HiQ™

Getting Results with HiQ™

NATIONAL INSTRUMENTS™

Internet Support

E-mail: support@natinst.com
FTP Site: ftp.natinst.com
Web Address: http://www.natinst.com

Bulletin Board Support

BBS United States: 512 794 5422
BBS United Kingdom: 01635 551422
BBS France: 01 48 65 15 59

Fax-on-Demand Support

512 418 1111

Telephone Support (USA)

Tel: 512 795 8248
Fax: 512 794 5678

International Offices

Australia 03 9879 5166, Austria 0662 45 79 90 0, Belgium 02 757 00 20, Brazil 011 288 3336,
Canada (Ontario) 905 785 0085, Canada (Québec) 514 694 8521, Denmark 45 76 26 00, Finland 09 725 725 11,
France 01 48 14 24 24, Germany 089 741 31 30, Hong Kong 2645 3186, Israel 03 6120092, Italy 02 413091,
Japan 03 5472 2970, Korea 02 596 7456, Mexico 5 520 2635, Netherlands 0348 433466, Norway 32 84 84 00,
Singapore 2265886, Spain 91 640 0085, Sweden 08 730 49 70, Switzerland 056 200 51 51, Taiwan 02 377 1200,
United Kingdom 01635 523545

National Instruments Corporate Headquarters

6504 Bridge Point Parkway   Austin, Texas 78730-5039   USA   Tel: 512 794 0100

Important Information

Warranty

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this manual is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation.

Trademarks

ActiveMath™, ComponentWorks™, CVI ™, HiQ™, HiQ-Script™, LabVIEW™, and natinst.com™ are trademarks of National Instruments Corporation.

Product and company names listed are trademarks or trade names of their respective companies.

WARNING REGARDING MEDICAL AND CLINICAL USE OF NATIONAL INSTRUMENTS PRODUCTS

National Instruments products are not designed with components and testing intended to ensure a level of reliability suitable for use in treatment and diagnosis of humans. Applications of National Instruments products involving medical or clinical treatment can create a potential for accidental injury caused by product failure, or by errors on the part of the user or application designer. Any use or application of National Instruments products for or involving medical or clinical treatment must be performed by properly trained and qualified medical personnel, and all traditional medical safeguards, equipment, and procedures that are appropriate in the particular situation to prevent serious injury or death should always continue to be used when National Instruments products are being used. National Instruments products are NOT intended to be a substitute for any form of established process, procedure, or equipment used to monitor or safeguard human health and safety in medical or clinical treatment.

Contents

Chapter 3
Getting Results with the HiQ Command Window

Chapter 4
Getting Your Data into HiQ

Chapter 5
Visualizing Data with 2D and 3D Graphs

Chapter 6
Analyzing Data with HiQ-Script

Chapter 7
Taking Advantage of ActiveX

Chapter 8
Converting a Notebook to a Problem Solver

Chapter 9
Getting Results as a MATLAB User

Appendix A
Customer Communication

Glossary

Index

Figures

Tables

About This Manual

The *Getting Results with HiQ* manual contains basic information you need to plan projects, understand the HiQ environment, navigate efficiently in the HiQ environment, and implement basic analysis and visualization features.

This manual was designed to teach you the fundamental features of HiQ through interactive discussions and examples. If you are new to HiQ, try reading this manual at your computer so you can test the discussion concepts and examples. If you are already a HiQ user and need information about completing a specific task, turn directly to the section discussing that task and complete the examples. Use the following questions and answers to assess where you should begin in this manual.

Are you new to HiQ and need to learn the interface elements and how to get results quickly?

Chapters 1–3 introduce HiQ, the primary interface element (the HiQ Notebook), and the Command Window. Try reading Chapters 2 and 3 at your computer so you can explore and experiment with the Notebooks and Command Window as you learn about their features and capabilities.

Do you understand the HiQ interface and Notebook metaphor but need information about constructing Notebooks and enhancing them?

In Chapters 4–8, you create notebooks from scratch and expand those Notebooks to include data visualization and analysis, ActiveX capabilities, and interactive components. Each chapter contains two examples—a Climate Notebook and a Seismic Notebook—that build on the results of the previous chapter. At the end of Chapter 8, you will have created a complete Problem Solver Notebook for both examples. See *Organization of This Manual* for a description of the contents of each chapter.

Do you need information to complete a specific task?

Use the Contents to locate the task, and complete the examples in that section to learn about it.

Are you a MATLAB user looking for a way to leverage your work?

Chapter 9, *Getting Results as a MATLAB User*, offers suggestions that can help you leverage the work you have already done in MATLAB. Try

reading this chapter at your computer so you can test the suggestions and examples.

Organization of This Manual

Getting Results with HiQ is organized as follows:

- Chapter 1, *Introduction to HiQ*, lists system requirements and installation instructions, introduces the HiQ environment, and describes ways in which you can apply HiQ analysis and visualization to your projects.

- Chapter 2, *Getting Results with a HiQ Notebook*, introduces the fundamental features of the HiQ Notebook interface and shows you how to interact with existing Notebooks.

- Chapter 3, *Getting Results with the HiQ Command Window*, presents the Command Window. It explains how to navigate the Command Window, get help from the Command Window, create and modify notebooks from the Command Window, and get immediate results using the Command Window.

- Chapter 4, *Getting Your Data into HiQ*, explains how you can get different types of data into HiQ. In this chapter, you build two Notebooks—the Climate Notebook and the Seismic Notebook—and use the Import Wizard to get your data into HiQ. You enhance these Notebooks over the next several chapters.

- Chapter 5, *Visualizing Data with 2D and 3D Graphs*, describes data visualization, which is an indispensable tool for communicating results, interpreting your analysis, and gaining an intuitive understanding of what your data represents. In this chapter, you enhance the Climate and Seismic Notebooks with 2D and 3D graphs displaying the data you imported in the previous chapter.

- Chapter 6, *Analyzing Data with HiQ-Script*, introduces HiQ-Script as an analysis tool. In this chapter, you analyze the data in the Climate and Seismic Notebooks with HiQ-Script from the Command Window or the Notebook.

- Chapter 7, *Taking Advantage of ActiveX*, explains how you can use different ActiveX technologies in HiQ. In this chapter, you enhance the Climate and Seismic Notebooks with ActiveX components to extend their capabilities.

- Chapter 8, *Converting a Notebook to a Problem Solver*, introduces problem solving Notebooks, which are useful tools for analyzing different data or performing different analysis on a specific data set. In

this chapter, you complete the Climate and Seismic Notebooks by converting them to Problem Solvers that can analyze a variety of different data sets.

- Chapter 9, *Getting Results as a MATLAB User*, summarizes the HiQ to MATLAB transitional tools—such as the HiQ Command Window, the automated M-File to HiQ-Script translator, and HiQ-Script for MATLAB users—and describes how MATLAB users can use these HiQ features to leverage work they have already done in MATLAB.

- Appendix A, *Customer Communication*, contains forms you can use to request help from National Instruments or to comment on our products and manuals.

- The *Glossary* contains an alphabetical list and description of terms used in this manual, including abbreviations, acronyms, metric prefixes, mnemonics, and symbols.

- The *Index* contains an alphabetical list of key terms and topics in this manual, including the page where you can find each one.

Conventions Used in HiQ Documentation

The following conventions are used in this manual:

<>	Angle brackets enclose the name of a key on the keyboard—for example, <Shift>. In HiQ-Script, angle brackets denote a HiQ-Script constant.
-	A hyphen between two or more key names enclosed in angle brackets denotes that you should simultaneously press the named keys—for example, <Control-Alt-Delete>.
»	The » symbol leads you through nested menu items and dialog box options to a final action. The sequence **Help»HiQ Help Topics** directs you to pull down the **Help** menu and select the **HiQ Help Topics** item. This symbol also represents the MATLAB prompt.
☞	This icon to the left of bold italicized text denotes a note, which alerts you to important information.
bold	Bold text denotes the names of menus and dialog box buttons or options.
bold italic	Bold italic text denotes a note.
<Control>	Key names are capitalized.

italic	Italic text denotes emphasis, a cross reference, or an introduction to a key concept.
`monospace`	Text in this font denotes text or characters that you should literally enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, parameters, functions, variables, filenames and extensions, and for statements and comments taken from programs.
`monospace italic`	Italic text in this font denotes optional parameters or indicates that you must enter the appropriate words or values in the place of these items.
paths	Paths in this manual are denoted using backslashes (\) to separate drive names, directories, folders, and files.

Related Documentation

The following documents contain information you might find helpful as you read this manual:

- *HiQ Reference Manual*
- *HiQ Online Help*, which you can access by selecting **HiQ Help Topics** from the **Help** menu

Customer Communication

National Instruments wants to receive your comments on our products and manuals. We are interested in the applications you develop with our products, and we want to help if you have problems with them. To make it easy for you to contact us, this manual contains comment and configuration forms for you to complete. These forms are in Appendix A, *Customer Communication*, at the end of this manual.

1

Introduction to HiQ

This chapter lists system requirements and installation instructions, introduces the HiQ environment, and describes ways in which you can apply HiQ analysis and visualization to your projects.

System Requirements

HiQ requires the following minimum system configuration:

- Windows 95 or Windows NT 4.0
- 486 CPU with coprocessor
- 8 MB RAM with Windows 95 or 16 MB RAM with Windows NT
- 256-color, 800 by 600 VGA display
- 40 MB free disk space

The following specifications are the recommended system configuration for HiQ:

- Windows NT 4.0
- Pentium 90 or higher
- 16 MB RAM with Windows 95 or 32 MB RAM with Windows NT
- Accelerated True Color, 1024 by 768 display
- 40 MB free disk space

Installing HiQ

Use the following procedure to install HiQ.

1. Insert the HiQ CD in the CD-ROM drive of your computer.
2. Follow the *Setup* instructions you see on your screen.

By default, the HiQ installation program creates a new folder, `C:\Program Files\National Instruments\HiQ`, that contains the following items:

- `Program` folder—`HiQ.exe`, HiQ help files, and related files.

- `Examples` folder—Example Notebooks demonstrating many of the analysis and visualization capabilities of HiQ, organized by category.

- `Readme.txt` file—Late-breaking information about HiQ, known limitations, and corrections.

☞ **Note**   *Be sure to read the* `Readme.txt` *file for the latest information about HiQ.*

Launching HiQ

Use one of the following methods to launch HiQ.

- Use the **Start** Button
  1. Click on the Windows **Start** button and select the **Programs** menu.
  2. Select the **National Instruments HiQ** submenu.
  3. Click on the HiQ icon.
- Use Windows Explorer
  1. Locate the HiQ executable, `HiQ.exe`, in Windows Explorer.
  2. Double click on `HiQ.exe`.

The HiQ Notebook

HiQ uses a notebook as the basic interface metaphor. The HiQ *Notebook* has features that you normally associate with a real engineering or scientific Notebook, such as pages, sections, and tabs. Figure 1-1, *HiQ Environment*, labels all of the interface elements in the HiQ environment.

An Organizational Tool

On a single page you can arrange information in a free-form manner much like in a typical drawing program. Each piece of information that you place on a Notebook page is known as an *object*. As with a real notebook, you can put many types of objects on a page: text, numerics, graphs, and more. HiQ offers these objects plus some that you do not find in a regular notebook. In fact, you are not limited to placing HiQ objects on your Notebook. You also can place any ActiveX control or object on a Notebook page, significantly increasing your flexibility within the HiQ environment.

You can interact with objects on a Notebook page in many ways. For example, if you have a three-dimensional graph on a Notebook page, you can rotate it, zoom in or out, and make any number of other changes to how that graph looks.

As your Notebook grows, you can add more pages. You also can group your work in sections to better organize a project.
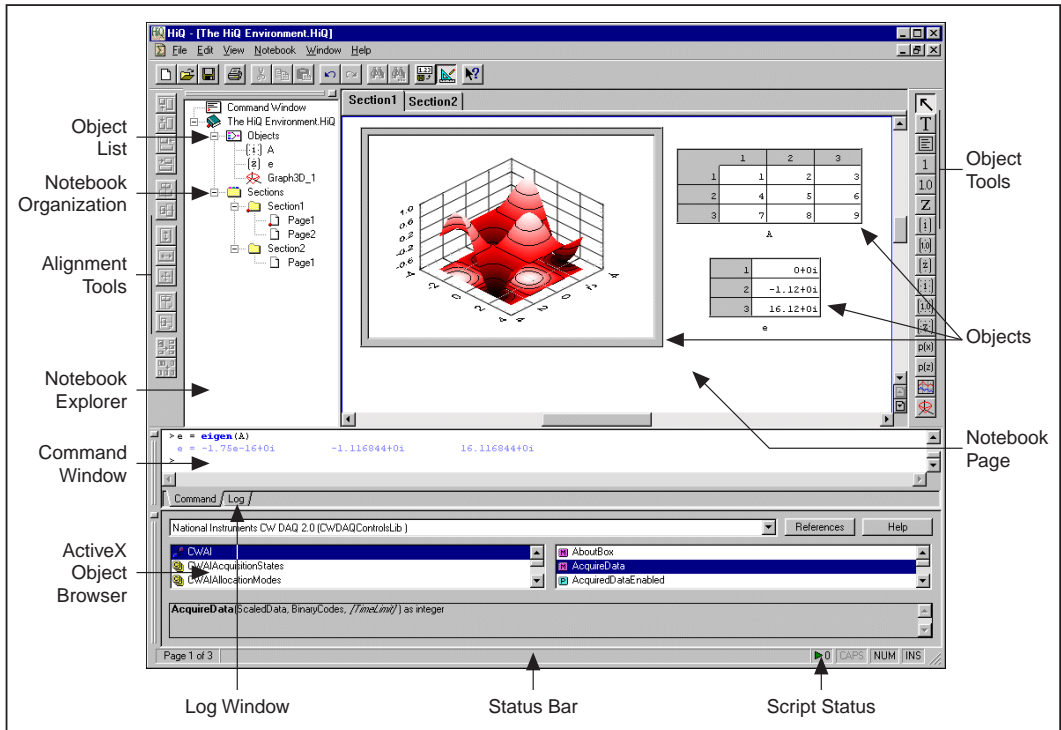
Figure 1-1. HiQ Environment

An Interactive Analysis Environment

The HiQ Notebook includes an object known as the HiQ-Script object. *HiQ-Script* is the built-in programming language that allows you to access the HiQ library of analysis and visualization functions.

With a script object on a Notebook page, HiQ emerges as a documentation tool and an interactive analysis environment. You can build a Notebook documenting what you have done and bring the analysis to life for your audience.

HiQ provides an extensive library of analysis and visualization functions. With these functions, you can visualize data in 3D objects, solve systems of ordinary differential questions, and solve many other analytical and visualization problems. Often, you can obtain your results using only HiQ built-in functions; however, when you need something more, you can create your own HiQ-Script functions to perform the exact analysis needed for your project.

HiQ also includes a comprehensive set of functions from the mathematical disciplines listed in this section. For a complete list of all functions, their usages, and descriptions as shown in Figure 1-2, refer to the HiQ online help or Chapter 7, *Function Reference*, of the *HiQ Reference Manual*.

Linear Algebra

HiQ provides comprehensive linear algebra functionality including the following items.

- Complete vector and matrix algebra using the natural, intuitive linear algebra syntax in HiQ-Script
- Vector and matrix norms such as Euclidean, Lp, and infinity
- Eigenvalue and eigenvector analysis
- Linear system solvers
- Matrix decompositions such as LUD, SVD, QRD, Schur, and others
- Vector and matrix transformations

Data Fitting and Interpolation

Data fitting and regression are excellent tools for gaining insight into real-world data. HiQ offers a flexible suite of data fitting and interpolation tools including the following items.

- Line, exponential, polynomial, and Gauss function data fitting
- Single- and multiple-dimension nonlinear data fitting
- Linear data fitting using a set of basis functions
- Polynomial interpolation
- Spline interpolation including cubic splines, b-splines, and polynomial splines
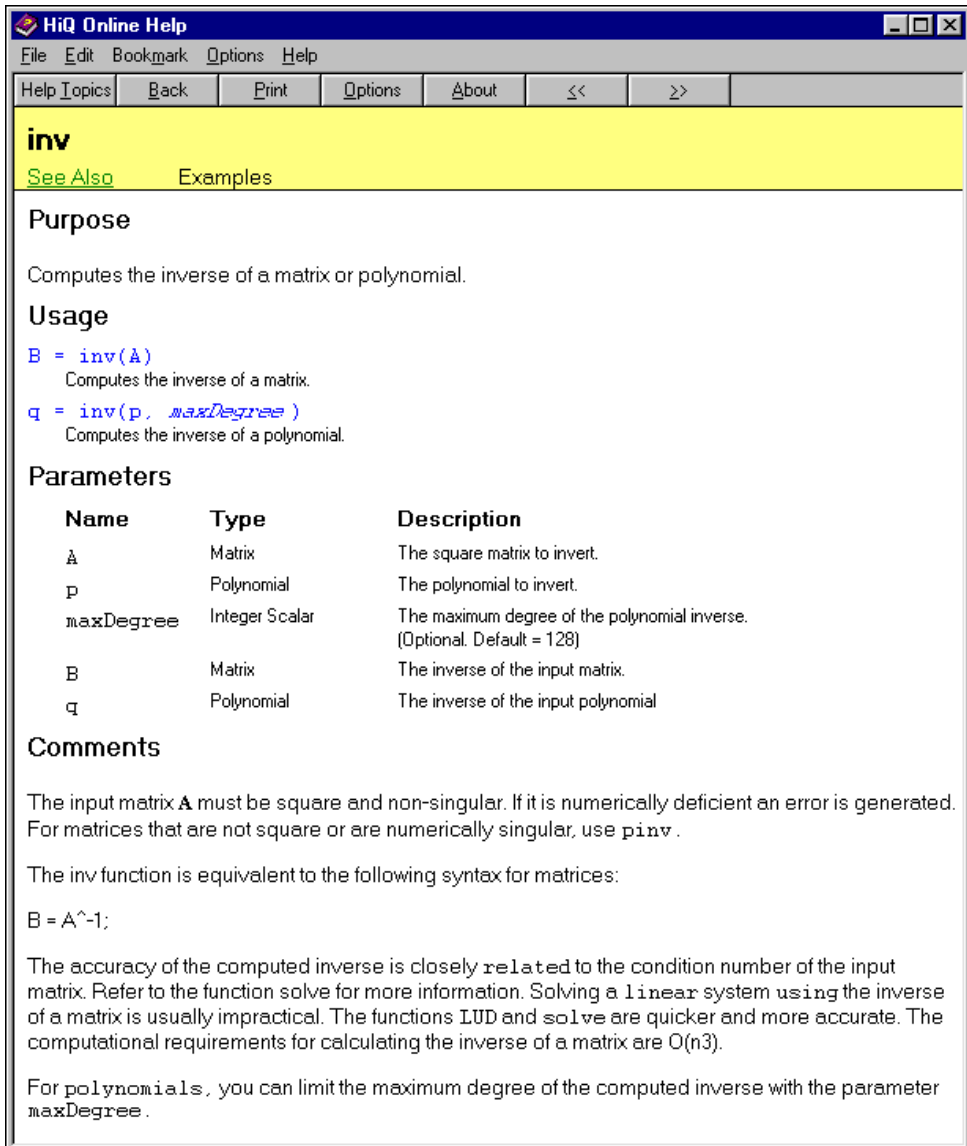
Figure 1-2. HiQ Online Help Describing the inv Function

Optimization

With HiQ optimization routines, you can minimize the cost of producing a product, maximize the capital gains in a financial portfolio, or minimize time required to perform a task. HiQ optimization capabilities include the following items.

•   Nonlinear single- and multiple-dimension optimization

•   Linear programming

Statistics

HiQ offers a complete set of descriptive statistics you can use to analyze a set of data.

•   Mean, standard deviation, variance, covariance, correlation, and more

•   Cumulative distribution and probability density functions of several types

•   Histogram, quartile, and range

Differential Equations

Dynamic systems such as the motion of an object, the results of a chemical reaction, and the reaction of complex economic systems can be described by a set of differential equations. HiQ can solve several types of differential equations including the following types.

•   Nonlinear initial-value problems including stiff systems

•   Linear and nonlinear boundary-value problems

Nonlinear Analysis

HiQ nonlinear analysis capabilities include the following items.

•   Roots of polynomials and nonlinear functions

•   Solving a system of nonlinear equations

Integration

HiQ integration capabilities include the following items.

•   Data integration

•   Nonlinear function integration

•   Polynomial integration

Differentiation

HiQ differentiation capabilities include the following items.

- Numeric multiple derivatives and partial derivatives

- Gradient, Laplacian, Jacobian, and Hessian

- Polynomial derivatives

Polynomials

HiQ has a native polynomial object that makes polynomial algebra as easy as linear algebra. With this object you can work with polynomials without worrying about the degree. Polynomial functions return polynomial objects and include the following items.

- Complete polynomial algebra using the natural, intuitive polynomial syntax in HiQ-Script

- All polynomial math including inverse, greatest common divisor, least common multiple, and more

- Integration

- Differentiation

ActiveX Environment

HiQ supports ActiveX technologies that allow you to customize Notebooks and increase productivity. For example, using drag-and-drop you can place objects on a Notebook from the Notebook Explorer, plot a vector or matrix in a graph, or print a Notebook without launching HiQ. You also can drag data from Excel into HiQ to create 3D graphs in HiQ. These ActiveX technologies let you work in your favorite environments and then collect all of your work in one place—a HiQ Notebook.

Because HiQ is an ActiveX container, you can embed ActiveX objects, such as Word documents and Excel Workbooks, and ActiveX controls, such as National Instruments ComponentWorks, to increase the functionality of a HiQ Notebook. Using ActiveX automation from HiQ-Script, you can take advantage of the capabilities of a control. Using ComponentWorks ActiveX controls, you can acquire data directly into a HiQ Notebook.

Because HiQ Notebooks are ActiveX objects, you can embed a Notebook as an object in other ActiveX containers. If you must prepare a lengthy, written technical report and prefer using Microsoft Word, you can embed a HiQ Notebook directly in your Word document.

Because HiQ is an ActiveX automation server, you can control HiQ from any ActiveX automation client such as LabVIEW, Visual Basic, or your own C programs. You can create and print customized reports with analysis and graphs directly from another application.

Because HiQ is an ActiveX automation client, you can control other applications from HiQ. For example, your Notebook can directly open an Excel Workbook, extract data, perform analysis, and create a 3D graph.

Putting It All Together

Imagine preparing a management study for a water reservoir. To report the results of your study you might use text to describe the problem, tables of data to support your claim, graphs to analyze the data, analysis to formulate projections or solutions, and conclusions to suggest ways to better manage the reservoir in the future. HiQ handles all these reporting, analysis, and visualization tasks. Furthermore, you can include an interactive script and graph so your audience can observe the effect of changing variables on inflow and outflow water levels. This interactive Notebook also becomes a tool for managing the reservoir in the future.

Experimenting with Example Notebooks

Looking at and experimenting with some example Notebooks gives you an idea of what you can accomplish with HiQ. In this manual, you learn how to use and create Notebooks to visualize and analyze data. You can browse the `Examples` folder, which contains a variety of Notebooks illustrating the analysis and visualization capabilities of HiQ.

Working with Multiple Windows

A HiQ Notebook uses the standard Multiple Document Interface (MDI) that Microsoft Office products like Word and Excel use. A main window exists within which multiple HiQ Notebooks may reside. From the **Windows** menu, you can maximize, overlay, or tile the Notebook windows you open.

Interacting with a Notebook and Running Scripts

A well designed Notebook should be self-documenting, describing how to interact with the Notebook. Interaction might be as simple as entering a start time and a stop time for simulation of a system of differential equations. Or a Notebook might instruct you to select a data file to import in order to calculate and graph the amount of rainfall in a particular area.

Most Notebooks include scripts that you can run to solve the type of mathematical or visualization problem the Notebook was designed to handle.

☞ **Note**   *To execute a script, right click on the Script object and select* **Run.**

Accessing Online Help

HiQ offers tooltips, context-sensitive help, and online help that you can access in the following ways.

- You can access the HiQ online help by selecting **HiQ Help Topics** from the **Help** menu. There you find information not available in this manual, such as complete function reference and answers to *How Do I...?* questions.

- Tooltips appear when you move the mouse cursor over a button in any toolbar. The text of the tooltip indicates the function of the button.

- A help topic appears in the status bar at the bottom of a HiQ Notebook as you move the mouse pointer over the menu items.

- You can click on the Help button, **?**, in other areas of the HiQ user interface, such as dialog boxes, to access context-sensitive help for the next item on which you click.

☞ **Note**   *Use the <F1> key to access context-sensitive help for a selected item or for an area where the cursor is located. For example, when your cursor is located in a script within the name of a HiQ built-in function, help for that function appears when you press <F1>.*

2

Getting Results with a HiQ Notebook

This chapter introduces the fundamental features of the HiQ Notebook interface and shows you how to interact with existing Notebooks.

Notebook Examples

This section presents two different HiQ Notebooks—a data fitting example and an expression evaluator Problem Solver—to demonstrate the use of the HiQ Notebook. The data fitting example, Population Fit, demonstrates how a Notebook performs a single data analysis, and the Expression Evaluator Problem Solver demonstrates how a Notebook can solve a class of problems. A Problem Solver Notebook accepts varied input from users without requiring them to alter the contents of the script.

Data Fitting Notebook

In this example, you work with a Notebook that performs data fitting analysis. The purpose of this example is to present the fundamental interface elements in HiQ, general parts of a Notebook, and common objects.

This Notebook demonstrates how HiQ analyzes a set of data, in this case population data, and graphically displays it. More specifically, the Population Fit Notebook graphically presents the change in U.S. population over time.

This Notebook contains a script and a graph. The script object contains HiQ-Script code that performs the data fit. The graph object displays the data fit after the script executes.

Opening a Notebook

1. To open a Notebook, use the **File»Open** command.

2. Open the `Examples` folder within the `HiQ` folder.

    The `Examples` folder contains Notebooks grouped by category. Later you can explore the different examples to get a feel for HiQ capabilities.

3. From the `Examples` folder, open the `Data Fitting` folder and select `Population Fit.HiQ`.

4. Click on **Open**. The Population Fit Notebook appears.
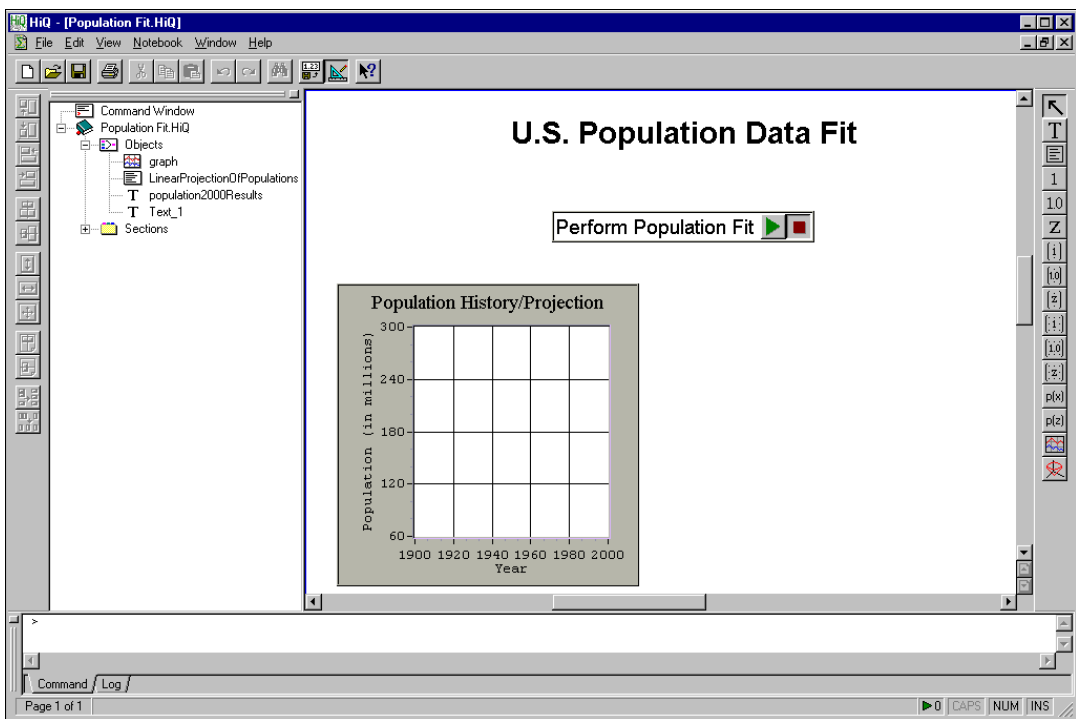
Figure 2-1. Population Fit Notebook

Running a Script

To run the script, click on the **Perform Population Fit** run button. The output of the script appears as a plot within the graph and represents the increase of U.S. population over time.
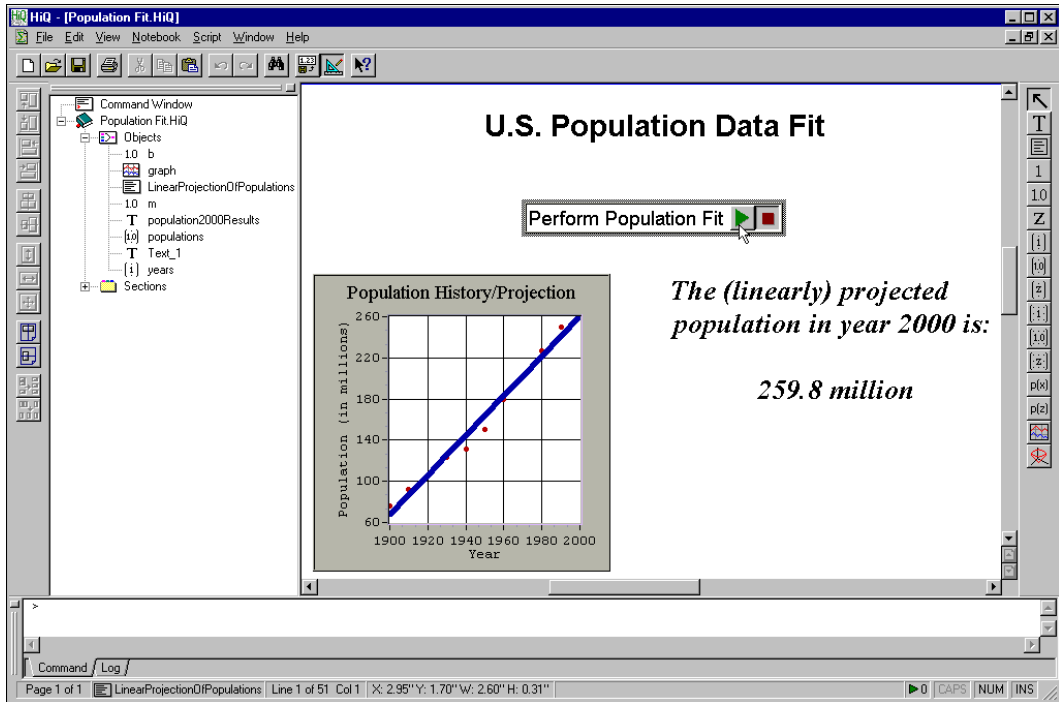
Figure 2-2. Running a Script in the Population Fit Notebook

Four objects are visible on the Notebook page: the graph, two text boxes, and the script. However, these are not the only objects in this Notebook, as you will learn in the next section.

Right click on the run button, select **View Source**, and scroll through the script at the bottom of the page to get a feel for how HiQ performs analysis. In this example, a single function `fit` performs the data fitting analysis. You can access most analysis capability in HiQ through calls to single functions like `fit`.

Viewing an Object List

Many, but not all, objects in a Notebook appear on the Notebook page. To see all objects in the Notebook, perform the following steps in the Notebook Explorer.

1. Click on the plus sign (+) to the left of the Notebook name (in this example, `Population Fit.HiQ`) to expand the contents of the Notebook.

2. Click on the plus sign (+) to the left of the word *Objects* to expand the Object List.
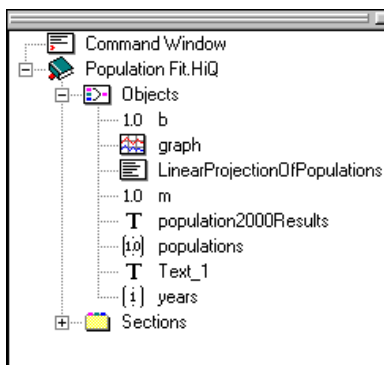
Figure 2-3. Expanding the Object List

The *Object List* contains all objects in the active Notebook. With the exception of the graph and HiQ-Script objects, all the objects in the list resulted from the execution of the script. As you can see, objects do not have to be visible to exist within a Notebook.
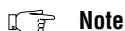
☞ **Note**   *Any object created by a script exists in the Notebook and appears in the Object List after that script runs. The only exception is objects with local scope. See the* local *section of Chapter 6, HiQ-Script Reference, in the HiQ Reference Manual for information about local scope.*

When HiQ-Script objects are generated, they appear only in the Object List. If you want the object on the Notebook page, select it in the Object List and drag-and-drop it on the page.

Expression Evaluator Problem Solver Notebook

Problem solvers are HiQ Notebooks designed to interact with users to analyze or solve a wide range of problems belonging to a certain class. The previous example Notebook, Population Fit, does not qualify as a problem

solver because it solves a single problem. Population Fit does not give you the option to fit populations from different countries or different time ranges. In order to analyze a different set of data you would need to modify the script. A problem solver is a HiQ Notebook that allows the user to enter a different set of data or analysis problem without having to modify script. The Expression Evaluator Notebook illustrates the problem solver features you can use in HiQ. Perform the following steps to learn more about the Expression Evaluator Notebook.

1. To open a Notebook, use the **File»Open** command.

2. Open the `Examples\Problem Solvers` folder within the `HiQ` folder.

3. Select `Expression Evaluator.HiQ` and click **Open**. The Expression Evaluator Notebook appears.
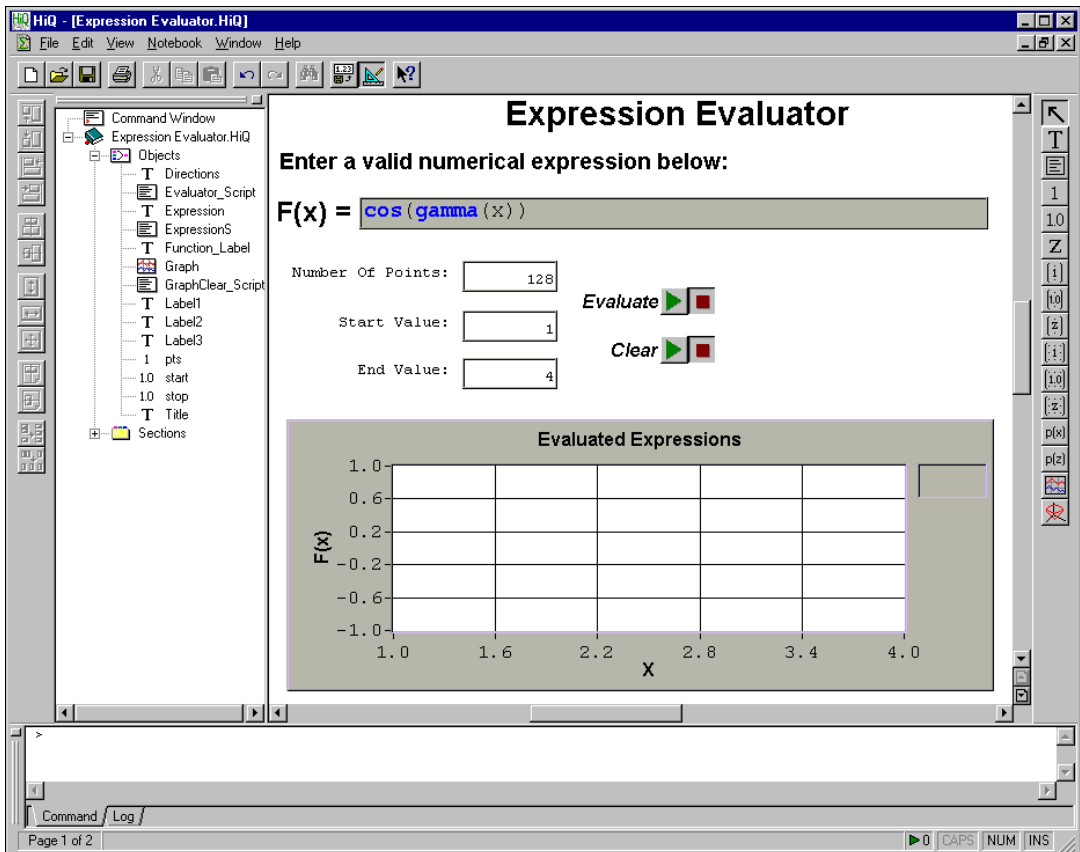
Figure 2-4. Expression Evaluator Notebook

Entering Data and Running Scripts

This Notebook can evaluate any mathematical expression and display its behavior graphically. Unlike the Population Fit Notebook, objects in the Expression Evaluator Notebook expect input from a user. The user enters a mathematical expression, the number of points to plot, and the interval of the domain. In this example, all objects have default values so you can just run the Notebook.

1. Click on the **Evaluate** button to evaluate the expression. The Notebook evaluates the expression over 128 points, starting at 1 and ending at 4. The graph displays the result.

2. Enter the mathematical expression `sin(x)*cos(x)` into the text object labeled **F(x) =**.

3. Click on the **Evaluate** button to evaluate the expression.
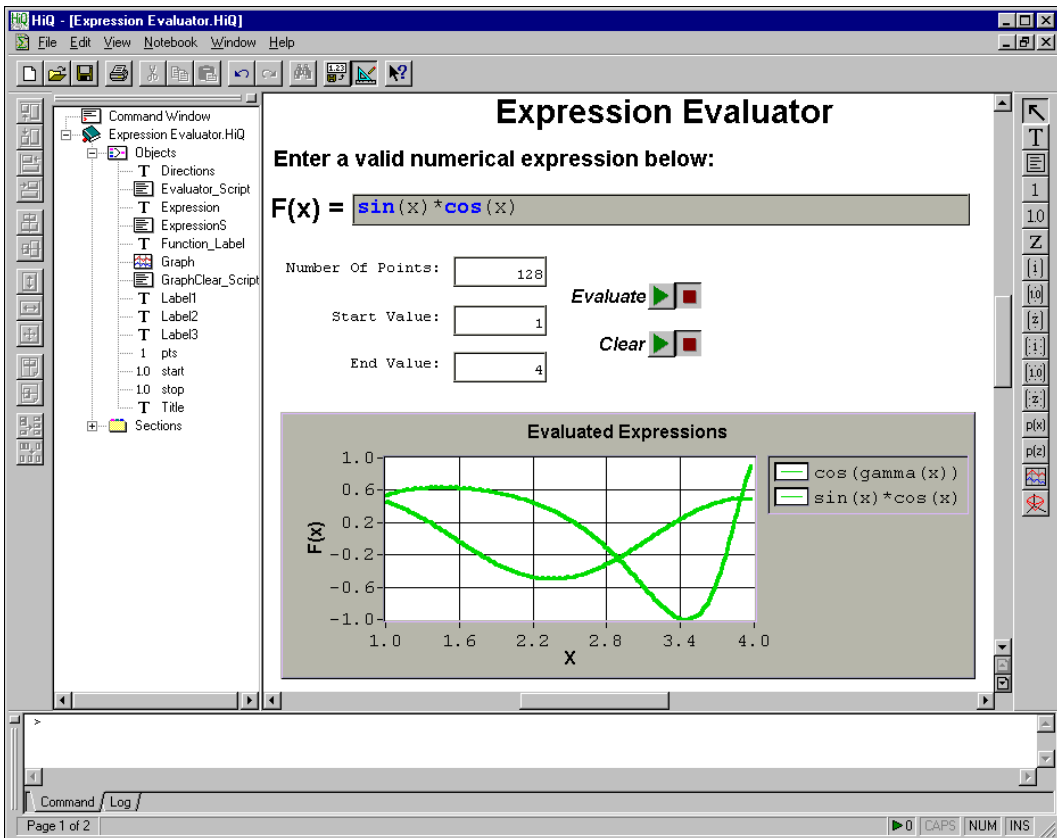
Figure 2-5. Evaluating Expressions with the Expression Evaluator Notebook

The expression you entered is fed into the numerical algorithm in the script. The plot that appears when you run the script is a graphical representation of the mathematical expression `sin(x)*cos(x)` that you entered. To observe the role of the numeric input of this Notebook, enter various values for the domain interval and number of plotted points and rerun the script.

Enter other mathematical expressions and run this Notebook to see how they display. Keep in mind that you must use valid operators and valid HiQ built-in functions. In this example, your expression must be in terms of *x*. Consult the HiQ online help or Chapter 7, *Function Reference*, of the *HiQ Reference Manual* for complete descriptions of operators and built-in functions.

Understanding Objects

All the items on a Notebook page are called objects. Notice that any object you click on becomes active and displays a border, as shown in Figure 2-6. You can interact with any activated object directly on the Notebook page.
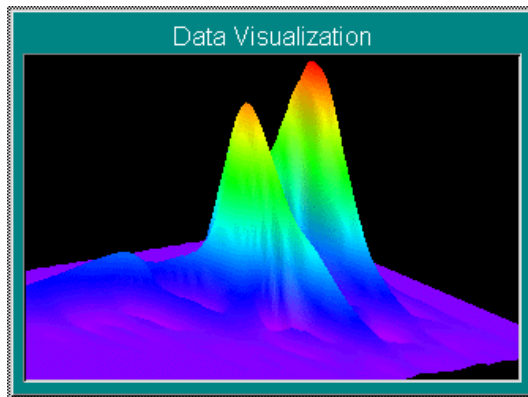
Figure 2-6. Object in Active Mode

What is an object in HiQ? Anything contained within a Notebook is an object. The following list includes some useful facts about objects.

- An object is associated with a Notebook but is not necessarily visible on the Notebook page until you place it there.
- All interactive tools of HiQ involve either the Notebook or its objects.
- HiQ uses several major categories of objects: numeric, graphic, HiQ-Script, text, and ActiveX. Each of the objects in these categories has a distinct purpose in HiQ.

Deleting Objects and Object Views

Objects in your HiQ Notebooks can exist without being visible on the Notebook page. For example, you might want to display the plot of a data set without displaying the numeric object that is the source of the data.

☞ **Note**   *Remember the distinction between objects and object views when deleting. You can delete an object entirely from the Notebook or delete only a view of that object from the Notebook page.*

Deleting the View of an Object from a Page

Deleting the view of an object from the Notebook page is not the same as deleting the object. When you delete a view, the object remains in the Notebook Object List. You can still access the object through a script or in the Object List.

To delete a view, right click on the object on the Notebook page and use the **View»Delete** command. Alternatively, you can select the object view and press the <Delete> key. The selected view is removed from the Notebook page, but the object remains in the Notebook.

Deleting an Object from a Notebook

To entirely delete an object from a Notebook, right click on the object and use the **Object»Delete** command. Alternatively, you can select the object view and press <Shift-Delete>. The object and all of its views are removed completely from the Notebook.

3

Getting Results with the HiQ Command Window

This chapter presents the Command Window. It explains how to navigate the Command Window, get help from the Command Window, create and modify notebooks from the Command Window, and get immediate results using the Command Window.

For more information about the HiQ Command Window, see Chapter 2, *HiQ Command Window*, in the *HiQ Reference Manual*. For information about HiQ-Script, see Chapter 5, *HiQ-Script Basics*, in the *HiQ Reference Manual*.

Navigating the Command Window

The Command Window provides immediate access to all analysis and visualization functions in HiQ-Script. If you are looking for quick answers or you are performing exploratory analysis, use the Command Window to display your results immediately.

To experiment with the Command Window, open a new Notebook. By default, the Command Window is attached to the bottom of the HiQ window. You can adjust the position of the Command Window by clicking on the double gray border on the left and dragging it. The window detaches from the bottom frame and becomes a floating window, as shown in Figure 3-1.

☞ **Note** *To return the Command Window to its default position, click in the Command Window title bar and drag the window toward the bottom of the HiQ window. As you drag the window, notice the gray frame. When the window is in a position where it can be docked in the HiQ window, the frame becomes thinner. When you see the thin frame, release the mouse button. This procedure works for the toolbars and other detachable windows, such as the ActiveX Object Browser and the Explorer.*
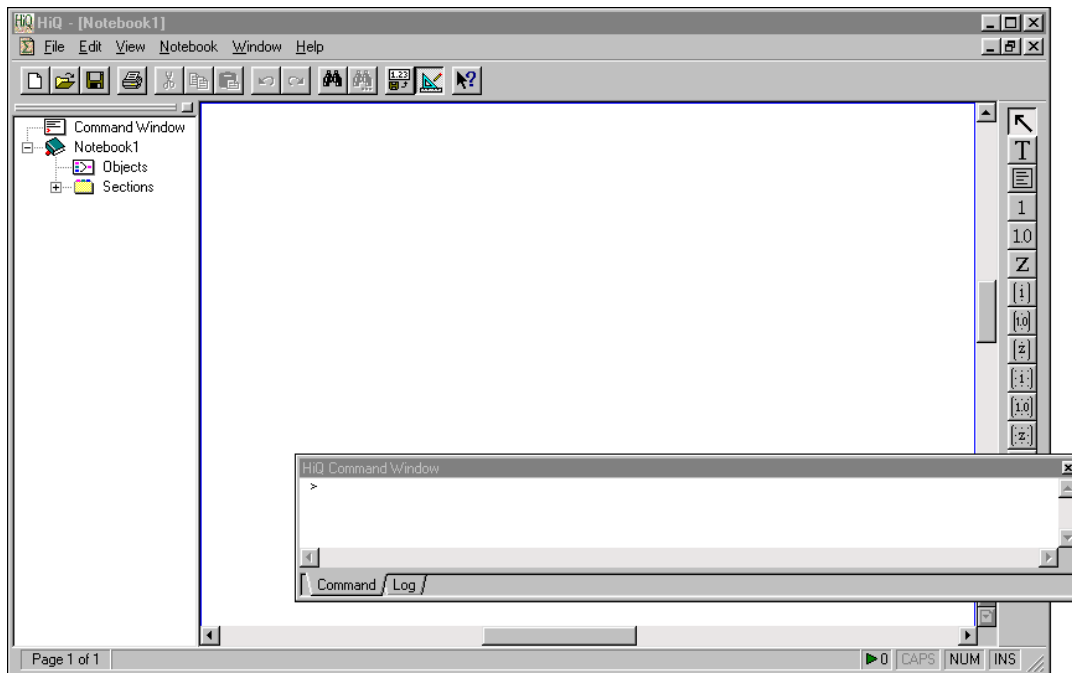
Figure 3-1. Floating Command Window

The Command Window prompt, >, indicates that the Command Window is ready to receive a command. You can create different objects at the prompt. For example, you can create a Matrix object named A by typing the following command at the prompt:

```
> A = {1,2,3;4,5,6;7,8,9}
```

Notice that the Command Window echoes the result:

```
A=
1     2     3
4     5     6
7     8     9
```

☞ **Note**   *If you do not want the Command Window to display the results, include a semicolon after the command.*

You can create a polynomial object named p by typing the following command at the prompt:

```
> p = {poly:"x^2+2x+1"}
```

The value for the polynomial object `p` is echoed by the Command Window:

```
p = x^2 + 2x + 1
```

If an object is too large to view in the Command Window, HiQ displays the object in an individual floating window. For example, create a graph by typing the following command at the prompt:

```
> MyGraph = createGraph(A)
```

☞ **Note**   *HiQ highlights all built-in functions, such as* `createGraph`*, in blue to indicate that you have correctly entered the function name.*

HiQ creates a graph named `MyGraph` of the matrix A and displays the graph in its own window.

Notice that each HiQ-Script command assigns a new object name, such as `p`, `A`, or `MyGraph`. All objects in HiQ have a name; however, you do not have to name every object you create in the Command Window. For example, you can quickly determine the norm of a matrix by typing in the following command at the prompt:

```
> norm(A)
```

The Command Window echoes the result:
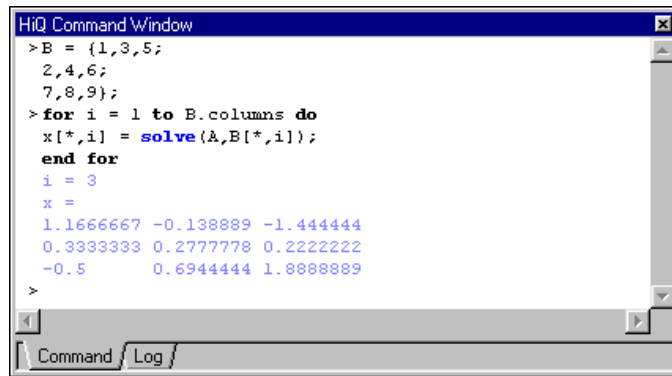
```
ans = 16.848103
```

Because you did not specify the object name, HiQ chose a default name. In this case, the default object name is `ans`.

☞ **Note**   *You can change the default name in the Command Window property pages. Right click in the Command Window and edit the* **Name** *option in the* **Default Object Assignment** *area.*

Because the Command Window allows multiline commands, you can issue commands over several lines to help you visually create a matrix or organize a script, as in Figure 3-2.

Figure 3-2. Multiline Commands in the Command Window

You can recall and edit previous commands by using the arrow keys. The Command Window accepts all special Command Window commands and all HiQ-Script built-in functions.

Getting Immediate Results with the Command Window

The Command Window offers access to all of the analysis and visualization capabilities of HiQ-Script. The following two examples show you how to use the Command Window to quickly compute answers. The first example computes the solution of a linear system of equations. The second example simulates a system of ordinary differential equations.

Solving a Linear System of Equations

Linear algebra is a widely used area of analysis for describing linear dynamic systems, computing the best fit to a set of data, optimizing equations, and many other areas. One of the most common operations in linear algebra is solving a system of linear equations.

Consider a set of three linear equations with three unknowns:

$$1\mathbf{x}_1 + 2\mathbf{x}_2 + 3\mathbf{x}_3 = 6$$
$$4\mathbf{x}_1 + 5\mathbf{x}_2 + 6\mathbf{x}_3 = 2$$
$$7\mathbf{x}_1 + 8\mathbf{x}_2 + 9\mathbf{x}_3 = 7$$

These equations can be represented in linear algebra syntax as

$$\mathbf{Ax} = \mathbf{b}$$

where **b** is a vector containing the values on the righthand side of the equations, **A** is a matrix containing the values of the coefficients on the lefthand side of the equations, and **x** is a vector of unknown quantities.

To create the matrix **A** and the vector **b**, type the following at the prompt:

```
> A = {1,2,3;4,5,6;7,8,9};
> b = {v:6,2,7};
```

One obvious way to solve this linear algebra equation for the vector **x** is to multiply the vector **b** by the inverse of the matrix **A**:

```
> x = inv(A) * b
```

This method works only if the matrix **A** has an inverse. In this case, **A** is singular, and, therefore, it is not invertible. HiQ displays the following message to convey this information: `The matrix is not invertible because it is numerically singular.`

Multiplying the vector **b** on the left by the inverse of **A**, as above, is semantically equivalent to dividing the vector **b** on the left by the matrix **A**. HiQ uses the backslash operator (\) to indicate left division and chooses an appropriate method for solving the equation:

```
> x = A\b
```

In this case, the left divide operator is equivalent to calling the built-in function `solve`:

```
> x = solve(A, b)
```

Both of the solutions share the same result:

```
x=-2.027778 0.0555556  2.1388889
```

Solving a System of Ordinary Differential Equations

Scientists and engineers use differential equations in many areas to describe the dynamics of a physical, chemical, or electrical system. HiQ contains a set of methods for numerically solving first-order ordinary differential equations for both initial values and boundary values.

For example, consider a second-order, initial-value system described by the following equations:

$$\frac{dx_1}{dt} = -5x_1 + 5$$

$$\frac{dx_2}{dt} = -10x_2 + 10$$

These equations describe how two states, x_1 and x_2, change in time. This example solves for the states x_1 and x_2 as time t moves from 0 to 2. The initial value for the states x_1 and x_2 at time = 0 are 4 and 9, respectively.

`ODEIVP` is the HiQ built-in function for solving ordinary differential equations (initial value problems). To get started, type `ODEIVP` at the Command Window prompt and press <F1> to display the online help for this function. Study the first function listed under the *Usage* heading:

```
[tOut, X] = ODEIVP(fct,x0,start,stop,stepSize,
                IVPAlg,absTolr,relTolr,outStep,callback)
```

☞ **Note**   *A HiQ function call consists of a list of input parameters enclosed by parentheses to the right of the function name and a list of output objects enclosed by square brackets to the left of the equal sign.*

The first five parameters for the `ODEIVP` function are required and consist of a user function that describes the differential equations, a vector representing the initial values of the states, a starting value, a stopping value, and a step size for the solution. The remaining parameters are optional. (As a convention, all optional parameters appear in an italic font.)

To create the user function for the differential equations, click on the parameter `fct` in the online help parameter table to view a detailed window with a template for the user function `fct`, as shown in Figure 3-3.

In the Command Window, press the <Esc> key to clear the current command. Copy this function and paste it at the Command Window prompt.
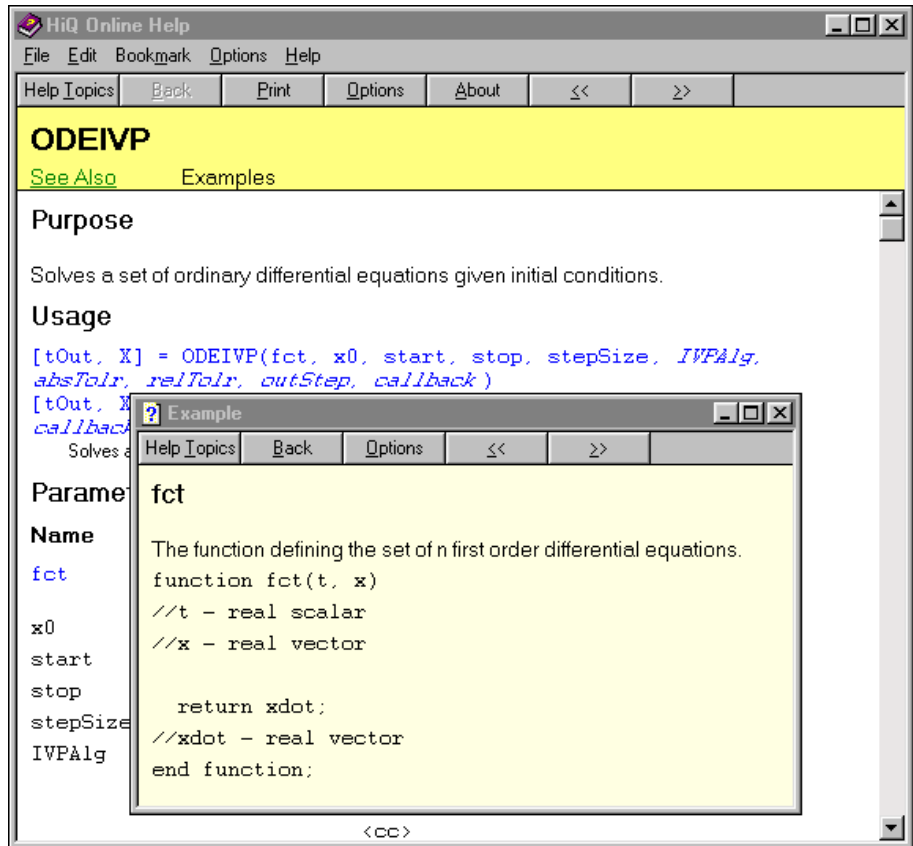
Figure 3-3. Accessing the fct Function Template

Complete the function as follows. Insert the cursor in the last line and press <Enter>.

```
> function fct(t, x)
    //t − real scalar
    //x − real vector
    xdot[1] = −5*x[1] + 5;
    xdot[2] = −10*x[2] + 10;
    return xdot;
    //xdot − real vector
end function;
```

If you have already pressed <Enter> and need to edit what you typed, recall the function with the up-arrow key and then edit it. After you make your

edits, insert the cursor at the end of the function and press <Enter>. HiQ creates a user function named `fct`.

To create the vector of initial values of the states of the two differential equations, type the following command:

```
> x0 = {v: 4, 9};
```

To solve the differential equations starting at 0, ending at 2, and with an output step size of 0.2, type the following command:

```
> [t, X] = ODEIVP(fct, x0, 0, 2, .2);
```

HiQ solves the differential equation and returns the independent solution values in the vector `t` and the solutions to the two differential equations in the first and second columns of the matrix `X`.

Finally, visualize the results using the `createGraph` and `addPlot` functions:

```
> g = createGraph(t, X[*,1])
```

```
> addPlot(g, t, X[*,2])
```

The first line creates a new graph named `g` with a new plot. The second line adds a new plot to the graph `g`. HiQ creates a floating view of the graph, as shown in Figure 3-4.

When you save this Notebook, all of the objects you have created are saved with it. You can load the Notebook later and change the parameters in the differential equations, alter the starting and stopping values of the solution, or place the objects on the Notebook page to create a document. You do not have to recreate results because you save them as you save the Notebook.
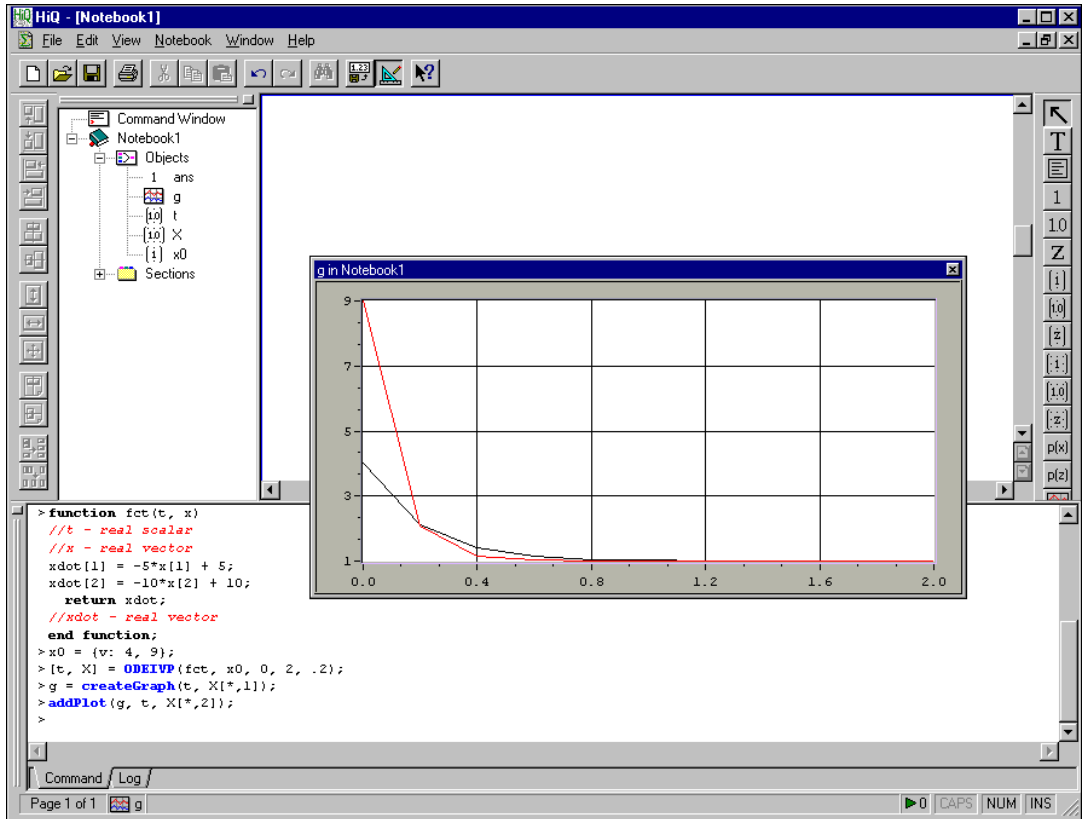
Figure 3-4. Creating an Object View

Getting Help from the Command Window

When working in the Command Window, you can access the online help using any of the following methods.

- Press the <F1> key when you are in the Command Window, and the help for the Command Window is displayed.

- Type a function name at the prompt, place your cursor in the function name, and press <F1>. The online help opens the topic information for that function.

- Type `help` at the command prompt, and the help for the Command Window is displayed.

- Type `help` *keyword* at the command prompt, where *keyword* is the function, script keyword, or help file entry for which you are looking. The online help opens the topic information related to the keyword.

Getting Notebook Information from the Command Window

A Notebook is an organizational tool as well as a documentation tool. HiQ saves all objects created in a Notebook with the Notebook so you can access them in a later session to continue your analysis.

This section describes how you can use the Command Window with an existing Notebook to quickly find answers to questions about Notebook data. For this discussion, open and refer to the `Command Window.HiQ` example in the `Examples\Getting Results` folder. This Notebook contains a matrix of data and a 3D graph representing seismic data collected at a seismographic station, as shown in Figure 3-5.
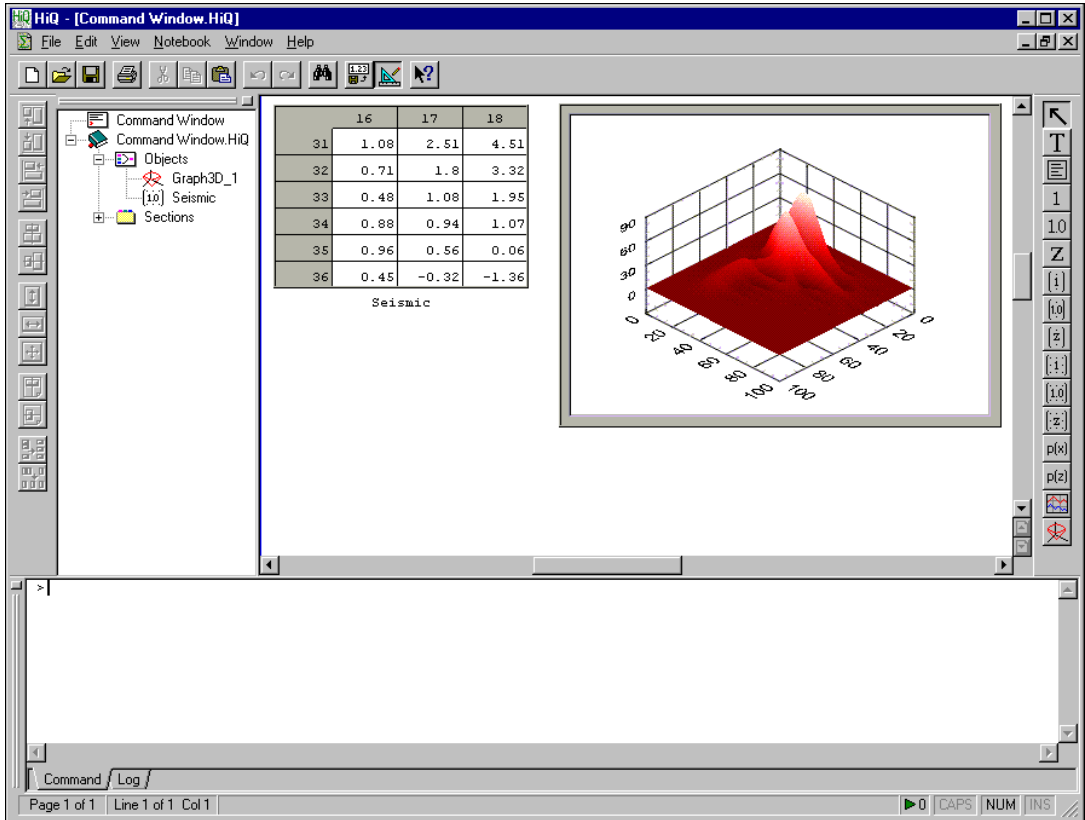
Figure 3-5. Command Window Notebook

In this example, you create a 2D graph of a signal and compute its average.

Suppose you want to know what happened thirty seconds into the seismic experiment. Specifically, you want to graph the data and calculate the average. First, you need to extract the desired data from the Seismic matrix. Type the following command at the command prompt to create a new vector of data from row 30 of the Seismic matrix:

```
> slice = Seismic[30,*]
```

This line of HiQ-Script creates a new vector object named slice using row 30 and all of the columns of the Seismic matrix. This object is visible in its own window floating above the Notebook. You can scroll down to view all elements of the vector.

Although a tabular view of this data is useful, a graph can convey this information faster and more effectively. Type the following command at the prompt to create a graph of `slice`:

```
> myGraph = createGraph(slice)
```

With this command, you have created a new graph object named `myGraph` that displays a plot of the object `slice`. If you need to make this graph larger, resize the window.

To compute the average, or arithmetic mean, of the elements in `slice`, type the following command at the prompt:

```
> ave = mean(slice)
```

The new scalar object `ave` represents the average of all elements in the vector `slice`. Notice that you can see the `ave` object in the Command Window immediately after you type the command. To view the `ave` object in its own window, type the following command at the prompt:

```
> view ave
```

To dismiss the floating window view, click on the close box in the title bar of the view. The `view` command is one of several Command Window commands you can use to manipulate objects from the Command Window. In the next section, you use the Command Window to modify this Notebook.

Modifying a Notebook from the Command Window

The Command Window is a convenient place to create or modify a Notebook as you generate results. For example, you can place notebook views of the objects created in the previous section by using the `place` command. Place the new objects on the page by typing the following commands:

```
> place slice
```

```
> place myGraph
```

```
> place ave
```

These commands place three new views on the Notebook: one view for slice, one view for myGraph, and one view for ave.

Use the following procedure to move objects on the Notebook page:

1. Right click on the object you want to move and drag the object to a new position.

2. When you have the object positioned where you want it, release the right mouse button.

Use the following steps to resize an object:

1. Click on the object to activate it.

2. Position the mouse on the object corner to activate the resize cursor, as in the following figure.
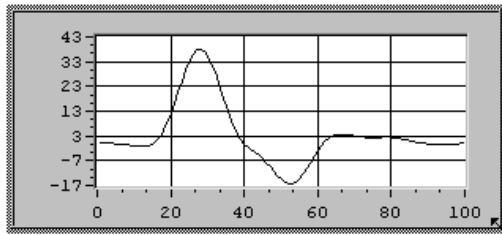
Figure 3-6. Resizing an Object

3. Click and drag the object border to the appropriate size.

Figure 3-7 shows the modified Notebook, which now analyzes and documents the results of a seismic experiment. To print this Notebook for a report or hardcopy documentation, use the **File»Print** command.
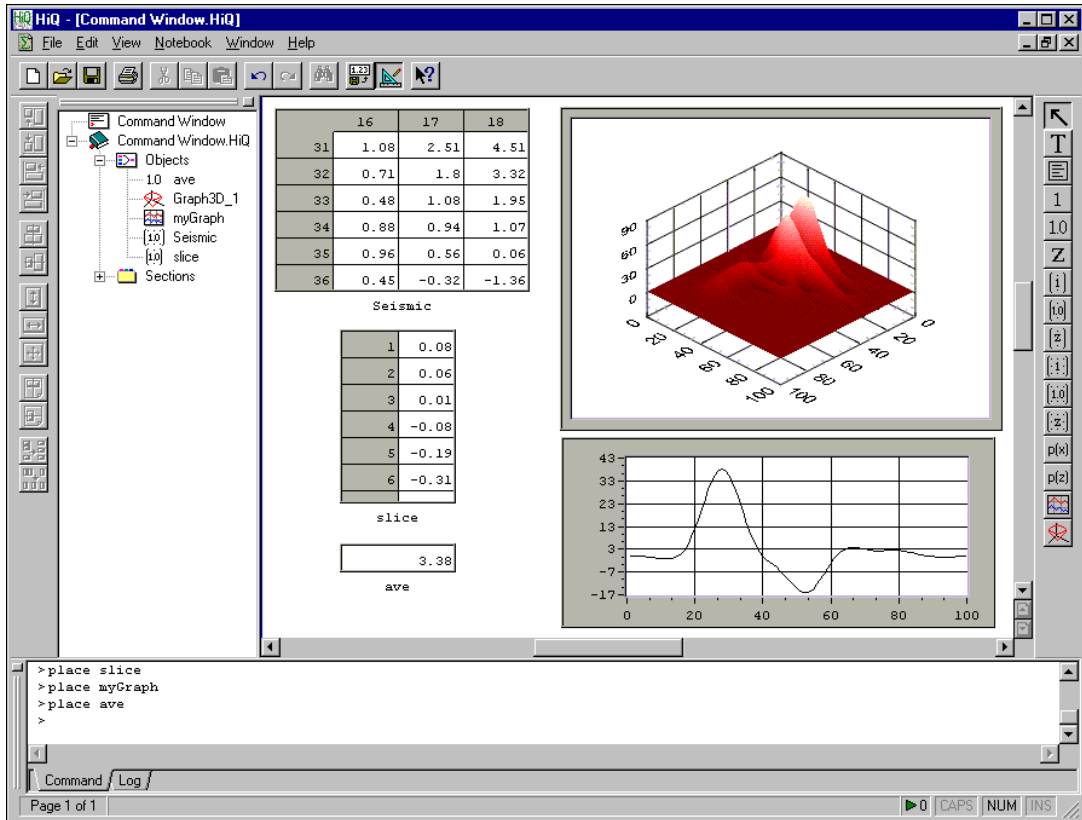
Figure 3-7. Modifying a Notebook from the Command Window

4

Getting Your Data into HiQ

This chapter explains how you can get different types of data into HiQ. In this chapter, you build two Notebooks—the Climate Notebook and the Seismic Notebook—and use the Import Wizard to get your data into HiQ. You enhance these Notebooks over the next several chapters.

Many Notebooks you create with HiQ involve analyzing and visualizing data. Usually this data exists in a file on your hard drive. When you need to import text or binary data from a file, you can take advantage of the HiQ Import Wizard.

Although you can use the default import settings in most cases, the Import Wizard allows you to define the data format in your file. When you change any of the import setting options in the Import Wizard, the data Preview window automatically updates to indicate the format of the imported data.

Climate Notebook: Importing Data

Over the next several chapters, you build a Notebook that performs a data fit on climate data from a file. As you conclude each task, save the Notebook to use it in the next chapter. If you forget to save, you can start with a prebuilt Notebook at the beginning of each chapter.

Consider a meteorological data set containing yearly rainfall and temperatures. To get this the data into HiQ, import `climate.dat` using the following steps.

1. Open a new file (**File»New**) and save the new notebook as `climate.HiQ` (**File»Save As**).

2. Select **Import Wizard** from the **Notebook** menu, and click on the **Browse** button.

3. Select the file climate.dat in the Examples\Data folder and click
    on **Open**. The Preview window displays the raw, unformatted contents
    of the file, as shown in Figure 4-1.
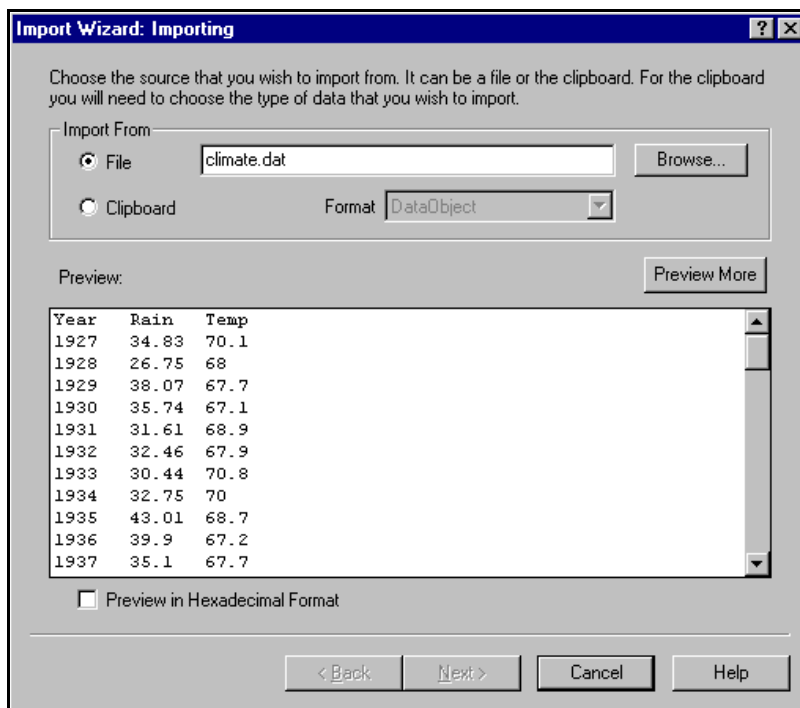
Figure 4-1. Importing Climate Data with the Import Wizard

In this example, the contents of the file are readable because the data
was saved in ACSII format.

4. Click on **Next>**. On the second page of the Import Wizard, HiQ
    imports a portion of the file and displays the data in the Preview
    window.

    On this page, you can change the name of the object (HiQ uses the
    filename by default). You also can start importing the data on a row
    other than 1, which is useful if the data file contains a header that you
    do not want to import. In this example, HiQ ignores the first line
    because it does not contain any numbers.

5. Select **Generate Script** and name the script ImportData, as shown in Figure 4-2. Although this script is not required to import the data, later you use it to complete the Notebook as a problem solver.
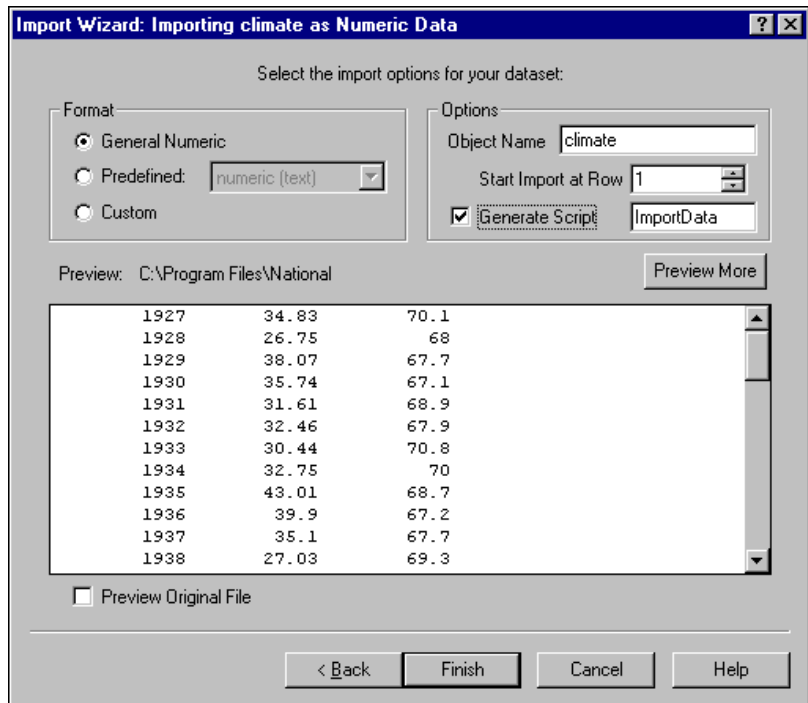
Figure 4-2. Generating a Script with the Import Wizard

6. Click on **Finish** to import the data.

7. Save the Notebook.

Figure 4-3 shows the Climate Notebook with the imported data. The climate data, which was imported as a matrix object, is visible on the Notebook page. The script generated as a result of the import is not visible on the Notebook page, but it is listed in the Explorer Object List.

You can move, resize, and modify the appearance of any object on a Notebook page. For example, to quickly expand the matrix to display all of its data, right click on the matrix and select **Show All Elements**.
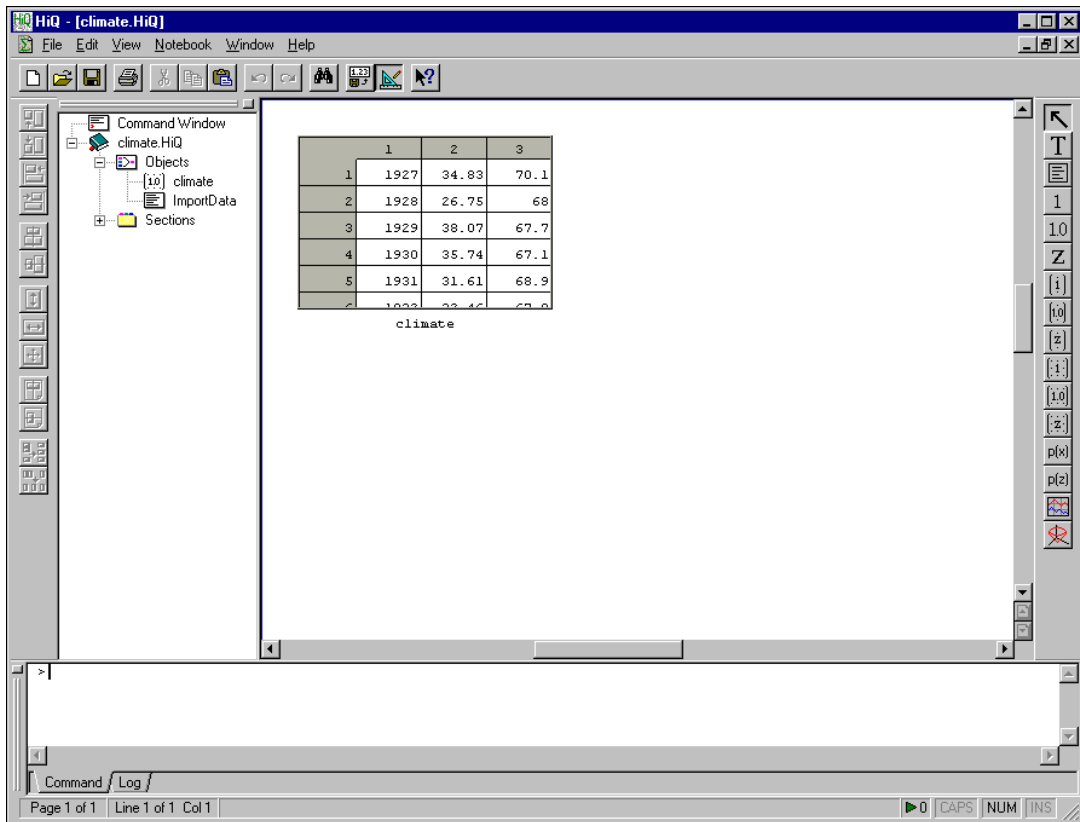
Figure 4-3. Climate Notebook with the Imported Data

Seismic Notebook: Importing Data

Over the next several chapters, you create a Notebook that visualizes and analyzes seismic data from a file. As you conclude each task, save the Notebook to use it in the next chapter. If you forget to save, you can start with a prebuilt Notebook at the beginning of each chapter.

Consider a file of binary data representing data collected from a seismic event. Because a raw binary data file contains no format information, you must know the structure of the data to successfully import it. In this example, the data represents a matrix of values with 100 rows and 100 columns and was written in IEEE Real 64 format. To import this data, perform the following steps.

1. Open a new file (**File»New**) and save the new notebook as
    `seismic.HiQ`. (**File»Save As**).

2. Select **Import Wizard** from the **Notebook** menu and click on the
    Browse button.

3. Select the `Seismic.bin` file from the `Examples\Data` folder and
    click on **Open**. The Preview window displays the raw, unformatted
   contents of the file. Scroll through the Preview window to view the
    binary data.

☞ **Note**   *You might need to select* **All Files** *from the* **Files of type** *list to display all
available files.*
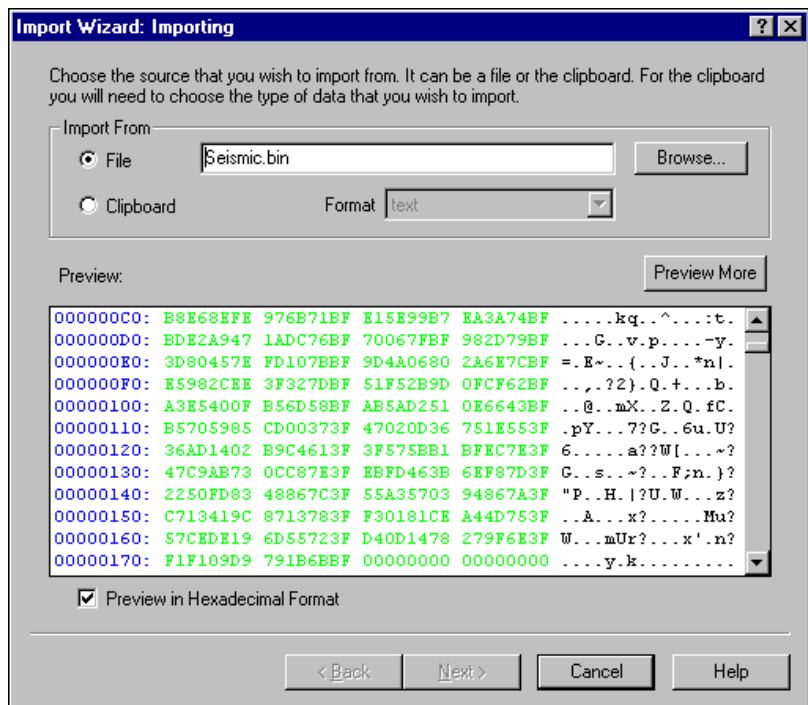
Figure 4-4. Importing Raw Seismic Data with the Import Wizard

In this example, the file is not readable because the data is saved in an
unreadable binary format. You must first tell HiQ how to format the
data before the Preview window is readable.

4. Click on **Next**. HiQ recognizes that the file is binary and automatically
    selects **Predefined: numeric (binary)** for the format.

5. Click on **Next**. On the third page, HiQ imports a portion of the file and displays the data in the Preview window. To verify that the data is correct, scroll through the column of numbers. If the data had been written using a different binary format, you would have set the correct **Data Type** and **Byte Ordering** options to specify the format.

6. Click on **Next**. Import this data as a matrix with 100 columns, as shown in Figure 4-5.
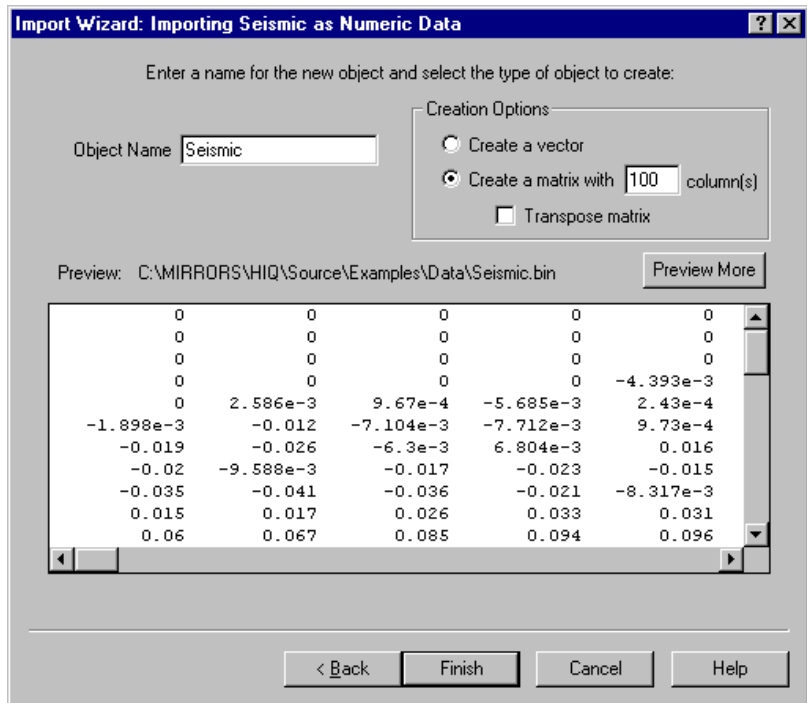
Figure 4-5. Importing Formatted Seismic Data

7. Click on **Finish** to import the data.
8. Save the Notebook.

Figure 4-6 shows the Seismic Notebook with the imported data. The seismic data, which was imported as the matrix Seismic, is visible on the Notebook page.
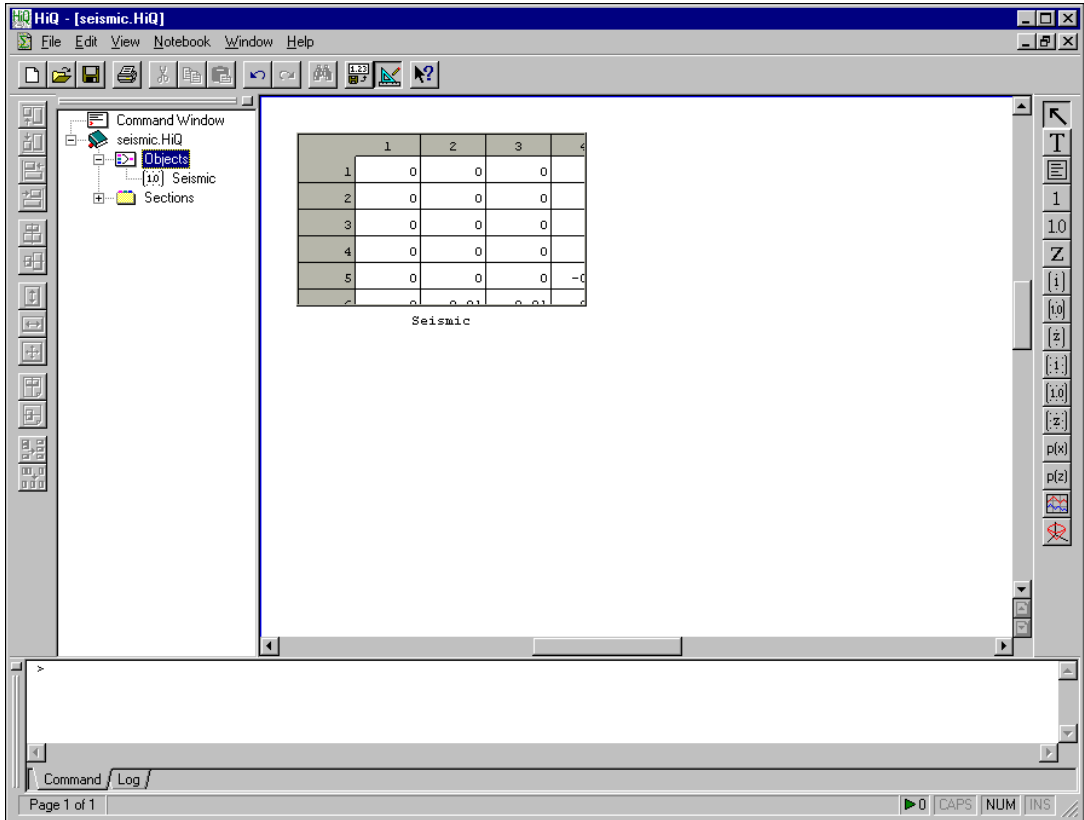
Figure 4-6. Seismic Notebook with Imported Data

Importing Data as Different Objects

Using the Import Wizard, you can import data in different formats as different objects. This section contains procedures for importing data in a text file as a Text object or Numeric object, importing data in a binary file as a Numeric object, and importing data in a custom format with the custom import option.

☞ **Note** *The Import Wizard dialog box includes advanced settings for formatting data. Use the* **Next** *and* **Back** *buttons to access all of the options. Click on the* **Finish** *button in the last option panel to accept all your settings.*

Importing a Text File as a Text Object

To import a text file into your Notebook, use the following procedure.

1. Select **Import Wizard** from the **Notebook** menu and click on the
   Browse button.

2. Navigate to the text file you want to import, select it, and click **Open**.

3. From the Import Wizard, select the appropriate import options.

   a. If the file has a `.txt` extension, click on **Finish** when you are
      ready to import the file.

   b. If the file is not a `.txt` file, click on the **Predefined** option and
      select **text** from the list of predefined formats. Click on **Finish**
      when you are ready to import the file.

   Your text file is now imported as a text object and is accessible through the
   Object List.

☞ **Note**   *Use the data Preview window of the Import Wizard to inspect how HiQ will import
the data. You might need to modify the import settings to obtain a special format.*

Importing a Text File as a Numeric Object

To import a text file into your Notebook as a numeric object, use the
following procedure.

1. Select **Import Wizard** from the **Notebook** menu and click on the
   Browse button.

2. Navigate to the text file you want to import, select it, and click **Open**.

3. Use the Preview window to ensure that your data is going to import
   correctly. If your data appears correctly in the Preview window, click
   on **Finish**. Otherwise, follow the remaining steps to customize the
   formatting of your data.

4. To customize the formatting of your text file, click on the **Predefined**
   option and select **numeric (text)** from the list of predefined formats.

5. Click on **Next>** to select the appropriate delimiter method.

6. Click on **Next>** again to select the appropriate type of numeric object
   to create.

7. When your settings are complete, click on **Finish** in the last panel of
   the Import Wizard.

   Your text file is now imported as a numeric object and is accessible through
   the Object List.

Importing a Binary File as a Numeric Object

To import a binary file to your Notebook as a numeric object, use the following procedure.

1. Select **Import Wizard** from the **Notebook** menu and click on the **Browse** button.

2. Navigate to the binary file you want to import, select it, and click **Open**.

3. Click on the **Predefined** option and select **numeric (binary)** from the list of predefined formats.

4. Click on **Next>** to select the type and byte ordering of your data.

5. Click on **Next>** again to select the appropriate type of object to create.

6. When your settings are complete, click on **Finish** in the last panel of the Import Wizard.

Your binary file is now imported as a numeric object and is accessible through the Object List.

Importing with the Custom Import Mode Option

The custom import option is useful when you know how to write import formats in HiQ-Script. Refer to the documentation on the `import` built-in function in the online help or Chapter 7, *Function Reference*, in the *HiQ Reference Manual*.

☞ **Note**   *If you need to export data from HiQ, use the Export Wizard, which works similarly to the Import Wizard. To access the Export Wizard, right click on the object you want to export and select **Export**.*

5

Visualizing Data with 2D and 3D Graphs

This chapter describes data visualization, which is an indispensable tool for communicating results, interpreting your analysis, and gaining an intuitive understanding of what your data represents. In this chapter, you enhance the Climate and Seismic Notebooks with 2D and 3D graphs displaying the data you imported in the previous chapter.

Climate Notebook: Visualizing Rainfall and Temperature Data in a 2D Graph

You can graph most data in two dimensions, where the data is plotted against a single, independent variable. In this example, you plot temperature and rainfall data on a 2D graph.

Creating a Graph

1. Open the climate.HiQ Notebook if you completed the Notebook example in Chapter 4, *Getting Your Data into HiQ*. Otherwise, open climate5.HiQ from the Examples\Getting Results folder.

2. To add a 2D graph object to the Notebook, click on the 2D graph tool in the HiQ Tools toolbar.

☞ **Note** *You also can create objects using the* **Notebook»Create** *command. For this example, select* **Notebook»Create»2D Graph.**

3. Click and drag on the Notebook page where you want the new graph to appear, as shown in Figure 5-1.
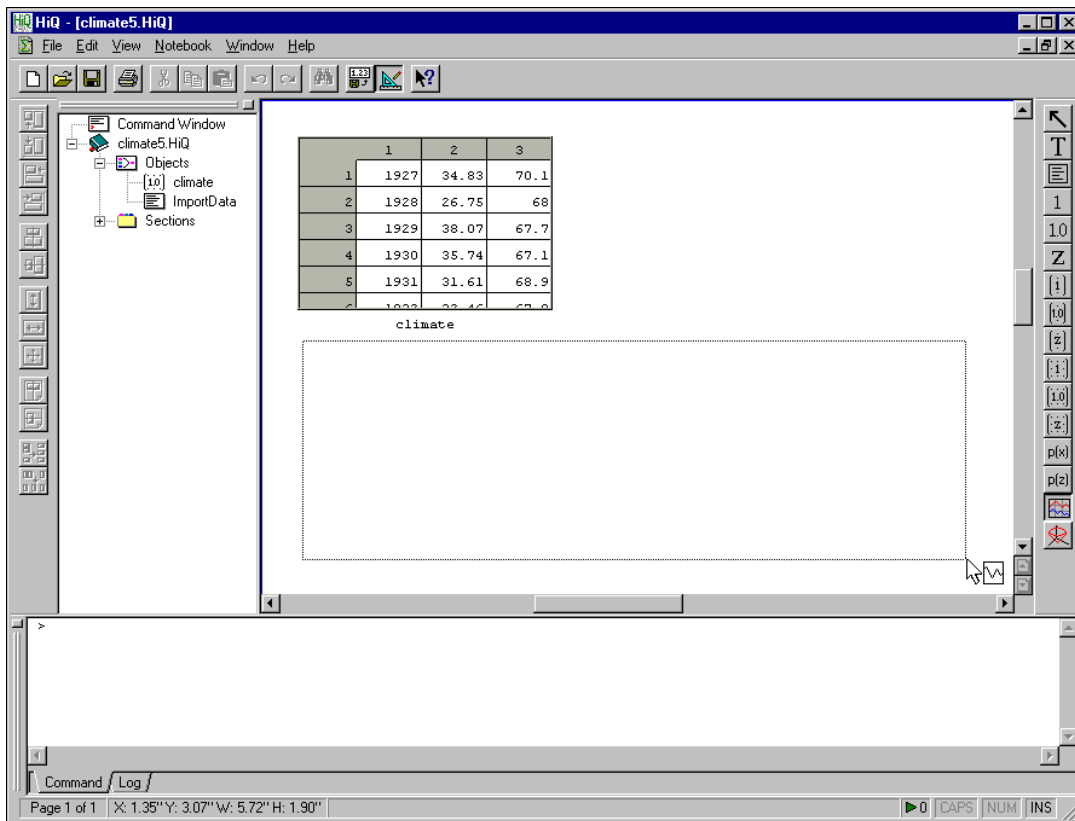
Figure 5-1. Clicking and Dragging to Place a 2D Graph Object on a Notebook Page

4. Change the name of the graph object in the Explorer. Select the object in the Object List, place the cursor in the object name, and replace `Graph2D_1` with `graph` to rename the object.

Plotting a HiQ Data Object

Add a new plot to the graph as described in the following steps. After completing the first six steps, compare your New 2D Plot dialog box with Figure , which shows the correct settings.

1. Right click on the graph and select **New Plot**.

2. Select **Data** as the **Range** option for your graph and click on the **Browse** button to the right. The range is the y data to plot.

3. Select climate from the **From Matrix** option, choose **Column**, and type 2. Click **OK**.

4. Select **Data** for the **x Domain** option for your graph and click on the **Browse** button to the right. The domain is the x data to plot.

5. Select climate from the **From Matrix** option, choose **Column**, and type 1. Click **OK**.

6. Select **Generate Script** and name the script Plot1. Although this step is not required to plot the data, you can use this script later to complete the Notebook as a problem solver.
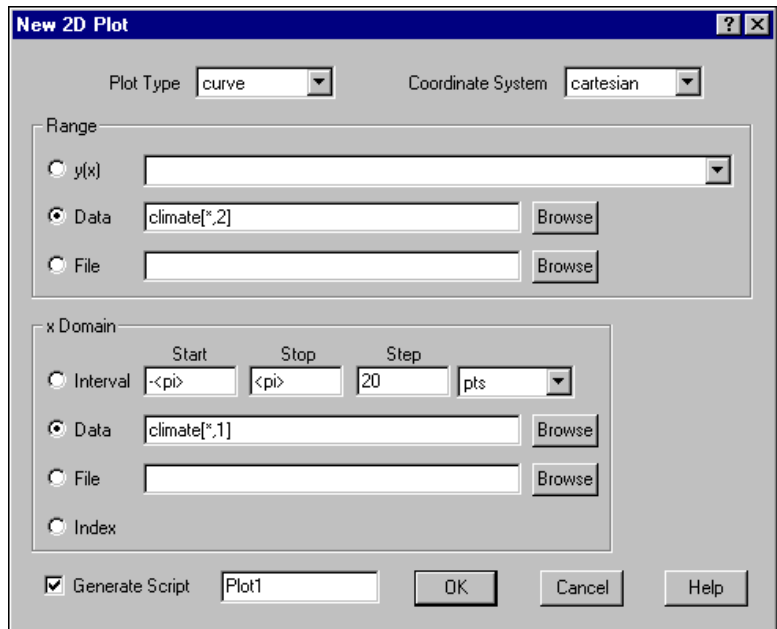
Figure 5-2. Adding a Plot to a Graph

7. Click on **OK** to add the new plot.

Figure 5-3 shows the graph and new line plot, which represents the yearly rainfall for the last 71 years.
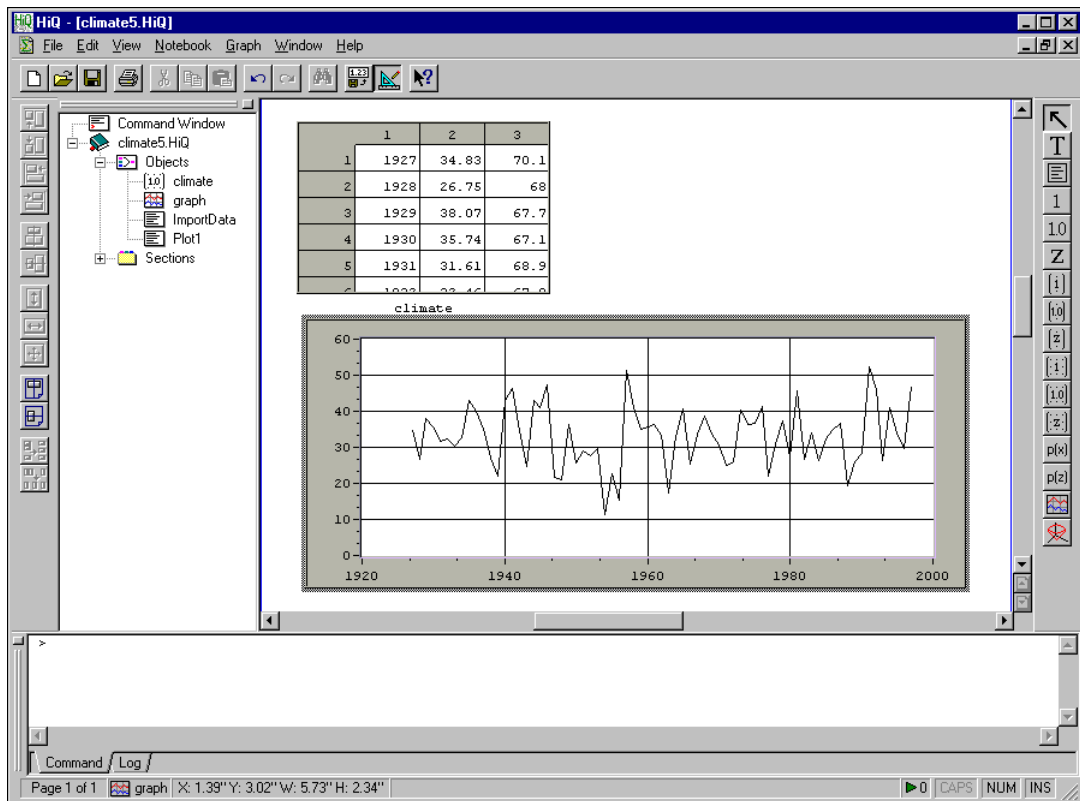
Figure 5-3. Plotting Climate Data on a 2D Graph

Modifying the Graph and Plot

Many times the reports and presentations you create require graphs with special formatting. With HiQ, you can format your graphs to match your needs. In this example, rainfall data might better be represented by a bar graph. To modify the properties of a plot and graph, use the following procedure.

1. Right click on the graph and select **Properties** to access the 2D graph property pages, as shown in Figure 5-4.
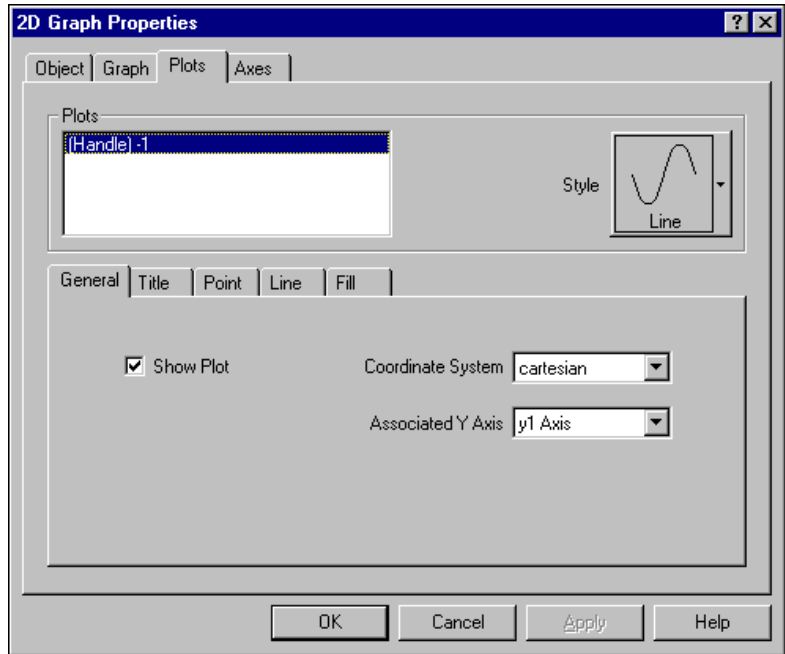
Figure 5-4. 2D Graph Property Pages

2. To modify plot properties, click on the Plots tab to change the plot style to a vertical bar (**Vert. Bar**) in the **Style** area. This style changes the plot to a vertical bar plot.

3. To modify graph properties, click on the Graph tab.

4. Enter Austin Climate for the graph **Title**.

5. To modify axis properties, click on the Axes tab.

6. In the **Axes** list box, select **xAxis** and enter Year in the **Title** field.

7. Set **Axes** to **y1Axis**. (y1 denotes the first y-axis. Two-dimensional graphs can have up to eight y axes. You use multiple y-axes later in this chapter.

8. Enter Rainfall (inches) for the **Title** of the y1 Axis.

9. Click on **OK**.

Figure 5-5 shows the Climate Notebook with the modified plot. The bar graph effectively communicates the trend in rainfall totals for 71 years.
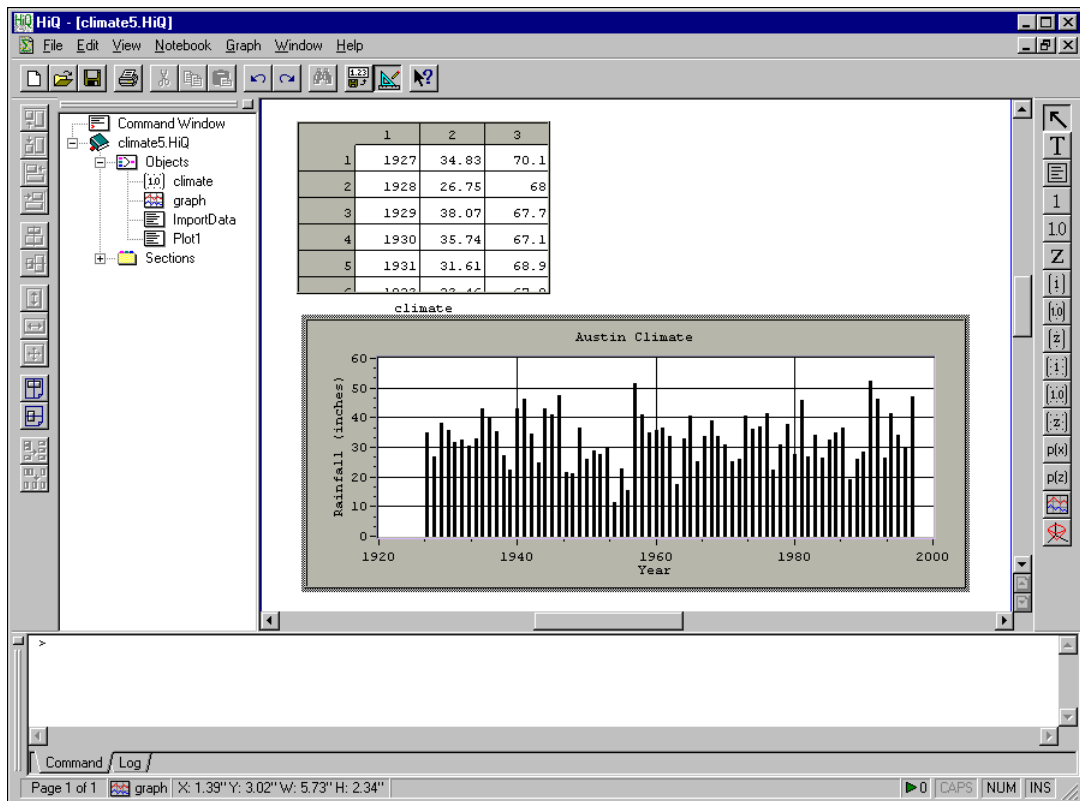
Figure 5-5. Climate Notebook After Modifying the Plot

You can add multiple plots to this graph to view the same data in different forms or to compare different data sets. In this example, you might want to add a new plot containing temperature data, as the next section explains.

Working with Multiple Plots in a Graph

A HiQ graph can display multiple plots. In this section, you add a new plot of temperature data to the same graph that you created in the previous section.

1. Add a second plot of the rainfall data, specifying the third column for the range of data instead of the second column. Select the **Generate Script** option on the New 2D Plot dialog box and name the script Plot2. Figure 5-6 shows the correct settings for this plot.
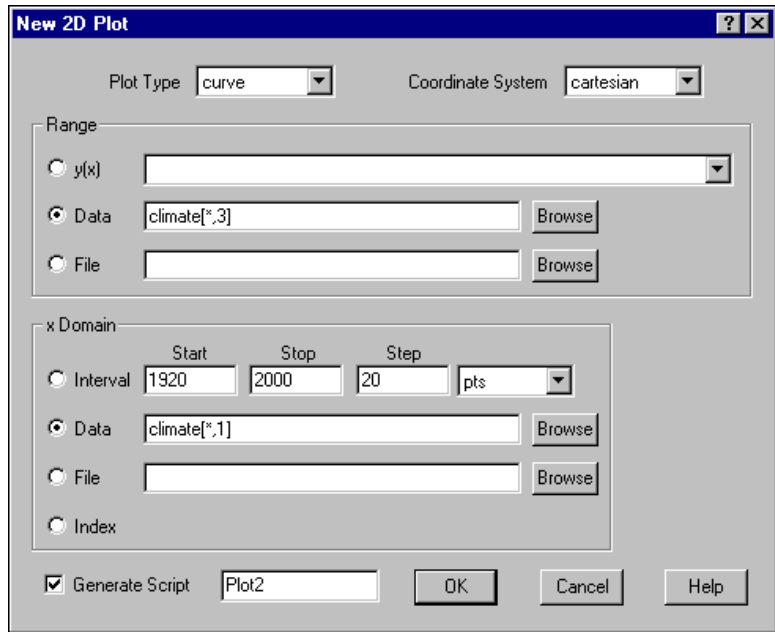
Figure 5-6. Adding a Second Plot to the Graph

Later you use this script to complete the Notebook. To review this procedure, see *Plotting a HiQ Data Object*.

The graph now has a second plot representing the average temperature over the same time period. Because the y-axis is labeled for the rainfall plot, we need to add a second y-axis for the temperature data. This second axis displays the temperature scale on the right side of the graph.

2. Right click on the graph object and select **Properties**.

3. Click on the Plots tab. Select the rainfall plot `(Handle)-2` from the list of available plots in the Plots property page.

4.   On the General tab on the Plots property page, choose **y2 Axis** as the **Associated Y Axis**, as shown in Figure 5-7.
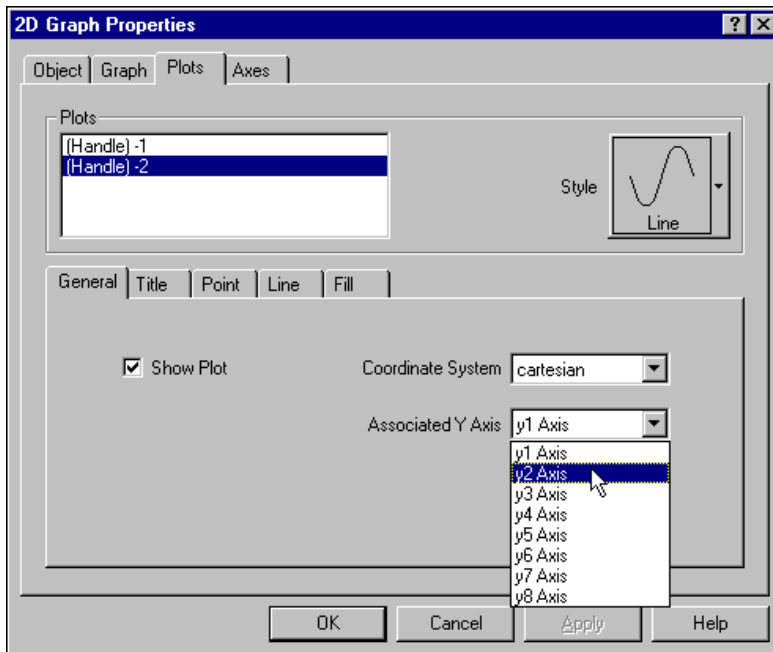
Figure 5-7. Adding a Second Y-Axis

5.   Click on the Axes property page and select the **y2 Axis**.

6.   Click on the Title tab and name the y2 axis Temperature (degrees F).

7.   On the Range tab, set the **Mode** to **Manual** with a **Minimum** of 50 and a **Maximum** of 80, as shown in Figure 5-8.
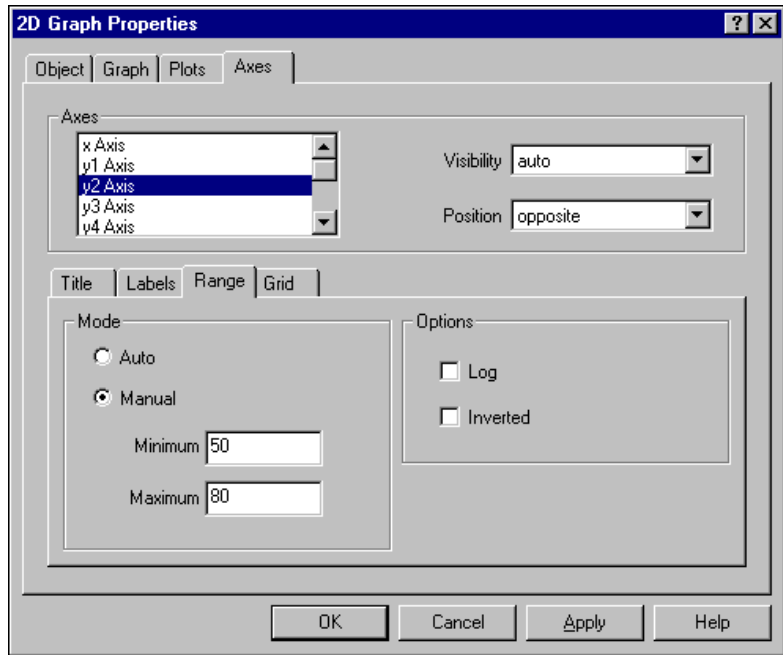
Figure 5-8. Setting Range Properties for a Y-Axis

8. Click on **OK**.

9. Save the Notebook as climate.HiQ.

Figure 5-9 shows the Climate Notebook with the second plot representing temperature data. This Notebook effectively presents two related data sets with different ranges using multiple y axes on a single graph.
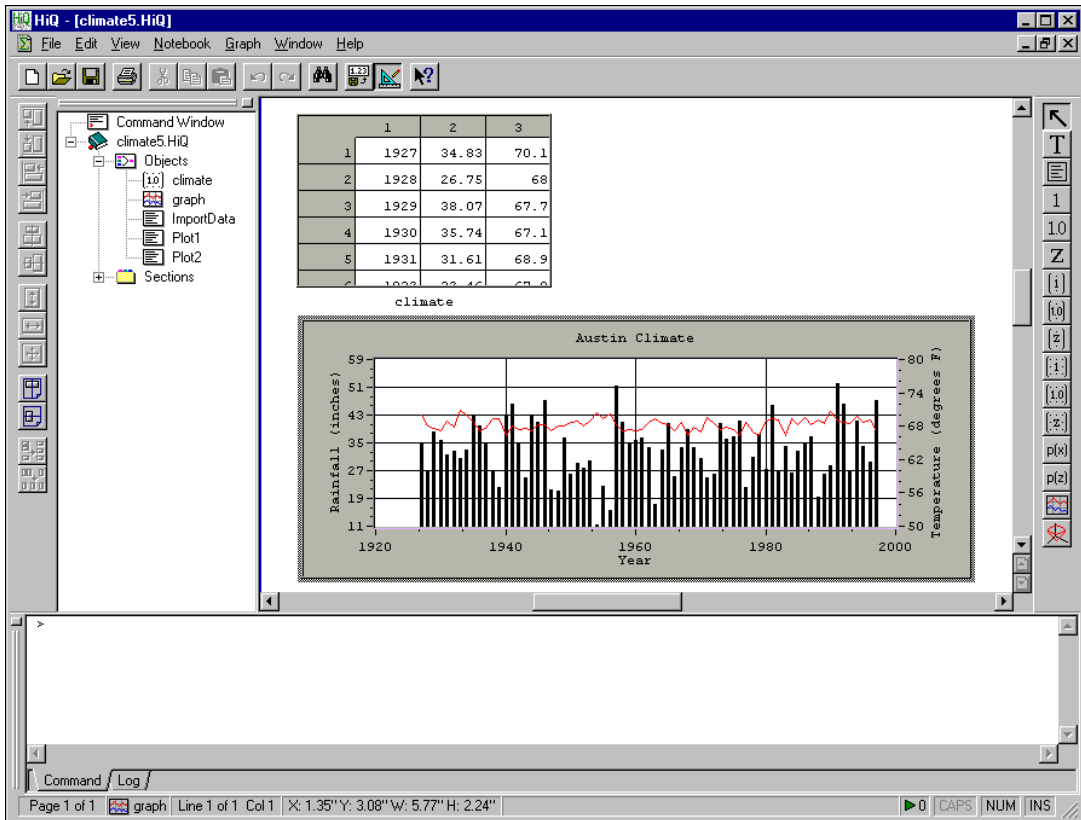
Figure 5-9. Climate Notebook with Two Related Data Plots

Seismic Notebook: Visualizing Seismic Data in a 3D Graph

For many real-world data sets—for example, terrain contours, the motion of an aircraft in three dimensions, the temperature distribution on a surface, and joint time-frequency analysis—you need to visualize data in three dimensions. In this example, you create a 3D graph of the seismic data.

Visualizing Data in 3D

1. Open the `seismic.HiQ` Notebook if you completed the Notebook example in Chapter 4, *Getting Your Data into HiQ*. Otherwise, open `seismic5.HiQ` from the `Examples\Getting Results` folder.

2. To add a 3D graph object to the Notebook, click on the 3D graph tool in the HiQ Tools toolbar.

3. Click and drag on the Notebook page where you want to place the new graph.

4. Drag and drop the Seismic matrix from the Object List in the Explorer to the graph on the Notebook page. When your mouse looks like the one in Figure 5-10, you can drop the data on the graph.
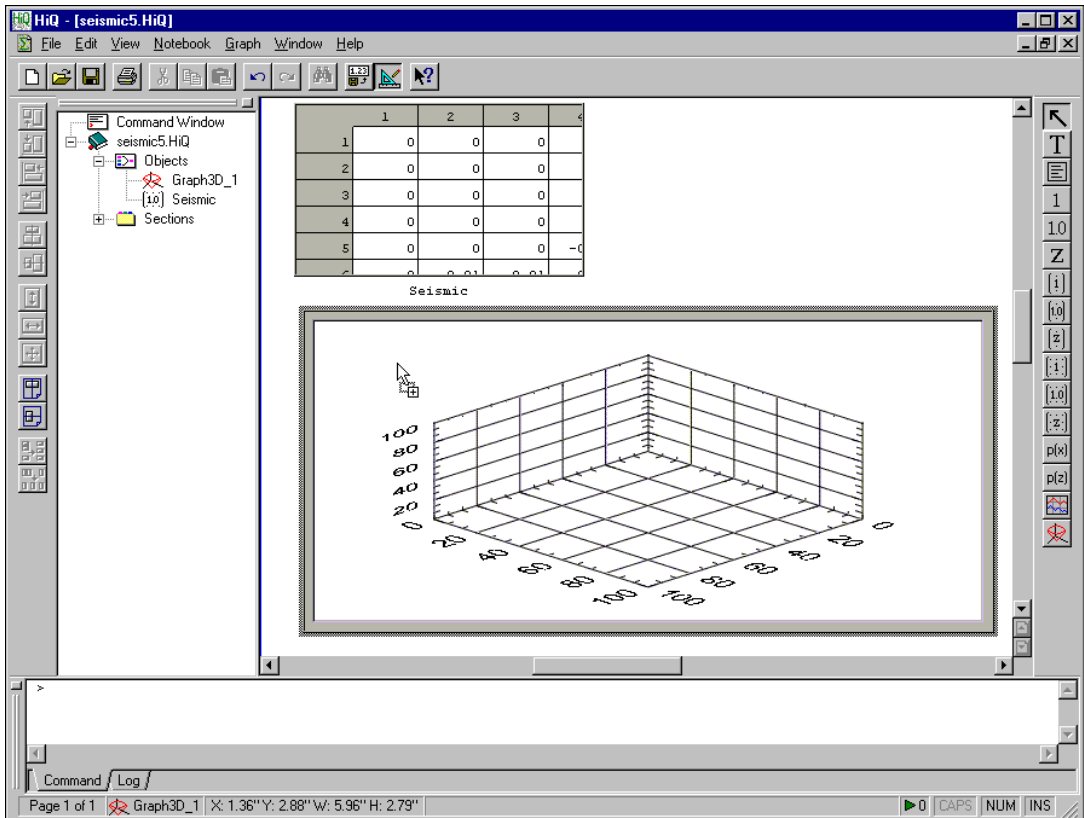
Figure 5-10. Using Drag-and-Drop to Plot Seismic Data

Modifying a 3D Graph and Plot

Three-dimensional graphs have all 2D graph properties plus several other properties of their own. You can access all properties from the Properties dialog box. For example, you might want to change the color of the surface plot and project the data to both the XZ and YZ planes.

1. Right click on the graph and select **Properties**.

2. Select the Graph tab and enter `Seismic Activity` for the graph **Title**.

3. Select the Axes tab and set the axis titles according to the following table:

Axis	Title
x	Time (sec)
y	Frequency (Hz)
z	Magnitude

☞ **Note**   *To switch between the axes, select the axis in the* **Axes** *list. To set the title for an axis, type the axis title in the* **Title** *field.*

4. From the Plots page, select the General tab and change the **Color Map** to **color spectrum**.
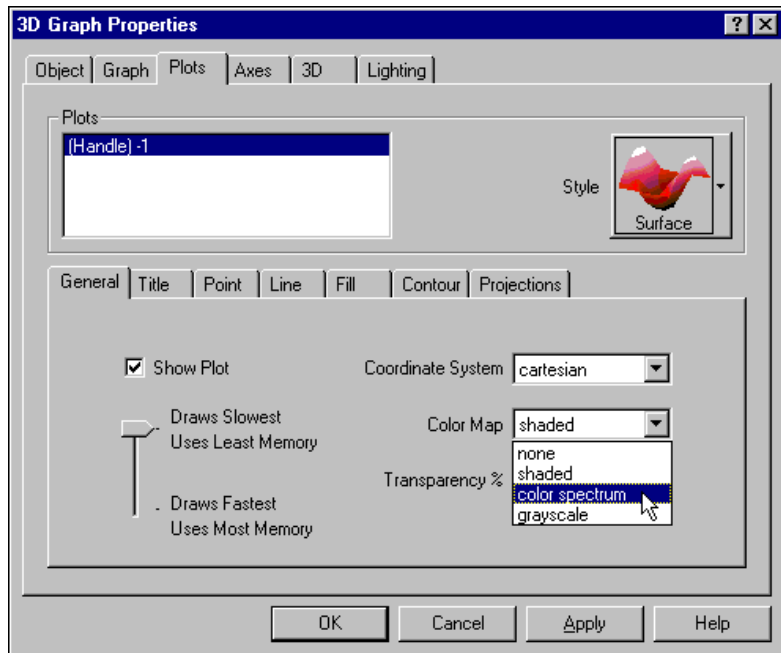
Figure 5-11. Setting the Plot Color Map

5. Click on the Projections tab and select **Project to X-Z Plane** and
    Project to Y-Z Plane.

6. Click on **OK**.

7. Save the Notebook as `seismic.HiQ`.

Figure 5-12 shows a graph similar to the one you have created that displays
data regarding seismic activity as a function of time and frequency. You can
experiment with the other 3D property pages—3D and Lighting—to
become more familiar with the flexibility of display formats for 3D graphs.
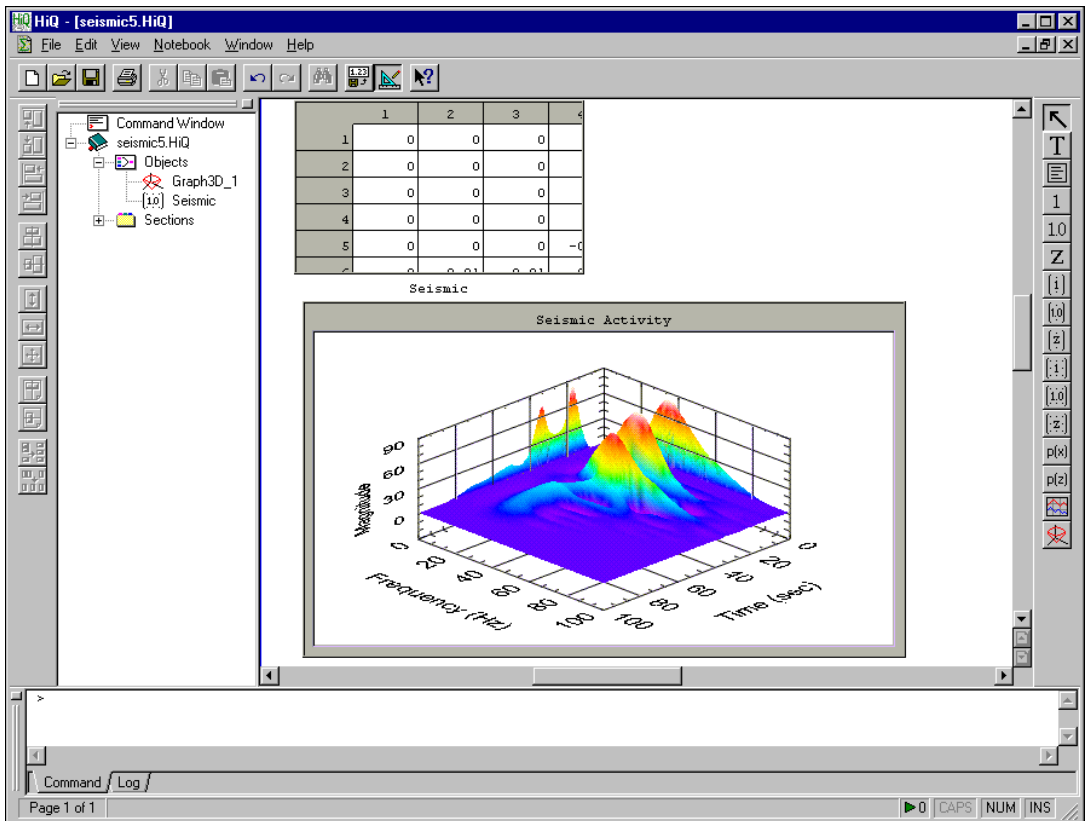
Figure 5-12. Seismic Notebook with Data Visualized in a 3D Graph

Rotating, Zooming, and Panning a 3D Graph

After putting a 3D graph into active mode, you can change the viewing position of it using combinations of the mouse, the <Alt> key, and the <Shift> key.

☞ **Note**   ***Click on a graph object to put it into active mode (the mode for editing the graph); a shaded border appears around the object. If you do not see the shaded border, the object is probably in selected mode (the mode for sizing and moving the object as a whole). Deselect the object by clicking on a blank space in the Notebook page and then click on the object again to put it into active mode.***

To rotate the graph, click and hold anywhere in the graph area and drag the mouse up, down, left, or right to rotate the graph up, down, left, or right respectively.

To zoom the graph, click and hold anywhere in the graph area while pressing the <Alt> key. Dragging the mouse down zooms the graph in and dragging the mouse up zooms the graph out.

To pan the graph, click and move the mouse anywhere in the graph area while pressing the <Shift> key.

6

Analyzing Data with HiQ-Script

This chapter introduces HiQ-Script as an analysis tool. In this chapter, you analyze the data in the Climate and Seismic Notebooks with HiQ-Script from the Command Window or the Notebook.

HiQ-Script, a programming language built for analysis, has many advanced features to make programming easier. With HiQ-Script, you have access to all HiQ built-in analysis functions, plus you have the power to create your own functions and programs using the HiQ-Script programming environment. In the first example, you use HiQ-Script to analyze rainfall data. In the second example, you use HiQ-Script to compute the energy in a signal.

Climate Notebook: Analyzing Rainfall Data

In this data analysis example, your goal is to add analysis to an existing Notebook that contains climate data to determine specific statistics, such as the average yearly rainfall totals and the average temperatures over the past 71 years.

You can access HiQ-Script in the Command Window or from a Script object on a Notebook page. The Command Window returns immediate results. The Script object encapsulates a series of HiQ-Script statements to extend the analysis and visualization capabilities of a Notebook.

1. Open `climate.HiQ` if you completed the Notebook example in Chapter 5, *Visualizing Data with 2D and 3D Graphs*. Otherwise, open `climate6.HiQ` from the `Examples\Getting Results` folder.

2. To quickly compute the average rainfall and temperature using the Command Window, type the following two commands.

   ```
   > averageRain = mean(climate[*,2])
   averageRain = 33.042958
   > averageTemp = mean(climate[*,3])
   averageTemp = 68.302817
   ```

   HiQ creates two new objects that contain the average of the second and third columns of the climate matrix. The brackets denote the desired

indices of the matrix, the asterisk denotes all row elements, and the 2 and 3 denote the second and third columns of the matrix. You can continue to perform any analysis from the Command Window. Although obtaining results from the Command Window might satisfy your information needs, you can quickly and easily document the results on the Notebook page.

3. Check the Object List in the HiQ Explorer. As shown in Figure 6-1, the objects you created—`averageRain` and `averageTemp`—are now listed. To place any object from the Object List on the Notebook page, you can drag-and-drop the object from the Explorer to the Notebook page.
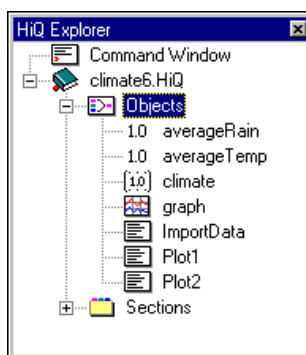
Figure 6-1. Climate Object List

4. Drag-and-drop the object `averageRain` on the Notebook page.

    You can resize this object and change any properties. For example, you might want to view exactly three decimal places of precision, including any trailing zeros.

5. Right click on the `averageRain` view, and select **Properties**. On the Numbers page, set the **Decimal Places** to 3 and uncheck **Remove Trailing Zeros**. Click on **OK**.

Setting Default View Properties

You can modify the properties of an object and make them the default properties so that any new views of an object of that same type have those properties.

1. Right click on the `averageRain` object view and choose **Defaults»Update Default Properties**.

2. Drag-and-drop the `averageTemp` object from the HiQ Explorer to the Notebook page near the `averageRain` object. Notice that this new view has the same size and decimal places as `averageRain`.

Figure 6-2 shows the Climate Notebook with the calculated average temperature and rainfall. You can print this Notebook to document the climate statistics over the last 71 years.
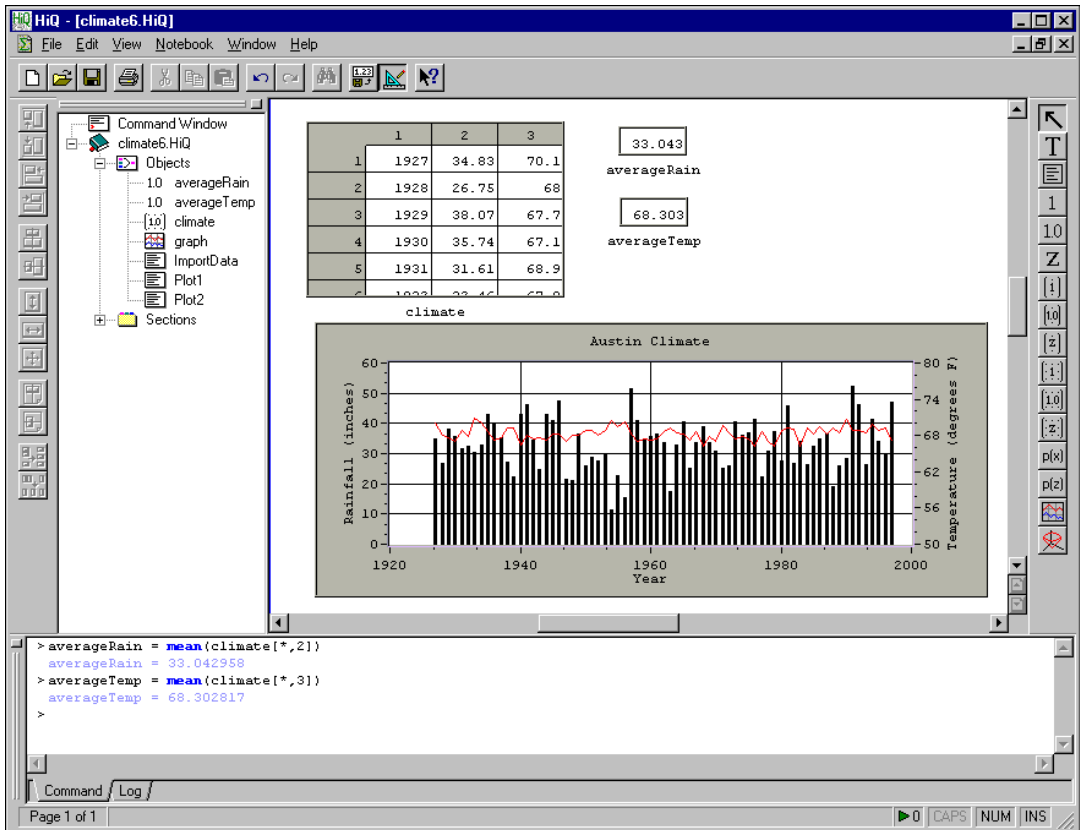
Figure 6-2. Climate Notebook After Analyzing Rainfall and Temperature Data

Looking for Trends

Finally, it might interest you to determine if there was a trend in the yearly rainfall data over the past 71 years. You can detect a simple trend by performing a linear data fit of the rainfall data. With the following steps, you can perform a data fit and graph the results.

The built-in functions in HiQ-Script are intuitively named. For example, you used the built-in function `mean` to compute the mean (or average) of a data set. The built-in function for performing a data fit is named `fit`, and the function for adding a plot to a graph is named `addPlot`.

1. At the Command Window prompt, type `fit` without pressing the <Enter> key. Notice that `fit` turns blue, indicating that HiQ recognizes this name as a built-in function.

2. With the cursor placed in `fit`, press the <F1> key. HiQ displays the online help for the `fit` function.

   This function can perform many kinds of data fits, from simple linear regression to complex multi-dimensional nonlinear fits. Because you are interested in the actual fitted data points in this example, you perform a line fit, which is the first usage listed for `fit`.

   For information about the value of each parameter, scroll down to the *Parameters* sections. Notice that the fifth return value returns the actual fitted data points.

3. To compute the fitted data, type the following script at the Command Window and press <Enter>.

   ```
   [,,,,fitData]=fit(climate[*,1],climate[*,2],<line>);
   ```

   The four commas on the left indicate that you do not want the first four values returned from the function.

☞ **Note**   ***HiQ-Script is case sensitive. Make sure you type the script exactly as it appears.***

4. Close the help window.

5. To add a plot of the fitted data to your graph, type the following script at the Command Window and press <Enter>.

   ```
   fitPlot=addPlot(graph,fitPlot,climate[*,1],fitData);
   ```

   The parameter `fitPlot` is a plot handle you can use to refer to the plot itself when you want to manipulate it later.

6. Save the Notebook as `climate.HiQ`.

Figure 6-3 shows the Climate Notebook, which now displays a plot representing the trend in rainfall over the last 71 years.
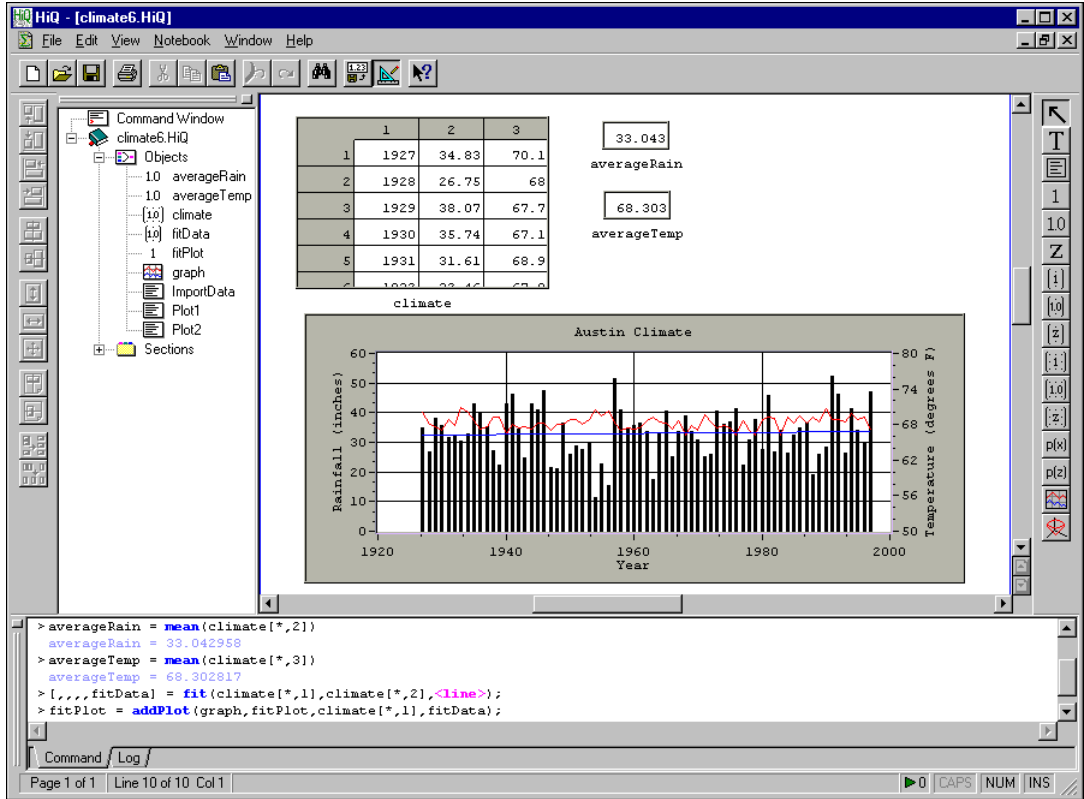
Figure 6-3. Climate Notebook Showing Rainfall Trends

Maintaining the Notebook

This Notebook now communicates the rainfall history over the past several years, as well as the average rainfall and average temperature. You can use this Notebook as a report for documentation or as a visual aid for a presentation.

If you repeat your analysis only once a year as new data becomes available, the steps you performed while creating this Notebook might suffice; however, if you need to repeat the analysis on a daily basis on many different sets of data, these steps might be inefficient. Because you can automate analysis in HiQ, you can create a Notebook that analyzes data when you push a button. You automate this analysis in Chapter 8, *Converting a Notebook to a Problem Solver*.

Seismic Notebook: Computing the Energy in the Signal

In this example, your goal is to compute the energy present in the seismic data at a particular point in time. Because the seismic data represents the power of the measured signal versus time and frequency, the energy in the signal at a specific time is the integral of the frequency signal at that time.

1. Open `seismic.HiQ` if you completed the Notebook example in Chapter 5, *Visualizing Data with 2D and 3D Graphs*. Otherwise, open `seismic6.HiQ` from the `Examples\Getting Results` folder.

2. Generate a vector representing the actual frequencies in the seismic data. Each column in the Seismic matrix represents the time domain data at a specific frequency. The first frequency point (the first column in the Seismic matrix) is 4 hertz, and the frequencies (columns) are separated by 4 hertz increments.

   Type the following line of HiQ-Script in the Command Window to create a vector of frequencies starting at four, incrementing by four, and containing as many elements as the Seismic matrix has columns.

   ```
   f = 4 * seq(Seismic.columns);
   ```

☞ **Note**   *HiQ-Script is case sensitive. Make sure you type the script exactly as it appears.*

3. Press <Enter>.

4. To compute the energy in the signal at time 30, type the following script in the Command Window and press <Enter>.

   ```
   energy = 1/(2*<pi>) * integrate(f, Seismic[30,*])
   ```

   The answer is echoed back in the Command Window, as shown in Figure 6-4.
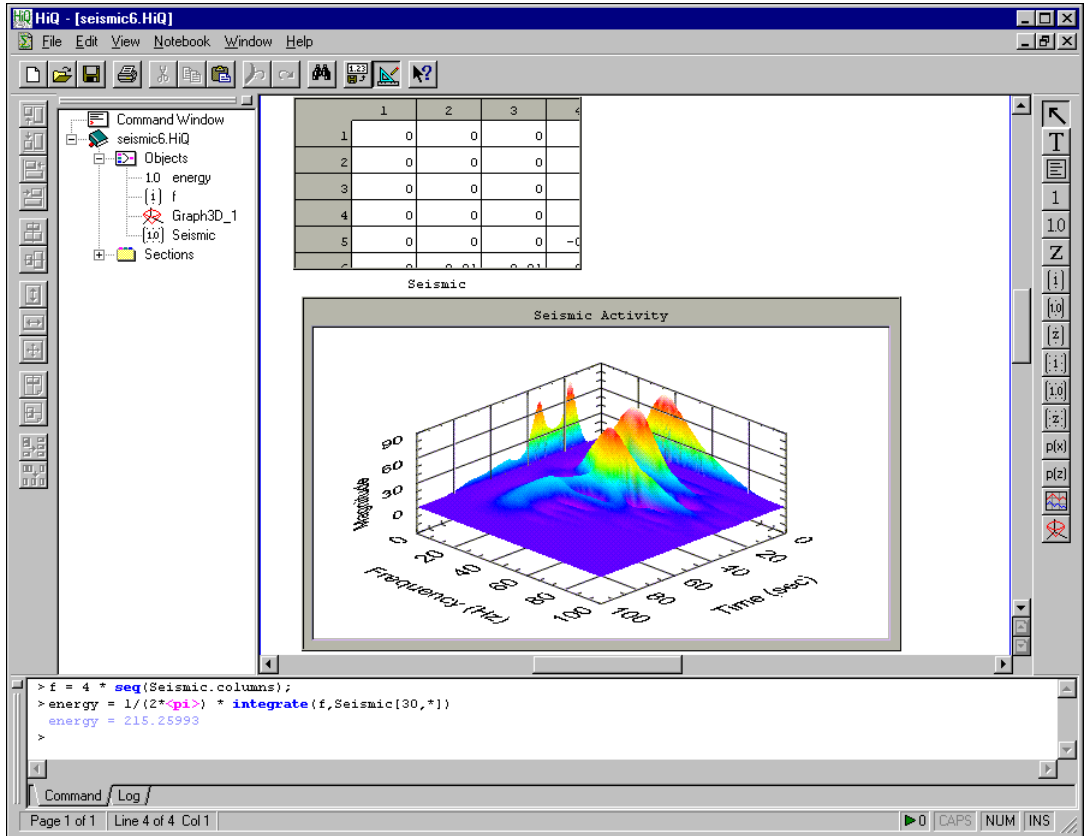
Figure 6-4. Seismic Notebook After Computing the Energy in a Signal

5. To find information about the integrate function, place the cursor in the function name and press the <F1> key.

6.   Click on the first usage for `integrate`. The help for this topic appears, as shown in Figure 6-5.
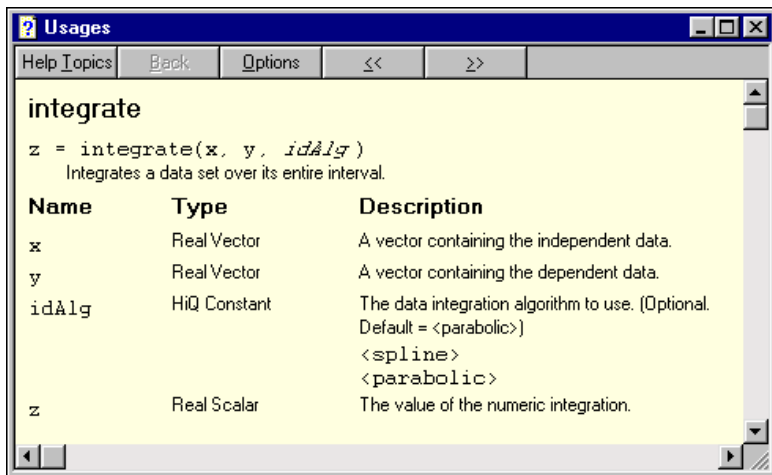
Figure 6-5. HiQ Online Help Describing the integrate Function

In this usage, `integrate` is integrating a data set specified by the second parameter with respect to the first parameter.

The data set is row 30 of the seismic data, which represents the frequency content of the seismic signal at 30 seconds. The `integrate` function allows you to optionally specify the algorithm to use for integrating the data. By default, the parabolic method is used.

7.   Close the help window.

Automating the Analysis

To repeat this analysis for any other time, you can recall the line of HiQ-Script and change the value 30 to the new time, or you can automate this analysis so that you can open the Seismic Notebook and compute the results by pressing a button.

1.   Place a new Script object on the Notebook page using the Script tool on the HiQ Tools toolbar.

2.   To compute the energy at time 30, type the following script in the Script object as shown in Figure 6-6.

```
energy = 1/(2*<pi>) * integrate(f, Seismic[30,*]);
```

☞ **Note**   *You can copy and paste this line from the Command Window to the Script object.*
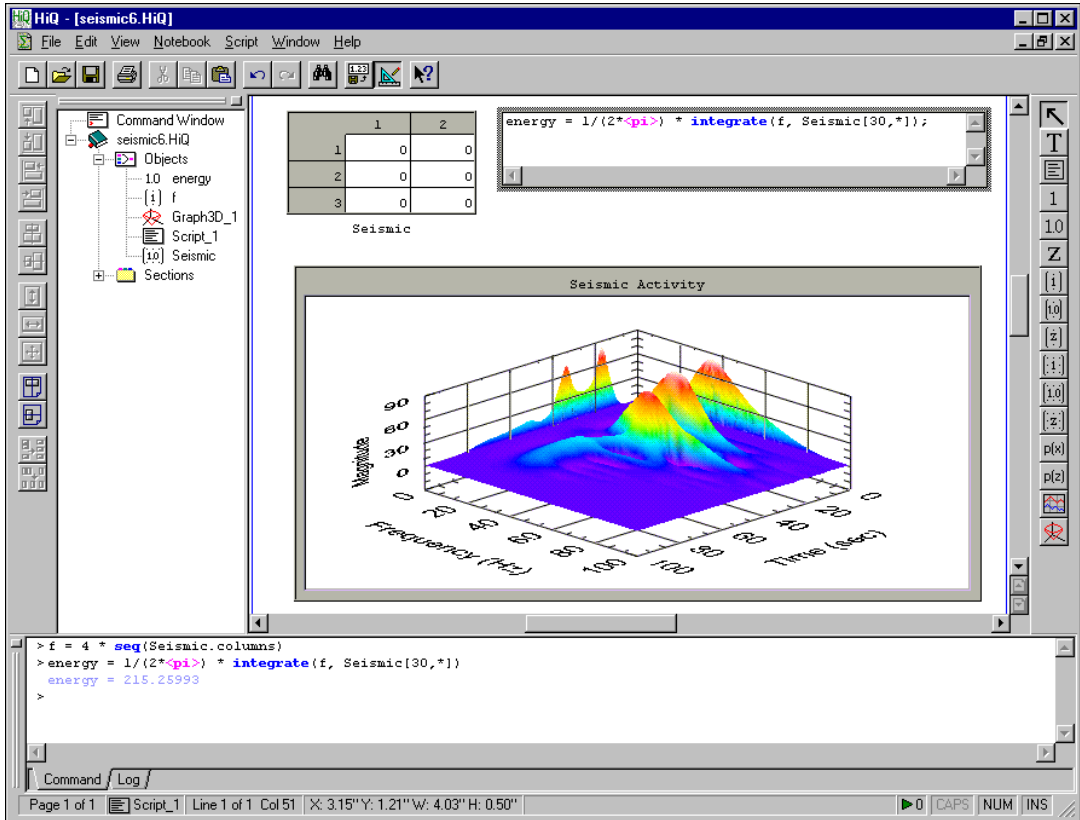
Figure 6-6. Adding Script to the Seismic Notebook to Automate Analysis

3. Right click on the script view and select **Properties**.

4. On the View page, select **Run View** and type Compute Energy for the **Caption**.

5. Click on **OK**.

6. Drag-and-drop a view of the energy object on the Notebook page.

7. Run the script by clicking on the **Compute Energy** run button.

8. Save the Notebook as seismic.HiQ.

Figure 6-7 shows the Seismic Notebook, which now computes and displays the energy contained in the signal at the thirtieth second.
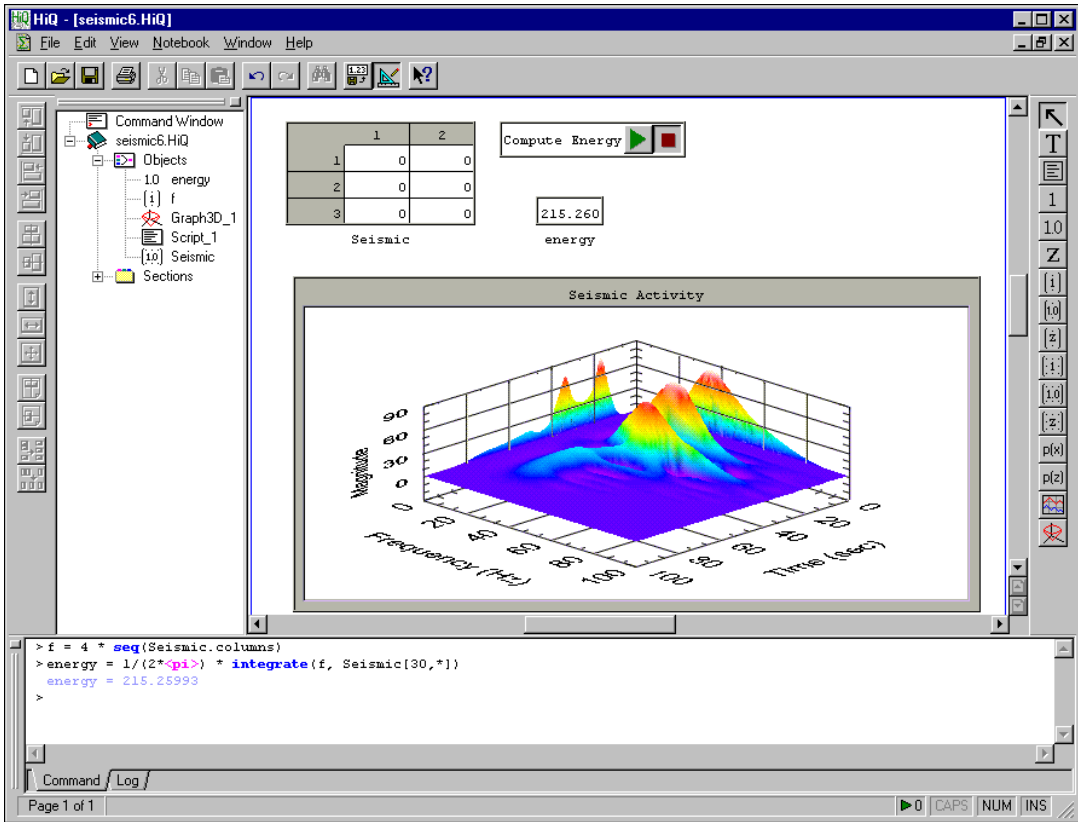
Figure 6-7. Seismic Notebook with Compute Energy Run Button

7

Taking Advantage of ActiveX

This chapter explains how you can use different ActiveX technologies in HiQ. In this chapter, you enhance the Climate and Seismic Notebooks with ActiveX components to extend their capabilities.

See Chapter 1, *ActiveX Connectivity*, in the *HiQ Reference Manual* for information about ActiveX connectivity and how you can use that technology in HiQ to communicate with other applications, embed objects from other applications in HiQ, embed HiQ Notebooks in other applications, control other applications from HiQ, control HiQ from other applications, and use ActiveX controls in HiQ

ActiveX allows you to customize your Notebook using any ActiveX control or object. For example, you can use an ActiveX control to access the Internet and download data from the Web or FTP sites in order to graph and analyze them.

ActiveX technologies add flexibility and versatility to your project. HiQ offers the following ActiveX technologies.

- **ActiveX Document Container**—As an ActiveX document container, HiQ allows you to embed documents and objects from other applications directly into your HiQ Notebook. For example, you can embed a Microsoft Word file, an Excel spreadsheet, or a PowerPoint presentation directly in your HiQ Notebook. These embedded objects are editable within the HiQ environment and can be stored within a HiQ Notebook or linked to an external file on disk.

- **ActiveX Document Server**—As an ActiveX document server, HiQ allows you to embed a HiQ Notebook into any other application that is an ActiveX document container. For example, you can embed a HiQ Notebook directly into a Microsoft Word document, an Excel spreadsheet, or a PowerPoint presentation. The embedded HiQ Notebook is editable within the other application and can be stored within the container document.

- **ActiveX Automation Client**—As an ActiveX automation client, HiQ allows you to run and control other applications from the HiQ environment. For example, you can create a HiQ Notebook that automatically launches Microsoft Word or Excel and then shares HiQ

data with the Word document or Excel spreadsheet to produce an automated report.

• **ActiveX Automation Server**—As an ActiveX automation server, HiQ allows you to run and control a HiQ Notebook from within any other application that is an ActiveX automation client. For example, you can create a program in another application, such as Microsoft Visual Basic, LabVIEW, or LabWindows/CVI, that automatically launches HiQ, opens a HiQ Notebook, and sends data to HiQ for automated analysis, visualization, and report generation.

• **ActiveX Controls Container**—As an ActiveX controls container, HiQ allows you to embed ActiveX controls directly into your HiQ Notebook. These embedded controls can be accessed both interactively and programmatically from within the HiQ environment. For example, you can embed a Microsoft Web Browser or National Instruments ComponentWorks control directly into your HiQ Notebook and then access that control to automatically gather data from the Internet or a physical measurement device.

☞ **Note**   *There are two types of objects in HiQ: native objects and ActiveX objects.*

Climate Notebook: Adding ActiveX Objects

Because HiQ Notebooks are ActiveX containers, you can embed ActiveX controls such as National Instruments ComponentWorks controls to extend the capabilities of your Notebook, including the ability to perform data acquisition from HiQ.

In this example, you add a button to the Notebook that indicates whether a script performs a data fit on the rainfall data. When the button is on, the script performs the data fit and plots the data fit. When the button is off, the script removes the data fit plot.

☞ **Note**   *To complete this example, you must have ComponentWorks installed on your computer. If you do not have ComponentWorks, install the ComponentWorks demo from the HiQ installation CD or download it from the National Instruments Web site at* `www.natinst.com`.

Adding a ComponentWorks ActiveX Control

1. Open `climate.HiQ` if you completed the Notebook example in Chapter 6, *Analyzing Data with HiQ-Script*. Otherwise, open `climate7.HiQ` from the `Examples\Getting Results` folder.

2. To place an ActiveX control in a Notebook, use the **Edit»Insert Control** menu.

3. Select **CWButton Control** in the Insert Control dialog box, and click on **OK**. The control is placed on the Notebook page.
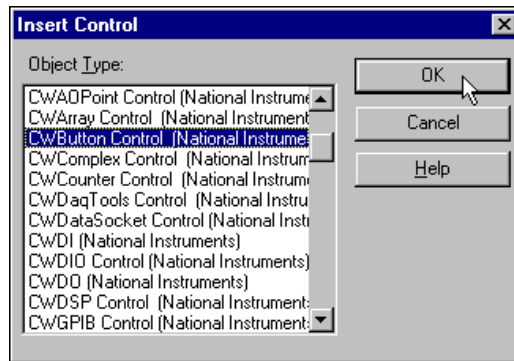
Figure 7-1. Insert Control Dialog Box

4. Resize the control as needed.

5. Right click on the new control and drag it if you need to reposition the control on the page.

6. Right click on the control and choose **Rename** to name the object `switch`.

☞ **Note**   *When you place an ActiveX control in a Notebook, the Notebook enters ActiveX Design Mode, which allows you to set properties for the control. Some controls only operate when ActiveX Design Mode is off. To test or operate a control, turn off ActiveX Design Mode by unchecking* **ActiveX Design Mode** *in the* **Notebook** *menu.*

7. Right click on the button control and choose **CWButton Control (National Instruments) Object»Properties**.

   The property pages that you see vary depending on the control. HiQ does not create the property pages for ActiveX controls. The property pages of the ActiveX control should clearly document its properties and methods.

8. On the Style page, choose the On/Off Toggle button, as shown in Figure 7-2.
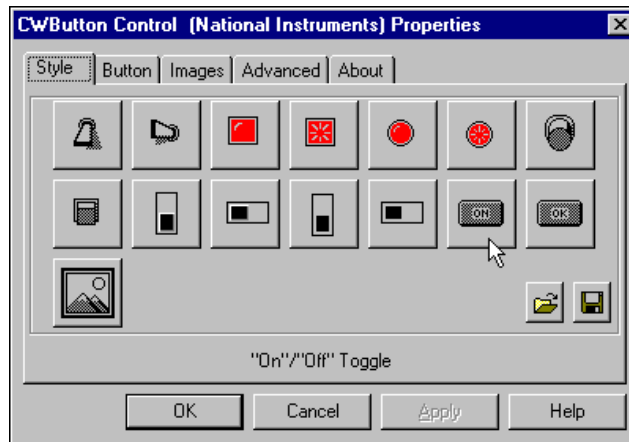
Figure 7-2. Selecting a Style for the CWButton Control

9. On the Button page, type `Perform Data Fit` for the **Caption** property.

10. Click on **OK**.

Programmatically Accessing the ActiveX Control

You can use the controls's properties and methods to programmatically access its capabilities from HiQ-Script. Use the HiQ ActiveX Object Browser to explore the methods and properties available in the control, as explained in following steps.

1. From the Command Window, type `browse switch` and press <Enter>. You also can right click on the switch object and select **Browse**.

   The Object Browser is displayed with two lists: the left pane displays the list of controls in a library with the particular control highlighted, and the right pane lists the methods and properties for the highlighted control.

2. Scroll down the methods and properties list in the right pane and select the `Value` property, which you will use to determine the position of the switch.

   Notice the syntax help and the property description for `Value` at the bottom of the ActiveX Object Browser. Use this information to access the control from HiQ-Script.

3. Place a HiQ-Script object on the Notebook.

4. To perform the data fit on the rainfall data when the switch is on and remove the plot when the switch is off, type the following HiQ-Script in the new HiQ-Script object.

```
//If switch is on, calculate the fit and plot it
//on the graph.
if switch.value == 1 then
    [,,,,fitData] = fit(climate[*,1],climate[*,2]);
    fitPlot =
    addPlot(graph,fitPlot,climate[*,1],fitData);
else
    //If switch is off, remove the fit plot.
    removePlot(graph,fitPlot);
end if;
```

5. Right click on the script and select **Properties**. On the View page, select **Run View** and type Compute Fit for the **Caption**, as shown in the following figure.
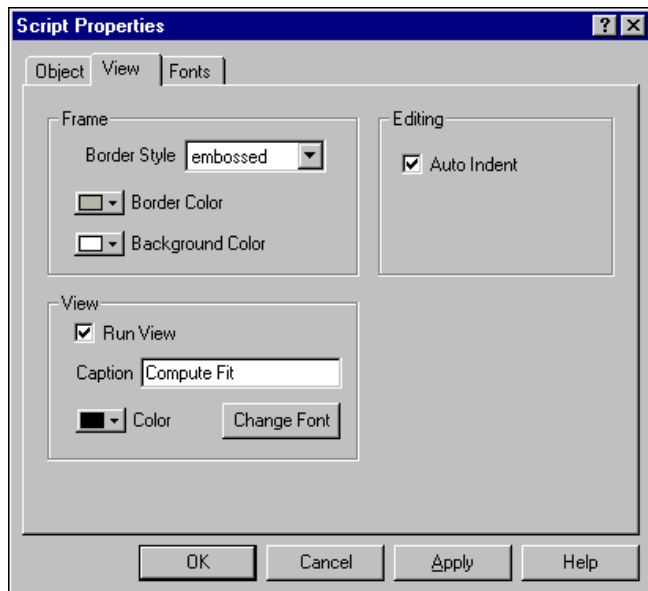
Figure 7-3. Setting View Properties for a Script Object

6. Click on **OK**.

7. Run the script to plot and analyze the data. Notice the linear data fit on the graph. You can remove the plot by clicking on the switch object and running the script again.

8. Save the Notebook as `climate.HiQ`.

Figure 7-4 shows the Climate Notebook with the CWButton control.
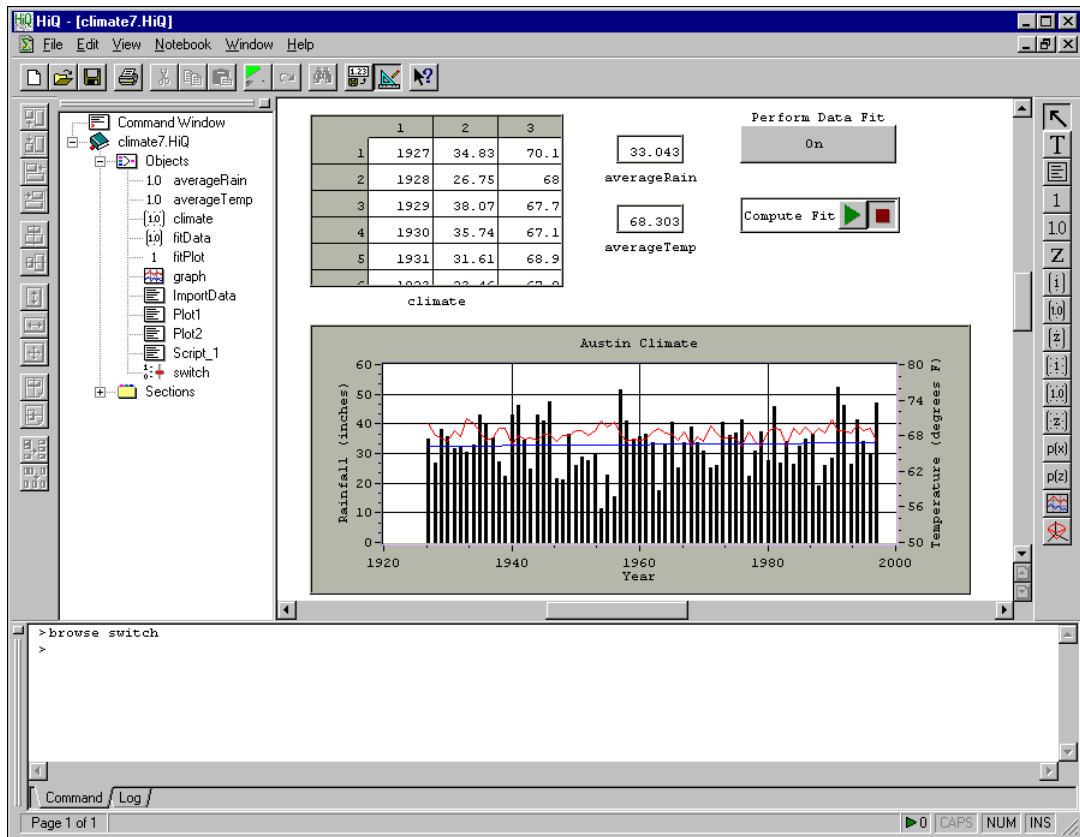
Figure 7-4. Climate Notebook with a ComponentWorks ActiveX Control

Adding a Microsoft Word Document

You also can embed ActiveX documents such as a Word document, an Excel spreadsheet, or PowerPoint slides in your Notebook. For this Notebook, add a Word document to summarize conclusions about this Notebook.

☞ **Note**   *To complete this example, you must have Microsoft Word installed on your computer.*

1. To place an ActiveX object in a Notebook, use the **Edit»Insert Object** menu item.

    When you insert an ActiveX object, you can either link the object to an existing file or embed a new object directly in HiQ. In both cases, the object appears on the Notebook page. While an embedded object stores its data in a Notebook file, a linked object does not. When a linked object file changes, the object displayed in the HiQ Notebook automatically updates. However, if the linked object file outside of the Notebook is moved or deleted, the object within the Notebook is no longer visible in HiQ.

2. Select a **Microsoft Word Document** object and the **Create New** option.
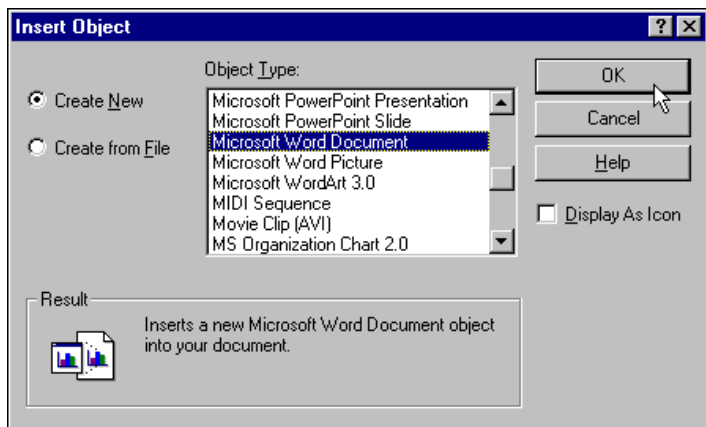
Figure 7-5. Insert Object Dialog Box Lists ActiveX Objects You Can Insert in HiQ

3. Click on **OK** to insert a new Microsoft Word document.

4. Use the power of the embedded Word document to enter and format text. For this example, type the following text:

    The average yearly rainfall is *increasing*, while the average temperature is remaining *steady*.

5. When you finish entering and formatting your text, click on the Notebook page (outside of the Microsoft Word Document object).

6. Save the Notebook as climate.HiQ.

Figure 7-6 shows the Climate Notebook with the embedded Microsoft Word Document.
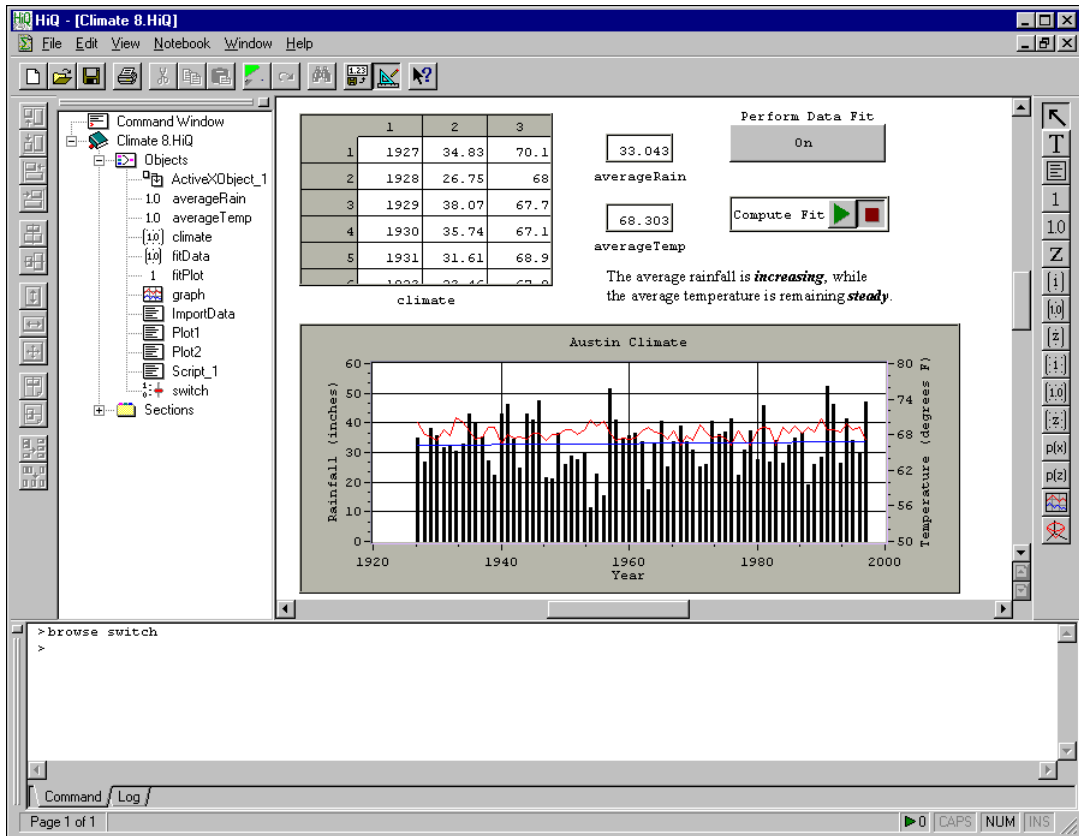
Figure 7-6. Climate Notebook with an Embedded Microsoft Word Document

Seismic Notebook: Adding a ComponentWorks Control

In this example, you add a ComponentWorks Slide control that specifies which algorithm to use for computing the energy in the signal.

☞ **Note**   *To complete this example, you must have ComponentWorks installed on your computer. If you do not have ComponentWorks, install the ComponentWorks demo from the HiQ installation CD or download it from the National Instruments Web site at* www.natinst.com.

Adding an ActiveX Control

1. Open `seismic.HiQ` if you completed the Notebook example in Chapter 6, *Analyzing Data with HiQ-Script*. Otherwise, open `seismic7.HiQ` from the `Examples\Getting Results` folder.

2. To place an ActiveX control in a Notebook, use the **Edit»Insert Control** command.

3. Select **CWSlide control**, and click on **OK**.

4. Right click on the slide and choose **Rename**. Rename the object `method`.

5. Right click on the slide and drag it if you need to reposition the control on the page.

You can specify which numerical integration algorithm you want to use with the Slide control. HiQ uses two methods to compute the integral of a data set: `spline` and `parabolic`. Refer to the *integrate* topic in the online help for more information about these methods. Complete the following steps to modify the Slide control properties.

6. Right click on the slide and choose **CWSlide Control (National Instruments) Object»Properties**. The control generates these property pages so you can customize the control.

7. On the Style page, choose **Vertical Slide** and **Value pairs only** for the Style.

Figure 7-7. Setting Style Properties for the ComponentWorks Slide Control

8. On the Pointer page, choose **None** for **Fill Style**.

9. On the Value Pairs page, delete all but two value pairs in the list. Change one of the pairs' name to `spline` and value to `0`, and change the other's name to `parabolic` and value to `1`.
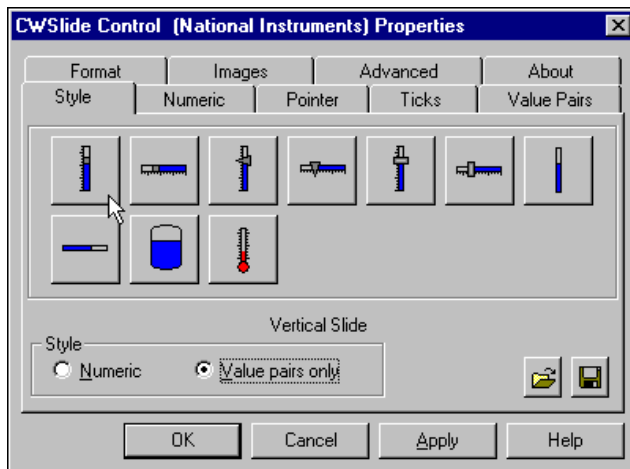
10. Click on **OK**.

11. Resize the control as needed.

Programmatically Accessing the ActiveX Control

To access the value of the slide, use the `Value` property. Because the `integrate` function expects a HiQ Constant as the third parameter to specify the method, you need to write a short user function to return the appropriate HiQ Constant based on the actual value of the slide.

1. Right click on the **Compute Energy** run button and choose **View Source**. The object containing the HiQ-Script is now visible in its own window.

2. To determine the HiQ Constant for the integration method using the slide, add a user function to the end of the script object as follows.
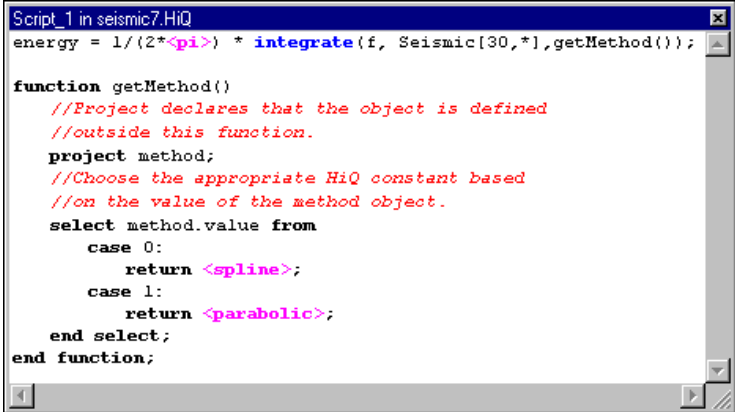
```
function getMethod()
    //Project declares that the object is defined
    //outside this function.
    project method;
    //Choose the appropriate HiQ constant based on the
    //value of the method object.
    select method.value from
    case 0:
        return <spline>;
    case 1:
        return <parabolic>;
    end select;
end function;
```

To extend your analysis, you can include user functions in your HiQ-Scripts. For example, the user function `getMethod` returns the correct HiQ constant given the position of the slide. This function is not automatically executed when you run the script. It is executed only when called as a function from elsewhere in your script. In fact, you can write this function in a separate script object if you do not want to include it in the script that computes the energy. The next step references this function in the call to `integrate`.

3. To compute the energy with the specified integration method, modify the call to the built-in function `integrate` at the top of the HiQ-Script.

```
energy = 1/(2*<pi>) * integrate(f, Seismic[30,*],
  getMethod());
```

The third parameter to the function `integrate` is a call to the user function `getMethod`. In the previous step, you defined this user function to return the HiQ constant `<spline>` or `<parabolic>`, depending on the position of the slide. Figure 7-8 shows the completed script.

Figure 7-8. HiQ-Script Accessing the User Function getMethod

4. Close the Script window.

5. Execute the script twice, using each integration algorithm to see how the algorithms affect the energy computation.

   To display more precision in the answer, change the **Decimal Places** property in the property pages for the energy object.

6. Save the Notebook as `seismic.HiQ`.

Figure 7-9 shows the Seismic Notebook with the ComponentWorks Slide control, which controls the integration method used.
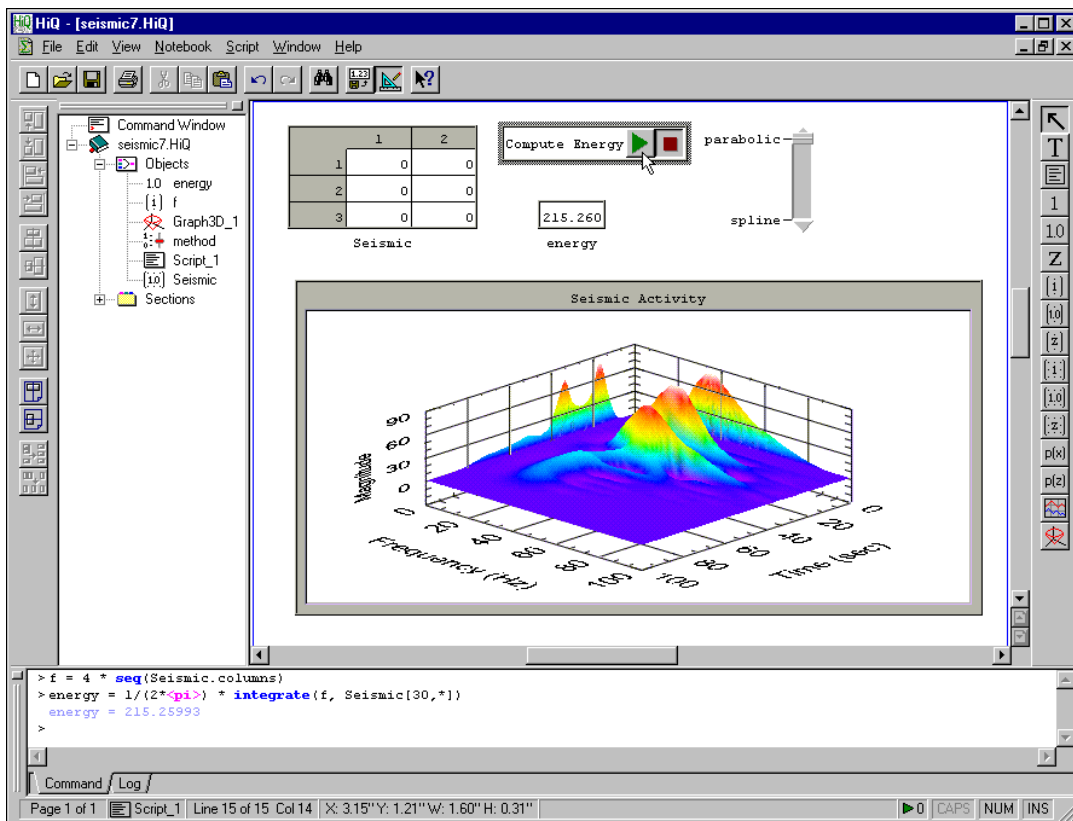
Figure 7-9. Seismic Notebook with the ComponentWorks Slide Control

8

Converting a Notebook to a Problem Solver

This chapter introduces problem solving Notebooks, which are useful tools for analyzing different data or performing different analysis on a specific data set. In this chapter, you complete the Climate and Seismic Notebooks by converting them to Problem Solvers that can analyze a variety of different data sets.

A *Problem Solver* is a Notebook that is flexible enough to work with different data or perform different analysis on a specific data set when you press a button. For example, a custom HiQ Notebook might be able to compute the best exponential data fit to the growth of a specific stock price in a specific file, but you would need to modify the Notebook if you wanted a different type of fit or a different data file. A Problem Solver Notebook allows you to choose the data file and the type of data fit to perform without having to redesign the Notebook.

Climate Notebook: Converting to a Problem Solver

In Chapter 7, *Taking Advantage of ActiveX*, you completed a Notebook that analyzed specific rainfall data. In the following Notebook example, you convert that Notebook to a Problem Solver that allows you to choose the data file, graph the data, and compute the mean and standard deviation when you press a button.

1. Open `climate.HiQ` if you completed the Notebook example in Chapter 7, *Taking Advantage of ActiveX*. Otherwise, open `climate8.HiQ` from the `Examples\Getting Results` folder.

   In previous chapters, you imported data, graphed the data, and finally analyzed the data. However, you can automate this process with HiQ-Script so you can quickly repeat your analysis with another data file. This Notebook already contains a Script object that computes a data fit. You can modify this script to import a data file and compute averages as well.

2. To edit the script represented by the **Compute Fit** run button, right click on the button and choose **View Source**.

   A floating window containing the script appears, as shown in Figure 8-1. You can position and size this window.


```
Script_1 in climate8.HiQ                                              ☒
//If switch is on, calculate the data fit and plot it on the graph.  ▲
if switch.value == 1 then
    [,,,,fitData] = fit(climate[*,1],climate[*,2]);
    fitPlot = addPlot(graph,fitPlot,climate[*,1],fitData);
else
    //If switch is off, remove the data fit plot.
    removePlot(graph,fitPlot);
end if;                                                              ▼
◄                                                              ►
```

Figure 8-1. Climate Notebook Script Displayed in a Floating Window

3. Select **Compile All Scripts** from the **Notebook** menu to compile all the HiQ-Scripts in this Notebook. This step compiles the previously generated scripts, Plot1, Plot2, and ImportData, and creates three new functions, Plot1_Run, Plot2_Run, and ImportData_Run.

4. Edit the Script object to import and graph the data by typing the following lines at the beginning of the script:

```
//Import the climate data file.
ImportData_Run();
```

```
//Remove all plots from the graph.
removePlot(graph);
```

```
//Plot the rainfall data in the second column.
Plot1_Run();
```

```
//Plot the temperature data in the third column.
Plot2_Run();
```

5. Change the rainfall plot to a vertical bar by typing the following lines:

```
//Change the rainfall plot to a vertical bar.
graph.plot(-1).style = <verticalbar>;
```

6. Associate the temperature plot with the second y-axis and change the y-axis range:
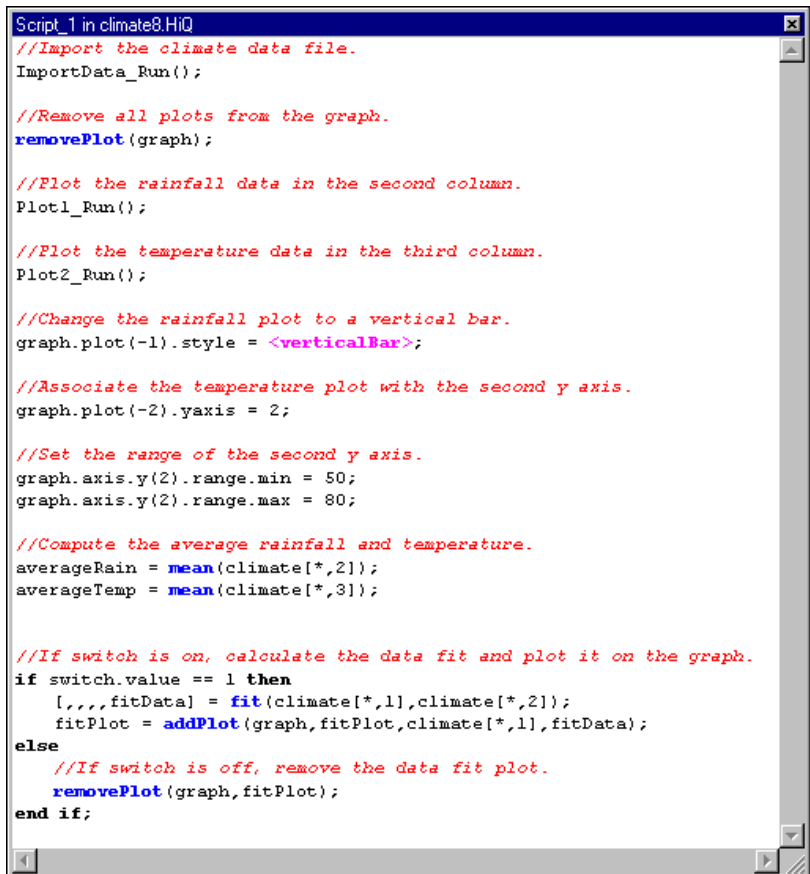
```
//Associate the temperature plot with
//the second y axis.
graph.plot(-2).yaxis = 2;
```

```
//Set the range of the second y axis.
graph.axis.y(2).range.min = 50;
graph.axis.y(2).range.max = 80;
```

7. Compute the mean of the rainfall and temperatures by typing the
    following lines:

```
//Compute the average rainfall and temperature.
averageRain = mean(climate[*,2]);
averageTemp = mean(climate[*,3]);
```

Figure 8-2 shows the final script in the script window.

Figure 8-2. Climate Notebook Final Script

8. Close the script window.

9. To test the Notebook, run the script by clicking on the **Compute Fit** run button and selecting the `climate.dat` data file.

10. Save the Notebook as `climate.HiQ`.

Figure 8-3 shows the Climate Notebook, which now imports a data file, graphs the data, computes the averages, and optionally performs a data fit when you press the **Compute Fit** run button.
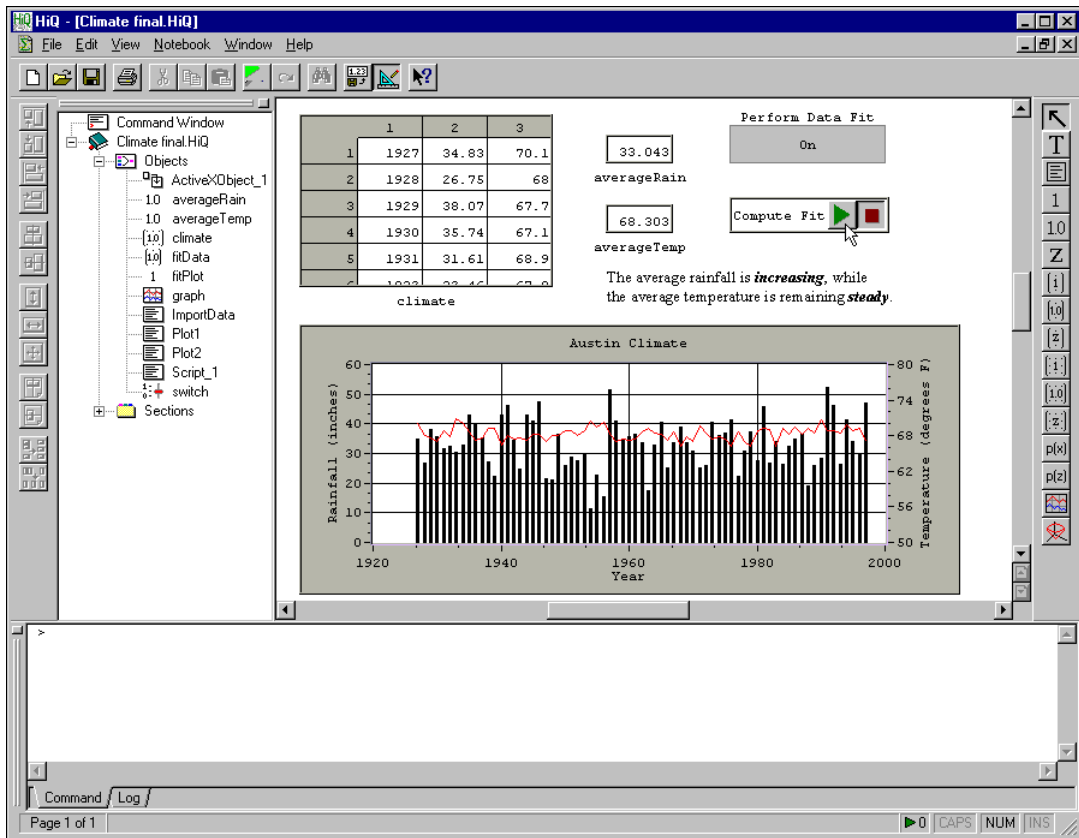
Figure 8-3. Climate Notebook as a Problem Solver

Seismic Notebook: Converting to a Problem Solver

In Chapter 6, *Analyzing Data with HiQ-Script*, you completed a Notebook that computed the energy contained in a joint time-frequency representation of a seismic data set at a particular time. In the following example, you convert that Notebook to a problem solver that allows you to choose the time value to compute the energy when you press the **Compute Energy** run button.

1. Open `seismic.HiQ` if you completed the Notebook example in Chapter 7, *Taking Advantage of ActiveX*. Otherwise, open `seismic8.HiQ` from the `Examples\Getting Results` folder.

2. Place a new Integer Scalar object on the Notebook page and rename it `t`.

3. Right click on the **Compute Energy** run button and select **View Source**.

4. To compute the energy at the time specified by the integer scalar object, modify the first line of the script by replacing `30` with `t`.

   ```
   energy = 1/(2*<pi>) * integrate(f, Seismic[t,*],
   getMethod());
   ```

5. Close the script window.

6. To recompute the energy at the specified time, enter a time in the `t` object. For this example, enter `50` and click on the **Compute Energy** run button.

   The energy object updates its value according to the time you specify in `t`.

7. Save the Notebook as `seismic.HiQ`.

Figure 8-4 shows the Seismic Notebook, which serves as a Problem Solver that computes the energy at any time t using either of the integration methods.
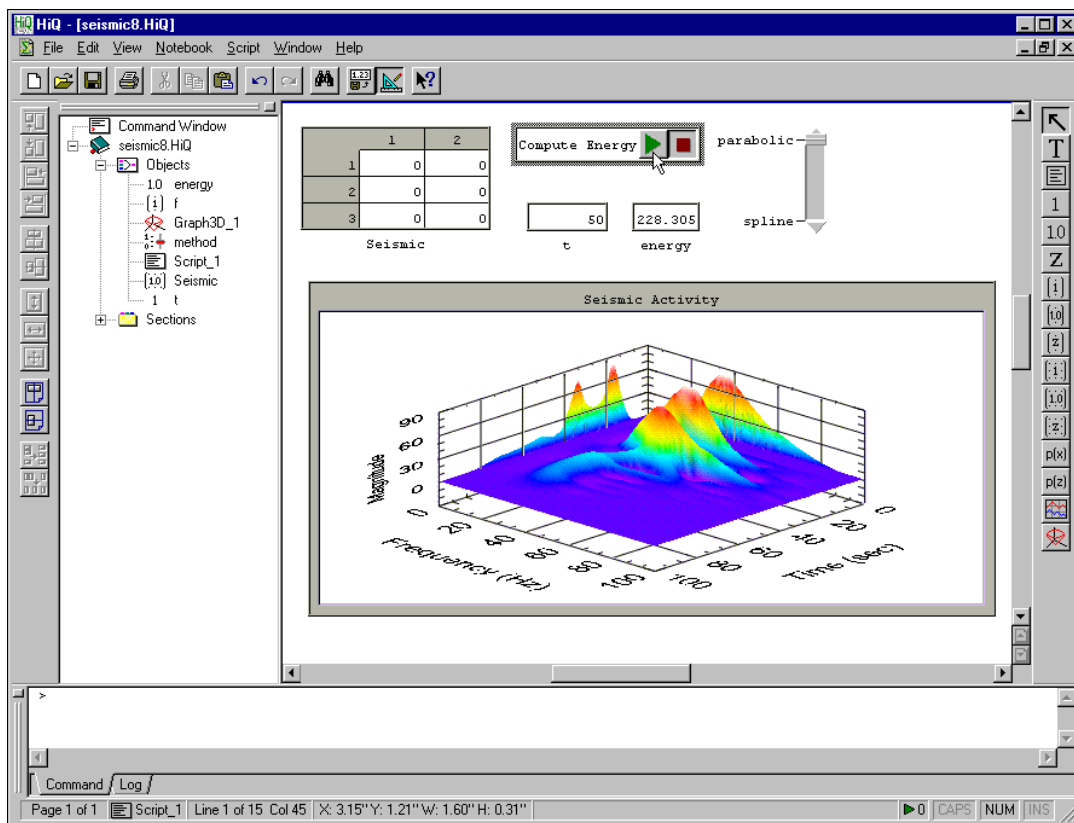
Figure 8-4. Seismic Notebook as a Problem Solver

9

Getting Results as a MATLAB User

This chapter summarizes the HiQ to MATLAB transitional tools—such as the HiQ Command Window, the automated M-File to HiQ-Script translator, and HiQ-Script for MATLAB users—and describes how MATLAB users can use these HiQ features to leverage work they have already done in MATLAB.

About HiQ

HiQ is designed specifically for scientists, engineers, and technical professionals. It offers flexible graphics for data visualization and includes a mature analysis library and scripting language that handles the rigorous requirements of current problem solving. The intuitive, natural user interface encourages the use of all the powerful capabilities in HiQ without the difficulties of traditional programming packages. In fact, this ease-of-use approach is the defining factor that separates HiQ from other analysis and data visualization packages.

Traditionally, when you changed from one analysis package to another, you encountered incredible obstacles, such as learning the features in the new package, transferring all work previously done in the prior package, fixing problems created by the transfer, and implementing functionality available in the old version that is not available in the new version. With HiQ 4.0, you will find transitioning to the HiQ environment easier than ever.

HiQ offers a transitional approach that allows you to continue working while learning the HiQ environment. Features that support this approach include the following:

- HiQ Command Window—The HiQ Command Window provides two interface modes: a HiQ mode that exploits all HiQ features and a 100% MATLAB 5 compatibility mode using your installed version of MATLAB 5 or greater.

- Automated M-File to HiQ-Script Translator—The translator and translation library are integral in supporting your progress in moving from MATLAB to HiQ.

- HiQ-Script—The HiQ-Script language is a programming language specifically designed for analysis. HiQ-Script is comparable to MATLAB M-File programming but contains additional programming constructs and built-in data types.

- HiQ Online Help for MATLAB Users—The online help is always available by pressing the <F1> key.

The goal of this chapter is to present ways in which HiQ and MATLAB can be used together. This chapter offers MATLAB-to-HiQ transition avenues to help you migrate at your own pace while still making optimal use of your previous work and valuable time.

HiQ Command Window

In MATLAB, you use the command line interface to interact with your analysis results. Because you can use the HiQ Command Window exactly as you do in MATLAB, you can get acquainted with HiQ while you continue to use the data and M-Files you have already created in MATLAB.

Accessing the HiQ Command Window

After starting HiQ, access the command line feature by opening the Command Window. Verify that the **Command Window** option in the **View** menu is checked. If not, select it.

If you are running HiQ for the first time, the Command Window appears in the lower section of the HiQ window. If you have used HiQ or moved the Command Window before, it appears in the last position it was placed.

At the top of the Command Window, you should see a > prompt, which is the prompt used by the HiQ command line while in HiQ mode. In the next section, you switch to MATLAB mode.

Accessing and Using MATLAB 5 Compatibility Mode

☞ **Note**   *To complete this example, you must have MATLAB 5 or greater installed on your computer.*

At the HiQ Command Window prompt, type the following command:

```
> matlab
```

After a few seconds, the MATLAB prompt, **»**, appears.

☞ **Note**   *Because the installation of MATLAB version 5.0 does not automatically register the application with the operating system, you might see the following error instead of the MATLAB prompt:*

```
Error:Unable to launch MATLAB. Verify that MATLAB is
properly installed.
```

If you receive this error, start MATLAB from the operating system with the following option:

```
MATLAB.exe /regserver
```

This option registers MATLAB with the operating system so other applications, including HiQ, can communicate with it.

Exit MATLAB and return to HiQ. Execute the `matlab` *command again in the Command Window to get the MATLAB prompt.*

Once you enter MATLAB mode, you can work as if you were in MATLAB. For example, you can perform all of the following tasks:

• Execute your existing M-Files by typing in their name and associated parameters.

• View the data objects by typing in their name and pressing return.

• Create and manipulate any data objects with the MATLAB programming language.

• Run any command defined in the MATLAB command line interface.

For example, try the following commands:

```
» x = eye(2)
» x
» size(x)
» x(2,2) = 5;
» cd
» help inv
» y=inv(x);
» sqrt(y)
```

Notice that information is displayed exactly as it is in MATLAB. The `inv` and `sqrt` functions used in this example are also HiQ built-in functions. In HiQ mode, the names of these functions are highlighted in blue to clearly indicate that you are calling a HiQ built-in function.

Try experimenting with various MATLAB commands. When the HiQ Command Window is in MATLAB compatibility mode, you have access to all MATLAB commands and programs from within HiQ.

The next section concentrates on sharing data and programs between HiQ and MATLAB using the MATLAB compatibility mode of the HiQ Command Window.

Sharing Data and Functionality

You have already seen an example of HiQ using MATLAB functionality with the `inv` and `sqrt` functions. However, this capability is not limited to functions shipping with MATLAB. Any M-File created by you or your colleagues is accessible using the same approach. In addition to running programs, you also can share data between HiQ and MATLAB.

To allow data transfer between applications, four commands have been added to the MATLAB mode: `get`, `getall`, `put`, and `putall`. With these commands, you can share any numeric or textual data created by either application.

- `get` *X*—Copies the variable name *X* from the current MATLAB workspace to an object in the active HiQ Notebook.

- `getall`—Performs a `get` on all variables in the MATLAB workspace.

- `put` *X*—Copies the HiQ object named *X* from the active HiQ Notebook to the current MATLAB workspace.

- `putall`—Performs a `put` on all HiQ objects in the active HiQ Notebook.

At the MATLAB prompt, type the following commands.

```
» r = rand(50);
» h = surf(r)
```

☞ **Note**   *When using MATLAB handle graphics, the* `get(h)` *function still operates as expected.*

A new window has been generated by MATLAB and named `Figure 1`. It contains a static representation of the random surface. Now, get the data from MATLAB to HiQ and graph it in HiQ with the following commands.

```
» get r
» hiq
```

The `hiq` command switches the interface mode from MATLAB to HiQ. Notice that the prompt changes (from » to >) when you switched modes. The remaining commands in this example are native HiQ commands.

```
> graph = createGraph(r)
> view graph
```

The result is a HiQ graph object that contains a dynamic representation of the random surface. For comparison, change the color map for the plot to color spectrum using the Graph property page. You can access the property page by right clicking on the graph and selecting **Properties**. To rotate the HiQ graph, click on the graph and drag.

Look at the data used to generate the graphs. In MATLAB mode, you type in the name of the variable, `r`, and press <Enter>. If you want to see only a portion of `r`, you index the variable for the range of interest. Unfortunately, this creates a new temporary object (in `ans`) and takes up MATLAB workspace. In HiQ mode, however, you have two options:

- Use the command line view, which provides results similar to MATLAB.

  ```
  > r
  ```

- Create a view of the object, which creates a window `r` that looks similar to an Excel spreadsheet. You can scroll through the elements of the matrix `r` using the scroll bars at the right and bottom. To edit individual values, highlight the cell and input a new value. For example, use this matrix view to change the corner elements of the matrix to 5.

  ```
  > view r
  ```

After editing the matrix, send the changes back to MATLAB and regraph the matrix to view the edits.

```
> matlab
» put r
» surf(r)
```

Notice that the four corners of the surface are spiked because the larger data values were added to the matrix.

Accessing and Using HiQ Mode

To change the HiQ Command Window from MATLAB mode to HiQ mode, type the following at the command prompt:

```
» hiq
```

Several commands, such as `cd`, `pwd`, `ls`, and `dir`, are identical in the HiQ Command Window and MATLAB. MATLAB makes no distinction in M-Files between command line commands and built-in functions. In HiQ, commands are not built-in functions. HiQ commands are valid only in the HiQ Command Window.

Use HiQ commands as you would operating system commands. For example, `delete` *objectName* frees the resources used by the object with the given name.

By treating commands differently from built-in functions, you can create user-defined functions in your scripts using the same names as the commands without limiting the use of either one in the Command Window. If the command name is the same as an object name, command overwriting occurs as it does in MATLAB. To continue using the command in this situation, precede the command with the pound sign (#). The first two lines treat `cd` as a scalar, while the last executes the `cd` command.

```
> cd = 5;
> y = 3 * cd;
> #cd
```

Many commands in the Command Window control other features of HiQ, such as the HiQ Notebook. The HiQ Notebook, the primary interface element in HiQ, provides a framework for analyzing, visualizing, and documenting solutions. You can think of the HiQ Notebook as an extension of the project notebook kept by an engineer or scientist with sections, pages, drawings and algorithms.

In HiQ, you can have multiple notebooks open at one time solving multiple problems. Each notebook has its own collection of objects and results. In this type of environment, the Command Window must attach to a particular notebook to allow the command line interface to alter the objects in that notebook. You must detach the Command Window to perform work independent of a notebook. For more information about attaching and detaching, see Chapter 2, *HiQ Command Window*, in the *HiQ Reference Manual*.

Frequently Asked Questions

The following questions address issues you might encounter while in HiQ mode. For a complete description of the HiQ Command Window and its commands, see Chapter 2, *HiQ Command Window*, in the *HiQ Reference Manual*.

How can I stop variables from displaying their contents when I execute a statement?

HiQ offers two display modes: terse and verbose. The Command Window is in verbose mode by default. If you prefer the MATLAB approach, you can set the **Trailing semicolon implies terse** mode option for the Command Window. Right click in the Command Window, select **Properties**, and set this property.

Where can I get help on a built-in function?

As in MATLAB mode, type `help` followed by the function name, or place the cursor in the text of the function name and press the <F1> key.

How do I clear a variable from my workspace?

To remove an object created by the Command Window, you can delete it from the Objects section in the HiQ Explorer or use the `delete` command in the Command Window.

How can I display the variables that were affected by the last command line execution?

The `whatChanged` command provides a list of all HiQ objects changed by the execution of the last line in the Command Window.

How can I prevent the Command Window from creating variables in my current workspace?

If you use the `detach` command, all HiQ objects created in the Command Window from that point forward are not associated with the active notebook.

How do I save variables created in the Command Window?

If the HiQ objects were created while the Command Window was attached to an active notebook, save the notebook, and all objects are saved with it.

How do I change the precision at which variables are displayed?

Right click on the Command Window and select **Properties**. In the property page, click on the Numbers tab. In the entry for **Decimal Places**, verify that the **Auto** option is not checked and set the precision required. If you prefer trailing zeros, uncheck the **Remove Trailing Zeros** option.

Where can I find a list of available commands for the Command Window?

Refer to Chapter 2, *HiQ Command Window*, in the *HiQ Reference Manual* for a complete list and description of HiQ Command Window commands for both HiQ and MATLAB modes.

Where can I find a list of available built-in functions for use in the Command Window?

Type `help` to evoke the HiQ online help. Select *Built-In Functions* on the Contents tab to display a list of all available HiQ functions listed by name or category.

How do I view a variable in its own window?

Use the `view` command followed by the name of the HiQ object.

Where can I find a list of variables accessible in my workspace?

Look at the Objects section in the HiQ Explorer, or type the `objects` command to get a list of available HiQ objects.

How can I tell if I am using the correct name of a built-in function?

By default, syntax highlighting is on. Syntax highlighting displays built-in function names in bright blue letters. If the cursor is in the text of the

function name, you can press <F1> to open the online help topic for that function.

How can I keep a log of results generated by the Command Window?

To save the entire contents of the Command Window, right click on the window, select **Save As**, and choose the name of the file in which you want the contents to be stored. To log specific messages to the Log Window, use the logMessage() function. To view this log, click on the Log tab in the lower left corner of the Command Window.

How do I reuse previous command line statements?

Use the up and down arrow keys to access the history of commands. If you type the initial part of the command, the arrow keys cycle through only the commands in the history that start with what you typed.

Comparing HiQ and MATLAB Commands

Many commands in MATLAB mode are defined in HiQ mode as well. Other commands are handled more efficiently or flexibly by other HiQ user interfaces. Table 9-1 contains an alphabetical collection of general commands in MATLAB mode and the best way to achieve the same results in HiQ.

Table 9-1. HiQ and MATLAB Command Comparisons

MATLAB Command	HiQ Equivalent
cd	Type cd in the Command Window.
clear *objectName*	Type delete *objectName* in the Command Window, where *objectName* is the name of the object you want to delete.
diary	Right click on the Command Window and select **Save As** to save the contents of the current Command Window. Type clear to clear the Command Window contents before beginning the session you want to record.
dir	Type dir or ls in the Command Window.
echo	Use the built-in function logMessage to record the supplied text in the Log Window.

MATLAB Command	**HiQ Equivalent**
edit	Edit the contents of a HiQ-Script through a view of the Script object. Think of this view as a built-in editor provided by HiQ specifically designed for editing HiQ-Script. Use one of the following methods to create a view of a HiQ-Script. • Type view *scriptName* on the command line, where *scriptName* is the name of an existing HiQ-Script you want to edit. • In the HiQ Explorer, find the name of the script in the tree structure under the Objects branch. You can double click on it or press \<Shift-Enter\> to create a view of the script in its own window. • Type place *scriptName* on the command line, where *scriptName* is the name of the script you want to edit. A view of the script is placed on the current notebook page.
format	Use the HiQ property pages. Right click on the Command Window, select **Properties**, and click on the Numbers tab to access the formatting properties for displaying numbers in the Command Window.
helpwin	Press the \<F1\> key anywhere in HiQ to bring up the HiQ online help. From there, you have access to all help resources. You also can type help from the HiQ Command Window to activate the help system.
helpdesk	Select **Help»Visit HiQ Web Page** to retrieve online product and technical information.
load	In HiQ, all objects are stored within the Notebook. When you open the Notebook, all objects associated with that Notebook are automatically opened. You can open a Notebook with the **File»Open** command.
more	Because views of HiQ objects are scrollable, creating a view of a particular object allows you to inspect any portion of the object dynamically. To specify where you want object views to appear, right click on the Command Window and select **Properties**. From the property pages, you can control which objects are displayed in the Command Window, which are viewed in their own windows, and the maximum size of a matrix or vector viewable in the Command Window. Matrices and vectors larger than the specified size are displayed in individual windows.
quit	Type quit in the Command Window or select **Exit** from the **File** menu.

Table 9-1. HiQ and MATLAB Command Comparisons (Continued)

MATLAB Command	HiQ Equivalent
save	In HiQ, all objects are stored within the Notebook. Saving a Notebook saves all of the objects associated with it. To save a Notebook, use **Save** or **Save As** from the **File** menu.
type	List the contents of a HiQ-Script through a view of the Script object. Think of this view as a built-in editor provided by HiQ specifically designed for inspecting and editing HiQ-Script. Use one of the following methods to create a view of a HiQ-Script. • Type view *scriptName* on the command line, where *scriptName* is the name of an existing HiQ-Script you want to view. • In the HiQ Explorer, find the name of the script in the tree structure under the Objects branch. You can double click on it or press <Shift-Enter> to create a view of the script in its own window. • Type place *scriptName* on the command line where *scriptName* is the name of the script you want to view. A view of the script is placed on the current Notebook page.
ver	Select **About HiQ** from the **Help** menu to get version and copyright information.
what	From the **File** menu, select **Open** to display the current directory storing HiQ notebooks. By default, the file filter is set to the Notebook extension .HiQ so that only HiQ Notebooks show in the directory listing.
whatsnew	Select **What's New** from the **Help** menu to read about new features in this version of HiQ.
who	Type objects in the Command Window or look at the Objects section in the HiQ Explorer.
whos *objectName*	Type whatis *objectName* in the Command Window or look at the Objects section in the HiQ Explorer.
web	Select **Visit HiQ Web Page** from the **Help** menu to visit the HiQ Web Page.

If you do not find a command in Table 9-1, it might be because that command is unique to the MATLAB environment. For example, path commands such as rmpath and addpath do not have meaning in the HiQ environment because a HiQ Notebook contains all of the objects and scripts associated with it.

Automated M-File to HiQ-Script Translator

The M-File to HiQ-Script translator is a tool you can use to convert your M-File programs to HiQ. Its primary goal is to automate the change in language syntax, including loops (such as while and for), conditional execution (such as if and switch), and structure subranging. The translator can map functions if the usage of the function is unambiguous. For example, some MATLAB functions map directly to built-in HiQ functions allowing the translator to replace the MATLAB function name with the HiQ function name. Many MATLAB functions can be emulated by a user-defined function in HiQ. In these cases, the MATLAB functions are prefixed with M__. When a script executes, the appropriate HiQ functions from the translation library execute to produce equivalent results.

The translation maintains the readability of the script in both function calls and common language syntax. Some language syntax used in M-Files does not match any syntax in HiQ-Script, in which case HiQ functions are used in place of the language syntax to produce the proper results.

Using the Translator

The Import Wizard is the standard interface in HiQ for importing any kind of data, including M-Files. Use the following procedure to import M-Files with the HiQ Import Wizard.

1.   Access the Import Wizard by selecting **Import Wizard** from the **Notebook** menu or click on the Import Wizard icon in the toolbar.

2.   In the **Import From** section of the Import Wizard, select the **File** option and click on **Browse** to select the M-File you want to translate.

     After selecting the file, notice that the contents of the beginning of the M-File appear in the Preview window.

3.   If the Preview window is displaying the M-File you want to translate, click the **Next** button at the bottom of dialog box to continue.

     If the file has the default MATLAB extension, `.m`, the Import Wizard automatically detects it as an M-File, and the **Predefined** option in the Format section selects the description **M-File to HiQ-Script**. In the upper right, notice the name of the script object that will hold the results of the translation. To see the results of the translation, look in the Preview window.

4.   Click the **Next** button again to view the Translation Output Options dialog, as shown in Figure 9-1.
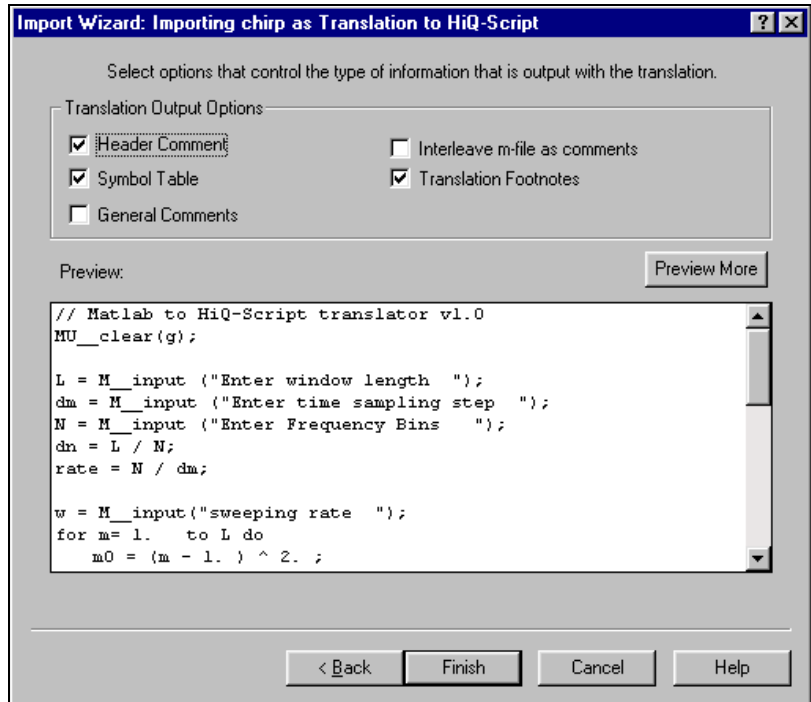
Figure 9-1. Import Wizard—Translator Output Options

5. Select the translation options that you want. The following list explains the options in more detail. Notice that the Preview window is automatically updated to reflect the new options.

- **Header Comments**—Inserts a header comment at the beginning of the translated file.

- **Symbol Table**—Includes a list of all symbols used in the translation. Use this option to verify the type descriptions of the translated objects and functions.

- **General Comments**—Includes explanations and general comments about the details of the translation.

- **Interleave M-File as Comments**—Interleaves each line of the original M-File with the corresponding translated line in HiQ-Script. Use this option to compare M-File script code with the automatically generated HiQ-Script code.

- **Translation Footnotes**—Appends additional translation footnotes to the end of the translation file. Use this option to obtain additional details about any limitations of the current translation.

6.   Click on **Finish** to complete the translation.

You also can translate M-Files through the view of a Script object. Although this approach has fewer options than the first, it provides a faster and more automated way to translate an M-File. The results of both procedures are the same.

1.   To translate a new M-File, place a new view of a new script object on the notebook page.

2.   Right click on the view and select **File»Import M-File**.

3.   Select the M-File you want to import. After you select the file, the translation begins and the resulting translation is stored in the current script object.

Working with the Results

If you translated your M-File using the Import Wizard, a view of that object is automatically placed on the Notebook page.

If you translated your M-File into an existing HiQ-Script view, the contents of the HiQ-Script are replaced by the translated output.

Table 9-2 compares how common scripting errors are reported in MATLAB and HiQ.

Table 9-2. Common Error Messages in MATLAB and HiQ

M-File Error Message	HiQ-Script Error Message
Assignment using a variable that does not exist: y = x;	
`??? Undefined function or variable 'x'.`	`The object "x" hasn't yet been assigned a value.`
Calling a function that is not defined: y = x(5);	
`Warning: Reference to uninitialized variable 'x'.`	`"x" is not a function or polynomial.`
Performing a language operation with incompatible object types:	
`??? Function '<op>' not defined for variables of class '<variableType>'.`	`Invalid operand types.`

To successfully execute a script, all functions in the script must be defined and compiled in the notebook. If you are translating an M-File that calls functions in other M-Files, you need to translate those files to HiQ-Script and compile them to make them available to all scripts in the current notebook.

HiQ-Script for MATLAB Users

Working in a new scripting language involves learning new syntax as well as new functionality. Table 9-3, *HiQ-Script for MATLAB Users*, compares M-Files and HiQ-Script to help you learn the HiQ syntax and functionality in the following categories:

- Script Comments

- Initializing Matrices

- Subscripting Matrices

- Expressions (for example, x = a + b/2; v = u.*w; E = B-C;)

- Operators (for example, .^, -, .\)

- Flow Control (for example, if, for, and while)

- Function Definitions

- Function Calls

Table 9-3. HiQ-Script for MATLAB Users

M-File	HiQ-Script
Script Comments	
% A comment in an M-file	// A comment in HiQ-Script
Initializing Matrices	

`M1 = [1];`
`M2 = [1 2 3 4 5];`
`M3 = [1; 2; 3; 4; 5];`
`M4 = [1 2 3; 4 5 6; 7 8 9];`
	`M1 = {1};`
	`M2 = {1, 2, 3, 4, 5};`
	`M3 = {1; 2; 3; 4; 5};`
	`M4 = {1, 2, 3; 4, 5, 6; 7, 8, 9};`

Table 9-3. HiQ-Script for MATLAB Users (Continued)

M-File	HiQ-Script
Subscripting Matrices	
`x = A(1,1);`	`x = A[1,1];`
`v = A(1,:);`	`v = A[1];`
`u = A(:,1);`	`u = A[:,1];`
`y = A(5);`	`y = (toVector(A'))[5];`
`p = A(1:5,4);`	`p = A[1:5,4];`
`q = A(2,3:7);`	`q = A[2,3:7];`
`M = A(3:5,4:9);`	`M = A[3:5,4:9]`
Expressions	
`r = 1+sqrt(5)/2;`	`r = 1+sqrt(5)/2;`
`s = abs(5-7i);`	`s = abs(5-7*<i>);`
`z = (4+2i)*(3-8i);`	`z = (4,2)*(3,-8);`
`x = exp(log(1e10*r));`	`x = exp(ln(1e10*r));`
`y = log(exp(1e10*s));`	`y = ln(exp(1e10*s));`
Operators	
`s = 5;`	`s = 5;`
`v = 1:.5:5;`	`v = seq(1, 5, .5);`
`M = [v; v; v; v];`	`M = {v; v; v; v};`
`r = s+s;`	`r = s+s;`
`t = s.*r;`	`t = s.*r;`
`u = v.*v;`	`u = v.*v;`
`A = M./M;`	`A = M./M;`
`B = M.\A;`	`B = M.\A;`
`C = A.^B;`	`C = A.^B;`
`D = C.';`	`D = trans(C);`
`E = D';`	`E = D';`
`w = u - v;`	`w = u - v;`
`n = v';`	`n = v';`
`p = [n, n .^2, 2 .^n];`	`p = {n, n .^2, 2 .^n};`

Table 9-3. HiQ-Script for MATLAB Users (Continued)

M-File	HiQ-Script
Flow Control	

<table>
<tr>
<td>

```
if errCode < 0
   disp('Invalid error code.');
elseif errCode == 0
   disp('No errors occurred.');
else
   warning('An error has
    occurred.');
end
```

</td>
<td>

```
if errCode < 0 then
   message("Invalid error code.");
else if errCode == 0 then
   message("No errors occurred.");
else
   warning("An error has
    occurred.");
end if;
```

</td>
</tr>
<tr>
<td>

```
switch errCode
   case 0
      disp('No errors reported.');
   case 1
      error('Error 1 occurred.');
   case 2
      error('Error 2 occurred.');
   otherwise
     error('Unhandled error code.');
end
```

</td>
<td>

```
select errCode from
   case 0:
      error("No errors reported.");
   case 1:
      error("Error 1 occurred.");
   case 2:
      error("Error 2 occurred.");
   default:
     error("Unhandled error code.");
end select;
```

</td>
</tr>
<tr>
<td>

```
i = 1;
absSum = 0;
while i <= size(A, 1)
  j = 1;
  while j <= size(A, 2)
   absSum = absSum + abs(A(i, j));
   j = j + 1;
  end
  i = i + 1;
  if absSum == inf
   break;
  end
end
```

</td>
<td>

```
i = 1;
absSum = 0;
while i <= A.rows do
  j = 1;
  while j <= A.columns do
   absSum = absSum + abs(A[i, j]);
   j = j + 1;
  end while;
  i = i + 1;
  if absSum == <inf> then
   exit while;
  end if;
end while;
```

</td>
</tr>
</table>

Table 9-3. HiQ-Script for MATLAB Users (Continued)

M-File	HiQ-Script
| ```
continue = 1;
absSum = 0;
for i = 1:size(A, 1)
  for j = 1:size(A, 2)
    absSum = absSum + abs(A(i, j));
    if absSum == inf
      continue = 0;
      break;
    end
  end
  if ~continue
    break;
  end
end
``` | ```
// multiple exits avoid 'continue'
absSum = 0;
for i = 1 to A.rows do
  for j = 1 to A.columns do
    absSum = absSum + abs(A(i, j));
    if absSum == <inf> then
      // multiple-exit support
      exit 2 fors;
    end if;
  end for;
end for;
``` |
| **Function Definitions** | |
| ```
function m = slope(P1, P2)

% SLOPE Slope between two points.
% SLOPE(P1) computes the slope
% between two points stored in
% the columns of P.
% SLOPE(P1, P2) computes the
% slope between two points stored
% in column vectors P1 and P2.
if nargin == 1
  m = P1(2,2)-P1(2,1);
  m = m/(P1(1,2)-P1(1,1));
else
  m = P2(2,1)-P1(2,1)
  m = m/(P2(1,1)-P1(1,1));
end
``` | ```
function slope(P1, P2)

// Compute the slope of two points
// where the points are given in
// columns of P1 or in two vectors,
// P1 and P2.

  if __ins == 1 then
    m = P1[2,2]-P1[2,1];
    m = m/(P1[1,2]-P1[1,1]);
  else
    m = P2[2]-P1[2];
    m = m/(P2[1]-P1[1]);
  end if;

    return m;
end function;
``` |
| ```
function [xmin, xmax] = minmax(x)

% MINMAX Compute the minimum and
% and maximum values in x.
xmin = min(min(x));
if nargout == 2
   xmax = max(max(x));
end
``` | ```
function minmax(x)

// Compute the minimum and
// maximum values in x.
  xmin = min(x);
  if __outs == 2 then
    xmax = max(x);
  end if;
  return xmin, xmax;
end function;
``` |

Table 9-3. HiQ-Script for MATLAB Users (Continued)

M-File	HiQ-Script
Function Calls	
`matrix = abs(matrix);`	`matrix = abs(matrix);`
`matrix = chol(matrix);`	`matrix = choleskyD(matrix);`
`scalar = cond(matrix);`	`scalar = cond(matrix);`
`scalar = cond(matrix, inf);`	`scalar = cond(matrix, <Li>);`
`vector = eig(matrix);`	`vector = eigen(matrix);`
`vector = eig(matrix, matrix);`	`vector = eigen(matrix, matrix);`
`[V,D] = eig(matrix);`	`[d,V] = eigen(matrix); D = diag(d);`
`[v,d] = eigs(matrix, 1);`	`[d,v] = eigenDom(matrix);`
`[v,d] = eigs(matrix, 1, scalar);`	`[d,v] = eigenSel(matrix, scalar);`
`matrix = eye(scalar);`	`matrix = ident(integer);`
`scalar = fmin('cos', 3, 4);`	`[,scalar] = optimize(cos, toVector(3.1));`
`scalar = fmins('fct', vector);`	`[,scalar] = optimize(fct, vector);`
`scalar = fzero('sin', [a b]);`	`scalar = root(sin, a, b);`
`scalar = fzero('sin', 3);`	`scalar = root(sin, toVector(3));`
`matrix = inv(matrix);`	`matrix = inv(matrix);`
`scalar = max(max(matrix));`	`scalar = max(matrix);`
`scalar = mean(mean(matrix));`	`scalar = mean(toVector(matrix));`
`scalar = norm(matrix);`	`scalar = norm(matrix);`
`scalar = norm(matrix, 'fro');`	`scalar = norm(matrix, <frob>);`
`scalar = norm(matrix, 1);`	`scalar = norm(matrix, <L1>);`
`scalar = plot(vector,vector);`	`[graph, integer] = createGraph(vector, vector);`
`R = qr(matrix);`	`R = QRD(matrix);`
`[Q,R] = qr(matrix);`	`[R,Q] = QRD(matrix);`
`vector = rand(6, 1);`	`vector = createVector(6, <random>);`

Table 9-3. HiQ-Script for MATLAB Users (Continued)

M-File	HiQ-Script
`matrix = rand(5, 4);`	`matrix = createMatrix(5, 4, <random>);`
`matrix = randn(3);`	`matrix = createMatrix(3, 3, <random>, <normal>);`
`scalar = rank(matrix);`	`integer = rank(matrix);`
`scalar = rank(matrix, 1e-4);`	`integer = rank(matrix, 1e-4);`
`vector = roots([1 2 3 4]);`	`vector = roots({poly: "x^3+2x^2+3x+4"});`
`[vector, ivector] = sort(vector);`	`[vector, integerVector] = sort(vector);`
`matrix = sqrt(matrix);`	`matrix = sqrt(matrix);`
`scalar = surf(matrix);`	`graph = createGraph(matrix);`
`vector = svd(matrix);`	`vector = sv(matrix);`
`[U,S,V] = svd(matrix);`	`[U,S,V] = svd(matrix);`

HiQ Online Help for MATLAB Users

HiQ offers online help specifically designed to help MATLAB users work in HiQ. You can access the online help in either of the following ways:

- Press <F1> and select the *MATLAB* topic on the Index tab.
- Type `help matlab` in the Command Window.

A

Customer Communication

For your convenience, this appendix contains forms to help you gather the information necessary to help us solve your technical problems and a form you can use to comment on the product documentation. When you contact us, we need the information on the Technical Support Form and the configuration form, if your manual contains one, about your system configuration to answer your questions as quickly as possible.

National Instruments has technical assistance through electronic, fax, and telephone systems to quickly provide the information you need. Our electronic services include a bulletin board service, an FTP site, a fax-on-demand system, and e-mail support. If you have a hardware or software problem, first try the electronic support systems. If the information available on these systems does not answer your questions, we offer fax and telephone support through our technical support centers, which are staffed by applications engineers.

Electronic Services

Bulletin Board Support

National Instruments has BBS and FTP sites dedicated for 24-hour support with a collection of files and documents to answer most common customer questions. From these sites, you can also download the latest instrument drivers, updates, and example programs. For recorded instructions on how to use the bulletin board and FTP services and for BBS automated information, call 512 795 6990. You can access these services at:

United States: 512 794 5422
Up to 14,400 baud, 8 data bits, 1 stop bit, no parity

United Kingdom: 01635 551422
Up to 9,600 baud, 8 data bits, 1 stop bit, no parity

France: 01 48 65 15 59
Up to 9,600 baud, 8 data bits, 1 stop bit, no parity

FTP Support

To access our FTP site, log on to our Internet host, `ftp.natinst.com`, as `anonymous` and use your Internet address, such as `joesmith@anywhere.com`, as your password. The support files and documents are located in the `/support` directories.

Fax-on-Demand Support

Fax-on-Demand is a 24-hour information retrieval system containing a library of documents on a wide range of technical information. You can access Fax-on-Demand from a touch-tone telephone at 512 418 1111.

E-Mail Support (Currently USA Only)

You can submit technical support questions to the applications engineering team through e-mail at the Internet address listed below. Remember to include your name, address, and phone number so we can contact you with solutions and suggestions.

`support@natinst.com`

Telephone and Fax Support

National Instruments has branch offices all over the world. Use the list below to find the technical support number for your country. If there is no National Instruments office in your country, contact the source from which you purchased your software to obtain support.

Country	Telephone	Fax
Australia	03 9879 5166	03 9879 6277
Austria	0662 45 79 90 0	0662 45 79 90 19
Belgium	02 757 00 20	02 757 03 11
Brazil	011 288 3336	011 288 8528
Canada (Ontario)	905 785 0085	905 785 0086
Canada (Québec)	514 694 8521	514 694 4399
Denmark	45 76 26 00	45 76 26 02
Finland	09 725 725 11	09 725 725 55
France	01 48 14 24 24	01 48 14 24 14
Germany	089 741 31 30	089 714 60 35
Hong Kong	2645 3186	2686 8505
Israel	03 6120092	03 6120095
Italy	02 413091	02 41309215
Japan	03 5472 2970	03 5472 2977
Korea	02 596 7456	02 596 7455
Mexico	5 520 2635	5 520 3282
Netherlands	0348 433466	0348 430673
Norway	32 84 84 00	32 84 86 00
Singapore	2265886	2265887
Spain	91 640 0085	91 640 0533
Sweden	08 730 49 70	08 730 43 70
Switzerland	056 200 51 51	056 200 51 55
Taiwan	02 377 1200	02 737 4644
United Kingdom	01635 523545	01635 523154
United States	512 795 8248	512 794 5678

Technical Support Form

Photocopy this form and update it each time you make changes to your software or hardware, and use the completed copy of this form as a reference for your current configuration. Completing this form accurately before contacting National Instruments for technical support helps our applications engineers answer your questions more efficiently.

If you are using any National Instruments hardware or software products related to this problem, include the configuration forms from their user manuals. Include additional pages if necessary.

Name _____

Company _____

Address _____

_____

Fax (___) _____Phone (___) _____

Computer brand_____ Model _____Processor_____

Operating system (include version number) _____

Clock speed _____MHz RAM _____MB   Display adapter _____

Mouse ___yes ___no   Other adapters installed _____

Hard disk capacity _____MB Brand_____

Instruments used _____

_____

National Instruments hardware product model _____ Revision _____

Configuration _____

National Instruments software product _____ Version _____

Configuration _____

The problem is: _____

_____

_____

_____

_____

List any error messages: _____

_____

_____

The following steps reproduce the problem: _____

_____

_____

_____

_____

HiQ Hardware and Software Configuration Form

Record the settings and revisions of your hardware and software on the line to the right of each item. Complete a new copy of this form each time you revise your software or hardware configuration, and use this form as a reference for your current configuration. Completing this form accurately before contacting National Instruments for technical support helps our applications engineers answer your questions more efficiently.

National Instruments Products

Hardware revision _____

Interrupt level of hardware _____

DMA channels of hardware _____

Base I/O address of hardware _____

Programming choice _____

National Instruments software _____

Other boards in system _____

Base I/O address of other boards _____

DMA channels of other boards _____

Interrupt level of other boards _____

Other Products

Computer make and model _____

Microprocessor _____

Clock frequency or speed _____

Type of video board installed _____

Operating system version _____

Operating system mode _____

Programming language _____

Programming language version _____

Other boards in system _____

Base I/O address of other boards _____

DMA channels of other boards _____

Interrupt level of other boards _____

Documentation Comment Form

National Instruments encourages you to comment on the documentation supplied with our products. This information helps us provide quality products to meet your needs.

Title: *Getting Results with HiQ™*

Edition Date: April 1998

Part Number: 321886A-01

Please comment on the completeness, clarity, and organization of the manual.

_____

_____

_____

_____

_____

_____

_____

If you find errors in the manual, please record the page numbers and describe the errors.

_____

_____

_____

_____

_____

_____

_____

Thank you for your help.

Name   _____

Title   _____

Company _____

Address _____

_____

E-Mail Address _____

Phone (___) _____ Fax (___) _____

Mail to: Technical Publications   **Fax to:**   Technical Publications
            National Instruments Corporation   National Instruments Corporation
            6504 Bridge Point Parkway   512 794 5678
            Austin, Texas 78730-5039

Glossary

Prefix	Meanings	Value
p-	pico	10^{-12}
n-	nano-	10^{-9}
μ-	micro-	10^{-6}
m-	milli-	10^{-3}
k-	kilo-	10^{3}
M-	mega-	10^{6}
G-	giga-	10^{9}
t-	tera-	10^{12}

A

ActiveX (Microsoft ActiveX)	A programming system and user interface that lets you work with interactive objects. Formerly called OLE.
ActiveX control	A standard software tool that adds additional functionality to any compatible ActiveX container.
ActiveX Control object	An object in HiQ that represents an inserted ActiveX control.
ActiveX Interface object	An object in HiQ that represents an interface to an ActiveX automation server application. See the createInterface function in Chapter 7, *Function Reference*, of the *HiQ Reference Manual*.
ActiveX embedded object	An object placed into a container and unconnected to any other object or application. *See also* embedded.
ActiveX linked object	An object placed into a container and connected to another object or application in the same container or in a separate container. *See also* linked.
ActiveX object	An object in HiQ that represents an inserted ActiveX object.
argument	*See* parameter.

assignment	A script statement that sets a value to a variable.
attribute	*See* property.

B

built-in function	One of many programming utilities in HiQ-Script that perform analysis, graphical, or utility operations.

C

caret	*See* insertion point.
cell	In a matrix or vector, the intersection of a row and column that contains a numerical value.
Color object	A HiQ object that sets the color attributes of graphs and plots.
Command Window	*See* HiQ Command Window.
comment	An explanatory line or portion of a line in HiQ-Script.
compiled script	A HiQ object containing HiQ-Script language that has been converted to code that a computer uses to execute the program.
ComponentWorks	A collection of 32-bit ActiveX controls designed for building virtual instrumentation systems.
constant	A predefined value in HiQ-Script.
cursor	The pointer or other image that displays on screen to show the location of your mouse, trackball, or other pointing device.

D

data type	*See* numeric type.
debug	To check and correct invalid code in a HiQ script in order to eliminate errors during the compilation or execution of the script.
declaration	A HiQ-Script statement that defines the scope of variables. Can be either project or local.

dialog box	A window containing a message, interactive options, and buttons.
dock	To attach a detachable window or toolbar to the HiQ window.
drag-and-drop	To move an object to a specific location, using the mouse to click on, drag, and release the object. Depending on the location of release, a specific action can occur.

E

embedded	Inserted into a container object and unconnected to any other object or application. Compare this term to linked. *See also* ActiveX embedded object.
error message	An information box that appears when HiQ cannot complete an action due to an internal error, compile error, run-time error, or user error.
Explorer	*See* HiQ Explorer.
expression	A mathematical operator and its operands.

F

Font object	A HiQ object type used to set the font attributes of graphs and plots.
function	A block of code that performs a specific task in HiQ-Script; can be a HiQ built-in function or a user-defined function.
Function Object	An object that represents the executable pseudocode of a complied Script object.
function call	Specific HiQ-Script syntax that calls a user or built-in function with given parameters.

G

grab handle	A site on a selected object that you click on and drag to move, size, or reshape the object.
Graph object	The HiQ object that contains a two-dimensional or three-dimensional collection of plots. *See* plot.

H

handle	*See* grab handle.
HiQ Command Window	An intuitive window where you type HiQ-Script to get immediate results.
HiQ Constant object	A HiQ object that represents a constant in HiQ-Script.
HiQ Explorer	An interactive window displaying the objects, sections, and pages of an open Notebook.
HiQ Log Window	A window to which you can post messages from HiQ-Script.
HiQ Object Browser	A browser window in which you can view all the interfaces for ActiveX servers, objects, and controls installed on your computer.
HiQ-Script	The intuitive HiQ programming language for mathematics.
HiQ Tools toolbar	Part of the HiQ user interface that contains icons for objects you can place in a Notebook. Depending on your preference, the toolbar can appear on any edge of the interface, or as a floating palette.

I

IEEE	Institute of Electrical and Electronic Engineers.
initialization	In HiQ-Script, a designated syntax which creates a particular type of object, for example, vector, matrix, color, and others.
insertion point	The location where text will be inserted (also referred to as the *caret*).

K

keyword	A reserved word in HiQ Script, such as `if`, `then`, or `while`, that is used for constructing specific types of programming statements.

L

LabVIEW	Laboratory Virtual Instrument Engineering Workbench. Program development application based on the programming language G used commonly for test and measurement applications.
LabWindows\CVI	An integrated ANSI C environment designed for engineers and scientists creating virtual instrumentation applications.
linked	Inserted into a container object and connected with another object or application. Compare this term to embedded. *See also* ActiveX linked object.
local	Describes the scope of an object. An object with local scope is not associated with the Notebook.
Log Window	*See* HiQ Log Window.
loop	A statement in HiQ-Script consisting of keywords and nested statements that performs a repetitive function. Also known as an iteration statement.

M

MATLAB	Third-party software product for programming and working with data.
Matrix object	A HiQ numeric object containing an array of elements with rows and columns. *See* vector.
message box	A secondary window that appears containing information about the status of a HiQ operation. *See* dialog box.

N

Notebook	The workspace in HiQ that stores, organizes, and displays all the components of an analysis and visualization problem.
Numeric object	A HiQ object type that is defined in terms of a numeric type (integer, real, or complex) and an object type (matrix, vector, polynomial, or scalar).
numeric type	One of three types of a numeric object (integer, real, or complex).

O

object	An entity in a HiQ Notebook that contains data of a specific type, for example, numeric, graphic, text, or HiQ-Script. Objects work together in a Notebook to generate and display solutions to analysis and visualization problems. Objects are always stored in a Notebook, but are not always visible on a Notebook page. *See* object view.
Object Browser	*See* HiQ Object Browser.
Object List	A window in HiQ that displays all the objects in a HiQ Notebook.
object view	A HiQ object that is visible on a Notebook page.
OLE (Microsoft OLE)	Object Linking and Embedding. A programming system and user interface that lets you work with interactive objects. *See* ActiveX.
operand	An object (or objects) modified by an operator.
operator	Code in HiQ-Script that is specific to basic mathematical operations and structure of HiQ-Script language. Often represented as a symbol, for example, "/" represents division.

P

parameter	An independent variable passed to a user-defined or built-in function call in a parameter list.
Plot object	A HiQ object type that graphically represents a 2D or 3D function or data set used in conjunction with a graph object. *See* graph.
Polynomial object	A HiQ numeric object represented by an equation in the polynomial form $ax^n + bx^{n-1} + cx^{n-2} + ...$
pop-up menu	A context-sensitive menu that you can access by right-clicking with your mouse on an object or on the Notebook.
Problem Solver	A HiQ Notebook containing objects, including HiQ-Script, that allows you to interactively perform analysis of data and display results for a broad class of problems. For example, the expression evaluator problem solver can display the results of any expression.

project	Describes the scope of an object. An object with project scope is saved with the Notebook.
Properties dialog box	A window in HiQ with tabbed pages (property pages) where you can quickly set a wide variety of attributes for a given object.
property	Attributes of a HiQ object. Examples include the color of a plot, the size of a matrix, and the type of any object.
property page	A tabbed subsection of a property dialog box containing a collection of object attributes.

S

Scalar object	A HiQ numeric object represented as a number. *See* vector and matrix.
scope	In HiQ-Script, a HiQ-object declaration that specifies whether the object is available to the entire Notebook (project) or is temporarily available (local).
script	A block of code that performs a certain task. *See* HiQ script and compiled script.
Script object	A HiQ object type containing HiQ-Script and from which you compile and execute your program.
section tabs	An organizing tool in a HiQ Notebook that lets you label and quickly access different parts of your Notebook.
selection handle	A graphical control point of an object that provides direct manipulation support for operations of that object, such as moving, sizing, or scaling.
selection tool	The mouse cursor in HiQ shaped like a standard pointer arrow that lets you select, move, and manipulate objects on the Notebook.
shortcut key	A key or combination of keys that you press to invoke a command.
Standard toolbar	Site on the user interface of HiQ that contains basic utility tools, for example, **Save**, **Open**, **Cut**, **Paste**, and, **Print**. Depending on your preference, the toolbar can appear on any edge of the interface, or as a floating window.
statement	In HiQ-Script, a line of code consisting of various keywords, functions, and/or operators that performs a certain task.

status bar	A region, usually the bottom of a window, containing information about HiQ and any selected object.
symbol	The name for *objects* in HiQ for the Macintosh.

T

Text object	A HiQ object type where you can enter and edit text.
toolbar	Site on an application interface that contains various buttons and other controls. Depending on your preference, a toolbar can appear on any edge of the interface, or a floating window.
tooltip	A small, descriptive pop-up window that appears when you position the mouse cursor over a toolbar icon.
type	A classification of an object based on its characteristics, behavior, and attributes.

U

Untyped object	A HiQ object that has not been assigned a value.
user-defined function	A function that a user creates in HiQ-Script to perform customized analysis, graphical, or utility operations.

V

Vector object	A HiQ numeric object containing an array with one row or column. Compare with matrix.

Index

Numbers

2D graphs, Climate Notebook example,
5-1 to 5-10
    creating graphs, 5-1 to 5-2
    modifying graph and plot, 5-4 to 5-6
    plotting data object, 5-2 to 5-4
    working with multiple plots, 5-6 to 5-10
3D graphs, Seismic Notebook example,
5-10 to 5-14
    modifying graph and plot, 5-12 to 5-13
    rotating, zooming, and panning, 5-15

A

ActiveX controls
    Climate Notebook, 7-2 to 7-8
        adding ComponentWorks ActiveX
          control, 7-3 to 7-4
        adding Microsoft Word document,
          7-6 to 7-8
        programmatically accessing ActiveX
          control, 7-4 to 7-6
    Seismic Notebook, 7-8 to 7-12
        adding ComponentWorks ActiveX
          control, 7-9 to 7-10
        programmatically accessing ActiveX
          control, 7-10 to 7-12
ActiveX environment, 1-7 to 1-8
ActiveX technologies, 7-1 to 7-2
    automation client, 7-1 to 7-2
    automation server, 7-2
    controls container, 7-2
    document container, 7-1
    document server, 7-1
automating HiQ-Script analysis, 6-8 to 6-9

B

binary file, importing as numeric object, 4-9
bulletin board support, A-1

C

Climate Notebook
    adding ActiveX controls, 7-2 to 7-8
        ComponentWorks ActiveX control,
          7-3 to 7-4
        Microsoft Word document, 7-6 to 7-8
        programmatically accessing ActiveX
          control, 7-4 to 7-6
    analyzing data with HiQ-Script, 6-1 to 6-5
        looking for trends, 6-3 to 6-4
        maintaining the notebook, 6-5
        setting default view properties,
          6-2 to 6-3
    converting to Problem Solver, 8-1 to 8-4
    importing data, 4-1 to 4-4
    visualizing data in 2D graph, 5-1 to 5-10
        creating graphs, 5-1 to 5-2
        modifying graph and plot, 5-4 to 5-6
        plotting data object, 5-2 to 5-4
        working with multiple plots, 5-6 to 5-10
Command Window. *See* HiQ
  Command Window.
ComponentWorks ActiveX control
    adding to Climate Notebook, 7-3 to 7-4
    adding to Seismic Notebook, 7-9 to 7-10
customer communication, *xiv,* A-1 to A-2

D

data, importing. *See* importing data.
data fitting and interpolation functions, 1-4
data fitting Notebook, 2-1 to 2-4
    opening, 2-2
    running scripts, 2-3
    viewing object list, 2-4
data visualization. *See* visualizing data.
differential equations
    functions for, 1-6
    solving in HiQ Command Window,
      3-5 to 3-9
differentiation functions, 1-7
documentation
    conventions used in documentation,
      xiii-xiv
    organization of manual, *xii-xiii*
    related documentation, *xiv*
    where to start with the manual, *xi-xii*

E

electronic support services, A-1 to A-2
e-mail support, A-2
Export Wizard (note), 4-9
Expression Evaluator Notebook, 2-4 to 2-7
    entering data and running scripts,
      2-6 to 2-7
    illustration, 2-5

F

fax and telephone support numbers, A-2
Fax-on-Demand support, A-2
FTP support, A-1
functions, 1-4- to 1-7
    data fitting and interpolation, 1-4
    differential equations, 1-6
    differentiation, 1-7
    integration, 1-6
    linear algebra, 1-4

    nonlinear analysis, 1-6
    online help (figure), 1-5
    optimization, 1-6
    polynomials, 1-7
    statistics, 1-6

G

graphs
    Climate Notebook 2D graph example,
      5-1 to 5-10
      creating graphs, 5-1 to 5-2
      modifying graph and plot, 5-4 to 5-6
      plotting data object, 5-2 to 5-4
      working with multiple plots,
        5-6 to 5-10
    Seismic Notebook 3D graph example,
      5-10 to 5-14
      modifying graph and plot,
        5-12 to 5-13
      rotating, zooming, and panning, 5-15

H

help. *See* online help.
HiQ
    ActiveX environment, 1-7 to 1-8
    environment (figure), 1-3
    installing, 1-1 to 1-2
    launching, 1-2
    system requirements, 1-1
HiQ Command Window, 3-1 to 3-14
    > prompt, 3-2
    adjusting position in Notebooks, 3-1
    changing default name for objects
      (note), 3-3
    default position, 3-1
    getting help, 3-10
    hiding results using semicolon (note), 3-2
    highlighting of built-in functions
      (note), 3-3