Gateway for Windows™
Reference Manual

Version 8.6.0

285 Davidson Ave., Suite 302 • Somerset, NJ 08873-4153
Telephone: 732-560-1377 • Outside NJ 800-524-0430
Fax: 732-560-1594

Internet address: **http://www.tbred.com**

Published by:
Thoroughbred Software International, Inc.
285 Davidson Ave., Suite 302
Somerset, New Jersey 08873-4153

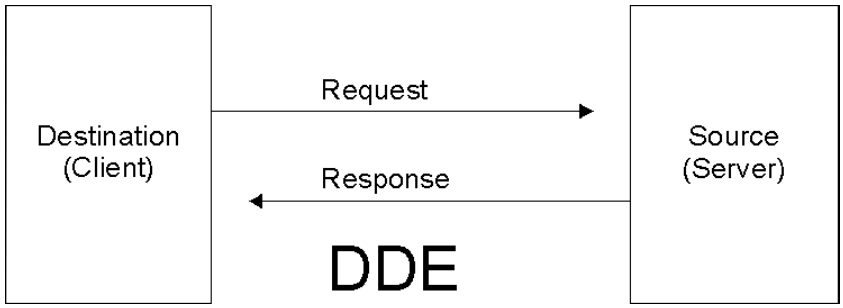
Document Number: GWW8.6.0M01

GWW - GATEWAY FOR WINDOWS™

Dynamic Data Exchange (DDE)

Microsoft provides the Dynamic Data Exchange (DDE) mechanism to support communications between any two applications running in the Windows 2000/NT/XP environments. Thoroughbred Gateway for Windows extends that communication channel beyond the limits of the Windows workstation to the host environment. This allows the host application developer to communicate directly with Windows-based software using existing DDE interfaces.

To communicate with a DDE application you need only construct a DDE instruction or multiple instructions. Communication can be two-way; you can send and receive information.



The Destination (Client) sends a Request to the Source (Server).
The Source (Server) sends a Response to the Destination (Client).

DDE follows the Client/Server model for communications control, but uses the terms: destination and source. The destination initiates a conversation, while the source responds to the request.

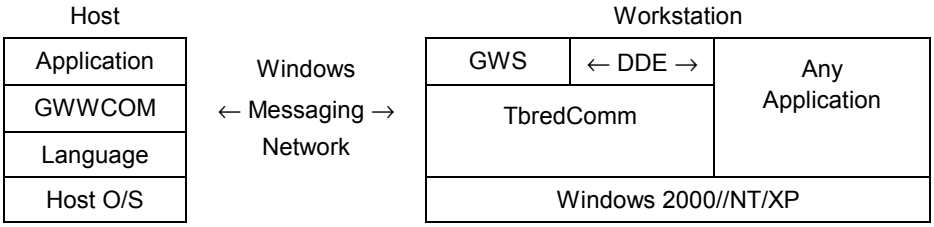
There are three basic types of DDE connection: hot, warm, and cold links.

- Hot and warm links force the source to notify the destination anytime changes in data values occur after the data has initially been transmitted to the Destination.

- Cold links require that the Destination requests an update when it is ready. Gateway for Windows supports cold DDE links, since it is integrating a host interrupt-driven environment with workstation event-driven applications.

Host to PC Communication

Gateway for Windows extends the DDE communication channel beyond the boundaries of a Windows workstation, to application software running within an operating system environment. The host and Windows workstation can be connected via serial or local area network (LAN) communication paths.



Note: To properly communicate using DDE, the source application directory must be set in your DOS Path variable. Under Win 2000/NT/XP, activate a DOS prompt, and edit the AUTOEXEC.BAT file SET PATH statement to include this directory. Under Windows NT, right-click the **My Computer** icon on the Desktop, select **Properties** from the popup menu, then choose the **Environment tab**. Find the PATH variable and add the source application directory to it.

Gateway for Windows uses GWWCOM, a Thoroughbred Dictionary-IV API running in the host environment. It controls all communications with the PC. In the host, 3GL or 4GL programs may be written containing CALLs to GWWCOM.

The Gateway Communications and VIP Gateway for Windows modules in the workstation accept all DDE communications from the host and distribute them within the workstation as required. These modules also control requests from workstation applications for communication with an application in the host.

Multiple DDE Channels

Gateway for Windows can process an unlimited number of DDE conversations at the same time. The conversations can involve any number of applications (memory permitting). With Gateway for Windows you may also have multiple conversations with the same application.

Non-DDE Applications

If a workstation application does not support a DDE interface, the host application is still able to command the PC application, using keyboard focus. In this case the host operates the application as a keyboard operator. No response can be received from the application.

Windows 2000/NT/XP Environments

Applications written in Thoroughbred development languages that have been designed to operate with VIP Gateway for Windows from standard host environments (UNIX, OpenVMS, etc.) can also operate without modification from the Thoroughbred host Windows environments (2000/NT/XP). VIP Gateway for Windows provides the VIPWIN module to replace the Gateway Communications module. It is entirely transparent to the application software.

How Applications are Controlled with DDE

Stages of DDE Conversation

There are three stages in a typical DDE conversation between two applications: initiation, transaction, and termination.

Initiation: the Destination (Client) requests a conversation with the Source (Server) by sending a Windows **INITIATE** message. Windows starts the application if it is not already active. The Source responds to this message by informing the Destination whether or not the requested conversation could be established.

Transaction: The Destination requests desired transactions. The Source then processes these requests. The Destination can send three types of messages.

REQUEST	requests a specified data item from the Source.
EXECUTE	requests that the Source execute the specified commands.
POKE	sends a particular piece of information to the Source.

Termination: At anytime, either the Source or the Destination can end the conversation by sending a Windows **TERMINATE** message. The other application then answers with a **TERMINATE** confirmation.

Components of DDE Messages

DDE messages are constructed from three components.

Source Name: A conversation begins when the Destination establishes a conversation with a Source. The Destination must, therefore, know to what name the Source will respond.

The Source name has two components:

- The name used to start the applications
- The name to which it responds through the DDE subsystem.

If these are the same name, you need only give it once. You must establish the proper Source name from the Source application's documentation. Most applications use some form of their application name as the first part of their Source name, combined with an .exe, .com, .bat, or .pif extension. For example, Microsoft Excel responds to the source name excel.exe.

Topic: A DDE conversation must also have a topic. The topic describes something in the source application that the Destination wants to access. You must establish the topic name(s) required by the Source application. For example, Microsoft Excel recognizes the name of any open document (with the extension) or the SYSTEM topic.

Item: Each DDE request, other than **EXECUTE** which operates on the application only, must reference an item. The main purpose of this item name is merely to match a Destination request to the proper Source response. The Source dictates the format of an item name, but both Destination request and Source response must reference the same item name. You must establish the type of item names the Source application requires. For example, Microsoft Excel can use a row/column reference (i.e. "R1C1") as an item.

DDE Source in the Host

Many practical exploitations of VIP Gateway for Windows have been achieved simply by controlling workstation-based applications from host software. This has enabled developers to use the presentation capabilities of any Windows-based product to significantly enhance the functionality of the host application.

However, it is equally simple to create DDE aware host applications that can respond to requests from applications running in the workstation. In this case the host application operates as the Source.

As an example, assume that a Visual Basic, Excel, Word for Windows, or some other DDE-capable process running in the Windows workstation would like to obtain information from files contained on a UNIX host. The workstation product uses standard DDE instructions to send requests to the host environment through VIP Gateway for Windows.

The source process on the host receives and interprets these **REQUEST**, **EXECUTE** procedures as specified by the requests, and then returns the information to the destination application.

Successful use of this capability requires the host application developer to correctly interpret the messages that the destination transmits, and have the source process active on the host machine for the appropriate port used by the workstation.

Programmer Guide

VIP Gateway for Windows allows many 3GL or 4GL host-based applications to communicate with Microsoft Windows workstation products.

VIP Gateway for Windows supports both DDE and non-DDE applications in the PC. However, with DDE applications communications can be two-way, and it is easier to build well-integrated applications.

To be able to take advantage of the DDE capability of workstation applications, developers must become familiar with the DDE commands documented with that application, and then issue or respond to the host commands with their application. VIP Gateway for Windows handles the communication of the host commands to the application, whether the application is in the host or the workstation.

Communications between DDE-aware applications are through messages. These messages are transmitted between the host and workstation environments by the components of VIP Gateway for Windows.

GWWCOM Communication

VIP Gateway for Windows uses a standard host public program to provide all DDE interaction: GWWCOM. This program can be called from any supported 3GL or 4GL program. The program will be operable assuming the following:

- Gateway for Windows is the active workstation communication program; and there is enough workstation memory available for operation of Gateway for Windows and the source application.

- A call to GWWCOM without meeting all of the above criteria met will result in a NO-OP; no activity will occur other than the return of an error code.

GWWCOM Call Format

```
CALL "GWWCOM", MSG$[ALL]

MSG$[0]  Return Status
MSG$[1]  Function Code
MSG$[2]  Source Name
MSG$[3]  Source Topic
MSG$[4]  Source Item(s)
MSG$[5]  Data (Send and/or Receive)
```

Detailed Descriptions MSG$

MSG$[0]: Return Status

Return Status	Meaning
. (period)	OK – Operation Successful
ERR(VIP): Description	VIP Error – Unable to Initiate Communication or VIP Not Installed
ERR(GWW): Description	DDE Error in Gateway for Windows
ERR(TSI): Description	Language Error

MSG$[1]: Function Code

Function Code	Meaning
I	Initiate DDE Conversation If application is not active on workstation, start the application.
T	Terminate DDE Conversation Leave application in current state. If you wish to terminate the application, you must do it through the DDE EXECUTE command supported by the source or through Windows keystrokes command.
P	Send (POKE) Data To Source Send a single element of data, or table of data, contained in MSG$[5] to the Item(s) specified in MSG$[$].
R	Request Data From Source Retrieve a single element of data, or table of data, for the items specified in MSG$[4], returning data in MSG$[5].
F	Focus Keystrokes To Application Send the keystrokes contained in MSG$[5] to the source application.
E	Execute the Source Command(s) in MSG$[5].
X	Examine the Windows Environment Indicate if the source application contained in MSG$[2] is active. Return the status in MSG$[5].
IN or FN	Initiate Conversation: If application not operating, start the application in Window Style "N" 1,5,9 Normal, with Focus 2 Minimized, with Focus (Default) 3 Maximized, with Focus 4,8 Normal, without Focus 6,7 Minimized, without Focus The result of any "without focus" option will be to have the operation performed invisibly. You must overtly focus the application before the operator can see it. If the application is currently operating, the focus can only be changed by an overt DDE command to the Windows Program Manager. The requested focus in initiation will be ignored.

MSG$[2]: Source Name

The source name is the Windows application with which communication is desired ("EXCEL.EXE" for Excel). It must end in .exe, .com, .bat, or .pif, other than the reserved source object ME.

Gateway for Windows assumes that the source name is the same as the executable program name minus the extension. For example, EXCEL.EXE translates to source name EXCEL. Where the initiation source name is different from the name used for DDE communications, enclose the source name in parentheses, for example MYPROG.EXE(SUPERC).

MSG$[3]: Source Topic

The source topic is the Windows application topic (SYSTEM or Open Document Name with extension for Excel). This may also be the keyword SERVER when initiating or terminating the host environment as a server process.

MSG$[4]: Source Item

The source items are the functions “P” & “R” or referenced item(s). They are formatted as: c|Item1||Item2|...0 (row/column reference for Excel).

MSG$[5]: Data

Function “P” - The data to send formatted as: c|data1||data2|...0

Function “R” - The data received formatted as: c|data1||data2|...0

Function “E” - The source commands to execute

Function “F” - The keystrokes to send to the source application

Function “X” - The application status, returned as:

|Y| = Yes - The application is active.

|N| = No - The application is not active.

Multiple Item Send and Retrieve

Multiple items can be sent or retrieved by providing a string for the source item (functions “P” & “R”) and source data (function “P”) with each entry enclosed in piping (|). Two Excel Cells could be sent by sending the following:

```
MSG$[1] = "P"
MSG$[2] = "EXCEL.EXE"
MSG$[3] = "SHEET1"
MSG$[4] = "|R1C1||R1C2|"
MSG$[5]= "|A||1|"
```

Data can be retrieved as follows:

```
MSG$[1] = "R"
MSG$[2] = "EXCEL.EXE"
MSG$[3] = "SHEET1"MSG$
[4] = "|R1C2||R1C1|"
```

After CALL:

```
MSG$[0] = "."
MSG$[5] = "|1||A|"
```

With multiple item entries, DDE failure will result in no data being returned (function “R”) or an unknown number of entries being complete (function “P”).

“ME” as Both Destination and Source

For source name you may also use the keyword “ME” for the function code “F” (Focus). This allows you to designate your host application as both the sender of keystrokes and as the receiver of the same keystrokes.

Keystrokes sent with the “ME” source name, would be returned to the host Task approximately 1 second after being sent. This allows you to call Gateway for Windows with an entire series of keystrokes, get a good response (MSG$[0] = “.”), and then branch to your normal application input code. The keystrokes will then be returned to the host program just as if they were entered from the keyboard.

Using function code “F” with the “ME” source name, but with no keystrokes, will result in focusing the Windows Program Manager to the host program Windows terminal task. This ensures that subsequent entries from the keyboard will apply to your UNIX host task.

Sending invalid keystrokes will result in a “no-op” in Gateway for Windows if you are using the “ME” source name. Attempting to send invalid keystrokes to any other source application will result in a normal Gateway for Windows error returned in MSG$[0].

Creating a Keystroke File in the Workstation

Keystrokes (or any data) may also be recorded in a file on the workstation prior to execution by sending a file command using the item variable MSG$[4].

O.FILENAME creates a new file;
I.FILENAME reads the keystrokes from an existing file created using “O.”

Filename may specify any valid path on the workstation.

Re-Sizing the "ME" Window

The "ME" terminal window can be re-sized from the host by sending a suitable Windows command in MSG$[5]. You may choose from:

```
MSG$[5]="WINDOW RESTORE"
MSG$[5]="WINDOW MINIMIZE"
MSG$[5]="WINDOW MAXIMIZE"
MSG$[5]="WINDOW MOVE(XXX,YYY,CH_WIDTH,LINE_LENGTH"
```

where xxx and yyy are screen co-ordinates, CH_WIDTH is the required character width and LINE_LENGTH is the number of lines required to be displayed on the screen.

SendKeys Format

The following describes the format for Visual Basic SendKeys, used by Gateway for Windows to send keystrokes to an application. The instruction format is taken directly from the Microsoft Visual Basic Language Reference.

Each key is represented by one or more characters. To specify a single keyboard character, use the character itself. To represent more than one character, append each to the one preceding it. For example, to represent the letters A, B, C use "ABC" for keytext.

The plus sign (+), caret (^), tilde (~), and parenthesis () have special meanings to Sendkeys. To specify one of these, enclose it inside braces {}. Brackets [] have no special meaning, but must also be enclosed in braces. To send a brace use {{} or {}}.

To send keys combined with any combination of SHIFT, CTRL, and ALT keys, precede the key code with one or more of the following:

```
SHIFT: + CTRL: ^ ALT: %
```

To obtain, for example, SHIFT with E and C use parentheses: "+(EC)".

To specify characters that are not displayed when you press a key (such as ENTER or TAB) use the codes in the following table.

Key	Code	Key	Code	Key	Code
Back Space	{BACKSPACE} or {bs} or {bksp}	Break	{BREAK}	Caps Lock	{CAPSLOCK}
Clear	{CLEAR}	Del	{DELETE} or {DEL}	Down Arrow	{DOWN}
End	{END}	Enter	{ENTER} or ~	Esc	{ESCAPE} or {ESC}
Help	{HELP}	Home	{HOME}	Ins	{INSERT}
Left Arrow	{LEFT}	Num Lock	{NUMLOCK}	Page Down	{PGDN}
Page Up	{PGUP}	Print Screen	{PRTSC}	Right Arrow	{RIGHT}
Scroll Lock	{SCROLLLOCK}	Tab	{TAB}	Up Arrow	{UP}
F1	{F1}	F2	{F2}	F3	{F3}
F4	{F4}	F5	{F5}	F6	{F6}
F7	{F7}	F8	{F8}	F9	{F9}
F10	{F10}	F11	{F11}	F12	{F12}
F13	{F13}	F14	{F14}	F15	{F15}
F16	{F16}				

Note: SendKeys cannot send keystrokes to an application that is not designed to run in Microsoft Windows. Also, SendKeys cannot send the Print Screen key to any application.

Special SendKeys

Gateway for Windows also recognizes two special SendKey formats:

{PAUSE} causes Gateway for Windows to wait four seconds before sending all of the keystrokes preceding the {PAUSE}. This is particularly valuable when using Gateway for Windows to focus keystrokes to an application using 'CI' (Host Clear Input Buffer). Keys to be sent after the {PAUSE} will not be affected by the 'CI'. You can also create a timed running demonstration using this capability. Multiple pauses can be included as {PAUSE}{PAUSE}, creating a total pause time of approximately 8 seconds.

{M:....Message Text....:M} Enclosing a message between message braces {M: and :M} will cause the message to be displayed on the monitor in a Windows Message Box before the key string preceding the message is sent to the host. If a {PAUSE} command does not immediately precede the message, one will be inserted automatically by Gateway for Windows. This allows you to include explanation text along with your self-running demo without modifying your application.

Implementation Guidelines

Adding Gateway for Windows task communication to your application can be accomplished with a minimum of 3GL or 4GL code.

All Applications

For all Windows applications with which you wish to interface, you will need to become familiar and comfortable with the normal operation of the product from the keyboard. You cannot possibly hope to interface properly with an application with which you are not familiar.

Applications Supporting DDE

For applications supporting DDE, you will need to know the format of instructions the application will accept through DDE. Each application is different, and each application supplier documents these instructions in a different manner. Excel, for example, documents each DDE instruction in a separate publication call “Function Reference.” Most of the instructions in this publication can be sent from another application using DDE.

> If you are in doubt about the correct DDE instruction to send, you can often turn to the operation of the application itself for examples. Many Windows applications, including Excel, have a Macro Record capability, where anything you accomplish with the mouse or from the keyboard is recorded in a file as a Macro Instruction. Thus you can: (1) turn on Macro Recording, (2) format your data the way you wish, (3) turn off Macro Recording, (4) examine the Macro File to determine the appropriate formatting instructions, and (5) include the host instructions in your Gateway for Windows communication from your host application.

GWWLOC

VIP Gateway for Windows includes a File Location DDE Server application that can be very useful when communicating with workstation applications. GWWLOC locates a specific filename anywhere on the user's workstation or attached file system. If GWWLOC can locate the file, the complete directory path is made available to the host destination program.

Most PC-based applications require a complete path to be given when a file is to be manipulated. When the host application programmer is sure of the file name required, but not of the entire directory structure within the PC file system, GWWLOC becomes invaluable.

GWWLOC will search for a file in the following order:

- The current directory
- The WINDOWS directory
- The WINDOWS SYSTEM directory
- The VIP directory
- All directories specified in the PATH environment variable
- All directories mapped into the network

Instructions for using GWWLOC are provided in the Other Applications ReadMe file distributed with the software.

Suggested Implementation Approaches

The following suggestions will help you minimize the amount of coding required for interface with most applications.

Isolate Communication Code into One Public Program

While you can include Gateway for Windows communication code in each application process from which you wish to communicate, that is not really an efficient manner to handle this communication. A better way is to develop a single public program for each application with which you wish to communicate. This program can then be called from any application process, supplying the desired parameters for the function required. In this manner, Windows communication is focused into one program, and the number of lines of code required from the application process can be minimized.

As an example, the sample program GWWXCEL described in the following pages contains all the code necessary to create a fully functional Excel Spreadsheet. This program can be called from any application program with a single statement, only requiring you to supply the data to be charted and the associated titles.

Communication Libraries for popular workstation products, containing the required software plus helpful advice in using Gateway for Windows, are also published as part of the VIP family of products and are available through your usual sales representatives.

Avoid Over Engineering

Most Windows applications allow a great deal of flexibility, particularly in the area of display formatting. As an example, Excel allows the operator to display data using Bar Charts, Pie Charts, 3D, etc., usually with only one or two keystrokes. You should not be concerned with building or allowing selection of all of these options from the host; simply choose a single default format, build the chart using the necessary commands for that format, then let the operator override that format, using the standard application capabilities, after the initial display.

Demonstration Programs Distributed

Gateway for Windows is delivered with a number of demonstration programs included on the distribution media. Inspection of these programs is the best way to become familiar with VIP Gateway for Windows programming techniques, and developers are encouraged to run and review these demonstration programs.

Two of these programs are included here. They are written in Thoroughbred Basic. You can adapt them to your own programming language as necessary.

GWWDEM1

This is an example of a host program communicating with Microsoft Excel. This program:

1. Initiates communication with Excel, starting it if it is not already active.
2. Transmits a series of data to the spreadsheet.
3. Retrieves the contents of a calculated field in the spreadsheet.
4. Creates a 3-Dimensional color graph from the spreadsheet.
5. Locates and re-sizes the display.

GWWXCEL

This is an example of a standard public program to use for generic Excel communications.

GWWXCEL can be used to automatically interface to Microsoft Excel with a simple CALL statement.

The examples using Excel in this manual are designed to work with Excel 97 or earlier. Some changes may be required for later versions.

```
00010 REM"GWWDEM1 Build Excel Graph using GWW"
00020 BEGIN;                    ! begin
      SETERR 9990;                 ! seterr for end
      SETESC 9990;                 ! setesc for end
      PRINT 'CS';                     ! clear screen
      VIPSC$="|";                  ! set separation character
      GWW$=CGV("GWW",ERR=25);  ! get gww common global
      IF GWW$(1,1)<>"Y"            ! if gww not set on
         GOTO 25                   ! run gww initiate
            ELSE                   ! else
            IF GWW$(4,1)<>"N"      ! if sep char <> "N"
               VIPSC$=GWW$(4,1)    ! set sep char to workstation std
            FI;                    ! endif
         GOTO 30                   ! go process
      FI                           ! endif

00025 RUN "GWW"                 ! run gww initiate program

00030 IF SSN <> 108012345 AND SSN <> 104012345 GOTO 40

00035 INPUT 'CS', @(0,1),"Use Excel, Lotus123 or QuattroPro?
         (E/L/Q):        ", O$;
      IF O$ = "L" OR O$ = "l"
            RUN "GWWDEMA",ERR=30
         ELSE
            IF O$ = "Q" OR O$ = "q"
               RUN "GWWDEMQ",ERR=30
            FI
      FI

00040 PRINT 'CS',
      @(0,1),"GWWDEM1 - Gateway to Windows To Excel Communication
         Example 1",
      @(0,3),"This example program will do the following:",
      @(0,5)," 1. Initiate communication with Excel, starting it",
      @(0,6),"    if it is not already active on the workstation",
      @(0,7)," 2. Transmit a series of data to a spreadsheet",
      @(0,8)," 3. Retrieve the contents of a field in the
                  spreadsheet",
      @(0,9)," 4. Transmit a series of formatting commands to ",
      @(0,10),"   create a 3 dimensional multi-color graph from",
      @(0,11),"   the spreadsheet",
      @(0,12)," 5.Locate and size the graph to a specific location",
      @(0,13),"   on the screen",
      @(0,14)," 6.Show you the data and instructions being
                  transmitted",
      @(0,15),"   to produce this display"

00042 PRINT @(0,16)," "

00043 INPUT "Run Gateway to Windows Excel Communication Example (Y/N)?:
      ",Y$

00050 IF CVT(Y$,32) = "N" OR CTL=4 GOTO 9990
00060 IF CVT(Y$,32) <> "Y" GOTO 10
```

```
00070 INPUT "Enter the title to use for your Excel Chart: ", Y$
00080 IF CTL=4 GOTO 10
00090 IF Y$ = "" Y$ = "VIP Gateway To Windows"

00100 PRINT 'CS',

00110 WINDOW CREATE (75,4,0,1) "BORDER=LG", "TITLE=Gateway to
         Windows",
      "BORDERATR=FG", "NAME=GWW"
01000 PRINT @(2,2),"Starting Excel",

! Only statements with an asterisk indications ------------------**
! are required for Gateway To Windows communication               **

01005 DIM MSG$[6];                ! set communication variables    **
      EXNAME$="EXCEL.EXE"         ! set excel source program name

01010 MSG$[1]="I6";               ! Function: Initiate(NoFocus,Minim)**
      MSG$[2]=EXNAME$;            ! Appl: Excel                    **
      MSG$[3]="SYSTEM";           ! Topic: System                  **
      MSG$[4]="";                 ! Item: None                     **
      MSG$[5]="";                 ! Data: None                     **
      GOSUB 2000                  ! Call GWWCOM                    **

01030 IF MSG$[0] = "."                  ! if good call
            PRINT @(0,2),'CL',          ! print
               @(2,2),"Excel Started",  ! confirmation
         ELSE                           ! else
            PRINT @(0,2),'CL',          ! print
               @(2,2),MSG$[0],          ! error
      FI;                               ! endif

      IF MSG$[0] <> "."                 ! if not good call
         INPUT "- CR TO CONTINUE",*,;   ! require acknowledge
         WINDOW DELETE ("GWW");         ! delete the window
         GOTO 10                        ! and restart
      FI                                ! endif

01100 PRINT @(0,2),'CL',                ! print
         @(2,2),"Building Graph.",      ! building graph message

! create window to show data being transmitted

01105 WINDOW CREATE(22,08,0,5) "BORDER=LG", "TITLE=Data
         Transmitted",
         "BORDERATR=FG", "NAME=GWR";

      WINDOW CREATE(22,3,0,13) "BORDER=LG", "TITLE=Data Retrieved",
         "BORDERATR=FG", "NAME=GWX";

      WINDOW CREATE(75,07,0,16) "BORDER=LG",
         "TITLE=Formatting Data Transmitted",
         "BORDERATR=FG", "NAME=GWF";
      WINDOW SELECT ("GWR")
```

```
01109 MSG$[1]="E";                    ! Function: Execute          **
      MSG$[2]=EXNAME$;                ! Appl: Excel                **
      MSG$[3]="SYSTEM";               ! Topic: System              **
      MSG$[4]="";                     ! Item: None                 **
      MSG$[5]="[ERROR(FALSE)]"        ! Data: Start with error off **
         +"[A1.R1C1(FALSE)]";            ! and R1C1 style
      GOSUB 2000

01110 MSG$[1]="P";                    ! Function: Poke Data         **
      MSG$[2]=EXNAME$;                ! Appl: Excel                **
      MSG$[3]="SHEET1";               ! Topic: Sheet1              **
      MSG$[4] = "|R1C2|"              ! Items: Set Items For Trans    **
         + "|R2C2|"                   !
         + "|R3C2|"                   !
         + "|R4C2|"                   !
         + "|R5C2|"                   !
         + "|R1C3|"                   !
         + "|R2C3|"                   !
         + "|R3C3|"                   !
         + "|R4C3|"                   !
         + "|R5C3|"                   !
         + "|R6C3|";                  !
      Q$=CHR(34);                     ! set quote character

      MSG$[5] = "|="+Q$+"A"+Q$+"|"    ! Data: Set Data Content       **
          + "|="+Q$+"B"+Q$+"|"        !
         + "|="+Q$+"C"+Q$+"|"         !
         + "|="+Q$+"D"+Q$+"|"         !
         + "|="+Q$+"E"+Q$+"|"         !
         + "|1|"                      !
         + "|2|"                      !
         + "|3|"                      !
         + "|4|"                      !
         + "|5|"                      !
         + "|=SUM(R1C3:R5C3)|";       ! Formula for Excel

      FOR X = 1 TO 5;                 ! print all data on char screen
         PRINT @(0,X-1),"R"+STR(X)+"C2",;
         PRINT @(4,X-1),"="+Q$+CHR(64+X)+Q$,;     !
         PRINT @(10,X-1),"R"+STR(X)+"C3",;
         PRINT @(15,X-1),STR(X),;     !
      NEXT X;
      PRINT @(0,5),"R6C3",;
      PRINT @(4,5),"=SUM(R1C3:R5C3)",;

      GOSUB 2000;              ! Call GWWCOM For all items         **
```

```
WINDOW SELECT ("GWX");              ! select retrieve window
PRINT @(0,0),"R6C3=",;              ! print column to retrieve

MSG$[1]="R";                        ! Function: Retrieve       **
MSG$[4]="|R6C3|";                   ! Item: Column ID          **
GOSUB 2000;                         ! call DDECOM              **

R$ = MSG$[5];                       ! set response
IF MSG$[0]="."                      ! if good response
      R$=R$(2,LEN(R$)-2             ! trim off "|" around
                                    ! response
   ELSE                             ! else
      R$ = MSG$[0]                  ! set response to error code
FI;                                 ! endif

PRINT @(5,0),R$,;                   ! print response

WINDOW SELECT ("GWF");              ! select FORMAT window
Q$ = CHR(34);                       ! set char for quote (")

MSG$[1]="E";                        ! Function: Execute        **
MSG$[2]=EXNAME$;                    ! Appl: Excel              **
MSG$[3]="SYSTEM";                   ! Topic: System            **
MSG$[4]="";                         ! Item: None               **

! Following statements conform to Excel requirements for DDE
! as documented in Excel Functions.
! They perform the following:
!    Activate the spreadsheet "sheet1"
!    Select cells build during dde conversation above
!    set proper alignment
!    set borders
!    define a new graph
!    Add a legend
!    Add some text
!    Set the graph format

X$ = "";
X$ = X$ + "[ACTIVATE(" + Q$ + "SHEET1" + Q$ + ")]";
X$ = X$ + "[FULL(TRUE)]";
X$ = X$ + "[SELECT(" + Q$ + "R1C2:R5C3" + Q$ + ")]";
X$ = X$ + "[ALIGNMENT(4)]";
X$ = X$ + "[BORDER(FALSE,FALSE,FALSE,TRUE,FALSE,FALSE)]";
X$ = X$ + "[NEW(2,1)]";
X$ = X$ + "[FORMAT.MAIN(3,1,0,50,TRUE,FALSE,FALSE)]";
X$ = X$ + "[ATTACH.TEXT(1)]";
X$ = X$ + "[FORMULA("+Q$+"="+Q$+Q$+Y$
   + Q$ + Q$ + Q$ + ")]";
X$ = X$ + "[FORMAT.FONT(0,1,FALSE," + Q$ + "HELV" + Q$
   +",18,FALSE,"
   + "FALSE,FALSE,FALSE)]";
X$ = X$ + "[SELECT(" + Q$ + Q$ + ")]";

! the following statements
!     (1) make the spreadsheet normal sized
!     (2) set the size of the spreadsheet in points
!     (3) set the location of the spreadsheet
```

```
X$ = X$ + "[APP.RESTORE()][APP.SIZE(330,200)][APP.MOVE(140,10)]";

PRINT @(0,0),X$,;

MSG$[5]=X$;                          !                          **
GOSUB 2000;                          ! call DDECOM              **

WINDOW SELECT ("GWW");

IF MSG$[0] = "."                     ! if good call
      PRINT @(0,2),'CL',             ! print
         @(2,2),"COMPLETE!",         ! confirmation
   ELSE                              ! else
      PRINT @(0,2),'CL',             ! print
         @(2,2),MSG$[0],             ! error message
FI;                                  ! endif

INPUT " - CR TO CONTINUE (Excel will terminate)", *;

WINDOW SELECT ("GWF");

PRINT 'CS',;

MSG$[1]="E";                         ! Function: Execute        **
MSG$[3]="SYSTEM";                    ! Topic: System            **
MSG$[5]="[CLOSE(FALSE)]              ! Data: Close graph        **
   +"[CLOSE(FALSE)]";                ! and spreadsheet          **

PRINT @(0,0),MSG$[5],;

GOSUB 2000;                          ! call DDECOM              **

MSG$[1]="E"                          ! Function: Execute        **
MSG$[3]="SYSTEM";                    ! Topic: System            **
MSG$[5]="[QUIT()]";                  ! Data: Quit Excel         **

PRINT @(0,1),MSG$[5],;

GOSUB 2000;                          ! call DDECOM              **

MSG$[1]="F";                         ! set focus
MSG$[2]="ME";                        ! back to this task
MSG$[5]="";                          ! with no keystrokes
GOSUB 2000;                          ! call DDECOM for focus

WINDOW SELECT ("GWW");

PRINT @(0,2),'CL',@(2,2),"EXCEL TERMINATED!",;

INPUT " - CR TO CONTINUE", *;

WINDOW DELETE ("GWW");               ! delete active window
WINDOW DELETE ("GWR");
WINDOW DELETE ("GWX");
WINDOW DELETE ("GWF");
GOTO 10                              ! go rerun
```

```
01800 IF VIPSC$ <> "|"                ! INT'L if not a standard sep
                                      ! char
         OLD$=MSG$[4];                ! using msg$[4]
         CHAR$="|";                   ! set look for char to standard
         RCHAR$=VIPSC$;               ! set replace char to override
         GOSUB 1900;                  ! set new char
         MSG$[4]=OLD$;                ! and replace
         OLD$=MSG$[5];                ! using msg$[5]
         GOSUB 1900;                  ! set new char
         MSG$[5]=OLD$                 ! and replace
      FI;                             ! endif
      RETURN                          ! return

01850 IF VIPSC$ <> "|"               ! *INT'L if not a standard sep char
         OLD$=MSG$[5];               ! using result
         CHAR$=VIPSC$;               ! set look for char to override
         RCHAR$="|";                 ! set replace char to standard
         GOSUB 1900;                 ! go replace
         MSG$[5]=OLD$                ! and restore result
      FI;                            ! endif
      RETURN                         ! return

01900 WHILE POS(CHAR$=OLD$)<>0;      ! INTL while look for char is in
                                     ! string
         SCPOS=POS(CHAR$=OLD$);      ! set its position
         OLD$(SCPOS,1)=RCHAR$;       ! replace with override char
      WEND;                          ! wend
      RETURN                         ! return

02000 REM"TRANSMIT TO GATEWAY"       ! call DDECOM                    **

02010 GOSUB 1800                     ! INT'L ONLY - check for sep char

02060 CALL "GWWCOM",MSG$[ALL]        ! call public                    **

02070 GOSUB 1850                     ! INT'L ONLY - check for sep char

02090 RETURN                         ! RETURN                         **

09990 RUN "VIPMENU",ERR=9991
09999 END
```

The following is an example of the results that can be obtained with proper implementation of Gateway for Windows using the program GWWXCEL.

Thoroughbred Basic Code Required

```
TITLES="|2005 Sales||2006 Sales||2007 Sales|";
DATA1$="|825000.00||1245000.00||1475000.00|";
DATA2$="";
NAME$="My Company Sales";
CALL"GWWXCEL","MYPROG",TITLE$,DATA1$,DATA2$,NAMES$,70,24,1,1;
```

```
00010 REM "GWWXCEL Build Excel Graph Using GWW Public Routine"
00015 SETERR 09990;                       ! seterr
      SETESC 09990                        ! setesc

00020 ENTER P$                          ! programname
            Y1$,                        ! referencename string
            Y2$,                        ! data string
            Y3$,                        ! 2nd data string (optional)
            T$,                         ! report title
            W,                          ! width in characters
            H,                          ! height in lines
            C,                          ! starting column
            R                           ! starting line

00025 SETERR 09990;                     ! seterr
      SETESC 09990;                     ! setesc
      Q$=CHR(34);                       ! set quote character
      VIPSC$="|";                       ! set separation character
                                        ! default
      GWW$=CGV("GWW",ERR=9990);         ! get gww global
      IF GWW$(1,1)<>"Y"                 ! ifot on
            GOTO 9990                   ! exit
         ELSE                           ! else
            IF GWW$(4,1)<>"N"           ! if sep charot =
               VIPSC$=GWW$(4,1)         ! set override sep char
            FI                          ! endif
      FI;                               ! endif
      DATAOPT$="1";                     ! set data option to 1
      IF CVT(Y3$,128) <> "" THEN        ! if 2nd data string
         DATAOPT$="2"                   ! set data option to 2
      FI                                ! endif

00026 T1$="";                       ! set 1st data heading to blank
      T2$="";                       ! set 2nd data heading to blank
      T0$="";                       ! set combined heading to blank
      T9$=T$+" ";                   ! add spaces to heading string
      IF POS(VIPSC$=T9$)=1 AND      ! if there is a start char
         POS(VIPSC$=T9$(2))<>0      ! and an end char
         T9$=T9$(2);                ! delete the start char
         T0=POS(VIPSC$=T9$);        ! get pos of end char
         T0$=T9$(1,T0-1);           ! set combined heading
         T9$=T9$(T0+1);             ! and truncate from string
         IF POS(VIPSC$=T9$)=1 AND ! if there is a start char
            POS(VIPSC$=T9$(2))<>0 ! and an end char
            T9$=T9$(2);             ! truncate start char
            T0=POS(VIPSC$=T9$);     ! get pos of end char
            T1$=T9$(1,T0-1);        ! set 1st data heading
            T9$=T9$(T0+1);          ! and truncate from string
            IF POS(VIPSC$=T9$)=1 AND    ! if there is a start char
               POS(VIPSC$=T9$(2))<>0    ! and an end char
               T9$=T9$(2);              ! truncate start char
               T0=POS(VIPSC$=T9$);      ! get pos of end char
               T2$=T9$(1,T0-1)          ! set 2nd data heading
            FI                          ! endif
         FI                             ! endif
      FI;                               ! endif
```

```
      IF T0$=""                        ! if combined heading blank
         T0$=T$                        ! set it to total heading string
      FI;                              ! endif
      IF T1$=""                        ! if 1st data heading blank
         T1$="Series 1"                ! set it to Excel Std Series 1
      FI;                              ! endif
      IF T2$=""                        ! if 2nd data heading blank
         T2$="Series 2"                ! set it to Excel Std Series 2
      FI                               ! endif

00029 IF W = 0                            ! if 0 width
         W = 75                           ! set to almost full width
      FI;                                 ! endif
      IF H = 0                            ! if 0 height
         H = 27                           ! set to almost screen height
      FI;                                 ! endif
      WIDTH=INT(W*72/12);                 ! calc width in points
      HEIGHT=INT(H*72/6);                 ! height
      HORZ=INT(C*72/12);                  ! characters
      VERT=INT(R*72/6)                    ! lines

00030 WINDOW CREATE (7,4,1,20) "NAME=EXCEL",  ! create action window
      "BORDER=LG", "BORDERATR=FG";            ! with attributes
      PRINT @(0,0),'BF',@(0,0),"(GWW)",'ER',; ! and print gww status
      SETERR 1190;                        ! seterr to del window, exit
      SETESC 1190                         ! setesc to del window, exit

00040 S$="SHEET1";                        ! start with excel sheet 1
      P$=CVT(P$,8+128);                   ! strip programname
      OPT$="NEW"                          ! set option =ew spreadsheet

01005 DIM MSG$[6];                        ! dim GWW pass string
      EXNAME$="EXCEL.EXE"

01006 MSG$[1]="X";                        ! function: get info
      MSG$[2]= EXNAME$;                   ! appl:     excel
      ERREXIT$="Y";                       ! set exit on error to yes
      GOSUB 02000;                        ! go get info from GWW
      R$=MSG$[5]                          ! get response from call

01010 MSG$[1]="I6";                  ! function: initiate,ormal,o focus
      MSG$[2]= EXNAME$;              ! appl: excel
      MSG$[3]="SYSTEM";              ! topic: system
      MSG$[4]="";                    ! item: one
      MSG$[5]="";                    ! data: one
      GOSUB 02000                    ! call GWW

01021 MSG$[1]="E";                        ! function: execute
      MSG$[3]="SYSTEM";                   ! topic: system
      MSG$[5]="[ERROR(FALSE)]"+           ! data: turn Excel errors off
         "[A1.R1C1(FALSE)]";              ! + R1C1 reference
      GOSUB 2000                          ! call GWW
```

```
01022 MSG$[1]="R"              ! function: retrieve
      MSG$[3]=S$;              ! topic: sheetame
      MSG$[4]=VIPS             ! set start sep char
         +"R1C1"               ! + item: row 1, column 1
         +VIPSC$;              ! + end sep char
      GOSUB 02000              ! call GWW to get current sheet info

01024 X5$=MSG$[5];                ! set response
      IF LEN(X5$)=2               ! if only two chars
            X5$=""                ! set response toone
         ELSE                     ! else
            X5$=CVT(X5$(2,LEN(X5$)-2),8+128)  ! set last programname
      FI;                         ! endif
      IF POS(P$=X5$)<>1           ! if curr program <> this sheet pgm
         IF X5$=""                ! if itso program
            OPT$="NEW";           ! set option =ew
            GOTO 01030            ! go process
         FI;                      ! endif
         S$="SHEET"+STR(NUM(S$(6))+1);  ! add 1 to sheet id
         MSG$[3]=S$;              ! set topic toext sheet
         ERREXIT$ = "N";          ! dont exit if error
         GOSUB 02000;             ! call GWW to get thisame
         IF MSG$[0]="."           ! if a good sheet
               GOTO 01024         ! goto check again
            ELSE                  ! else
               MSG$[1]="E";       ! function: execute
               MSG$[3]="SYSTEM";  ! topic: system
               MSG$[5]="[NEW(1,0)]";    !text: ew sheet
               GOSUB 02000;       ! call GWW to createew sheet
               GOTO 01030         ! go process
            FI                    ! endif
         ELSE                     ! else
            OPT$="CURR"           ! set option to curr sheet there
         FI                       ! endif

01030 X1$=Y1$+"  ";               ! add blanks to reference cells
      X2$=Y2$+"  ";               ! add blanks to 1st data cells
      X3$=Y3$+"  ";               ! add blanks to 2nd data cells
      R=0;                        ! set row to zero
      GOSUB 6000                  ! set alpha location references

01040 PRINT @(1,1),"P..",               ! print poke status in window
01110 MSG$[1]="P";                      ! function: poke
      MSG$[2]= EXNAME$;                 ! appl: excel
      MSG$[3]=S$;                       ! topic: sheetame
      MSG$[4]="";                       ! start item string asull
      MSG$[5]="";                       ! start data string asull
```

```
WHILE POS(VIPSC$=X1$)<>0          ! while there is a good start
                                  ! seperator
   AND POS(VIPSC$=X1$(2))<>0;     ! and a good end seperator
      P1=POS(VIPSC$=X1$);         ! get start sep
      X1$=X1$(2);                 ! truncate it
      P2=POS(VIPSC$=X1$);         ! get end sep
      R=R+1;                      ! add 1 to row
      MSG$[4]=MSG$[4]+VIPSC$      ! add to current string 1 sep
                                  ! char
         +"R"+STR(R)+"C2"         ! + row and column id
         +VIPSC$;                 ! + end sep char
      X1$=X1$(P2+1);              ! truncate this entry
WEND;                             ! wend

R=0;                              ! set row to zero

WHILE POS(VIPSC$=X2$)<>0          ! while there is a good
                                  ! seperator
   AND POS(VIPSC$=X2$(2))<>0;     ! and a good end seperator
      P1=POS(VIPSC$=X2$);         ! get start sep
      X2$=X2$(2);                 ! truncate it
      P2=POS(VIPSC$=X2$);         ! get end sep
      R=R+1;                      ! add 1 to row
      MSG$[4]= MSG$[4]            ! add to curr string
         + VIPSC$                 ! 1 sep char
         + "R"                    ! + row indicator
         + STR(R)                 ! + row id
         + "C3"                   ! + column
         + VIPSC$;                ! + end sep char
      X2$=X2$(P2+1);              ! truncate this entry
WEND;                             ! wend

IF DATAOPT$ <> "1"              ! if more than one data string
   R=0                          ! reset row counter
FI;                             ! endif

WHILE POS(VIPSC$=X3$)<>0        ! while there is a start char
   AND POS(VIPSC$=X3$(2))<>0;   ! and an end char
      P1=POS(VIPSC$=X3$);       ! get start char
      X3$=X3$(2);               ! truncate it
      P2=POS(VIPSC$=X3$);       ! get end char
      R=R+1;                    ! add 1 to row
      MSG$[4]=MSG$[4]           ! add to current msg string
         +VIPSC$                ! 1 sep char
         +"R"                   ! + row id
         +STR(R)                ! + rowumber
         +"C4"                  ! + column
         +VIPSC$;               ! + end sep char
      X3$=X3$(P2+1);            ! truncate this entry
WEND;                           ! wend
```

```
MSG$[5]=Y1$+Y2$+Y3$;           ! set data strings
GOSUB 02000;                   ! call GWW to send sheet id data
MSG$[4]= VIPSC$                ! createew string with 1 sep
                               ! char
         + "R1C1"              ! + r1c1 indicator
         + VIPSC$ + VIPSC$     ! + 2 sep chars
         + "R2C1"              ! + r2c1 indicator
          + VIPSC$ + VIPSC$    ! + 2 sep chars
         + "R3C1"              ! + r3c1 indicator
         + VIPSC$;             ! + end sep char
MSG$[5]= VIPSC$                ! set data string with 1 sep
                               ! char
         + P$                  ! + sourcename
         + VIPSC$ + VIPSC$     ! + 2 sep chars
         + S$                  ! + sheetame
         + VIPSC$ + VIPSC$     ! + 2 sep chars
         + T0$                 ! + title
         + VIPSC$;             ! + end sep char

GOSUB 02000;                   ! call GWW to set this data
MSG$[1]="E";                   ! function: execute
MSG$[2]= EXNAME$;              ! appl: excel
MSG$[3]="SYSTEM";              ! topic: system
MSG$[4]="";                    ! item: one
X$="";                         ! initialize transmit string
IF DATAOPT$<>"1"               ! ifot one data column
      COL$="C4"                ! select over to column 4
   ELSE                        ! else
      COL$="C3"                ! select over to column 3
FI;                            ! endif
IF OPT$="NEW"                  ! if aew sheet
   X$=X$+"[ACTIVATE()]";       ! activate excel sheet
   X$=X$+"[FULL(TRUE)]";       ! set full option
   X$=X$+"[SELECT("+Q$+"R1C2:R"+STR(R)+COL$+Q$+")]"; ! select
                               ! all data
   X$=X$+"[ALIGNMENT(4)]";     ! set alignment
   X$=X$+"[BORDER(FALSE,FALSE,FALSE,TRUE,FALSE,FALSE)]"; ! and
                               ! border
   X$=X$+"[NEW(2,1)]";         ! create aew graph
   IF DATAOPT$="1"             ! if only one data column
         X$=X$+"[GALLERY.3D.COLUMN(4)]";   ! type 3D column
         X$=X$+"[FORMAT.MAIN(8,1,0,50,TRUE,FALSE,FALSE)]"
      ELSE                     ! else
         X$=X$+"[GALLERY.COLUMN(1,TRUE)]"; ! type 2D column
         X$=X$+"[FORMAT.MAIN(3,1,0,50,TRUE,FALSE,FALSE)]"
   FI;                         ! endif
```

```
   X$=X$+"[ATTACH.TEXT(1)]";      ! attach for the title
   X$=X$+"[FORMULA("+Q$+"="+Q$+Q$+T0$+Q$+Q$+Q$+")]";   ! the
                                  ! reportname
   IF DATAOPT$<>"1"               ! ifot single data graph
                                  ! created
      X$=X$+"[ACTIVATE("+Q$+S$+Q$+")]"; ! select each sheet
      X$=X$+"[SELECT("+Q$+"R1C2:R"+STR(R)+"C3"+Q$+")]";
      X$=X$+"[NEW(2,1)]";
      X$=X$+"[GALLERY.3D.COLUMN(4)]";
      X$=X$+"[FORMAT.MAIN(8,1,0,50,TRUE,FALSE,FALSE)]";
      X$=X$+"[ATTACH.TEXT(1)]";
      X$=X$+"[FORMULA("+Q$+"="+Q$+Q$+T1$+Q$+Q$+Q$+")]";
      X$=X$+"[ACTIVATE("+Q$+S$+Q$+")]";
      X$=X$+"[SELECT("+Q$+"R1C2:R"+STR(R)+"C2,R1C4:R"+
         STR(R)+"C4"+Q$+")]";
      X$=X$+"[NEW(2,1)]";
      X$=X$+"[GALLERY.3D.COLUMN(4)]";
      X$=X$+"[FORMAT.MAIN(8,1,0,50,TRUE,FALSE,FALSE)]";
      X$=X$+"[ATTACH.TEXT(1)]";
      X$=X$+"[FORMULA("+Q$+"="+Q$+Q$+T2$+Q$+Q$+Q$+")]"
   FI                             ! endif
FI;                               ! endif
X$=X$+"[SELECT("+Q$+Q$+")]";      ! select the graph
X$=X$+"[APP.RESTORE()]";          ! make it big
X$=X$+"[APP.SIZE("+STR(WIDTH)+","+STR(HEIGHT)+")]"; ! size it
X$=X$+"[APP.MOVE("+STR(HORZ)+","+STR(VERT)+")]";    ! and move
                                  ! it
IF OPT$ = "NEW"                   ! if this isew sheet
   X$=X$+"[ACTIVATE("+Q$+S$+Q$+")]"; ! activate the data sheet
   X$=X$+"[HIDE()]";              ! and hide it
   X$=X$+"[ARRANGE.ALL()]"        ! then arrange the rest
FI;                               ! endif

X$=X$+"[APP.RESTORE()]";          ! restore the base sheet
MSG$[5]=X$;                       ! set execute items
GOSUB 02000;                      ! call GWW to format graph
MSG$[1]="F";                      ! function: focus to ensure
                                  ! display
MSG$[2]= EXNAME$;                 ! appl: excel
MSG$[3]="SYSTEM";                 ! topic: system
MSG$[4]="";                       ! item: one
MSG$[5]="%{ }{ENTER}";            ! alt, space, enter
GOSUB 02000                       ! send these keys to activate
                                  ! graph
```

```
01190 WINDOW DELETE ("EXCEL");          ! delete the status window
      GOTO 09990                        ! and exit
02000 REM"GWW Transmit"                 ! call GWW
02010 PRINT @(1,1), MSG$[1](1,1),".X",; ! print function code
                                        ! transmitted
      CALL"GWWCOM",MSG$[ALL];           ! call GWW with all parms
      PRINT @(1,1),"...",;              ! clear function code
                                        ! transmitted
      IF MSG$[0]<>"." AND ERREXIT$="Y"  ! ifot a good call & exit
                                        ! on err yes
            EXITTO 1190                 ! exit at this time
         ELSE                           ! else
            ERREXIT$="Y";               ! set exit onext error to yes
            RETURN                      ! return
      FI                                ! endif

06000 REM "make sure refs are alpha     ! check reference string for
                                        ! alphas
06010 Y5$=Y1$+"  ";                     ! set reference string + 2
                                        ! blanks
      Y1$="";                           ! clear original ref string
      WHILE POS(VIPSC$=Y5$)=1 AND       ! while there is a start char
            POS(VIPSC$=Y5$(2))<>0;      ! and an end char
         Y1$ = Y1$                      ! add to previously decode
                                        ! string
            + Y5$(1,1)                  ! the start char
            + "="                       ! + excel = sign for formula
            + Q$;                       ! + quote for text
         Y5$ = Y5$(2);                  ! truncate the start char
         Y5  = POS(VIPSC$=Y5$);         ! find the end char
         Y1$ = Y1$                      ! add to decoded string
            + Y5$(1,Y5-1)               ! the remainder of the
                                        ! formula
            + Q$                        ! + end quote for text
            + Y5$(Y5,1);                ! + end character
         Y5$ = Y5$(Y5+1);               ! and set the remainder of
                                        ! source
      WEND;                             ! wend
      RETURN                            ! return
09990 EXIT                              ! exit graphic routine
09999 END                               ! end
```