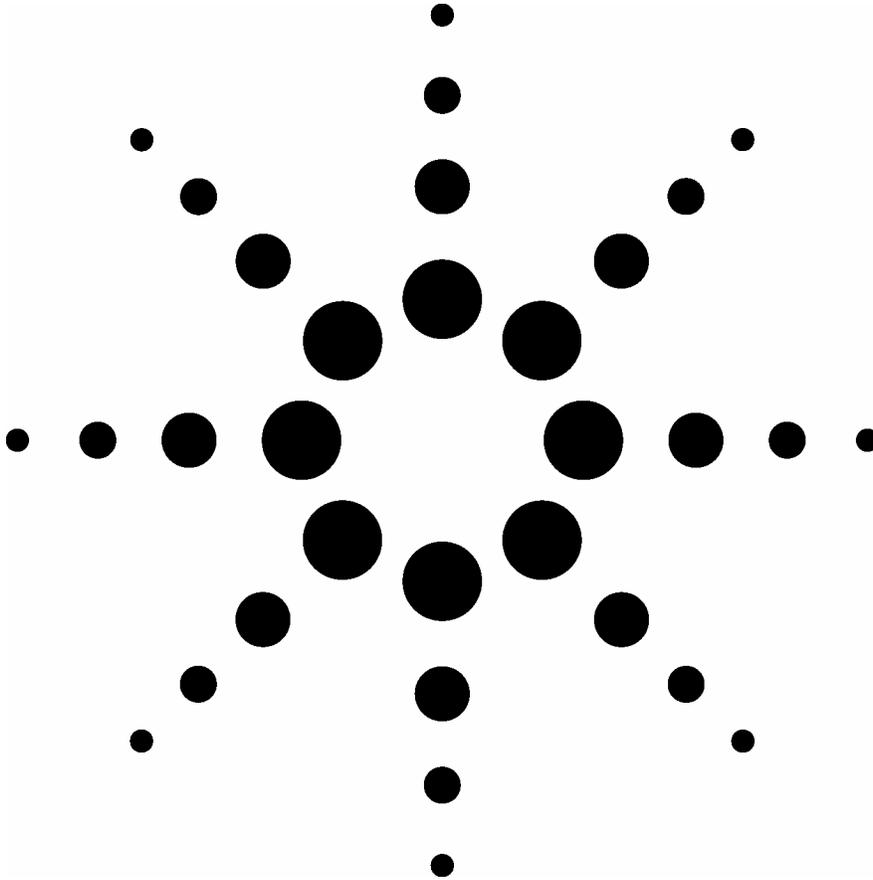


# Accessing Microsoft® Excel from VEE

White Paper



**Agilent Technologies**

## **Overview**

Integrated access to Microsoft® Excel has been incorporated into Agilent VEE 7.5. Over 40 new menu selections are available to instantiate VEE objects that perform basic and sophisticated Excel operations. These new objects insulate developers from the messy (Active X) programming details of communicating with Excel. The exercises in this lab will show you just how easy it is to add Excel to your VEE repertoire.

## **Objective**

In this lab you will learn how to:

- Launch and close Excel from VEE
- Send data from VEE into Excel
- Retrieve data from Excel directly into VEE
- Plot multiple traces on an Excel chart
- Save an Excel spreadsheet to a file

## **Requirements**

- Agilent VEE Pro 7.5
- Microsoft Office 2000 or higher
- Basic understanding of Microsoft Excel

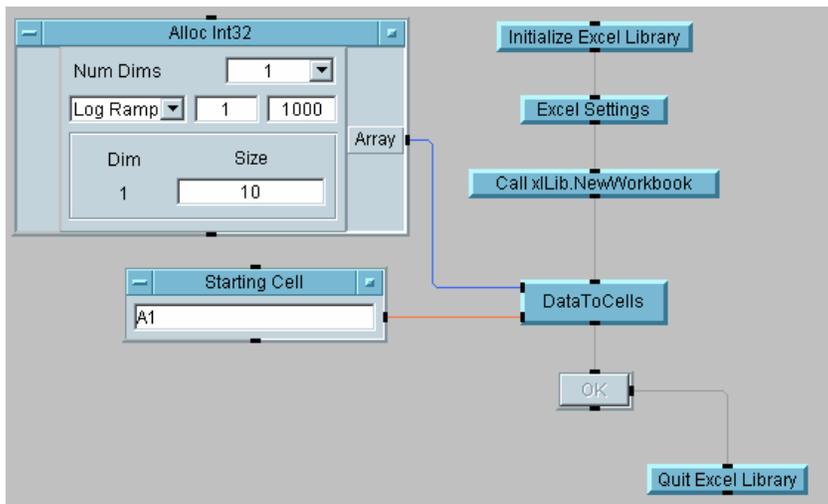
## Exercise 1

Write a VEE program that generates a small array and sends it to a column of cells in a sheet.

- **Step 1 (Generate the data)**
  - Ensure VEE has a clean slate ([File => New](#))
  - Load object that creates an array of integer values ([Data => Allocate Array => Int32](#))
  - Change the *LinRamp* value to *LogRamp*.
  - Change the final value from *10* to *1000*.
- **Step 2 (Initialize the Excel library)**
  - Load object that initialize the Excel Library ([Excel => Initialize Excel Library](#))
- **Step 3 (Configure settings such that Excel launches Maximized)**
  - Load settings object ([Excel => Settings](#))
  - Open the settings object ([double-click on the settings object icon](#))
  - In the record object, change the *WindowMode* field value to *MINIMIZED*
  - Close or minimize the settings object window
- **Step 4 (Create a new Excel Workbook)**
  - Load object that creates a new Excel workbook ([Excel => New Workbook](#))
- **Step 5 (Determine which cell in the current sheet will receive the first data value)**
  - Load object that creates text ([Data => Constant => Text](#))
  - Type a cell index (e.g. "A1" or "C7") into the text field of this text constant object
  - Retitle the constant to "Starting Cell" ([double-click on constant object's title bar and type in the text "Starting Cell"](#))
- **Step 6 (Send data to Excel)**
  - Load object that sends data from VEE into Excel starting at the named cell ([Excel => Send Data => Data To Cells](#))
- **Step 7 (Add a button for pausing the VEE program)**
  - Load OK button object ([Flow => Confirm \(OK\)](#))
- **Step 8 (Close Excel)**
  - Load object that closes Excel and unloads the Excel library ([Excel => Quit Excel Library](#))

- **Step 9 (Wire the objects)**

- Your wired program should look something like this:



- **Step 10 (Save & Run the program)**

- Save the program and name it **ExcelLab1.vee** (File => SaveAs...)
- Notice that an Excel icon is visible in the Windows Taskbar. Open it up and example the contents in the first book. The values (1, 2, 5, 10, 22, 46, 100, 215, 464, 1000) should be inserted beginning at the cell index you selected in your program.
- Minimize Excel and return to the VEE program.
- Click on the **OK** button in your program.
- Notice that Excel is no longer visible in the Windows Taskbar.

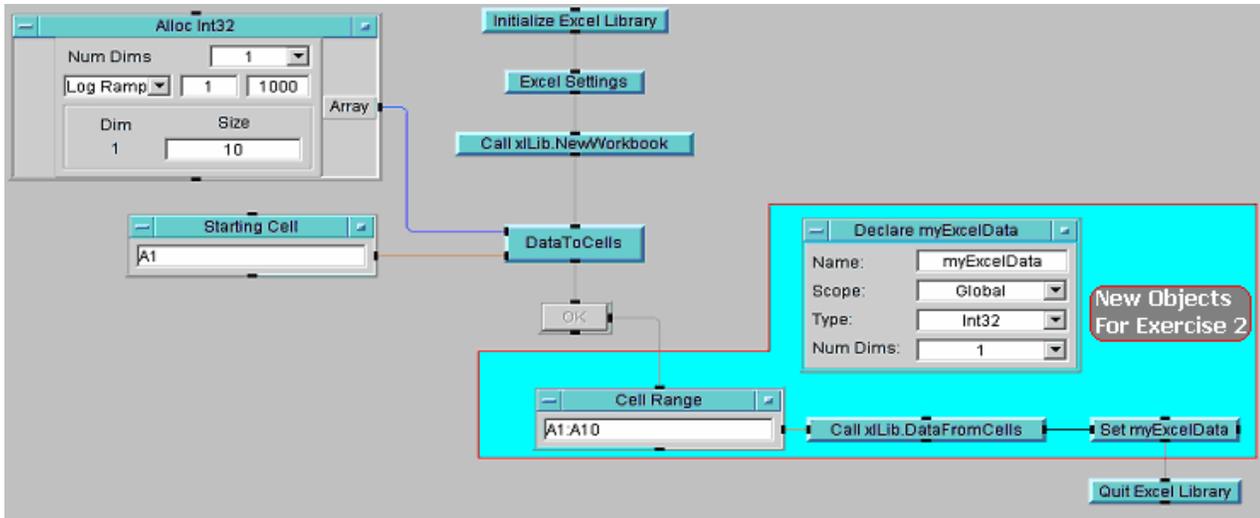
## Exercise 2

Extend the program to retrieve the data from Excel and assign a VEE variable to it.

- **Step 1 (Declare a VEE variable to hold the data)**
  - Load object that declares a VEE variable (**Data => Variable => Declare Variable**)
  - Change the name field value from *global1* to *myExcelData*.
  - Change the type field value to *Int32*.
  - Change the number of dimensions (Num Dims) from 0 to 1.
- **Step 2 (Specify the cell range)**
  - Load object that specifies a text string (**Data => Constant => Text**)
  - Type in the starting cell, followed by a colon, followed by the last cell. For example, if you selected “C1” to send your data, then type in the text “C1:C10”. If you aren’t sure, run your program again and look at which cells contain data.
  - Retitle the constant to “Cell Range” (**double-click on constant object’s title bar and type in the text “Cell Range”**)
- **Step 3 (Get data from Excel)**
  - Load object that gets data from Excel (**Excel => Get Data => Data From Cells**)
- **Step 4 (Assign the VEE variable to the values returned from Excel)**
  - Load formula object (**Device => Formula**)
  - Retitle the formula object to “Set myExcelData” (**double-click on the formula title bar and type in the text “Set myExcelData”**)
  - Rename the formula input from *A* to *CellData* (**double-click on the *A* terminal and type in the text “CellData”, then hit <Enter>**)
  - In the formula expression field, replace the text “2\*A+3” with “myExcelData = CellData[\* ,0]” (**NOTE: If your data were two dimensional, you could simply type in “myExcelData = CellData”. However, since this exercise is using a one-dimensional array, this expression selects all the elements (\*) of the first column (0) of a two-dimensional array.**)
  - Minimize the formula to conserve space (**click on the upper right corner of the formula object**)

- **Step 5 (Wire the objects)**

- The three objects which retrieve data from Excel should be inserted between the **OK** button and *Quit Excel Library* UserObject as follows:



- **Step 6 (Save & Run the program)**

- Save the program and name it **ExcelLab2.vee** (**File => SaveAs...**)
- Run the program, and click on the **OK** button, which will now retrieve the values that were just sent to Excel.
- Notice that almost instantly, Excel is no longer visible in the Windows Taskbar.

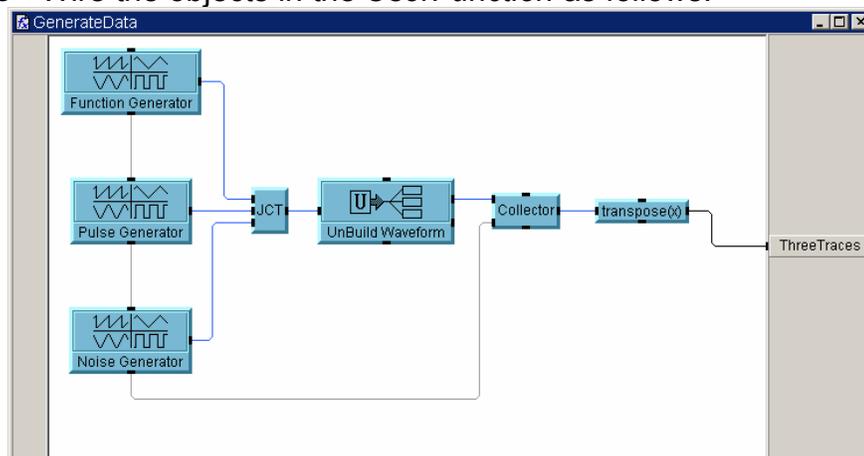
- **Step 7 (Verify the results)**

- Click on the *View Variables* toolbar button (it looks like a globe and is located directly beneath the *Excel* menu pick)
- Double-click on the variable *myExcelData*.
- Verify that the corresponding values are (1, 2, 5, 10, 22, 46, 100, 215, 464, 1000).

### Exercise 3

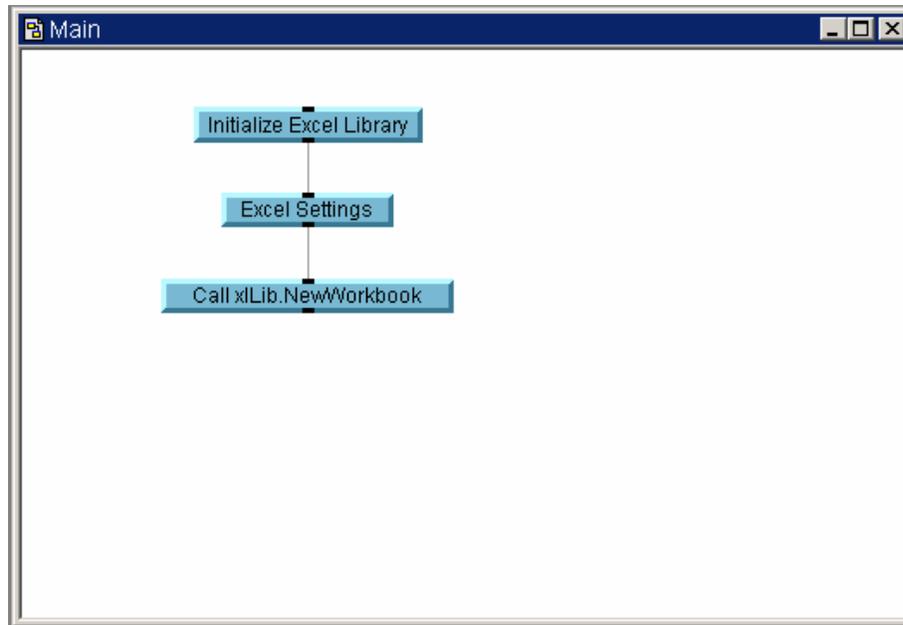
Plot multiple traces on an Excel chart.

- **Step 1 (Generate the data – a 256x3 two-dimensional array)**
  - Close any open excel workbooks
  - Ensure VEE has a clean slate (**File => New**)
  - Create a *UserFunction* which will end up containing the code which will generate the data (**Device => UserFunction**)
  - Set the title of the *UserFunction* to “GenerateData”
    1. Right-click on the UserFunction title bar and select **Properties**
    2. Set the *Title* property value to *GenerateData*
  - Load and minimize the three Virtual Source objects into the *UserFunction*:
    1. **Device => Virtual Source => Function Generator**
    2. **Device => Virtual Source => Pulse Generator**
    3. **Device => Virtual Source => Noise Generator**
  - Load a *Junction* object into the *UserFunction* (**Flow => Junction**)
  - Add an extra input to the *Junction* (**right-mouse-click on the icon, select Add Terminal => Data Input...**)
  - Extract the array data information by loading an *UnBuild Waveform* object (**Data => UnBuild Data => Waveform**)
  - Combine the 1-D arrays together into a 2-D array by loading a *Collector* object (**Data => Collector**)
  - Transpose the 3x256 array to a 256x3 array by calling the transpose function:
    1. **Click on the Function & Object Brower toolbar button – Fx**
    2. Under **Type**, select *Built-In Functions*
    3. Under **Category**, select *Matrix*
    4. Under **Functions**, select *transpose*
    5. **Click on Create Formula button**
  - Add an output to the *UserFunction* named **ThreeTraces**:
    1. **Move the cursor the UserFunction output area**
    2. **Hit Control-A** (short cut for add terminal)
    3. **Double-click on the new terminal**
    4. **Type in the text “ThreeTraces” and hit <Enter>**
  - Wire the objects in the *UserFunction* as follows:



- **Step 2 (Create an Excel Workbook)**

- Load excel objects into Main (**not** the *GenerateData* UserFunction you just created) that initialize the library, establish settings, and create a new workbook:
  1. Load the Initialize Excel Library object ([Excel => Initialize Excel Library](#))
  2. Load Excel Settings object ([Excel => Settings](#))
  3. Load New Workbook Excel object ([Excel => New Workbook](#))
  4. Wire the objects such that they execute in order as follows:



- **Step 3 (Send data to Create Line Chart object)**

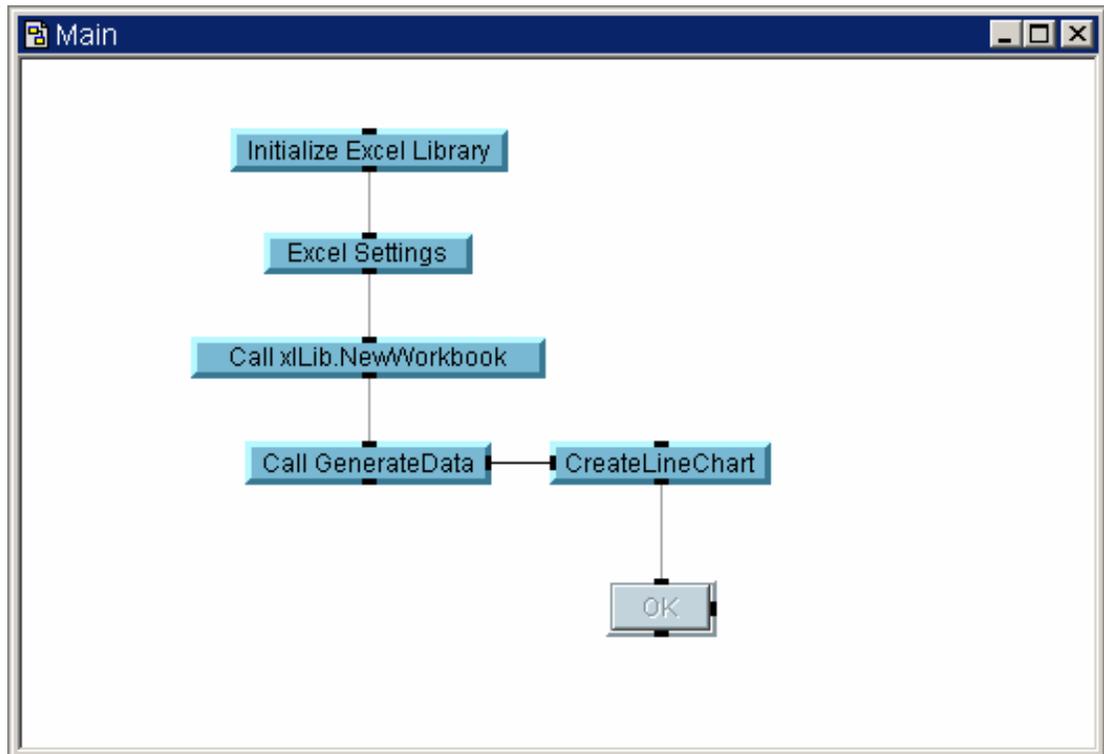
- Create a call reference to the **GenerateData** *UserFunction* (click on the *UserFunction* toaster (Fx) menu, and select [Generate => Call](#), and place the Call object in the Main program window)
- Load the object which creates an Excel line chart ([Excel => Create Chart => Create Line Chart](#))
- Wire the output of the *Call* object to the input of the *CreateLineChart* *UserObject*.

- **Step 4 (Configure the *CreateLineChart* *UserObject*)**

- Open up the *CreateLineChart* *UserObject* (double-click on the *CreateLineChart* icon)
- Modify the **Line Chart Settings** field values as follows:
  - **Title** = "Three Traces"
  - **TitleFontSize** = 24
  - **Names** = "Sine, Pulse, Noise, ..."
- Minimize the *CreateLineChart* *UserObject*

- **Step 5 (Finish the program)**

- Load *OK* button object ([Flow => Confirm \(OK\)](#))
- Wire the sequence output of the *CreateLineChart* icon to the sequence input of the *OK* button.
- Wire the sequence output of the *Call xLib.NewWorkbook* object to the sequence input of the *Call GenerateData* object.
- Your main program should look something like this:



- **Step 6 (Save & Run the program)**

- Save the program and name it **ExcelLab3.vee** ([File => SaveAs...](#))
- Run the program, and study the chart & make sure it looks reasonable
- Re-configure some of the line chart settings and re-run the program

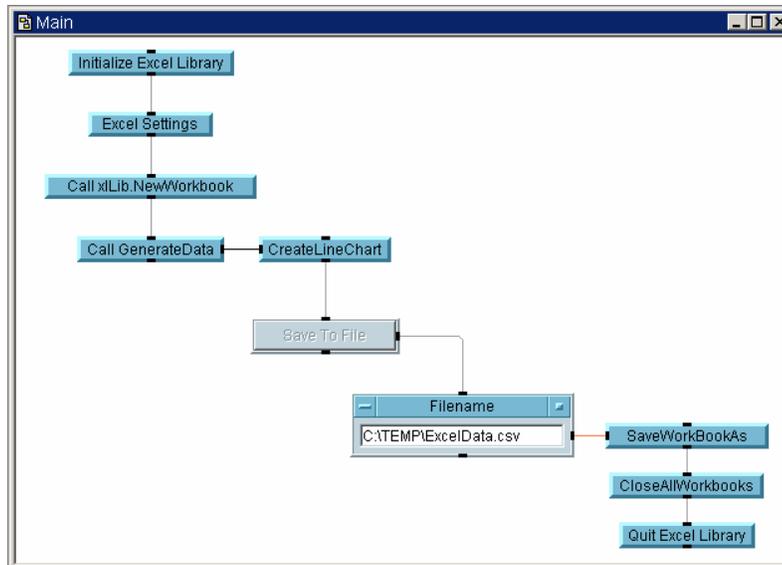
#### **Exercise 4**

Extend the previous program to save the data in the Excel spreadsheet to an ASCII file.

- **Step 1 (Rename the OK button to Save To File)**
  - Click the right mouse button on the **OK** button and select **Properties**
  - Change the *Title* property value from “OK” to “Save To File”
- **Step 2 (Determine the filename to write data into)**
  - Load object that will contain the filename (**Data => Constant Text**)
  - In the text constant value field, type in the text “C:/Temp/ExcelData.csv”
  - Retitle the constant to “Filename” (**double-click on constant object’s title bar and type in the text “Filename”**)
- **Step 3 (Load Excel objects that save a workbook to file, closes all workbooks, closes Excel and the excel library)**
  - Load the save workbook object (**Excel => Save Workbook As**)
  - Load object which closes all workbooks (**Excel => Close All Workbooks**)
  - Load object that closes Excel and unloads the Excel library (**Excel => Quit Excel Library**)
- **Step 4 (Configure the SaveWorkBookAs UserObject to write a comma separated list ASCII file)**
  - Open up the *SaveWorkBookAs* UserObject (**double-click on the SaveWorkBookAs icon**)
  - Set the **Workbook Type** value field to *CSVWindows*
  - Minimize the *SaveWorkBookAs UserObject*

- **Step 5 (Complete the wiring)**

- Wire the OK button output to the sequence input of the Text Constant
- Wire the Text Constant output to the *SaveWorkbookAs* input
- Wire the *SaveWorkbookAs* sequence output to the *CloseAllWorkbooks* sequence input
- Wire the *CloseAllWorkbooks* sequence output to the *Quit Excel Library* sequence input
- The final program should look something like this:



- **Step 6 (Save & Run the program)**

- Save the program and name it **ExcelLab4.vee** (File => SaveAs...)
- Run the program and click on the *Save To File* button.
- Open up the windows explorer, and ensure that a file named *ExcelData.csv* is located in the *Temp* directory on the *C:* drive.
- Click the right mouse button on the file and select **Edit**.
- The file should be loaded into Notepad.

**Agilent Technologies' Test and Measurement Support, Services, and Assistance**

Agilent Technologies aims to maximize the value you receive, while minimizing your risk and problems. We strive to ensure that you get the test and measurement capabilities you paid for and obtain the support you need. Our extensive support resources and services can help you choose the right Agilent products for your applications and apply them successfully. Every instrument and system we sell has a global warranty. Two concepts underlie Agilent's overall support policy: "Our Promise" and "Your Advantage."

**Our Promise**

Our Promise means your Agilent test and measurement equipment will meet its advertised performance and functionality. When you are choosing new equipment, we will help you with product information, including realistic performance specifications and practical recommendations from experienced test engineers. When you receive your new Agilent equipment, we can help verify that it works properly and help with initial product operation.

**Your Advantage**

Your Advantage means that Agilent offers a wide range of additional expert test and measurement services, which you can purchase according to your unique technical and business needs. Solve problems efficiently and gain a competitive edge by contracting with us for calibration, extra-cost upgrades, out-of-warranty repairs, and on-site education and training, as well as design, system integration, project management, and other professional engineering services. Experienced Agilent engineers and technicians worldwide can help you maximize your productivity, optimize the return on investment of your Agilent instruments and systems, and obtain dependable measurement accuracy for the life of those products.

**Agilent Email Updates**

**[www.agilent.com/find/emailupdates](http://www.agilent.com/find/emailupdates)**

Get the latest information on the products and applications you select.

**Agilent Direct**

**[www.agilent.com/find/agilentdirect](http://www.agilent.com/find/agilentdirect)**

Quickly choose and use your test equipment solutions with confidence.

**For more information on Agilent Technologies' products, applications or services, please contact your local Agilent office. The complete list is available at:**

**[www.agilent.com/find/contactus](http://www.agilent.com/find/contactus)**

**Phone or Fax**

**United States:**

(tel) 800 829 4444

(fax) 800 829 4433

**Canada:**

(tel) 877 894 4414

(fax) 800 746 4866

**China:**

(tel) 800 810 0189

(fax) 800 820 2816

**Europe:**

(tel) 31 20 547 2111

**Japan:**

(tel) (81) 426 56 7832

(fax) (81) 426 56 7840

**Korea:**

(tel) (080) 769 0800

(fax) (080) 769 0900

**Latin America:**

(tel) (305) 269 7500

**Taiwan:**

(tel) 0800 047 866

(fax) 0800 286 331

**Other Asia Pacific Countries:**

(tel) (65) 6375 8100

(fax) (65) 6755 0042

Email: [tm\\_ap@agilent.com](mailto:tm_ap@agilent.com)

Contacts revised: 05/21/05

Product specifications and descriptions in this document subject to change without notice.

© Agilent Technologies, Inc. 2006

Printed in USA, March 6, 2006

5989-4911EN

