ease of use and security should be considered for each situation.

Open Database Connectivity (ODBC)

Quality Analyst lets you connect to many external databases (such as those created by Oracle or SQL Server) using ODBC drivers. ODBC drivers are available from several sources. Usually, when a database is installed, the ODBC driver for the database can be installed as well. Microsoft Office for Windows includes several ODBC drivers. ODBC drivers can also be purchased directly from several different companies.

When you install Quality Analyst, ODBC drivers for Oracle and SQL Server are installed. The installation program will not, however, replace any ODBC drivers that are already installed on your system.



ODBC data sources are configured using the 32-bit ODBC program in the Windows Control Panel. Control Panel is found under Settings on the Start button. When setting up databases for use by Quality Analyst, be sure to use the 32-bit ODBC icon, shown at left. Depending on what drivers are installed on your system, it may be labeled “ODBC” or “32-bit ODBC.”

Database Access Methods

Using NWA Quality Analyst’s External Database Setup dialog and the QAEDIT.INI file, several different Database access methods and options are available:

- 1 NWA Quality Analyst may connect directly to the database, or ODBC can be used (see the “Connection” list box in the External Database Setup dialog).
- 2 The entire database may be opened, or individual data tables may be attached to a temporary database created by Quality Analyst (see the **Open Database** setting on the External Database tab of the Quality Analyst Settings dialog box).

- 3 Data tables attached to a temporary database can be renamed to match Jet Database Engine naming restrictions (see the **Modify Table Names for Jet Engine** setting on the External Database tab of the Quality Analyst Settings dialog box).
- 4 Data can be retrieved from a single database table, or a user-defined SQL query can be executed (see *Specifying a Data Table* on page 118).
- 5 A user-defined SQL query may be evaluated by the Jet Database Engine, or passed to the external database for evaluation (see the **Pass SQL to Database for processing** setting on the External Database tab of the Quality Analyst Settings dialog box).
- 6 Results can be returned as keys to the data, or as the actual data records (see the **Dynaset** and **Snapshot** settings on the External Database tab of the Quality Analyst Settings dialog box).

Which Options Are Best?

Not all options are available for all of the databases. This makes finding the “best” options challenging.

If Microsoft Access is available, it can be used to define a database that attaches the tables of external databases. This allows Microsoft Access to handle all the details of connecting to remote databases; Quality Analyst need only connect to the Access database.

If Microsoft Access is used to connect external databases, tables from separate databases can be joined in a single Quality Analyst Data Set through Access, even if different kinds of databases are used. Also, if an Access database is used, SQL queries can be created in Microsoft Access using its Query-By-Example (QBE) technology and pasted into the SQL text box on Quality Analyst’s External Database Setup dialog box. Not only is this an easy way to generate SQL queries, but Microsoft Access optimizes the query using its “Rushmore” data-access technology.

The disadvantage of using Microsoft Access is that Access must be available to each user, and Access must be used to initially establish the connections to the other databases.

Assuming that Microsoft Access is not used as an intermediary to connect to other databases, the following paragraphs discuss which options can be used with various databases.

Remote Databases (SQL Server, Oracle, etc.)

ODBC is the only valid type of connection for these databases.

Table and field names in these databases don't always match Jet Database Engine restrictions; each table name has a prefix consisting of an owner name and a period. Whereas Jet expects database field names in the format "table.field", these databases return "owner.table.field". This conflict can be resolved in the following ways:

- If a single table is attached to a temporary database (**Open Database** setting cleared), Quality Analyst can modify the internal table name (**Modify Table Names for Jet Engine** setting selected) so it is acceptable to Jet. In this case, "owner.table" is changed to "owner_table".
- If an SQL query is used and the database is opened directly (**Open Database** setting selected), the SQL query must be passed through to the database for evaluation (**Pass SQL to Database for Processing** setting selected). It is not possible to write the query in a form that Jet can interpret. The query must be expressed in the database's native SQL dialect.
- If an SQL query is used and the database tables are attached to a temporary database created by Quality Analyst (**Open Database** setting cleared), Quality Analyst can modify the internal table names and SQL query (**Modify Table Names for Jet Engine** setting selected) to a form that Jet will understand. In the SQL query, "owner.table" is automatically changed to "owner_table".

Microsoft Access Databases

Microsoft Access databases can only be connected directly through the Jet Database Engine—not through ODBC.

Paradox, dBase, and FoxPro Databases

Quality Analyst connects to Paradox, dBase, and FoxPro databases only directly through the Jet Database Engine—not through ODBC. This requires that you clear the **Pass SQL to Database for Processing** check box.

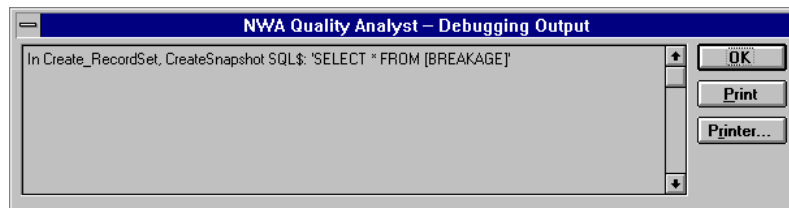
Other Desktop Databases

Other desktop databases such as Lotus Approach may be used if an appropriate ODBC driver is available. With the 16-bit version of Quality Analyst, ODBC may not function with the Jet 2.0 / Visual Basic 3.0 compatibility layer. SQL pass-through (**Pass SQL to Database for Processing** selected) will only function if the database is opened (**Open Database** selected).

Debug and Message Logging

When you encounter problems in setting up or using an external database connection, logging the commands sent by Quality Analyst and the messages returned by the different database components can help troubleshoot the problem.

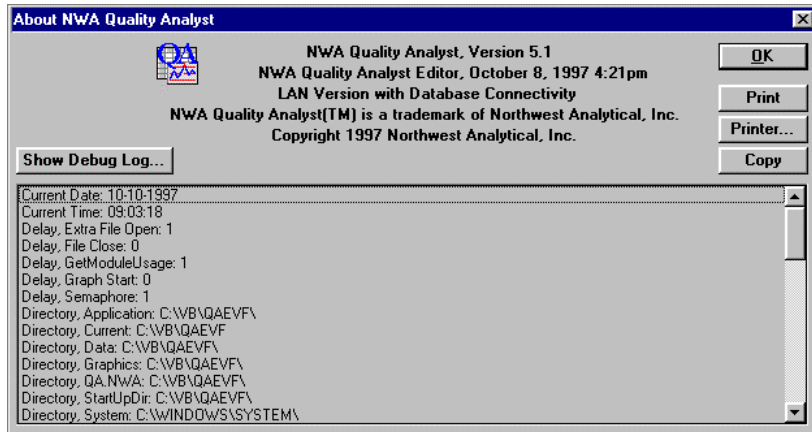
When **Debug Window** is selected on the External Database tab of the Quality Analyst Settings dialog box, Quality Analyst displays, in a message box, all commands it sends and any replies it receives from the various external database components. The user must acknowledge each message by selecting **OK**. In this dialog box, you can also **Print** the log and set up the **Printer**.



Example debugging output

When **Debug Logging** is selected on the External Database tab of the Quality Analyst Settings dialog box, all debug information is saved to the file LOG.TMP in the Quality Analyst folder. This is true even if **Debug Window** is *not* selected.

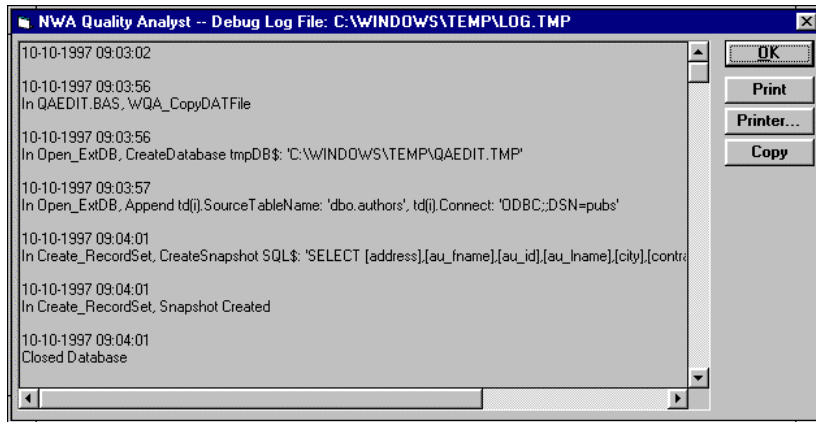
The debug log file is an ASCII text file and may be read by any editor or word processor. To display the debug log file in Quality Analyst, first open the **Help** menu and select **About Quality Analyst**.



Quality Analyst Help About dialog box showing Show Debug Log button

Note: You can print the information in this dialog box or copy it to the Clipboard by clicking **Print** or **Copy** respectively.

If **Debug Logging** is selected, this dialog box will include a **Show Debug Log** button. Click it to display the debug log file.



Example debug log file

To print the debug information, click **Print**. To copy the information to the Clipboard, click **Copy**. If you need technical assistance from NWA, faxing or e-mailing this information to us may help us resolve your problem.

Getting Technical Support for Connectivity

Quality Analyst with Connectivity lets you connect to a wide variety of external databases using any of several techniques. Though it makes establishing these connections relatively easy, because of the large number of possible types of connections, establishing and maintaining those connections may not always proceed trouble-free.

If you encounter problems connecting Quality Analyst to an external database, try the following, in this order:

- 1 Refer to the section *Configuring and Troubleshooting* which begins on page 151.
- 2 Consult your Information Systems support staff.
- 3 Select **Debug Window** or **Debug Logging** on the External Database tab of the Quality Analyst Settings dialog box, restart Quality Analyst, and retry. See if the debug messages indicate a solution. (Refer to the section *Debug and Message Logging* on page 161 for more information.)

- 4 Print the debug log and call NWA Technical Support with the print-out in hand. We may ask you to fax us the print-out.

INDEX

Index

\$

\$ASKFILTER (Replaceable Parameter), 71, 96
 \$COMMENT (Run File command), 45
 \$DATA (Run File command-line option), 108
 \$FILE (Replaceable Parameter), 71, 95
 \$FILTER (Run File command), 49
 \$MATCHFILTER (Replaceable Parameter), 71, 96
 \$PLOT (Run File command-line option), 108
 \$QANWA (Run File command-line option), 108
 \$RANGEFILTER (Replaceable Parameter), 71, 96
 \$REPORTDATE (Run File command-line option), 108
 \$RUN (Run File command-line option), 104, 105, 106, 108
 \$STRING (Replaceable Parameter), 71, 96
 \$TEMP (Run File command-line option), 108
 \$TITLE (Run File command-line option), 108
 \$VAR (Replaceable Parameter), 71, 95

/

/CLOSE (CHART subcommand), 39
 /CONTINUE (CHART subcommand), 39
 /DELAY (CHART subcommand), 39
 /PRINT (CHART subcommand), 39

@

@ (Run File command-line option), 108
 example, 109

A

AIAG capability indices, 38, 56
 ANOVA (Run File command), 10
 ANOVAR (Run File command), 11
 APPEND (Run File command), 11
 Append editor, 12
 AskAlpha (Run File command), 13
 AskAlphaMatchFilter (Run File command), 15
 AskAlphaRange (Run File command), 17
 AskAlphaRangeFilter (Run File command), 19
 AskAlphaSelectFilter (Run File command), 22
 AskDate (Run File command), 13
 AskDateMatchFilter (Run File command), 15
 AskDateRange (Run File command), 17
 AskDateRangeFilter (Run File command), 19
 AskDateSelectFilter (Run File command), 22
 AskDATFile (Run File command), 25
 AskInteger (Run File command), 13
 AskIntegerMatchFilter (Run File command), 15
 AskIntegerRange (Run File command), 17

AskIntegerRangeFilter (Run File command), 19

AskIntegerSelectFilter (Run File command), 22

AskList (Run File command), 26

AskNumber (Run File command), 13

AskNumberMatchFilter (Run File command), 15

AskNumberRange (Run File command), 17

AskNumberRangeFilter (Run File command), 19

AskNumberSelectFilter (Run File command), 22

AskOpenFile (Run File command), 29

AskSaveFile (Run File command), 31

AskTime (Run File command), 13

AskTimeMatchFilter (Run File command), 15

AskTimeRange (Run File command), 17

AskTimeRangeFilter (Run File command), 19

AskTimeSelectFilter (Run File command), 22

AskVariable (Run File command), 33

ATTRIBUT (Run File command), 35

automating procedures. *See* Run Files

B

batch operation. *See* Run Files

box plots, 36

BOXPLOT (Run File command), 36

C

CALC (Run File command), 38, 149

calculated variables

- and the Append editor, 12
- calculating, 149

c-charts, 35

CHART (Run File command), 39

CHITEST (Run File command), 40

CLEANUP (Run File command), 42

command-line options

- Run File, 94, 108, 143

commands

- Run File, 3

comma-separated-value files

- importing, 43, 44

comments

- Run File, 76

CONNECT (Run File command), 45, 148

connection type

- external databases, 117

CORVAR (Run File command), 46

CUSUM (Run File command), 47

D

data filters

- external databases, 128
- in Run Files, 49

Data Sets

- connectivity, 115
- using external databases, 115

database connectivity, 113. *See also* external databases

date

- fields, 127

direct connection

- external databases, 114

directories, 87

E

errors

- Run Files, 110

EWMA (Run File command), 47

executing

- Run Files, 102

exponentially weighted moving average charts, 47

external databases

- assigning database fields, 122
- configuring, 151
- connection type, 117
- data source, 117
- data table, 118
- date and time fields, 127
- direct connection, 114

direct-connection sources, 117
 filtering, 128
 in Run Files, 147
 modifying, 144
 ODBC connection, 114
 ODBC sources, 118
 preparing to use, 114
 selecting new, 116
 troubleshooting, 151
 using, 113, 143
 using SQL, 119
 EXTRACT (Run File command), 49
 extract definition files
 using, 49

F

file paths, 87
 filtering
 external databases, 128, 137
 fixed-field definition files, 43
 fixed-field files
 importing, 43
 FLIP (Run File command), 51
 folders, 87
 FromFilter (Replaceable Parameter), 138

G

graphics files, 9
 GROUP (CHART subcommand), 39
 GROUP (Run File command), 51

I

importing
 comma-separated-value files, 43, 44
 fixed-field files, 43
 IR (Run File command), 53

K

KRUSKAL (Run File command), 54

L

LANDSCAPE (CHART subcommand),
 39
 LastDaysFilter (Replaceable Parameter),
 138
 LastMonthsFilter (Replaceable
 Parameter), 138
 LikeFilter (Replaceable Parameter), 138

M

MatchFilter (Replaceable Parameter),
 138
 MatchFilter form of Ask command, 15
 MCAPREP (Run File command), 54
 median
 box plots, 38
 capability reports, 56
 MERGE (Run File command), 56
 missing-data symbol
 changing, 44
 MPCAP (Run File command), 58
 MR (Run File command), 60
 MSORT (Run File command), 61
 MULTREG (Run File command), 62

N

NEWVAR (Run File command), 63
 NONE (CHART subcommand), 39
 normal probability plots
 creating, 63
 NORMPLOT (Run File command), 63
 np-charts, 35

P

parameters
 replaceable, 95
 Run File, 3
 PARETO (Run File command), 64
 PAUSE (Run File command), 65
 PCAP (Run File command), 67
 p-charts, 35
 POLREG (Run File command), 68

PRINT (Run File command), 69
 process capability histograms
 creating, 67
 PROMPT (Run File command), 70

Q

quantile plots
 creating, 63

R

Range form of Ask command, 17
 RangeFilter form of Ask command, 19
 ranges
 filter, 71, 96
 REDUCE (Run File command), 73
 REGRESS (Run File command), 74
 REM (Run File command), 76
 Replaceable Parameters, 92, 95
 and (Ask) settings, 135
 and connectivity, 149
 example, 99, 100, 108
 in SQL statements, 135, 137
 in the command line), 108
 naming, 93, 96
 passing from command line, 94
 prompting for at run time, 94
 setting within Run Files, 94
 REPORT (Run File command), 76
 report format definition files, 77
 RUN (Run File command), 77
 run charts, 78
 Run File commands, 3
 \$COMMENT, 45
 \$FILTER, 49
 ANOVA, 10
 ANOVAR, 11
 APPEND, 11
 AskAlpha, 13
 AskAlphaMatchFilter, 15
 AskAlphaRange, 17
 AskAlphaRangeFilter, 19
 AskAlphaSelectFilter, 22
 AskDate, 13
 AskDateMatchFilter, 15

AskDateRange, 17
 AskDateRangeFilter, 19
 AskDateSelectFilter, 22
 AskDATFile, 25
 AskInteger, 13
 AskIntegerMatchFilter, 15
 AskIntegerRange, 17
 AskIntegerRangeFilter, 19
 AskIntegerSelectFilter, 22
 AskList, 26
 AskNumber, 13
 AskNumberMatchFilter, 15, 22
 AskNumberRange, 17
 AskNumberRangeFilter, 19
 AskOpenFile, 29
 AskSaveFile, 31
 AskTime, 13
 AskTimeMatchFilter, 15
 AskTimeRange, 17
 AskTimeRangeFilter, 19
 AskTimeSelectFilter, 22
 AskVariable, 33
 ATTRIBUT, 35
 BOXPLOT, 36
 CALC, 38, 149
 CHART, 39
 CHITEST, 40
 CLEANUP, 42
 CONNECT, 45, 148
 CORVAR, 46
 CUSUM, 47
 EWMA, 47
 examples, 8
 EXTRACT, 49
 FLIP, 51
 general form, 8
 GROUP, 51
 IR, 53
 KRUSKAL, 54
 MatchFilter form of Ask, 15
 MCAPREP, 54
 MERGE, 56
 MPCAP, 58
 MR, 60
 MSORT, 61
 MULTREG, 62
 NEWVAR, 63

NORMPLOT, 63
 PARETO, 64
 PAUSE, 65
 PCAP, 67
 POLREG, 68
 PRINT, 69
 PROMPT, 70
 Range form of Ask, 17
 RangeFilter form of Ask, 19
 REDUCE, 73
 REGRESS, 74
 REM, 76
 REPORT, 76
 requirements, 9
 RUN, 77
 RUNCHART, 78
 SCATTER, 79
 SelectFilter form of Ask, 22
 SET, 80
 SHELL, 81
 SINGCOL, 82
 SUMMARY, 83
 SUMSTAT, 84
 SYSTEM, 85
 TITLE, 85
 TTESTS, 85
 USETEMPDIR, 87, 148
 WALDRUN, 88
 WEIBULL, 90
 XRS, 90
 Run File Interpreter, 2
 command line, 104
 error messages, 110
 executing, 103
 Run Files
 and external databases, 142, 143, 147
 command-line options, 108
 converting old, 111
 creating, 2
 errors, 110
 examples, 97, 150
 executing, 102
 how they work, 2
 scheduling execution of, 106
 using old, 111
 RUNCHART (Run File command), 78

S

SCATTER (Run File command), 79
 scatter diagrams, 79
 scheduling Run Files, 106
 SelectFilter form of Ask command, 22
 SET (Run File command), 80, 135
 SHELL (Run File command), 81
 shortcuts
 program, 105
 SINGCOL (Run File command), 82
 SQL, 119, 134, 137, 149
 SUMMARY (Run File command), 83
 summary files, 9, 83
 SUMSTAT (Run File command), 84
 SYSTEM (Run File command), 85

T

Task Scheduler
 and Run Files, 106
 temporary folders, 87
 text output files, 9
 time
 fields, 127
 TITLE (Run File command), 85
 ToFilter (Replaceable Parameter), 138
 troubleshooting
 external databases, 151
 TTESTS (Run File command), 85

U

u-charts, 35
 USETEMPDIR (Run File command),
 87, 148

V

variable-definition files, 63
 V-mask, 47

W

WALDRUN (Run File command), 88

WEIBULL (Run File command), 90
wild cards, 132

X

XRS (Run File command), 90