

FORTRAN 77
Language Reference Booklet

RT PC Graphics Development Toolkit

Programming Family



**Personal
Computer
Software**

59X8617

IBM RT PC FORTRAN 77 Language Reference Booklet

This reference booklet describes the IBM RT PC FORTRAN 77 and IBM RT PC FORTRAN 77 Subset syntax required to write an application program using the IBM RT PC Graphics Development Toolkit. Any general or special considerations for the FORTRAN language are described in this booklet. It is important that you keep this booklet in a safe place. This booklet is the only source of information that specifically describes the IBM RT PC FORTRAN 77 interface to the Toolkit.

Writing in FORTRAN 77

Writing in the FORTRAN 77 language requires a few special considerations. They are as follows:

- **Array numbering.** Array tables in the *IBM RT PC Graphics Development Toolkit* show the array index starting with 1 and continuing to n.

For some routines the size of an array for one parameter is based on the value of another parameter. This is shown in the "Data Types:" by using the parameter name in the array size. This notation is used only to show the relationship between the parameters and does not imply actual coded values or refer to storage allocation.

- **Data Types.** Use **integer*4** for signed 32-bit integers and **character** for characters. Be sure to designate the names of the call to a Toolkit routine as integer*4 in your application program. Below is an example that defines character strings, integer parameters, and routine names:

```
      character*40 temp1, temp2, temp3,  
      *temp4, temp5  
c     this defines five character strings  
c     40 characters long  
c  
      integer*4 devhdl, echo, xin, yin  
c     this defines four 32-bit integers  
c  
      integer*4 vopnwk, vsthgt, vslcol  
c     this defines three routine names as  
c     32-bit integers
```


FORTRAN 77 Syntax

All parameters are of integer*4 type, unless defined otherwise in "Data Types:" after the routine syntax description.

In the "Routines" section of this booklet, two entries have been included to show the syntax of some routines; one for FORTRAN 77 and one for its subset. These FORTRAN 77 subset routines are described in order to provide information to assist in transferring programs that have been written for the IBM FORTRAN V2.00 Compiler.

A double asterisk (**) following a generic routine name indicates that the routine is device-dependent. Using these routines in an application program makes that application program device-dependent.

Throughout the "Routines" section of this booklet, the input parameters are italicized and the output parameters are shown in regular type.

Routines

Application Data (FORTRAN 77)

status = vappl
(*handle, functn, datcnt, appdat*)

Data Types: character*n functn
 integer*4 appdat(datcnt)

Application Data (FORTRAN 77 Subset)

status = vapplS
(*handle, funcnt, functn, datcnt, appdat*)

Data Types: character*1 functn(funcnt)
 integer*4 appdat(datcnt)

Clear Workstation

status = vclrwk
(*handle*)

Close Workstation

status = vclswk
(*handle*)

Copy Page**

status = vcpage
(*handle, source, destin*)

Copy Pels**

status = vcppeL
(*handle, xy*)

Data Types: integer*4 xy(6)

Cursor Down**

status = vcurdn
(*handle*)

Cursor Home**

status = vcurhm
(*handle*)

Cursor Left**

status = vcurlf
(*handle*)

Cursor Right**

status = vcurrT
(*handle*)

Cursor Up**

status = vcurup
(*handle*)

Direct Cursor Address**

status = vcurad
(*handle*, *row*, *column*)

Display Graphic Input Cursor**

status = vdspcr
(*handle*, *x*, *y*)

Enter Cursor Addressing Mode**

status = vencur
(*handle*)

Erase to End of Line**

status = vereol
(*handle*)

Erase to End of Screen**

status = vereos
(*handle*)

Exit Cursor Addressing Mode**

status = vexcur
(*handle*)

Get Pels**

status = vgtpel
(*handle*, *xy*, *parray*)

Data Types: integer*4 xy(4)
 integer*4 parray(n)

Hardcopy**

status = vhdcpy
(*handle*)

Input Choice (request mode)

status = vrqchc
(*handle*, *chin*, *chout*)

status = 0 request unsuccessful
 > 0 request successful
 = -1 an error has occurred

Input Choice (sample mode)

status = vsmchc
(*handle*, *chout*)

status = 0 sample unsuccessful
 > 0 sample successful
 = -1 an error has occurred

Input Locator (request mode)

status = vrqloc
 (*handle, xyin, ink, rband, echdl, xyout, term*)

Data Types: integer*4 xyin(2)
 integer*4 xyout(2)
 character*1 term

status = 0 request unsuccessful
 > 0 request successful
 =-1 an error has occurred

Input Locator (sample mode)

status = vsmloc
 (*handle, xyin, xyout, pressed, released, keystate*)

Data Types: integer*4 xyin(2)
 integer*4 xyout(2)

status = 0 sample unsuccessful
 > 0 sample successful.
 =-1 an error has occurred

Input String (request mode)

status = vrqstr
 (*handle, maxlen, echomd, echoxy, string*)

Data Types: integer*4 echoxy(2)
 character*1 string(n)

status = 0 request unsuccessful
 > 0 number of characters returned
 =-1 an error has occurred

Input String (sample mode)

status = vsmstr
 (*handle, maxlen, echomd, echoxy, string*)

Data Types: integer*4 echoxy(2)
 character*1 string(n)

status = 0 sample unsuccessful
 > 0 number of characters returned
 =-1 an error has occurred

Input Valuator (request mode)

status = vrqval
 (*handle, valin, echdl, valout*)

status = 0 request unsuccessful
 > 0 request successful
 =-1 an error has occurred

Input Valuator (sample mode)

status = vsmval
(*handle*, valout)

status = 0 sample unsuccessful
> 0 sample successful
=-1 an error has occurred

Inquire Addressable Character Cells

status = vqchcl
(*handle*, rows, columns)

Inquire Alpha Text Capabilities

status = vqacap
(*handle*, alphcp)

Data Types: integer*4 alphcp(15)

Inquire Alpha Text Cell Location

status = vqacel
(*handle*, row, column, proflg, xout, yout)

Inquire Alpha Text Font Capability

status = vqafnt
(*handle*, fontin, sizein, capaby)

Data Types: integer*4 capaby(7)

status = 0 font unavailable
> 0 font available
=-1 an error has occurred

Inquire Alpha Text Position

status = vqapos
(*handle*, xout, yout)

**Inquire Alpha Text String Length
(FORTRAN 77)**

status = vqalen
(*handle*, string)

Data Types: character*n string

status ≥ 0 length
=-1 an error has occurred

**Inquire Alpha Text String Length
(FORTRAN 77 SUBSET)**

status = vqalns
(*handle*, count, string)

Data Types: character*1 string(count)

status ≥ 0 length
=-1 an error has occurred

Inquire Cell Array

status = vqclry
(*handle*, *xy*, *rowlen*, *numrow*, *elmrow*, *rowuse*, *value*,
colors)

Data Types: integer*4 xy(4)
 integer*4 colors(N)

Inquire Color Representation

status = vqclor
(*handle*, *indin*, *setflg*, *rgb*)

Data Types: integer*4 rgb(3)

status ≥ 0 actual index selected
 = -1 an error has occurred

Inquire Current Cursor Text Address**

status = vqcura
(*handle*, *row*, *column*)

Inquire Current Fill Area Attributes

status = vqfatt
(*handle*, *attrib*)

Data Types: integer*4 attrib(4)

Inquire Current Graphic Text Attributes

status = vqtatt
(*handle*, *attrib*)

Data Types: integer*4 attrib(10)

Inquire Current Polyline Attributes

status = vqlatt
(*handle*, *attrib*)

Data Types: integer*4 attrib(4)

Inquire Current Polymarker Attributes

status = vqmatt
(*handle*, *attrib*)

Data Types: integer*4 attrib(4)

Inquire Cursor Text Mode**

status = vqcurm
(*handle*)

status ≥ 0 current mode
 = -1 an error has occurred

Inquire Error

status = vqerr()

Inquire Graphic Color Burst Mode**status = vqgcm
(*handle*)status ≥ 0 actual mode selected
= -1 an error has occurred**Inquire Page****status = vqpage
(*handle*, grmode, curmod)Data Types: integer*4 grmode(3)
integer*4 curmod(3)**Message** (FORTRAN 77)**status = vmsg
(*handle*, *msg*, *wait*)

Data Types: character*n msg

Message (FORTRAN 77 SUBSET)**status = vmsgs
(*handle*, *msgcnt*, *msg*, *wait*)

Data Types: character*1 msg(msgcnt)

Open Workstationstatus = vopnwk
(*workin*, *handle*, *workot*)Data Types: integer*4 workin(19)
integer*4 workot(66)**Output Alpha Text (FORTRAN 77)**status = vatext
(*handle*, *string*, *xout*, *yout*)

Data Types: character*n string

Output Alpha Text (FORTRAN 77 Subset)status = vatxts
(*handle*, *count*, *string*, *xout*, *yout*)

Data Types: character*1 string(count)

Output Arcstatus = varc
(*handle*, *x*, *y*, *radius*, *begang*, *endang*)

Output Bar

status = vbar
(*handle, xy*)

Data Types: integer*4 xy(4)

Output Cell Array

status = vclary
(*handle, xy, rowlen, elmrow, numrow, wrmode, colors*)

Data Types: integer*4 xy(4)
integer*4 colors(n)

Output Circle

status = vcircl
(*handle, x, y, radius*)

Output Cursor Addressable Text (FORTRAN 77)**

status = vctext
(*handle, string*)

Data Types: character*n string

Output Cursor Addressable Text
(FORTRAN 77 SUBSET)**

status = vctxts
(*handle, count, string*)

Data Types: character*1 string(count)

Output Filled Area

status = vfarea
(*handle, count, xy*)

Data Types: integer*4 xy(2*count)

Output Graphic Text (FORTRAN 77)

status = vgtxt
(*handle, x, y, string*)

Data Types: character*n string

Output Graphic Text (FORTRAN 77 SUBSET)

status = vgtxts
(*handle, x, y, count, string*)

Data Types: character*1 string(count)

Output Pie Slice

status = vpiesl
(*handle, x, y, radius, begang, endang*)

Output Polyline

status = vpline
(*handle, count, xy*)

Data Types: integer*4 xy(2*count)

Output Polymarker

status = vpmark
 (*handle*, *count*, *xy*)

Data Types: integer*4 xy(2*count)

Put Pels**

status = vptpel
 (*handle*, *xy*, *parray*)

Data Types: integer*4 xy(2)
 integer*4 parray(n)

Read Cursor Movement Keys**

status = vrdcky
 (*handle*, *inpmid*, *direct*, *key*)

Data Types: character*1 key

Remove Graphic Input Cursor**

status = vremcr
 (*handle*)

Reverse Video Off**

status = vrvoff
 (*handle*)

Reverse Video On**

status = vrvon
 (*handle*)

Set Alpha Text Color Index

status = vsacol
 (*handle*, *indin*)

status ≥ 0 index selected
 = -1 an error has occurred

Set Alpha Text Font and Size

status = vsafnt
 (*handle*, *fontin*, *sizein*, *fontcp*)

Data Types: integer*4 fontcp(8)

status = 0 font unavailable
 > 0 font selected
 = -1 an error has occurred

Set Alpha Text Line Spacing

status = vsaspc
(*handle*, *spacin*)

status \geq 0 spacing selected
= -1 an error has occurred

Set Alpha Text Overstrike Mode

status = vsaovr
(*handle*, *modein*)

status \geq 0 mode selected
= -1 an error has occurred

Set Alpha Text Pass Through Mode

status = vsapas
(*handle*, *modein*)

status \geq 0 mode selected
= -1 an error has occurred

Set Alpha Text Position

status = vsapos
(*handle*, *xin*, *yin*, *xout*, *yout*)

Set Alpha Text Quality

status = vsaql
(*handle*, *modein*)

status \geq 0 mode selected
= -1 an error has occurred

Set Alpha Text Subscript Superscript Mode

status = vsasub
(*handle*, *modein*)

status \geq 0 mode selected
= -1 an error has occurred

Set Alpha Text Underline Mode

status = vsaund
(*handle*, *modein*)

status \geq 0 mode selected
= -1 an error has occurred

Set Background Color Index

status = vsbcol
(*handle*, *indin*)

status \geq 0 index selected
= -1 an error has occurred

Set Character Height

status = vsthgt
(*handle*, *rqhgt*, *chrwid*, *celwid*, *celhgt*)

status ≥ 0 height selected
= -1 an error has occurred

Set Color Representation

status = vscolr
(*handle*, *indin*, *rgbin*, *rgbout*)

Data Types: integer*4 rgin(3)
 integer*4 rgbout(3)

status ≥ 0 actual index selected
= -1 an error has occurred

Set Cursor Text Attributes**

status = vcratt
(*handle*, *reqatt*, *selatt*)

Data Types: integer*4 reqatt(4)
 integer*4 selatt(4)

Set Cursor Text Color Index**

status = vcrcol
(*handle*, *forreq*, *bacreq*, *forse*, *bacsel*)

Set Cursor Text Mode**

status = vscurm
(*handle*, *mode*)

status ≥ 0 mode selected
= -1 an error has occurred

Set Fill Color Index

status = vsfcol
(*handle*, *indin*)

status ≥ 0 fill index selected
= -1 an error has occurred

Set Fill Interior Style

status = vsfint
(*handle*, *stylein*)

status ≥ 0 style selected
= -1 an error has occurred

Set Fill Style Index

status = vsfstl
 (*handle, indin*)

status \geq 0 index selected
 = -1 an error has occurred

Set Graphic Color Burst Mode**

status = vsgcm
 (*handle, mode*)

status \geq 0 actual mode selected
 = -1 an error has occurred

Set Graphic Text Alignment

status = vstaln
 (*handle, horin, vertin, horout, vertot*)

Set Graphic Text Color Index

status = vstcol
 (*handle, indin*)

status \geq 0 index selected
 = -1 an error has occurred

Set Graphic Text Font

status = vsfstnt
 (*handle, fontin*)

status \geq 0 font type selected
 = -1 an error has occurred

Set Graphic Text String Baseline Rotation

status = vstrot
 (*handle, angin*)

status \geq 0 angle selected
 = -1 an error has occurred

Set Line Edit Characters

status = vsedch
 (*handle, lindel, chrdel*)

Data Types: character*1 lindel
 character*1 chrdel

Set Page**

status = vspace
 (*handle, grin, curin, grout, curout*)

Data Types: integer*4 grin(2)
 integer*4 curin(2)
 integer*4 grout(2)
 integer*4 curout(2)

Set Pen Speed**

status = vpspd
(*handle, speed*)

status \geq 0 pen speed selected
= -1 an error has occurred

Set Polyline Color Index

status = vsicol
(*handle, indin*)

status \geq 0 index selected
= -1 an error has occurred

Set Polyline Line Type

status = vsityp
(*handle, typin*)

status \geq 0 type selected
= -1 an error has occurred

Set Polyline Line Width

status = vslwid
(*handle, width*)

status \geq 0 width selected
= -1 an error has occurred

Set Polymarker Color Index

status = vsmcol
(*handle, indin*)

status \geq 0 index selected
= -1 an error has occurred

Set Polymarker Height

status = vsmhgt
(*handle, hgtin*)

status \geq 0 height selected
= -1 an error has occurred

Set Polymarker Type

status = vsmtyp
(*handle, typin*)

status \geq 0 type selected
= -1 an error has occurred

Set Writing Mode

status = vswrmd
(*handle*, *modein*)

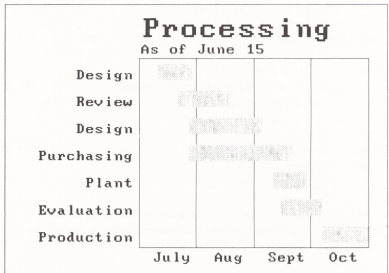
status \geq 0 actual mode selected
= -1 an error has occurred

Update Workstation

status = vupdwk
(*handle*)

Program Example

This program demonstrates how to create and display a Gantt chart. The output from this program should appear as follows:



PROGRAM GANTT

```
C *** This is a program to use the Graphics Development
C *** Toolkit FORTRAN binding to draw a Gantt chart
C ***

C *** every variable is an integer in this program
implicit integer*4 (a-z)
common xmax, ymax

dimension xy(10), savary(66)
dimension echoxy(2), workin(19)
dimension stdate(7), enddat(7)
character * 20 tasks(7), ticks(4), dummy
dimension taskn(7), tickn(4)

C *** create the device
C *** input array
data workin /0, 1, 1, 3, 1, 1, 1, 0, 0, 1, 1, 68,
+          73, 83, 80, 76, 65, 89, 32/

C *** the echo location
C *** for request work
data echoxy /0, 0/

C *** create the array of task names
data tasks /'Production', 'Evaluation',
+          'Plant', 'Purchasing',
+          'Design', 'Review', 'Design'/
data taskn /10, 10, 5, 10, 6, 6, 6/
```

```

C *** create the array of start dates for the bars
data stdate /83, 72, 70, 48, 48, 45, 40/

C *** do the same for end dates
data enddat /95, 83, 79, 75, 67, 60, 49/

C *** create the array of dates for the horizontal axis
data ticks /'July', 'Aug', 'Sept', 'Oct'/
data tickn /4, 3, 3, 3/

C *** open the workstation
status = vopnwk(workin,devhan,savary)
xmax = savary(52)
ymax = savary(53)

C *** set the constants for
C *** the grid
xy(2) = fytr(11)
xy(4) = fytr(81)
C *** for the points along the axis for grid lines,
C *** set the elements in the array for the grid line
C *** and draw the line
do 10 i = 51, 81, 15
  xy(1) = fxtr(i)
  xy(3) = xy(1)
  status = vpline(devhan,2,xy)
10 continue

C *** set text alignment to top center
status = vstaln(devhan,1,2,horout,vrtout)

C *** set the character height for the tick labels
status = vsthgt(devhan,fytr(5), xhgt, xwidth,
+               cwidth,chg)

C *** set an index into the array of months
j = 1

C *** for the labels on the horizontal axis
C *** write out a month
do 20 i = 44, 89, 15
  status = vgttext(devhan,fxtr(i),fytr(11),
+                 ticks(j)(1:tickn(j)))
  j = j + 1
20 continue

C *** set text alignment to middle right
status = vstaln(devhan,2,1,horout,vrtout)

C *** set an index into the arrays of tasks
j = 1

```

```

C *** for the points along the vertical axis
C *** write out the labels
do 30 i = 16, 76, 10
    status = vgtxt(devhan,fxtr(34),fytr(1),
+                tasks(j)(1:taskn(j)))
    j = j + 1
30 continue

C *** set text alignment back to the default lower left
status = vstaln(devhan,0,0,horout,vrtout)

C *** write out the subtitle to the chart
status = vgtxt(devhan,fxtr(36),fytr(83),
+            'As of June 15')

C *** set a new character height for the title
status = vstght(devhan,fytr(10),xhgt,xwidth,
+            cwidth,chggt)

C *** write out the title
status = vgtxt(devhan,fxtr(36),fytr(89),
+            'Processing')

C *** set the fill pattern to a 45 degree hatch
status = vsfstl(devhan,2)
status = vsfint(devhan,3)

C *** set an index into the arrays of dates
j = 1

C *** for the location of the bars
C *** create the array for drawing the bars;
C *** draw the bar
do 40 i = 13, 73, 10
    temp = i + 6
    xy(2) = fytr(i)
    xy(4) = fytr(temp)
    xy(1) = fxtr(stdate(j))
    xy(3) = fxtr(enddat(j))
    status = vbar(devhan,xy)
    j = j + 1
40 continue

C *** call routine to create box coordinates
call box(36,96,11,81,xy)

C *** draw a frame around the chart
status = vpline(devhan,5,xy)

C *** call routine to create box coordinates
call box(0,100,0,100,xy)

C *** draw a border around the page
status = vpline(devhan,5,xy)

C *** wait for the viewer before we close the workstation
status = vrqstr(devhan,2,0,echoxy,dummy)

```

```

C *** close down the workstation
      status = vclswk(devhan)
      stop
      end

      subroutine box(xmin,xmax,ymin,ymax,xy)
C *** a subroutine to create box coordinates from
C *** mins and maxs

C *** every variable is an integer
      implicit integer*4 (a-z)
      dimension xy(*)

C *** set values for all the corners
      xy(1) = fxtr(xmin)
      xy(2) = fytr(ymin)
      xy(3) = fxtr(xmax)
      xy(4) = xy(2)
      xy(5) = xy(3)
      xy(6) = fytr(ymax)
      xy(7) = xy(1)
      xy(8) = xy(6)
      xy(9) = xy(1)
      xy(10) = xy(2)
      return
      end

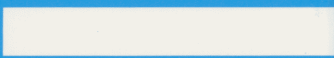
      integer*4 function fxtr(iper)
      implicit integer*4 (a-z)
      common xmax, ymax

C *** function that converts from percentages to
C *** NDC for the x axis
C
      fxtr = iper / 100.0 * xmax
      return
      end

      integer*4 function fytr(iper)
      implicit integer*4 (a-z)
      common xmax, ymax

C
C *** function that converts from percentages to
C *** NDC for the y axis
C
      fytr = iper / 100.0 * ymax
      return
      end

```



©IBM Corporation 1986
All rights reserved.

International Business
Machines Corporation
Dept. 997, Bldg. 998
11400 Burnet Rd.
Austin, Texas 78758

Printed in the
United States of America

59X8617