

CONVERGENT TECHNOLOGIES

RELEASE NOTICE FOR 3.1 VAM

Revised August 25, 1989

SECTION	TITLE	PAGE
1.0	Description of Programs	2
2.0	Changes from Prior Version	3
2.1	SPRs Closed in This Release	3
2.2	Changes from Release 3.0	3
3.0	Contents of Distribution Diskette(s)	5
4.0	Installation Procedures	6
4.1	Stand-Alone Hard Disk Systems	6
4.2	Master Hard Disk Systems	6
4.3	Stand-Alone Floppy Disk Systems	8
5.0	Required Files	8
6.0	System Software Compatibility	8
6.1	Workstation Environment	8
7.0	Hardware Information	9
7.1	Hardware Configurations Supported	9
7.2	Special Hardware Requirements	9
8.0	Resource Requirements/Utilization	10
8.1	Memory Requirements/Utilization	10
8.2	Disk Requirements/Utilization	10
9.0	Restrictions	11
10.0	Supplementary Information	11
10.1	Standard Character Map Support	11
10.2	Character Map Resolutions	11
11.0	Documentation Updates	12
12.0	Known Errors and Omissions	31

1.0 Description of Programs

This Release Notice describes VAM, the Video Access Method version 3.1. This section and the sections that follow provide a description of the product, contents of the Distribution Diskette, installation procedures, and other information pertinent to this release of VAM.

This product is the installed video service loaded by CTOS/VM to perform all VAM level video support. Since this includes support of graphics video hardware it also contains the RVAM services with a set of extensions to support new hardware features. It installs as any other CTOS server and serves System Common and Request level interfaces required by the video procedures listed in the CTOS Reference Manual. Additionally it serves the Debugger System Common procedures to provide generic debugger video. The product supports a wide variety of monitors as described in the VGA Hardware Specification.

This product is required for operation of the GC-x04 hardware on any 286 or higher NGEN product. It provides high quality monochrome and color graphics on any NGEN monitor (with the exception of the VC-001) or any VGA compatible off-the-shelf monitor. Additionally it provides VGA compatible video to PC-Emulator applications.

VAM provides procedural interfaces accessible to all CTOS application programmers allowing full use of the GC-x04 video hardware in character, graphics, and mixed modes. As in all VAM products, it provides enough information to the programmer to enable video independent CTOS applications to be written.

2.0 Changes from Prior Version

This is the first version of VAM fully separated from the CTOS/VM Operating System. Previous versions (3.0 and earlier) were bundled with the CTOS/VM software package.

This VAM release includes support for the GC-x04 Video hardware family, including both the GC-004 and the GC-104 graphics modules.

This product supports the standard VGA hardware interface as described in the VGA Technical Reference from IBM. The application level video interfaces are described in the CTOS/VM Reference. RVAM is generally interfaced to the application using one of two graphics libraries. These are described in the Graphics Programmer's Guide and the CCGI+ Programmer's Guide.

2.1 SPRs Closed in This Release

SPR 14248 The InitVidFrame call, when called to initialize a frame to 132 column width on a monitor only capable of 80 character width, returned an error code 0. It now returns error code 501.

SPR 14719 The QueryFrameCharsAndAttrs call would GP fault if requesting information for the entire virtual screen (iFrame = 0FFH). The failure occurred when requesting both text and attrs, just text, or just attrs.

2.2 Changes from Release 3.0

- A. This is the first version of VAM packaged separately from the CTOS/VM Operating System. Previous versions (3.0 and earlier) were bundled with the CTOS/VM software package.
- B. This VAM release includes support for the GC-x04 Video hardware family, including both the GC-004 and the GC-104 graphics modules.
- C. This product supports the standard VGA hardware interface as described in the VGA Technical Reference from IBM. The additional features of the VGA+ chip are described in the GC-004/GC-104 Graphics Controller Technical Reference Manual. The application level video interfaces are described in the CTOS/VM Reference. RVAM is generally interfaced to the application using one of two graphics libraries. These are described in the Graphics Programmer's Guide and the CCGI+ Programmer's Guide.

- D. This product provides NGEN compatible video on the GC-x04 hardware. It also provides the complete graphics interface with access to the hardware graphics of the VGA+ chip. Additionally this product supports a wider range of monitors and more resolutions and colors than was available on any previous NGEN graphics controller. It also allows VGA compatible PC-Emulator operation for higher speed PC video.
- E. The product provides full VGA support to the PC-Emulator.
- F. This product provides full support of the GHA hardware on the VGA+ chip.
- G. This product now supports the 720x348 character and graphics modes on the VM-003 monitor when attached to a GC-x04
- H. The GHA capability of the VGA+ provides greatly enhanced line drawing and BitBlt capabilities within the VGA memory space. This feature is used when operations are done on the display and its use is transparent to the user.
- I. Very slight aesthetic changes have been made to the font file [sys]<sys>1024x768_80Sys.font to improve the font appearance for the half-bright attribute on the VM-003.
- J. 3.1 VAM supports program access to all 256 possible VGA compatible graphics colors. Previously the Ctos.lib procedure *ProgramColorMapper* was very specific to the NGEN color hardware. Specifically it supported 3 palettes of 8 values each, with 1 byte for each value, representing 2 intensity bits for each *red*, *green* and *blue*. To support the GC-x04 color capabilities, the Computer Graphics Interface (CGI) model is used to provide access to more color entries and also to allow a device independent way to define colors. This model allows support for virtually any **mapped** color hardware. These enhancements are incorporated into the *ProgramColorMapper* call. This approach provides the necessary features without any new requests or added support machinery.

This new ProgramColorMapper procedural interface is available from the Ctos.lib on the target system if the optional Ctos library update is requested during VAM installation. It is documented in section 11.0 of this release notice.

3.0 Contents of Distribution Diskette(s)

The VAM Distribution Diskette is your master copy, and has been shipped write-protected. It should not be write-enabled, nor should it be used as a working copy.

The Distribution Diskette contains the following files in the <CT> directory:

InstallVdm.run	1024x768_146LocSys.Icon
Vdm_Bm.run	1024x768_146locsys.icon-box
Vdm_Ch.run	1024x768_146sys.font
Vdm_VGA.run	1024x768_80LocSys.Icon
DisplayConfigMsg.Bin	1024x768_80locsys.icon-box
DisplayConfigMsg.Txt	1024x768_80Sys.font
t1Sys.font	132x29_CharSys.font
Vdm>Config.Sys	132x34_CharSys.font
720x348_80Sys.font	720x348_80LocSys.Icon
80x34_CharSys.font	720x348_80locsys.icon-box

In addition to various standard system files, the Distribution Diskette contains the following files in the <Sys> directory:

HdInstall.sub	QueryVideo.Obj
Color.Obj	Request.Vam.Sys
Delete.flS	UpdateCTOSLib.Sub
Legalese	VAMRqLabl.Obj
Obj.flS	

4.0 Installation Procedures

VAM should only be used on Convergent Information Processing Systems which are equipped with hard disk (or dual floppy disks), or in clusters where the master is equipped with hard disk.

Note: If updating on existing systems, VAM should be installed prior to attaching the GC-x04 Graphics Controller to the workstation.

Use the installation procedures described below (after updating the Standard Software revision level, if required). Characters that you must type are shown in **boldface**. Special keys, such as RETURN and GO, are shown in upper case.

Note: This installation optionally updates (by performing a Librarian addition) the 11.3 Ctos.lib on the target system to include 3.1 VAM-consistent programmatic interface. The user is requested, following the software installation, to perform this step.

4.1 Stand-Alone Hard Disk Systems

- A. Signon and set path at the standalone workstation. If the Signon form is displayed, fill it in and press GO. Set the path as follows:

```
Command Path RETURN
Path
  [Volume]                Sys RETURN
  [Directory]             Sys RETURN
  [Default file prefix]   RETURN
  [Password]              (if any) GO
  [Node]
```

If your hard disk has a volume password on [Sys], fill this password into the [Password] field before pressing GO.

- B. Type the Install command as follows:

```
Command Install RETURN
Install
  [Floppy (default = [f0])] GO
```

- C. Respond to the instructions to change either the Floppy drive or the path in which to install as appropriate.
- D. Reboot your workstation when you are instructed to do so.

4.2 Master Hard Disk Systems

- A. Signon and set path at the master workstation. If the Signon form is displayed, fill it in and press GO. Set the path as follows:

```
Command Path RETURN
Path
  [Volume] Sys RETURN
  [Directory] Sys RETURN
  [Default file prefix] RETURN
  [Password] (if any) GO
  [Node]
```

If your hard disk has a volume password on [Sys], fill this password into the [Password] field before pressing GO.

- B. Type the Install command as follows:

```
Command Install RETURN
Install
  [Floppy (default = [f0])] GO
```

- C. Respond to the instructions to change either the Floppy drive or the path in which to install as appropriate.
- D. Reboot your workstation when you are instructed to do so.

4.3 Stand-Alone Floppy Disk Systems

- A. Signon and set path at the standalone workstation. If the Signon form is displayed, fill it in and press GO. Set the path as follows:

```
Command Path RETURN
Path
  [Volume]                Sys RETURN
  [Directory]             Sys RETURN
  [Default file prefix]  RETURN
  [Password]              (if any) GO
  [Node]
```

If your hard disk has a volume password on [Sys], fill this password into the [Password] field before pressing GO.

- B. Type the Install command as follows:

```
Command Install RETURN
Install
  [Floppy (default = [f0])] GO
```

- C. Respond to the instructions to change either the Floppy drive or the path in which to install as appropriate.
- D. Reboot your workstation when you are instructed to do so.

5.0 Required Files

The files contained on the release diskette are not all required for operation. They are required as follows:

A bootable hard or floppy disk must contain the file InstallVdm.run.

Additionally, bootable disks on workstations using character mapped video must contain the file Vdm_Ch.run.

Bootable disks on workstations using bit mapped video must contain the file Vdm_BM.run.

Bootable disks on workstations using VGA Compatible video must contain the file Vdm_VGA.run.

6.0 System Software Compatibility

6.1 Workstation Environment

This product requires CTOS/VM 2.2 or above to operate. It is compatible with 2.3 Context Manager/VM, 12.1 Graphics Library, 2.3 Mouse Server, 4.0 PC Emulator, 2.1 Art/Chart Designer, and 2.0 CCGI+ Library.

7.0 Hardware Information

To achieve full functionality, this product requires the GC-004 or GC-104 hardware running on a 286, 286i, 386 or 386i NGEN Workstation. This 3.1 VAM will also work correctly on the GC-001, GC-x03, and GC-102 Graphics Controllers when used with a compatible CPU product.

7.1 Hardware Configurations Supported

7.2 Special Hardware Requirements

The GC-x04 supports a variety of monitors when the system is running CTOS. The table below shows which monitors support which resolutions through VAM access. More resolutions (full VGA compatibility) are supported under PC Emulation. A (*) denotes the default mode for the monitor (its initial mode at system boot time).

<u>Monitor</u>	<u>Resolution</u>	<u>Comments</u>
VM-001/2	720x348x4	* 80x29 (80x34)
	924x348x4	132 col. true char map (132x29, 132x34)
VM-004	720x348x4	80x29 only
	720x348x8	
VC-002	720x348x4	* 80x29 (80x34)
	924x348x4	132 col. true char map (132x29, 132x34)
VM-003	1024x768x2	*80x38, true char map
	1024x768x2	146x38, emulated characters
	720x348x2	80x29 only
VGA mono	720x348x4	* 80x29 (80x34)
	720x348x8	gray scale, emulated characters (80x29)
	924x348x4	132 col. true char map (132x29, 132x34)
		(this includes the Unisys VA1 monitor)

VGA Color	720x348x4	* (see VGA mono)
	720x348x8	256 color, emulated characters
	924x348x4	132 column true char map
Multi-Sync	720x348x4	* (see VGA mono)
	720x348x8	256 color, emulated characters
	924x348x4	132 column true char map
8514	720x348x4	(see VGA mono)
	1024x768x4	*80x38, true char map
	1024x768x4	146x38, emulated characters
	720x348x8	256 color, emulated characters (this includes the Unisys CA1 and GS1 monitors)

Note that the 1024x768x4 modes show 2 entries. In the 80x38 character mode, overlapping characters and graphics are supported, giving a true character map. In the 146x38 mode the display is full graphics with characters emulated in the bitmap.

8.0 Resource Requirements/Utilization

8.1 Memory Requirements/Utilization

The 3.1 VAM VGA compatible run file (vdm_VGA.run) occupies 154K bytes of memory.

8.2 Disk Requirements/Utilization

The 3.1 VAM installation uses an additional 230 sectors (115 K bytes) for the vdm_vga run file. All other files already exist on the target system and are replaced by run files or configuration files of similar size.

9.0 Restrictions

The VM-003 monitor has no half-bright capability so the hardware dithering feature of the VGA+ is used to emulate it in the 80x38 character mode. Dithering removes every other line of the font to produce a half bright effect.

GC-001 color graphics is supported procedurally by Graphics.lib 12.0.1, but Graphics.lib version 12.1 is required to gain access to any more than the first 8 graphics colors. RVAM supports all 256 possible graphics colors.

10.0 Supplementary Information

10.1 Standard Character Map Support

The VGA+ fully emulates the NGEN character map. It directly emulates the placement of characters in the map as well as the display of attributes and support of the Font Ram. As long as the procedural interface is used (*LoadFontRam*, *PosFrameCursor*, *LockVideo/UnlockVideo* and the VAM calls), and the *rgpVidMemLine* structure is used for direct map access this mode is supported in most cases.

The uses of *rgpVidMemLine* is not supported under 2 conditions. The first is when an application tries to use another memory-mapped X-Bus module with the GC-x04. This problem has never been experienced in practice. It would be rare to have this come up since most X-Bus modules are controlled by servers and servers can't use video.

The other condition under which *rgpVidMemLine* cannot be used is in those display modes where the character map must be emulated in the bitmap. This occurs on the VM-003 monitor in the 146x38 mode only and in all modes when using more than 4 bit-per-pixel graphics. This second case causes no problem since all applications initialize with a maximum of 4 bits-per-pixel. The only way to increase this is through use of the *ResetVideoGraphics*, a new procedure.

The existing system fonts are used on all character map displays, where the The 9x12 characters use *t1sys.font* and the alternate EV resolutions (80x34, 132x29, 132x34) use the fonts created for that device. All other resolutions use raster fonts.

10.2 Character Map Resolutions

All current monitors are supported. The VM-001/2 is supported with a true character map in 80x29, 80x34, 132x29 and 132x34. The VC-002 is supported with a true character map in 80x29 (both EV and GC-001 color). VM-003 is supported in 80x38, 146x38 and 80x29. Due to a hardware limitation the 146x38 character mode is emulated in the bitmap as on the GC-003 with 3.1 VAM. The 80x29 mode on the VM-003 uses *Tlsys.font*. The hardware actually requires a 9x24 character cell in this mode, so the font height is doubled using pixel replication upon load. This font manipulation is user transparent in the *LoadFontRam* call.

11.0 Documentation Updates

The programmatic interface for this product is described in the CTOS/VM Reference and in the Raster Video Access Method (RVAM) Specification.

- A. `QueryFrameCharsAndAttrs` returns `Erc 500` when requesting information for the entire real screen (`iFrame=0FEH`). The current CTOS/VM Reference Manual is incorrect.
- B. When using a GCx04 and a VM003 monitor, character appearance is aesthetically unsatisfactory for the following collection of simultaneous attributes: Reverse Video, Blinking, and Half-Bright. This attribute combination is not recommended on this monitor/controller combination.
- C. The GC-x04-1 (256K version) cannot be used for hi-resolution mode (1024x768x4) with off the shelf monitors including the 8514, the GS1 and the CS1. This is due to memory limitations of the 256K version. When the above monitors are attached they default to 720x348x4.
- D. On a GC-x04 with a VM-003, a graphics application performing a `SetScreenVidAttr` call to set reverse video clears graphics and redisplay the screen.
- E. On a GC-x04 (revision 4 ASIC chip) with a VM-003 monitor, the struck-through attribute is missing when the half-bright attribute is also set. This is because half-bright on that monitor is accomplished using a dithering method which interferes with the pixel row necessary for display of the struck-through attribute.
- F. `Erc 7654` is returned when doing Screen Setup `<GO>` from a hi-resolution Executive if the file `1024x768_146LocSys.icon` is missing from the `[sys]<sys>` directory.

The following documentation updates the CTOS/VM Reference Manual.

ProgramColorMapper

ProgramColorMapper(pNewPalette, sNewPalette, pNewControl, sNewControl, pOldPaletteRet, sOldPaletteRet, pOldControlRet, sOldControlRet):ercType

Caution: *This operation works on workstation hardware only. Do not use it on Shared Resource Processor (SRP) hardware.*

Description

ProgramColorMapper sets and queries the color palette and/or color control structure. Through the ProgramColorMapper interface, the caller can enable alphanumeric and/or graphics displays and can define customized colors in the color palette(s) for the displays. The caller can choose to set up color using either a single-palette or three-palette format. The single palette format allows selection from a greater range of colors and color intensities on hardware that supports a large number of colors and intensities.

(The parameter descriptions below assume that you are familiar with the ProgramColorMapper structures. For details on the formats of the palette and control structures and examples of how to use ProgramColorMapper, see “Using Color” in the *CTOS Programmer’s Guide*.)

Using this operation causes ASCB.fExecScreen to be set to 0.

Procedural Interface

ProgramColorMapper (*pNewPalette*, *sNewPalette*, *pNewControl*, *sNewControl*, *pOldPaletteRet*, *sOldPaletteRet*, *pOldControlRet*, *sOldControlRet*): *ercType*

where

pNewPalette
sNewPalette

describe the new palette.

If the new control structure described by *pNewControl* and *sNewControl* defines the three-palette format (the control structure field *bFormat* is 0 or *sNewControl* is less than 6), *sNewPalette* may have one of the following sizes:

Size	Meaning
0	No palette change.
8	Replace the alpha palette.
16	Replace the alpha and graphics1 palettes.
24	Replace all three palettes.

If the new control structure defines the single-palette format (the control structure field *bFormat* is 1 and *sNewControl* is greater than 5), *sNewPalette* may have one of the following sizes:

Size	Meaning
0	No palette change.
3*x	Replace x of the 3-byte entries in the palette starting at the palette entry specified by the value of <i>wIndexStart</i> in the control structure.

pNewControl
sNewControl

describe the new control structure. *sNewControl* may be one of the following values:

Size	Meaning
0	No change
5	Three-palette format
8	Single-palette format if the field <i>bFormat</i> is 1

pOldPaletteRet
sOldPaletteRet

describe the memory area where a copy of the current palette will be returned. If *sOldPaletteRet* is 0, no current palette is returned. The operation works correctly even if *pOldPaletteRet* is the same as *pNewPalette*.

pOldControlRet
sOldControlRet

describe the memory area where a copy of the current control structure will be returned. If *sOldControlRet* is 0, no control information is returned.

Request Block

ProgramColorMapper is an object module procedure.

ProgramColorMapperRq has no procedural interface. You must make a request block and issue the request using Request, Request Direct, or RequestRemote.

Caution: *This operation is for internal use only and is subject to change in future operating system releases. It is for use on workstation hardware only. Do not use it on Shared Resource Processor (SRP) hardware.*

Description

ProgramColorMapperRq sets and queries the palette and/or control structure. The request is called by the object module procedure, ProgramColorMapper. If an error code is returned, ProgramColorMapperRq executes linked-in code to change the hardware. This allowed ProgramColorMapperRq to be served by the Context Manager to perform context-specific color operations.

With current releases of the operating system, this request is served by installable VAM. (See Appendix B, "Operating System Features," in *CTOS/VM Concepts* for a list of the operating system versions that support installable VAM.) The request is filtered by the Context Manager to installable VAM to perform the hardware-specific operations, while the Context Manager performs the context-specific ones. As a result, ProgramColorMapperRq may return error code 33 ("Service not installed").

For a description of the parameters to ProgramColorMapperRq, see the description of the corresponding library procedure, ProgramColorMapper in this chapter.

Procedural Interface

ProgramColorMapperRq has no procedural interface. You must make a request block and issue the request using one of the request operations listed above

where

pNewPalette
sNewPalette

See ProgramColorMapper.

pNewControl
sNewControl

See ProgramColorMapper.

pOldPaletteRet
sOldPaletteRet

See ProgramColorMapper.

pOldControlRet
sOldControlRet

See ProgramColorMapper.

Request Block

Offset	Field	Size (Bytes)	Contents
0	sCntInfo	1	0
1	rtCode	1	0
2	nReqPbCb	1	2
3	nRespPbCb	1	2
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	287
12	pNewPalette	4	
16	sNewPalette	2	
18	pNewControl	4	
22	sNewControl	2	
24	pOldPaletteRet	4	
28	sOldPaletteRet	2	
30	pOldControlRet	4	
34	sOldControlRet	2	

QueryVidHdw (pBuffer, sBuffer): ercType

Caution: *This operation works on workstation hardware only. Do not use it on Shared Resource Processor (SRP) hardware.*

Description

QueryVidHdw places information describing the level of video capability of the workstation in the specified memory area. When writing software that must work on several workstation models, use QueryVidHdw to determine the level of video capability present before calling the Reset Video operation.

QueryVidHdw is similar to Query Video. Both operations return video capability information in the same format described below. QueryVidHdw, however, fills in certain fields only for the specified workstation operating system version. Query Video fills in all fields. (See Query Video for more information.)

The fields that are filled in by QueryVidHdw are listed following the video format below.

The format of the video information returned is as follows:

Offset	Field	Size (Bytes)	Contents
0	level	1	Level of video capability: 0 = IWS 1 = reserved 2 = AWS 3 = reserved 4 = NGEN with character map 6 = bit map workstations
1	nLinesMax	1	Maximum number of lines.
2	nColsNarrow	1	Number of columns in narrow mode
3	nColsWide	1	Number of columns in wide mode (for example, 132 for workstations with 132 column capability and 146 for workstations with 146 column capability).
4	graphicsVersion	1	Level of bit map capability: 0 = none 1 = for IWS Graphics workstations 2 = for monochrome graphics (AWS or GC-001) 3 = for Color Graphics workstations (AWS or GC-001)

Offset	Field	Size (Bytes)	Contents
			5 = GC-003 with VM-003
			6 = GC-003 with VM-001 or VM-002
5	nPixelsHigh	2	Number of pixels high for the graphics version of the bit map.
7	nPixelsWide	2	Number of pixels wide for the graphics version of the bit map.
9	saGraphicsBoard	2	Segment address of the memory segment that is assigned to the bit map.
11	ioPort	2	Input/output port of the Graphics Board memory.
13	wBytesPerLine	2	Number of bytes in one raster line for a bit map video display.
15	nCharHeight	1	Character cell height in pixels.
16	nCharWidthNarrow	1	Character cell width in pixels in narrow mode.
17	nCharWidthWide	1	Character cell width in pixels in wide mode.
18	pBitmap	4	Memory address of the bit map structure. (See below for the format of the bit map structure.)
22	pFont	4	Memory address of the font currently being used by the video display. The font may be a character font on a character map workstation without graphics or a raster font on a workstation with a bit map display. (See the <i>Graphics Programmer's Guide</i> for more information on raster fonts.)

Offset	Field	Size (Bytes)	Contents
26	bModuleType	1	Number that is assigned to the module controlling the video display (for the X-Bus).
27	bModulePos	1	Number that specifies the position of the video display controlling the X-Bus module (for example, 1 indicates the first module to the right of the workstation).
28	wModuleEar	2	I/O port address that provides access to the video display controlling module.
30	nYCenter	2	Offset of the first character cell in a video display from the top of the display.
32	nXCenterNarrow	2	Offset of the first character cell in a video display from the left side of the display in narrow mode.
34	nXCenterWide	2	Offset of the first character cell in a video display from the left side of the display in wide mode.
36	wxAspect	2	ratio between x and y
	wyAspect	2	physical screen dimensions. One value is 32768. The other value is less than 32768.
40	nPlanes	1	Number of graphics planes.
41	fColorMonitor	1	Flag. If TRUE, the color monitor is supported.
43	nColors	2	Number of colors supported.
44	fBackgroundColor	1	Flag. TRUE means background color capability.

Offset	Field	Size (Bytes)	Contents
45	fHardwareCharMap	1	Flag. TRUE means a character-mapped workstation.
47	nAlternateLinesMax	2	Maximum number of lines for monitors capable of changing the number of lines. (For example, workstations with Enhanced Video (EV) can change the number of lines from 29 to 34.)
49	nAlternateCharHeight	2	Character cell height in pixels for workstations with the ability to change character height. (For example, workstations with EV have this ability.)
51	wVidRelease	2	Video software release number.
53	wVidVersion	2	Video software internal version number.
55	fGenericColor	1	Flag. TRUE means the hardware supports the single-palette format returned by the ProgramColorMapper operation. (For details, see "Using Color" in the <i>CTOS Programmer's Guide</i> .)
56	fVariableCharMap	1	Flag. TRUE means the hardware reformats (changes the size of) the character map as the mode changes.
57	fMultiSystemFonts	1	Flag. TRUE means more than one system font is supported.
58	fUseWsLine	1	Flag. TRUE means use wsLine in the Video Control Block (VCB). FALSE means use sLine instead. (See "Video Control Block" in the Chapter 4, "System Structures.")
59	wLevelsPerPrimary	2	Number of color levels per primary color (red, green, or blue).
61	nExtraRasterLines	2	Number of raster lines below the display bit map.

The following fields (indicated by their offsets) are filled in by QueryVidHdw for each operating system version:

Offsets	Operating System Version
0-11	CTOS 9.9 and prior operating system versions
0-38	CTOS II 1.0
0-44	CTOS/VM 2.0
45-54	CTOS/VM 2.2
55-61	CTOS/VM 3.0

(For details on release numbers, see CurrentOsVersion.)

The format of the Bitmap field in the video structure above is as follows:

Offset	Field	Size (Bytes)	Contents
0	cbLineBm	2	Same as sBytesPerLine.
2	nWidthBm	2	Same as nPixelsWide.
4	nHeightBm	2	Same as nPixelsHigh.
6	fScreenBm	1	Flag that is always TRUE.
7	bRegBase	1	Same as upper byte of ioPort.
8	bPlanes	1	Number of planes in bitmap.
9	bSegs	1	Always 1.
10	wSegHeight	2	Same as nHeightBm.
12	pBm	1	Physical bitmap address.

Request Block

Offset	Field	Size (Bytes)	Contents
0	sCntInfo	1	6
1	RtCode	1	0
2	nReqPbCb	1	0
3	nRespPbCb	1	1
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	21
12	reserved	6	
18	pBuffer	4	
22	sBuffer	2	

ResetVideoGraphics (nCols, nLines, fAttr, bSpace, psMapRet, nPixelsWide, nPixelsHigh, nPlanes, fBackgroundColor): ercType

Caution: *This operation works on workstation hardware only. Do not use it on Shared Resource Processor (SRP) hardware.*

Description

ResetVideoGraphics functions in the same way as the ResetVideo operation but allows the caller more options. (See the description of ResetVideo.) With ResetVideoGraphics the caller can, in addition, set the graphics bit map resolution, the number of planes for color, and the background color mode on workstations with VGA capability.

NOTE: When using ResetVideoGraphics to change the column format of the character map or the resolution of the bit map, your program also must call PdSetVirtualCoordinates or PdSetCharMapVirtualCoordinates to reset the virtual coordinates of the pointing device. For details on these operations, see the Mouse Services Manual.

The values you specify for *nCols*, *nLines*, *nPixelsWide*, and *nPixelsHigh* are used by this operation to select a display mode that can minimally support them and will be at least the sizes specified. If this cannot be done, however, error code 501 ("Invalid argument to VDM") is returned. Any value can be 0, in which case the value will not be used to decide on a mode. If all values are 0, a default will be chosen that is the same as the initial mode of the monitor at boot time.

Following are examples:

nCols	80
nLines	29
nPixelsWide	0
nPixelsHigh	0
nPlanes	0

With the above values, ResetVideoGraphics selects a monitor resolution of 80 columns by 29 lines and an appropriate width and height in pixels for the graphics resolution. In addition, the operation, selects an appropriate number of planes (because 0 is specified for *nPlanes*.)

nCols	0
nLines	0
nPixelsWide	720
nPixelsHigh	348
nPlanes	4

With the above values, ResetVideoGraphics selects an appropriate monitor resolution, 720 pixels wide by 348 pixels high for the graphics resolution, and 4 planes. If you specified a graphics resolution greater than that supported by the monitor (such as 1024 by 768 pixels for a monitor that only supports 720 x 348), error code 501 is returned. If, however, you specified a graphics resolution less than the highest resolution supported by the monitor (such as 720 x 348 pixels for a monitor that supports 1024 x 768), ResetVideoGraphics will accept the lower pixel values.

nCols	80
nLines	29
nPixelsWide	1024
nPixelsHigh	348
nPlanes	4

With the above values, ResetVideoGraphics selects a mode that is *at least* 80 columns by 29 lines and 1024 pixels wide by 348 pixels high. Say, for example, the monitor supported the following two modes:

80 x 29 and 720 x 348

and

80 x 38 and 1024 x 768

In this case, ResetVideoGraphics would select the latter mode.

Procedural Interface

ResetVideoGraphics (*nCols*, *nLines*, *fAttr*, *bSpace*, *psMapRet*, *nPixelsWide*, *nPixelsHigh*, *nPlanes*, *fBackgroundColor*): *ercType*

where

nCols

specifies the number of characters per line. (See QueryVidHdw.) If the value of *nCols* is 0, this parameter is ignored, and any character width that can accommodate the resolution can be chosen.

nLines

specifies the number of lines per screen. (See QueryVidHdw.) If the value of *nLines* is 0, this parameter is ignored, and any character height that can accommodate the resolution can be chosen.

fAttr

is a flag that is TRUE if the character map is to include character attributes.

bSpace

specifies a character code that is blank in the font. This is used when the character map is initialized by the `InitCharMap`, `ResetFrame`, or `ScrollFrame` operation. (See the chapter entitled "Video" in *CTOS/VM Concepts* for more information.)

psMapRet

is the memory address of the word to which the required size of the character map is returned.

nPixelsWide

is the number of pixels wide for the bit map. If the value of *nPixelsWide* is 0, this parameter is ignored, and any horizontal resolution that can accommodate the character map can be chosen.

nPixelsHigh

is the number of pixels high for the bit map. If the value of *nPixelsHigh* is 0, this parameter is ignored, and any vertical resolution that can accommodate the character map can be chosen.

nPlanes

is the number of planes used for color graphics. If the value of *nPlanes* is 0, this parameter is ignored, and an appropriate number of planes is chosen for this resolution.

fBackgroundColor

is a flag that sets the background color mode. If *fBackgroundColor* is FALSE, all character backgrounds are color index 0. If *fBackgroundColor* is TRUE, character backgrounds are indexes 0 through 7, matching the value of the color bits in the attribute byte. (For details on color programming, see "Using Color" in the *CTOS Programmer's Guide*.)

Request Block

ssMapRet is always 2.

Offset	Field	Size (Bytes)	Contents
0	sCntInfo	1	10
1	RtCode	1	0
2	nReqPbCb	1	1
3	nRespPbCb	1	
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	33137
12	nCols	1	
13	nLines	1	
14	fAttr	1	
15	bSpace	1	
16	nPixelsWide	2	
18	nPixelsHigh	2	
20	nPlanes	1	
21	fBackgroundColor	1	
22	psMapRet	4	
26	ssMapRet	2	2

12.0 Known Errors and Omissions

- A. Several problems exist in color support in the version 11.3 Executive. Foreground color cannot be set from the Video and Screen Setup commands on analog monitors connected to GC4 in high resolution; applications run from the exec on analog mono monitors (vga) that change the color (gray scale) may cause the exec when it is again chained, to run in the color left by the application. Background color cannot be set. These problems will be corrected in a general release, version 12.0 Standard Software.

Note: An Executive corrected for most of these color support problems is available as a special version, s11.3.5 Exec.run. Contact technical support for more information. Using this version, background color may only be set on Extended Video systems (CP-0A2 and CP-0A3).

- B. When calling ScrollFrame, if the startline is greater than or equal to the lastline, unpredictable results will ensue (SPR 12478).
- C. The QueryFrameCharsAndAttrs call returns error code 500 when requesting information for the real screen, iFrame = 0FEH (SPR 13018).
- D. The [sys]<sys>Config.sys parameters in the :BMAAttr: field are ignored on a 286i CPU with a VM-001 monitor (SPR 15036).
- E. VidinfoType structure fields bModuleType and bModulePos are zero for GC-102.
- F. Reverse video does not work when used on a GC-x04 with a revision 3 ASIC chip using a VM-003.
- Note: To determine if you have such a module, inspect the bottom of the unit for its serial/model number label. Adjacent to the model number, there will be either the letters "AA" or "AB". Only those labeled AA are affected.**
- G. ResetFrame, when called with an illegal frame returns ErcOk.
- H. LoadFontRam, when called with a work area which is not a multiple of 512 Bytes returns ErcOk.
- I. QueryVidHdw returns saGraphicsBoard = 0 when called on GC-X04 hardware.
- J. Using SetStyleRamEntry to change bVal causes horizontal lines to flash across the screen momentarily.

- K. When the mouse cursor is turned off using ResetVideoGraphics to change from hi-resolution (1024x768) to lower resolution (720x348x4), a mouse cursor may remain in the upper left corner of the screen. This occurs on a GC-104 with an 8514 monitor. It has also been observed using Image Designer when the mouse is moved very quickly immediately following a zoom command.
- L. Pressing the ACTION-OVERTYPE key on a GC-x04 using either a GS1 or VM003 monitor causes the screen to revert to the default background (all white instead of all black).
- M. Immediately after booting a system with a GC-x04 and a VM-003 monitor, if the user logs on without Context Manager and executes a Screen Setup command specifying yes to Reverse Video and 80 columns, there is a vertical white band on both sides of the screen.
- N. Extended Multiplan does not run in color when using a GC-x04 with a color monitor. This is an Extended Multiplan problem.
- O. 2.2 Document Designer does not work in WYSIWYG on the GC-x04. This problem will be corrected in version 2.3 Document Designer. It has been corrected in a special version, s2.2.1 Document Designer. **Contact technical support for more information.**
- O. The ^Z Zoom feature of Document Designer is not supported with VAM 3.1 when using a GC-x04 and Document Designer 2.2. This problem will be corrected in version 2.3 Document Designer.
- P. Low resolution images do not display in Document Designer 2.2 when using VGA monitors. This problem will be corrected in version 2.3 Document Designer.
- Q. On a GC-x04 with a low res monitor, if the Exec is set to 132 column mode and an application switches to 80 column mode, the Exec is not reset to 132 column mode. This problem will be corrected in version 12.0 Executive
- R. On a GC-x04 with a Revision 3 ASIC chip, running with a VM-003, the debugger is invisible. Typing an Action-A, you will not see the debugger, however, you must press GO before regaining control of the screen/keyboard. To determine if you have such a graphics controller, see item F above.