RETOS Installation and User's Guide

AA–JU56A–TE

digital software

RETOS Installation and User's Guide

Order Number: AA–JU56A–TE

This manual describes how to install and use RETOS, the ReGIS to sixel graphics converter.

This is a new manual.

Operating System and Version: VAX/VMS, Version 4.4 or later

Software Version: RETOS, Version 1.0

digital equipment corporation
maynard, massachusetts

First Printing, July 1987

The READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

This document was prepared using an in-house documentation production system. All page composition and make-up was performed by TEX, the typesetting system developed by Donald E. Knuth at Stanford University. TEX is a trademark of the American Mathematical Society.

Contents

FIGURES

TABLES

Preface

This manual describes how to use RETOS, the ReGIS to sixel graphics converter. RETOS changes ReGIS files to sixel files for printing on sixel printers. The text also explains how to install RETOS and provides information on ReGIS (Remote Graphics Instruction Set) and sixel graphics.

Structure of This Document

The *RETOS Installation and User's Guide* has four chapters and three appendixes:

- Chapter 1 presents an overview of RETOS, the ReGIS to sixel graphics converter, instructions for installing the converter, and a sample installation.
- Chapter 2 explains how to use RETOS by describing the command syntax, including supported qualifiers, converter default values, and error messages.
- Chapter 3 describes ReGIS to sixel graphics conversion, including ReGIS commands, the handling of ANSI text and escape sequences, screen addressing, and color mapping, and shows examples of the ReGIS to sixel graphics conversion.
- Chapter 4 provides information on the sixel graphics protocol and the sixel output.
- Appendix A identifies the character sets and fonts that RETOS supports.
- Appendix B compares RETOS with other ReGIS devices.
- Appendix C shows sixel graphics printable dot patterns.

Intended Audience

The *RETOS Installation and User's Guide* is intended for:

- Users who produce Remote Graphics Instruction Set (ReGIS) files that require conversion to sixel graphics for printing on sixel printers
- Programmers who write applications that produce ReGIS files for printing on sixel printers

Conventions Used in This Document

The following conventions are used throughout this document:

Convention	Meaning
Uppercase notation	Type the word or letter exactly as shown.
Lowercase notation	Substitute a word or value of your choice.
< >	Indicates, in ReGIS graphics, that you must supply a specific value. The <values> in the angle brackets define the type of information you can use, but the brackets are not part of the ReGIS syntax. In W(M <n>) you supply a numeric value, W(M1) or W(M2), for example.
...	Indicates that the preceding item(s) can be repeated one or more times.
RET	Indicates that you should press the RETURN key.
CTRL/X	Indicates that, while pressing the key labeled CTRL, you simultaneously press another key, for example, CTRL/Z, CTRL/C, CTRL/O.
$ help retos	Indicates a DCL command. In this example, the system prints the "$" (default command prompt) and you type the DCL command "help retos".

Associated Documents

Further information about ReGIS, the sixel graphics protocol, and VMS installation procedures may be obtained from the following:

- *Guide to VAX/VMS Software Installation*
- *Guide to VAX/VMS System Management and Daily Operations*
- *VAX/VMS System Manager's Reference Manual* [1]
- *VT240 Series Programmer Reference Manual*
- *VAX/VMS System Messages and Recovery Procedures Reference Manual*

[1] As of VMS Version 4.4, this is the new title for *Guide to VAX/VMS System Management and Daily Operations.*

Chapter 1

RETOS Installation

RETOS, the ReGIS to sixel graphics converter, allows the printing of ReGIS graphics on sixel printers or terminals. ReGIS graphics created on a terminal can be run through the converter and changed into sixel graphics, then printed on your sixel printer. Following is a list of applications that produce ReGIS output:

- BASEVIEW
- DATATRIEVE
- DECgraph
- DECslide

The converter has command qualifiers that allow generation of output files for printing on sixel printers and video terminals. The following output devices are supported: LJ250, LJ252, LN03 PLUS, LN03, LA210, LA100, LA75, and LA50 printers, and VT340, VT330, VT241, and VT240 terminals.

NOTE

For the PRINTSERVER 40 and the LN03R SCRIPTPRINTER, DIGITAL recommends that you print ReGIS files with /PARAMETERS=(DATA_TYPE=REGIS) rather than converting them with RETOS and printing them with /PARAMETERS=(DATA_TYPE=ANSI).

The following sections contain instructions for installing RETOS on the VAX/VMS and MicroVMS operating systems. They also explain how to read the online release notes before installing the product. This document applies to RETOS Version 1.0 and all point releases in that version.

Keep this document with your distribution kit. You will need to refer to it again should you receive any maintenance updates or need to reinstall RETOS for any other reason.

The installation procedure is described step by step. For the most part, the dialog is self-explanatory. The system prompts you to answer questions and change volumes and waits for you to tell it to continue.

At the back of this document is a READER'S COMMENTS form. If you have any comments on this manual, please fill out the form and mail it.

For information on submitting Software Performance Reports (SPRs), see Section 1.12.

This document does not contain release notes information; instead, release notes are provided on line. You can read the release notes before you install RETOS and at any time after the product is installed.

NOTE

To gain access to the release notes before RETOS is installed, follow the installation procedure up to step 3, in which you may choose to read or print the release notes and discontinue the installation procedure.

To read the release notes after installation, type or print the file SYS$HELP:RETOSnnn.RELEASE_NOTES, where nnn is the version number, for example, 010 for Version 1.0.

The Help file also directs you to the release notes file. After the installation, simply type:

```
$ HELP RETOS RELEASE_NOTES
```

For more information on installing a product on VAX/VMS, see the *Guide to VAX/VMS Software Installation* and the *Guide to VAX/VMS System Management and Daily Operations.*

NOTE

As of VMS Version 4.4, *Guide to VAX/VMS System Management and Daily Operations* is retitled to *VAX/VMS System Manager's Reference Manual.*

1.1 Hardware Requirements

Hardware requirements for the RETOS installation are:

- A VAX operating system with one of the following for installation of the software:
  - 1600 BPI Magnetic Tape
  - RX50 Disk Drive
  - TK50 Tape Drive
- A MicroVax I, MicroVax II, VAXstation I, or VAXstation II system with:
  - RX50 Disk Drive or TK50 Tape Drive (software installation)
  - RD52, RD53, or RD54 System Disk

A graphics terminal or a printer is NOT required to install or run the software.

1.2 Software Requirements

The only software requirement for RETOS installation is that RETOS Version 1.0 requires VAX/VMS Version 4.4 or later or MicroVMS Version 4.4 or later. The installation procedure uses the VMSINSTAL utility.

1.3 Preparing to Install RETOS

Your bill of materials specifies the number and contents of your media. Be sure to check the contents of your distribution kit against the list in the RETOS Bill of Materials. The kit should include the following:

- Software labeled VAX/VMS RETOS, Version 1.0, in one of the above listed media forms (See Section 1.1.)
- *RETOS Installation and User's Guide*

To mount the distribution medium, see the *Guide to VAX/VMS Software Installation* pertaining to your system. This guide also details several options to the installation procedure; be sure to read it.

1.3.1 Access to the On-Line Release Notes

RETOS provides on-line release notes. To get access to them, you can choose either to display or print the release notes during the installation procedure.

The prompt to display or print the release notes appears in the installation procedure dialog, as noted in the description of step 3, only when you invoke the VMSINSTAL utility with the N option.

1.3.2 Installation Requirements

Before you can install RETOS, you must have the following:

- 600 blocks of free disk space during and after installation (minimum)
- Two megabytes of physical memory (recommended)
- Approximate time: 2 to 3 minutes, depending on your type of media and your system configuration.

1.4 Using VMSINSTAL

After you log in to the system manager's account to install RETOS, set the default device to SYS$UPDATE: and then invoke VMSINSTAL, a command procedure that executes the installation of this product. When you enter the command line to invoke VMSINSTAL, as described in step 2 of the installation dialog, you provide the following parameters:

1. facility_name_and_version_number *n* — The variable *n* represents the point release number of your particular product. For this release, RETOS010 is the facility_name_version_number.

   Refer to the label on the RETOS distribution kit or the RETOS Bill of Materials for the point release number of your kit. You can also enter an asterisk (*) in place of facility_name_and_version_number *n*. For example, the first parameter to VMSINSTAL for RETOS Version 1.2 could be either RETOS012 or *.

2. The device name — Device names have the form *ddn:*, where *dd* is the device code, and *n* is the unit number. The variable *device-id* is the device name used in examples in this document.

It is not necessary to use the console drive for installing RETOS. If you do use the console drive, when the installation is complete, be sure to replace any media you may have found in the drive. You cannot use the console drive if you are installing RETOS on a VAX 8600 or 8650.

3. OPTIONS N — Specify N with the OPTIONS keyword to specify that release notes be displayed or printed, for example:

```
$ @SYS$UPDATE:VMSINSTAL RETOSO10 device-id OPTIONS N
```

VMSINSTAL prompts you for the product and device names if you do not supply these parameters. However, if you omit the OPTIONS N parameter, VMSINSTAL does not display the release notes query.

When you invoke VMSINSTAL, it asks:

- Are you logged in to the system manager's account?

  DIGITAL recommends that you install layered software from the system manager's account.

- Do you have adequate quotas for installing layered products?

  VMSINSTAL checks for the following quota values:

  - ASTLM = 24
  - BIOLM = 18
  - BYTLM = 18000
  - DIOLM = 18
  - ENQLM = 30
  - FILLM = 20

  VMSINSTAL then checks to see if either of the following conditions exists and should be discontinued prior to resuming the installation:

  - Whether DECnet is up and running
  - If there are any users logged in to the system

  If VMSINSTAL detects either condition, VMSINSTAL gives you the opportunity to stop the installation procedure by prompting you to continue; to stop the installation, simply press RETURN. In most cases, it should not be necessary to stop the installation because DECnet is running or other users are logged in to the system.

1.5 Installing RETOS

This section contains excerpts from the installation dialog and explanatory text.

Enhancements to VMSINSTAL in future releases of VMS may result in additional prompts appearing during the course of the installation. Such prompts may not be documented in this installation section; however, the installation should proceed normally. Refer to the letter *Read Before Installing or Using* for any notes pertaining to installing RETOS.

Step 1. Log in to the system manager's account.

```
RET
Username: SYSTEM   RET
Password:          RET
```

Step 2. Invoke VMSINSTAL.

Invoke VMSINSTAL as follows:

```
$ @SYS$UPDATE:VMSINSTAL RETOSO10 device-id OPTIONS N

   VAX/VMS Software Product Installation Procedure

It is dd-mmm-yyyy at hh:mm.
Enter a question mark (?) at any time for help.
```

If users are logged into the system, VMSINSTAL lists them on your screen and asks if you want to continue:

```
% VMSINSTAL-W-ACTIVE, The following processes are still active:

* Do you want to continue anyway [NO]?Y

* Are you satisfied with the backup of your system disk [YES]? Y

The following products will be processed:

  RETOS V1.0

Beginning installation of RETOS V1.0 at hh:mm
%VMSINSTAL-I-RESTORE, Restoring product saveset A...
```

Step 3. Select release notes option.

```
Release Notes Options:
    (1) Display release notes
    (2) Print release notes
    (3) Both
* Select option [3]: RET
```

If you select option 1, VMSINSTAL displays the release notes immediately on your terminal. If you select option 2, VMSINSTAL prompts you for a queue name with the following prompt:

```
* Queue name [SYS$PRINT]:
```

After you enter a queue (or press RETURN to indicate the default output print device, SYS$PRINT), VMSINSTAL asks if you want to continue with the installation by displaying the following prompt:

```
Do you want to continue the installation [NO]?: Y
```

If you type NO or press RETURN, VMSINSTAL discontinues the installation.

If you type YES, VMSINSTAL continues with this message:

```
VMSINSTAL-I-RELMOVED, The products release notes have been successfully
removed to SYS$HELP
```

Step 4. Select installation options.

```
* Do you want to purge files replaced by this installation [YES]? RET
```

The installation procedure does not automatically purge files that are replaced during the installation. These files reside in SYS$SYSTEM and SYS$LIBRARY and constitute each release of RETOS. Purging is recommended; you must press RETURN to request that files be purged. If you do not want the files purged, type NO in response to the question. The system Help files are updated in either case.

```
* Do you want to run the IVP after installation [YES]? RET
```

The installation verification procedure (IVP) runs tests to ensure that the installation was successful. If you do not want to run the IVP, type NO in response to the question. Press RETURN to run the IVP after the installation. If this option was chosen, the installation procedure calls the IVP to verify that RETOS was successfully installed.

VMSINSTAL now moves files to their target directories and ends the installation.

```
%VMSINSTAL-I-MOVEFILES, files will now be moved to their
target directories...

        Installation of RETOS V1.0 completed at hh:mm

        VMSINSTAL procedure done at hh:mm

$ LOGOUT
   SYSTEM      logged out at dd-mmm-yyyy hh:mm:ss.ss
```

When the RETOS installation procedure is complete, you can choose to install more products or end the installation procedure. To end the installation procedure, type EXIT or press RETURN. If you removed any media from the console drive before beginning, you should replace it now.

1.6 Files Created or Modified During the Installation

Files moved to the target directory during a RETOS installation include the following:

- SYS$SYSTEM:[SYSEXE]RETOS.EXE
- SYS$SYSTEM:[SYSEXE.RETOS$]FONTAB.FDF
- SYS$SYSTEM:[SYSEXE.RETOS$]FNT07.EXE
- SYS$SYSTEM:[SYSEXE.RETOS$]FNT08.EXE
- SYS$SYSTEM:[SYSEXE.RETOS$]FNT10.EXE
- SYS$SYSTEM:[SYSEXE.RETOS$]FNT14.EXE
- SYS$SYSTEM:[SYSEXE.RETOS$]FNT16.EXE
- SYS$SYSTEM:[SYSEXE.RETOS$]FNT12X14.EXE
- SYS$SYSTEM:[SYSEXE.RETOS$]FNT12X29.EXE
- SYS$SYSTEM:[SYSEXE.RETOS$]FNT15X38.EXE
- SYS$SYSTEM:[SYSEXE.RETOS$]FNT8X38.EXE

In addition, SYS$HELP:HELPLIB.HLB is updated to include the new version of the RETOS help text; the RETOS release notes are copied to SYS$HELP:RETOS010.RELEASE_NOTES.

The following files have been created or modified by the installation procedure:

- [SYSEXE]RETOS.EXE
- [SYSMGR]RETOS_STARTUP.COM
- [SYSHLP]RETOS_HELP.HLB
- [SYSHLP]RETOS010.RELEASE_NOTES
- [SYSEXE.RETOS$]FONTAB.FDF
- [SYSEXE.RETOS$]FNT07.EXE
- [SYSEXE.RETOS$]FNT08.EXE
- [SYSEXE.RETOS$]FNT10.EXE
- [SYSEXE.RETOS$]FNT14.EXE
- [SYSEXE.RETOS$]FNT16.EXE
- [SYSEXE.RETOS$]FNT12X14.EXE
- [SYSEXE.RETOS$]FNT12X29.EXE
- [SYSEXE.RETOS$]FNT15X38.EXE
- [SYSEXE.RETOS$]FNT8X38.EXE
- [SYSTEST]RETOS_IVP.COM
- [SYSTEST]RETOS_TEST.PIC

1.7 Postinstallation Procedures

- VMSINSTAL deletes or changes entries in the process symbol tables during the installation. Therefore, if you are going to continue using the system manager's account, you should log out and log back in again to restore those symbols.
- Add the following line to the system startup command file (SYS$MANAGER:SYSTARTUP.COM) to keep the RETOS symbol and RETOS$FONT logicals available after each system reboot:

  ```
  $ @SYS$MANAGER:RETOS_STARTUP
  ```

  Execute this command on each CPU of the cluster, if you are installing RETOS in a cluster environment.

- Add the following command to the systemwide log-in procedure (normally SYS$MANAGER:SYLOGIN.COM) to define the RETOS symbol for system users:

```
$ RETOS :== $RETOS
```

1.8 Invoking RETOS

RETOS is now installed and can be invoked with the *RETOS/qualifier inputfile* command line. (See Chapter 2.)

The installation procedure assigns a global symbol, so that the RETOS command is recognized and processed. However, the symbol is not assigned for those users who are logged in. All logged-in users who want to use the RETOS command must log out of the system and log back in.

If RETOS will be used extensively on your system, you can reduce system overhead and memory requirements by using the Install utility to install RETOS as a shared image. See Section 1.11, Installing RETOS as a Shared Image, for more information.

1.9 Error Conditions

If the installation procedure or IVP fails for any reason, one of the following messages is displayed:

```
RETOS V1.0 Installation failed.
Please consult the RETOS installation chapter for further
information.

RETOS V1.0 Installation Verification Procedure failed.
Please consult the RETOS installation chapter for further
information.
```

An error during the installation can be caused by one or more of the following conditions:

- Insufficient disk space to complete the installation
- Insufficient system virtual page count parameter
- Insufficient AST quota
- Insufficient buffered I/O byte count
- Insufficient subprocess quota
- Insufficient open file quota

- Insufficient process paging file quota
- Insufficient process working set quota
- Insufficient system maximum working set
- Incorrect version of VAX/VMS or MicroVMS

For descriptions of the error messages generated by these conditions, see the *VAX/VMS System Messages and Recovery Procedures Reference Manual* and the *Guide to VAX/VMS Software Installation.*

If you are notified that any of these conditions exist, you should take the appropriate action, as described in the message. You may need to change a system parameter or increase an authorized quota value. If the installation fails, you must restart the installation procedure from step 2.

If installation fails due to an IVP failure, contact a DIGITAL field representative.

1.10 Maintenance Updates

DIGITAL may periodically issue maintenance update releases of RETOS. Each update consists of an installation kit. You should install this kit as described in this book or in any documentation that may accompany the maintenance update.

Each time a maintenance update is released, the version number changes. For example, if the current version is Version 1.0, the version number of the first maintenance update will be Version 1.1. In addition, the maintenance update will usually include changes to the release notes. The release notes state what changes have been made to RETOS since the previous release.

Read the release notes before you install RETOS and at any time after the product is installed. For information on reading the release notes before RETOS is installed, follow the installation procedure up to step 3. At that point, you can choose to type or print the release notes and discontinue installation. To read the release notes after installation, type or print the file SYS$HELP:RETOS010.RELEASE_NOTES.

The Help file also directs you to the release notes file. After installation, type:

```
$ HELP RETOS RELEASE_NOTES
```

1.11 Installing RETOS as a Shared Image

If you wish to install RETOS on a system that will use it extensively, you can reduce the system overhead and memory requirements by installing it as a shared image. To install RETOS as a shared image on a system that is currently running, use the Install utility while you are logged in under the system manager's account.

DIGITAL recommends that you perform the following steps on a system that has just been bootstrapped. If errors occur, consult the *Guide to VAX/VMS System Management and Daily Operations*.

Step 1. Invoke the Install Utility.

```
$ INSTALL :== $RUN SYS$SYSTEM:INSTALL/COMMAND_MODE
$ INSTALL
```

Step 2. Install RETOS as a shared image.

```
INSTALL> SYS$SYSTEM:RETOS.EXE/OPEN/SHARED
INSTALL> CTRL/Z
```

Step 3. Update the system start-up file to install RETOS.

Include the following lines in SYS$MANAGER:SYSTARTUP.COM (the site-specific start-up file) to ensure that RETOS is installed as a shared image each time the system is bootstrapped:

```
$ INSTALL :==RUN SYS$SYSTEM:INSTALL/COMMAND_MODE
$ INSTALL
SYS$SYSTEM:RETOS.EXE/OPEN/SHARED
```

Then, RETOS will be installed each time the system is bootstrapped.

1.12 Determining and Reporting Problems

If an error occurs while you are using RETOS, and you believe that the error is due to a problem in RETOS, submit a Software Performance Report (SPR). If you find an error in the RETOS documentation, fill out and submit the READER'S COMMENTS form appearing at the back of the manual in which the error was found.

When you prepare to submit an SPR, please do the following:

1. Describe as accurately as possible the state of the system and the circumstance when the problem occurred. Include in the description the version number of RETOS being used. (You can find the version number by entering the DCL command, HELP RETOS RELEASE_NOTES.) Illustrate the problem with specific examples.

   If you report a documentation error, specify the title of the manual and include the section and page number where the error occurred.
2. Reduce the problem to as small a size as possible.
3. Remember to include listings of any command files, relevant data files, and so on.
4. Provide a listing of the ReGIS file.
5. Submit sources on machine-readable media (floppy diskette or magnetic tape), if possible. All media will be returned to you when the SPR is answered.
6. Report only one problem per SPR. This will facilitate a more rapid response.
7. Mail the SPR package to DIGITAL.

Experience shows that many SPRs contain insufficient information to duplicate or identify the problem. Complete and concise information will help DIGITAL give accurate and timely service to software problems.

1.13 Sample Installation

Following is a sample installation of RETOS. Installation messages may vary due to system conditions and the options you choose.

```
Username: SYSTEM
Password:

    Last interactive login on Monday, 18-MAY-1987 21:59
    Last non-interactive login on Tuesday, 19-MAY-1987 03:00

$ set def sys$update
$ @vmsinstal retos010 bee$dub0:[retos.retos010] options n

VAX/VMS Software Product Installation Procedure V4.5

It is 19-MAY-1987 at 13:36.
```

```
Enter a question mark (?) at any time for help.
%VMSINSTAL-W-DECNET, Your DECnet network is up and running.
%VMSINSTAL-W-ACTIVE, The following processes are still active:
        WOPR CLUSTER
        COMBAR_1
        Proud 4MClub VP
        Marino - TD!!
        TORRES_1
* Do you want to continue anyway [NO]? yes
* Are you satisfied with the backup of your system disk [YES]? yes

The following products will be processed:

       RETOS V1.0

Beginning installation of RETOS V1.0 at 13:36

%VMSINSTAL-I-RESTORE, Restoring product saveset A...

Release Notes Options:

1.  Display release notes
2.  Print release notes
3.  Both 1 and 2

* Select option [3]: 2
* Queue name [SYS$PRINT]: alpha$print

Job retos010 (queue ALPHA$PRINT, entry 598) pending

* Do you want to continue the installation [N]? Y
%VMSINSTAL-I-RELMOVED, The products release notes have been successfully
moved to SYS$HELP.

During this installation, new files will be provided to replace existing
versions.  You may purge these older versions to save disk space, or keep
them if you feel they may be of use.  Purging is recommended.
* Do you want to purge files replaced by this installation [YES]? yes

Most products provide an Installation Verification Procedure (IVP)
which verifies the completeness and accuracy of the installation.
You may wish to run the IVP immediately after installation.
* Do you want to run the IVP after the installation [YES]? yes

%VMSINSTAL-I-SYSDISK, This product creates system disk directory VMI$ROOT:[SYSEXE.RETOS$]

      The RETOS$FONTS logical and the RETOS symbol will disappear the next
      time the system is rebooted.  To make them appear after each boot,
      add the following line to the system startup command file,
      SYS$MANAGER:SYSTARTUP.COM:

              $ @SYS$MANAGER:RETOS_STARTUP

      If you are installing RETOS into a cluster environment make sure that
      this command is executed on each CPU of the cluster.

      To define the symbol RETOS for each user who logs in, add the following
      command to the system-wide login procedure (normally
      SYS$MANAGER:SYLOGIN.COM):
```

```
                $ RETOS :== $RETOS
%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target directories...
        Installation of RETOS V1.0 completed at 13:42
        VMSINSTAL procedure done at 13:42
$ log
  SYSTEM       logged out at 19-MAY-1987 13:58:02.40
```

Chapter 2
Using RETOS

2.1 The Converter User

Users of RETOS generally fall into two groups:

- Those who use applications, such as DECslide or DECgraph, that produce ReGIS files
- Programmers who write applications that produce ReGIS output files

RETOS converts ReGIS to sixel graphics. Typically, a user previews a ReGIS file on a video terminal. Connected to the printer port of this terminal is a sixel printer. Since the printer supports sixels for bit map display, the ReGIS output file requires conversion to sixel graphics before printing.

Send your RETOS-converted file through your DIGITAL video terminal, to a printer attached to your printer port, by using the DCL TYPE command (*$ TYPE filename*) and the Printer Control Mode. To enter and exit Printer Control Mode, put the following escape sequences at the top and bottom of the converted file:

- To enter Printer Control Mode: **ESC [5 i**
- To exit Printer Control Mode: **ESC [4 i**

NOTE

Make sure that you set your terminal to NO BROADCAST before using Printer Control Mode. Otherwise, a message received on your terminal could interrupt the printing of your

file and create unwanted characters and white space in your drawing.

Another RETOS environment might find several users sharing a sixel printer, such as the LJ250, through a print queue. In either case, sharing an LJ250 through a print queue or using a sixel printer connected to your printer port, you access RETOS by using a command line at the $ (dollar sign) or system prompt. For a detailed description of the command syntax including supported qualifiers, see Sections 2.2 and 2.4.

To print your newly-converted sixel graphics file on a printer connected to a print queue, use the DCL PRINT command.

Using the terminal's Print Screen function, which dumps the terminal's screen to the printer, is an alternative method of printing your ReGIS file. However, using RETOS offers the following advantages:

- Page size and resolution compatibility of printer and terminal (The picture is scaled to fit on the output page.)
- Availability of the full range of colors and resolutions on printers like the LJ250
- Ease of sharing graphics files (through electronic mail, for example)
- Separation of conversion and printing (Files can be converted in a batch job and printed through a print queue without operator intervention.)
- Sharing of a printer by several users (use of a print queue rather than a terminal's printer port)

2.2 RETOS Command Syntax

RETOS command syntax is as follows:

```
$ RETOS/qualifiers inputfile
```

where:

RETOS	is the name of the ReGIS to sixel graphics converter
qualifiers	is zero or more command qualifiers
inputfile	is the name of the ReGIS file to be converted to sixel graphics

2.3 Format of the Input File

Ideally, RETOS input files contain pure ReGIS graphics. RETOS treats characters in this file as ReGIS commands with the exception of ANSI escape sequences and controls, which it parses and ignores.

Your input file can contain variable length records as long as no record is larger than 512 bytes.

See Chapter 3 for information on ReGIS command syntax.

2.4 Supported Command Qualifiers

This section describes the command syntax qualifiers supported by RETOS. Use these qualifiers to select characteristics for the ReGIS-to-sixels conversion. All command qualifiers are optional.

Command qualifiers can be abbreviated by using the least number of unique characters.

NOTE

Default values for each qualifier depend on the value of the /DEVICE qualifier you select. RETOS preset default values are those associated with /DEVICE=LJ250.

2.4.1 /BACKGROUND=bckgnd

The /BACKGROUND=*bckgnd* qualifier designates the background color of the output page. Select *bckgnd* from one of the keywords: BLACK, WHITE, or COLOR. RETOS does not write sixel data in the background color, which is normally BLACK for video terminals and WHITE for printers. If you specify /BACKGROUND=COLOR, RETOS writes sixel data in all colors, including black and white, and considers the background color to be BLACK. The default is /BACKGROUND=WHITE.

See Section 2.4.2 for a description of the effect of the /BACKGROUND qualifier with the /NOCOLORS qualifier.

2.4.2 /COLORS=n; /NOCOLORS

The /COLORS=*n* qualifier specifies the number of colors or intensities, including black and white, to display on the output device at the same time. Two is the minimum number and 256 is the maximum number. The default is /COLORS=256.

The /NOCOLORS qualifier indicates that RETOS produces black and white sixels rather than gray or colored sixels. In the output, a 0 represents pixels drawn in the background color; a 1 represents pixels drawn in any other color. Therefore, the output file contains no color information.

NOTE

When your destination device is a black and white printer, such as the LN03 PLUS, RETOS conversion works best if the input file is also black and white. Otherwise, anything that is black on the screen is white on the page, and anything that is not black on the screen is black on the page. This means, at best, that all bars in a graph or slices of a pie print black. At worst, the entire page prints black if you draw the picture with a white background and gray bars or gray pie slices. DECgraph and DECslide users, for example, should select monochrome output for RETOS conversion to be printed on a black and white device.

With the /NOCOLORS qualifier selected, the combination of /BACKGROUND and /REVERSE_VIDEO qualifiers determines the background color. See Table 2–1.

Table 2–1: Background Color with /NOCOLORS Qualifier

	/BACKGROUND Setting		
/REVERSE_VIDEO Setting	**BLACK**	**WHITE**	**COLOR**
/REVERSE_VIDEO	WHITE	BLACK	WHITE
/NOREVERSE_VIDEO	BLACK	WHITE	BLACK

NOTE

/COLORS=2 and /NOCOLORS are not equivalent. /COLORS=2 means that the entire picture is drawn in two colors. For example, you can draw in red and blue, or in yellow and pink, but you can use only two different colors on each

page. /NOCOLORS means that no color information is written to the output file; each pixel is either on or off. /NOCOLORS output is generally black and white.

2.4.3 /DEVICE=devicetype

The /DEVICE=*devicetype* qualifier names the destination device — printer or terminal. Select one of the following keywords as the *devicetype*: LJ250, LJ252, LN03_PLUS, LN03, LA75, LA50, LA210, LA100, VT340, VT330, VT241, or VT240. The default is /DEVICE=LJ250. Specifying /DEVICE=LJ252 has the same effect as specifying /DEVICE=LJ250.

This qualifier determines default values for other RETOS qualifiers. See Tables 2–2 and 2–3 for the specific qualifiers set for each device type.

Table 2–2: Qualifiers Selected with /DEVICE=printertype

Qualifier	LJ250	LA50	LA75	LA210	LA100	LN03	LN03_PLU
/ORIENTATION	=P†	=P	=P	=P	=P	=P	=P
/SIZE	=(8,10.5)	=(8,10.5)	=(8,10.5)	=(8,10.5)	=(8,10.5)	=(8,10.25)	=(8,10.25)
/RESOLUTION	=(90,90)	=(144,72)	=(144,144)	=(330,72)	=(132,72)	=(100,100)	=(300,300)
/MARGIN	=(0,0)	=(0,0)	=(0,0)	=(0,0)	=(0,0)	=(0,0)	=(0,0)
/COLORS	=256	—	—	—	—	—	—
/NOCOLORS	—	set	set	set	set	set	set
/BACKGROUND	=WHITE	=WHITE	=WHITE	=WHITE	=WHITE	=WHITE	=WHITE
/REVERSE_VIDEO	set	set	set	set	set	set	set
/NOREVERSE_VIDEO	—	—	—	—	—	—	—
/LEVEL	=2	=1	=2	=1	=1	=2	=2
/GRID_UNITS	=D‡	=D	=D	=D	=D	=PX:300§	=PX:300
/MONOCHROME	—	set	set	set	set	set	set
/NOMONOCHROME	set	—	—	—	—	—	—

§PX stands for PIXEL.

†P stands for PORTRAIT.

‡D stands for DECIPOINT.

Table 2–3: Qualifiers Selected with /DEVICE=terminaltype

Qualifer	VT240	VT241	VT330	VT340
/ORIENTATION	=P†	=P	=P	=P
/SIZE	=(8,4.8)	=(8,4.8)	=(8,4.8)	=(8,4.8)
/RESOLUTION	=(100,50)	=(100,50)	=(100,100)	=(100,100)
/MARGIN	=(0,0)	=(0,0)	=(0,0)	=(0,0)
/COLORS	=4	=4	=4	=16
/NOCOLORS	—	—	—	—
/BACKGROUND	=COLOR	=COLOR	=COLOR	=COLOR
/REVERSE_VIDEO	—	—	—	—
/NOREVERSE_VIDEO	set	set	set	set
/LEVEL	=1	=1	=2	=2
/GRID_UNITS	=D‡	=D	=D	=D
/MONOCHROME	set	—	set	—
/NOMONOCHROME	—	set	—	set

†P stands for PORTRAIT.

‡D stands for DECIPOINT.

NOTE

After selecting a destination device, print only the resulting sixel file on that kind of device. The /DEVICE qualifier automatically selects the right grid size and aspect ratio for the device you name. Printing or displaying the file on a different device type may produce poor results.

2.4.4 /GRID_UNITS=gru

The /GRID_UNITS=*gru* qualifier specifies the units for the horizontal grid size (distance between adjacent dots) in the output file. This applies only to level 2 devices. The variable *gru* is either DECIPOINTS or PIXELS:*n*. If *gru* is DECIPOINTS, RETOS gives the grid size in decipoints. If *gru* is PIXELS:*n*, where *n* is the (integer) horizontal hardware resolution in dots/inch, RETOS gives the grid size in pixels. RETOS calculates the horizontal grid size based on the horizontal resolution provided in the command line. The default is /GRID_UNITS=DECIPOINTS.

NOTE

For some devices, such as the LJ250, the grid size must be in decipoints. For other devices, such as the LN03, giving the grid size in pixels provides for maximum accuracy. A pixel is the smallest piece of an image as defined by the generating software. A decipoint is 1/720″.

2.4.5 /LEVEL=n

The /LEVEL=*n* qualifier indicates the sixel graphics conformance level of the output device. The variable *n* is either 1 or 2. The VT240 is level 1, and the LJ250 is level 2. See Tables 2–2 and 2–3 for the sixel graphics conformance level of other devices supported by RETOS. The default is /LEVEL=2.

NOTE

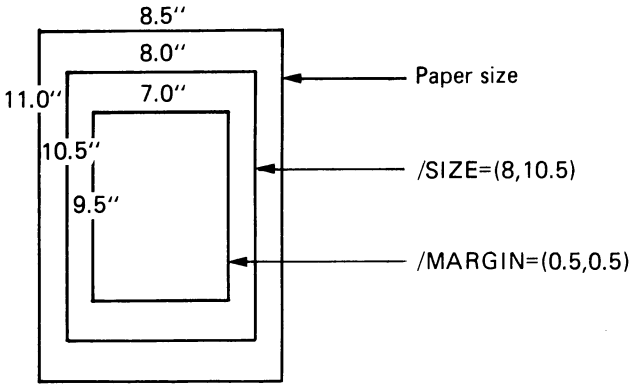
Level 1 or level 2 in this qualifier refers to sixels only. For example, a level 2 sixel device is not the same as a level 2 ANSI device. The VT340 is a level 2 sixel device and a level 3 ANSI device.

2.4.6 /MARGIN=(hmar,vmar)

The /MARGIN=(*hmar,vmar*) qualifier designates the horizontal and vertical margins for the output page in inches. Give these numbers in decimal fractions.

These margins add to the margins that output devices impose; therefore, they are within the boundaries defined by the /SIZE qualifier. For example, the LJ250's output page size is 8″ x 10.5″. By setting /MARGIN=(0.5,0.5), you get an additional margin of 0.5″ on all sides, and your graphics print in a 7″ x 9.5″ area (see Figure 2–1 below). The default for supported devices is /MARGIN=(0,0).

Figure 2–1: RETOS/DEVICE=LJ250/MARGINS=(0.5,0.5) inputfile



2.4.7 /MONOCHROME; /NOMONOCHROME

The /MONOCHROME qualifier identifies the output device as a monochrome device; that is, the device displays shades of gray instead of colors. RETOS initializes the color map as a gray scale and converts RGB color specifiers into appropriate gray values. The output file then contains only the lightness component of colors.

RETOS ignores the /MONOCHROME qualifier if /NOCOLORS is in effect; /NOCOLORS implies that the output device is monochrome.

NOTE

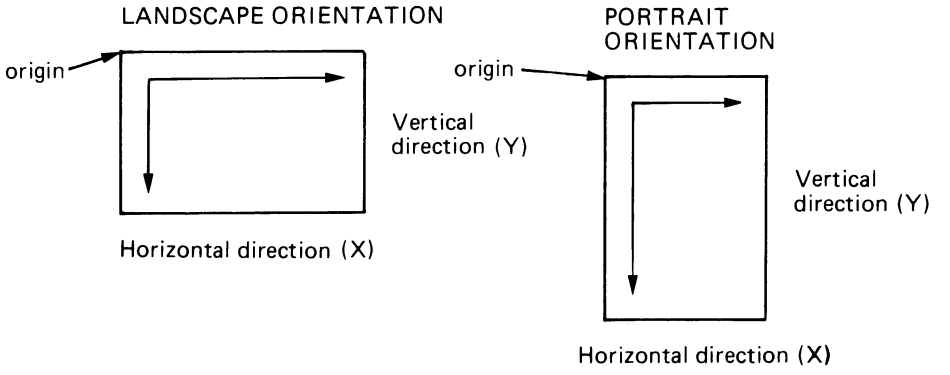
RGB, standing for red, green, and blue, is a way to express colors. RETOS converts RGB colors to HLS colors, so that LJ250 colors match more closely with video colors.

The /NOMONOCHROME qualifier identifies the output device as a color device. RETOS initializes the color map to the default color map (see Table 3–5) and writes HLS colors to the output file.

2.4.8 /ORIENTATION=lp

The /ORIENTATION=*lp* qualifier selects either portrait mode (the x axis is the horizontal axis) or landscape mode (the y axis is the horizontal axis) as the page printing orientation. The variable *lp* is one of the two keywords: LANDSCAPE or PORTRAIT. The default for supported devices is /ORIENTATION=PORTRAIT. Figure 2–2 shows the two orientation modes.

Figure 2–2: Page Printing Orientation



2.4.9 /OUTPUT=outputfile

The /OUTPUT=*outputfile* qualifier specifies the name of the sixel graphics output file. If you do not specify an output file name, the converter uses the ReGIS (input) file name with a .SIX file type. If you want to place the sixel (output) file in a directory other than the current directory, you must specify the directory name in the /OUTPUT qualifier.

2.4.10 /RESOLUTION=(hres,vres)

The /RESOLUTION=(*hres,vres*) qualifier specifies the horizontal and vertical resolutions of the output device in dots/inch. Provide these resolutions in decimal fractions. Normally, you do not need to change the resolution settings. If you do change the settings, specify only a resolution supported by the intended output device. The default is /RESOLUTION=(90,90).

2.4.11 /REVERSE_VIDEO; /NOREVERSE_VIDEO

The /REVERSE_VIDEO qualifier dictates inversion of the lightness component of colors in the output file. An HLS color of H50 L70 S20 becomes H50 L30 S20. Subtract the lightness value (L70) from 100 to calculate the /REVERSE_VIDEO lightness value (L30). This means that a picture designed for a video terminal having a black background has a white background when printed on white paper. If /NOCOLORS is in effect, specifying /REVERSE_VIDEO reverses the background color specified by /BACKGROUND; black becomes white and vice versa.

The /NOREVERSE_VIDEO qualifier specifies that the lightness component of colors not be inverted.

See Section 2.4.2 for the combined effect of this qualifier and the /NOCOLORS qualifier.

NOTE

HLS stands for hue, lightness, and saturation. This is a method of expressing colors in ReGIS and sixel graphics. For more information, refer to Chapter 3, Table 3–4.

2.4.12 /SIZE=(width,height)

The /SIZE=(*width, height*) qualifier specifies the sixel graphics output page width and height in inches. Provide the width and height in decimal fractions, as =(8,10.5). The default is /SIZE=(8,10.5).

These margins do not include margins that output devices impose. For example, the LJ250 imposes a 0.25″ margin. For a paper size of 8.5″ x 11″, the /SIZE qualifier for the LJ250 is =(8,10.5) or 8″ x 10.5″. When you select a destination with the /DEVICE qualifier, RETOS provides you with the proper output page size.

2.5 Default Values

Section 2.5.1 lists RETOS's device-dependent defaults. RETOS provides you with default file types, discussed in Section 2.5.2, and initial default qualifiers, explained in Section 2.5.3.

2.5.1 Device-Dependent Parameters

RETOS has the following device-dependent defaults:

Default coordinate system:	S(A[0,0][799,479])
Macrograph storage:	20,000 bytes
Position stack:	16 positions
Loadable alphabets:	3
Default text display size:	T(S[9,20])
Default text unit size:	T(U[8,20])

2.5.2 Default File Types

If you do not provide a file type for the ReGIS input file, the file type defaults to .PIC. If you do not specify the sixel output file, the default output file name is the input (ReGIS) file name with a .SIX file type extention.

RETOS writes the sixel graphics file to the current directory, unless you include a directory name in the /OUTPUT qualifier.

2.5.3 Initial Default Values

Each time you invoke RETOS, the default output device type is the LJ250 printer. Other initial values determined by this default qualifier are as follows:

- /ORIENTATION=PORTRAIT
- /SIZE=(8,10.5)
- /RESOLUTION=(90,90)
- /MARGIN=(0,0)
- /COLORS=256
- /BACKGROUND=WHITE
- /REVERSE_VIDEO
- /LEVEL=2
- /GRID_UNITS=DECIPOINTS
- /NOMONOCHROME

2.5.4 Selecting Default Values

If you make another /DEVICE qualifier selection, some of the preceding default values change. For values specific to the selected device, refer to Tables 2–2 and 2–3. Other output devices supported by RETOS include the following:

- LA50 printers
- LA75 printers
- LA100 printers
- LA210 printers
- LN03 printers
- LN03 PLUS printers
- VT240 terminals
- VT241 terminals
- VT330 terminals
- VT340 terminals

2.6 Error Messages

RETOS prints an error message before exiting if any of the following conditions occur:

- RETOS cannot open the input file.
- RETOS cannot open the output file.
- An unexpected I/O error occurs.
- The command syntax is invalid.

Following is a list of error messages produced by RETOS with a suggested action for each error. The variable *filename* indicates that the error message includes a file name.

% RETOS-F-OPENFILE, Error opening file *filename* for input

Explanation. RETOS could not open the ReGIS source file or a font file.

User Action. Reenter the command with the name of a source file you can read or ensure that you can read the font files in the RETOS$FONTS directory.

% RETOS-F-CREATEFILE, Error creating file *filename*

Explanation. RETOS could not create the sixel output file.

User Action. Reenter the command with a valid file name.

% RETOS-F-READFILE, Error reading from file *filename*

Explanation. RETOS encountered an RMS error when it tried to read from the ReGIS source file or from a font file.

User Action. Reenter the command after ensuring you can read the file.

% RETOS-F-WRITEFILE, Error writing to file *filename*

Explanation. RETOS encountered an RMS error when it tried to write to the sixel output file.

User Action. Reenter the command after ensuring you can write on the file.

% RETOS-F-CLOSEFILE, Error closing file *filename*

Explanation. RETOS encountered an RMS error when it tried to close the source file or output file or a font file.

User Action. If the output file is corrupted, reenter the command after ensuring that you can write on the file.

% RETOS-F-PARSEFILE, Error parsing file name *filename*

Explanation. RETOS encountered an RMS error when it tried to parse the given file name.

User Action. Reenter the command with a valid file name.

% RETOS-F-SEARCHFILE, Error searching for file *filename*

Explanation. RETOS encountered an RMS error when it tried to search for the given file.

User Action. Reenter the command with a valid name.

% RETOS-F-CONNECTFILE, Error connecting to file *filename*

Explanation. RETOS encountered an RMS error when it tried to connect to the record stream for the given file.

User Action. Reenter the command after ensuring that you can access the file.

% RETOS-F-BADFORMAT, File *filename* is in invalid format

Explanation. RETOS found a font file that was formatted incorrectly.

User Action. Reinstall RETOS from the distribution medium.

% RETOS-F-VMERROR, Error allocating virtual memory

Explanation. RETOS encountered an error when it called LIB$GET_VM. Your page size is too large for the system configuration.

User Action. Reenter the command with a lower resolution or a smaller page size, or contact the system manager for a larger page size quota. Refer to Section 2.8.2 for more information.

% RETOS-F-BARPARAM, Command qualifier value out of range

Explanation. A qualifier in the command line has a value that is too low or too high.

User Action. Reenter the command with valid qualifier values.

In addition, RETOS produces RMS error messages and other standard error messages providing further information about errors. RETOS does not report ReGIS errors.

2.7 Supported Output Devices

RETOS supports the following output devices: LJ250, LJ252, LN03 PLUS, LN03, LA75, LA50, LA210, and LA100 printers, and VT340, VT330, VT241, and VT240 terminals. To obtain compatibility among output devices, so that the same RETOS-converted sixel file can be printed on more than one device type, choose a resolution and conformance level supported by the desired output devices. For example, use /LEVEL=1/RESOLUTION=(144,72), if you want the output file to be compatible with the LA50 printer.

Selecting /DEVICE=LA50 in the RETOS command line allows you to print the output file on most printers except the LA100 and the LA210 printers. If you use /DEVICE=LA100, the picture looks right on the LA100 and LA210 printers.

2.7.1 LJ250 and LJ252 Printers

The LJ250 printer is a color ink jet printer. Output can be in 8 colors at 180 dots/inch or in 256 colors at 90 dots/inch (using half-toning). Other resolutions that the device supports include:

- 256 colors — (90,90), (90,45)
- 8 colors — (180,180), (180,90), (180,72), (144,72), (90,36), (72,72), (72,36), (36,36)

Specify the following qualifiers to obtain high resolution output:

```
/RESOLUTION=(180,180)/COLORS=8
```

The LJ252 printer is the same as the LJ250 printer except that the LJ252 printer has a parallel communications interface.

When printing transparencies on the LJ250 or LJ252, leave an inch at the top and bottom of the page. The printers cannot print the top inch of a single-fed sheet. Use /SIZE=(8,9) to allow for a .25″ margin on the left and right of the transparency and a 1″ margin on the top and bottom of an 8″ x 11″ page. Use the /SIZE qualifier rather than the /MARGIN qualifier, since the hardware imposes the margin.

2.7.2 LN03 PLUS and LN03 Printers

The LN03 PLUS and LN03 printers are laser printers producing 300 dots/inch output. Files converted with the /DEVICE=LN03 qualifier print correctly on the LN03 PLUS printer and vice versa. However, some high resolution sixel files cause picture complexity errors on the LN03 printer, so RETOS uses a default resolution of 100 dots/inch for the printer. Using a lower resolution when printing on the LN03 PLUS printer may also improve performance.

RETOS output for LN03 PLUS emulation on the PRINTSERVER 40 print systems and the LN03R SCRIPTPRINTER is not supported. DIGITAL recommends that you use /PARAMETERS=(DATA_TYPE=REGIS) in the PRINT command line to print ReGIS files on these print systems.

2.7.3 LA75 Printers

In addition to the default resolution of (144,144), the LA75 printer supports the following resolutions: (180,72), (90,36), (144,72), and (36,36). Notice that the horizontal resolution can be increased at the expense of the vertical resolution.

2.7.4 LA50 Printers

The LA50 printer supports two resolutions: (144,72) and (180,72). Select the appropriate resolution by setting a switch on the back of the printer. RETOS uses a default resolution of (144,72). If your printer is set to the 2.5:1 aspect ratio mode, use /RESOLUTION=(180,72).

2.7.5 LA210 Letterprinters

The LA210 Letterprinter supports the following resolutions: (330,72), (220,72), (165,72), (132,72), (110,72), (94,72), (83,72), and (74,72).

2.7.6 LA100 Printers

The LA100 printer supports only (132,72) resolution output.

2.7.7 VT340, VT330, VT241, VT240 Terminals

The VT340, VT330, VT241, and VT240 video terminals have various physical resolutions. RETOS uses 100 dots/inch as the video terminal horizontal resolution. For example, the VT340 has 800 horizontal pixels and 480 vertical pixels; its resolution is (100,100) and its size is (8,4.8).

Normally, you display ReGIS files on these terminals without a RETOS conversion, since these terminals support the ReGIS protocol.

2.8 Improving RETOS Performance

The following sections give hints on how to increase the speed of ReGIS-to-sixel conversion and how to avoid a system deadlock.

2.8.1 Increasing Conversion Speed

RETOS can take several minutes to convert high resolution files (180 dots/inch on the LJ250, for example). You can use the following techniques to speed up conversion:

- Reduce the output file resolution (use 90 dots/inch on the LJ250 instead of 180 dots/inch).
- Increase the working set size for RETOS users.
- Run RETOS on lightly loaded systems or at non-peak hours.
- Add more physical memory to your VAX system.

2.8.2 Avoiding VMS System Deadlock

RETOS runs as an unprivileged user process and should not, in theory, create a VMS system deadlock. Depending on system parameter settings and the size of the page file and swap file, RETOS could allocate a bit map so large that VMS goes into a deadlock. Using RETOS to convert files with extremely high resolutions and unusually large page sizes can stop the system. Using the /DEVICE qualifier for your output device does not cause this problem; overriding these values can.

Normally, if RETOS calls for too much memory, you receive an insufficient virtual memory error message as follows:

```
% RETOS-F-VMERROR, Error allocating virtual memory
```

However, if the user's page file quota (limit) and the VIRTUALPAGECNT system generation parameter are too high compared to the size of the page file, then RETOS can allocate a bit map too large for the page file, and the system stops. The page file quota limits the number of pages that your process can use in the system page file, which is shared by all system processes.

If the bit map is too large for the system swap file, other processes cannot create pages in the page file, and the system also deadlocks.

To calculate the virtual memory required for a RETOS bit map, /DEVICE=LN03PLUS, for example, do the following:

1. Multiply the x resolution by the y resolution (300 x 300).
2. Multiply by the x and y /SIZEs (300 x 300 x 8 x 10.25).

   This is the number of pixels in the bit map.
3. Divide by the number of pixels in a byte (300 x 300 x 8 x 10.25/8).

   Determine the pixels/byte as follows:

   - 8 for /NOCOLORS or 2 colors (LN03, for example)
   - 4 for 3 or 4 colors (VT240, for example)
   - 2 for 5 to 16 colors (VT340, for example)
   - 1 for 17 to 256 colors (LJ250, for example)

   This is the number of bytes required for the bit map (922500 bytes).
4. Divide by 512 to determine the number of pages in the bit map (300 x 300 x 8 x 10.25/8/512 or 1802 pages).

   Each page requires one block in the page file.
5. Add 150 blocks (pages) required by RETOS.EXE and 300 pages that RETOS allocates for fonts (1802 + 450 pages).

RETOS memory requirements for /DEVICE qualifiers should not cause problems on most systems. For /DEVICE=LN03PLUS, RETOS requires approximately 2250 pages. Two thousand pages equal one megabyte. For /DEVICE=LJ250, the RETOS default, the bit map is (90 x 90 x 8 x 10.5)/1 + (150 + 300) x 512 bytes, or about 1780 pages. However, if you override the default qualifiers, with /DEVICE=LN03PLUS/COLORS=8 for

example, RETOS requires more than 7500 pages. If the page file is too small, the system deadlocks.

To help prevent these deadlocks, ask your system manager to increase the size of the page file or reduce the page file quotas (limits) of RETOS users. With a lower page file quota, RETOS fails before it deadlocks the system. Reducing the VIRTUALPAGECNT parameter in SYSGEN has a similar effect; it reduces the virtual memory that can be allocated by any process in the system.

2.9 RETOS Hints for DECgraph and DECslide

The following sections tell how to use selected RETOS qualifiers for better results with DECgraph and DECslide files.

2.9.1 Creating Files for /NOCOLORS Devices

DECgraph and DECslide users must generate a *monochrome* file rather than a *color* file to create a ReGIS file that can be converted by RETOS for printing on /NOCOLORS printers, such as the LN03 PLUS. You can create a monochrome file interactively or by using a DCL command. Press the GOLD/M or PF1/M key to create the required file while using DECgraph or DECslide. DECgraph users can issue the following DCL command line to produce a monochrome file:

```
$ GRAPH/NOINTER/MONOCHROME graphics_file
```

With DECslide, use the following DCL command to produce a monochrome file:

```
$ SLIDE/NOINTER/MONOCHROME slide_file
```

2.9.2 Selecting Appropriate Backgrounds for Printing

By default, RETOS reverses the background color of files converted for graphic printers. That is, RETOS changes the background from black to white or from white to black. When you create drawings on a screen with a black background, the default value provides the appropriate background for the output page. RETOS reverses the background color to white. However, if your picture has a white background on the screen, you must change the default value to avoid an unwanted black background. Use the /NOREVERSE_VIDEO qualifier on the RETOS command line, as follows:

```
$ RETOS/DEVICE=printertype/NOREVERSE_VIDEO inputfile
```

2.10 RETOS Hints for DECpage

DECpage users can include RETOS output files in their documents intended for LN03 and LN03 PLUS printers. DECpage, Version 2.1, offers three sizes of diagrams: small, large, and landscape. To produce diagrams for these printers, select the appropriate RETOS command line when converting your DECpage files:

- For a small diagram:

  ```
  $ RETOS filename/DEVICE=LN03_PLUS/SIZE=(3,2)
  ```

- For a large diagram:

  ```
  $ RETOS filename/DEVICE=LN03_PLUS/SIZE=(5,3.2)
  ```

- For a landscape diagram:

  ```
  $ RETOS filename/DEVICE=LN03_PLUS/SIZE=(7.6,4.9)
  ```

Do not use the /ORIENTATION=LANDSCAPE qualifier when producing a landscape diagram, because DECpage rotates the picture.

NOTE

Normally, the LANDSCAPE graphic size is (7.5,4.8), but because of a DECpage bug, you must use /SIZE=(7.6,4.9).

Chapter 3

ReGIS Graphics

ReGIS (Remote Graphics Instruction Set) is a DIGITAL-developed graphics protocol. This chapter describes the ReGIS display structure, the ReGIS command structure, ReGIS commands supported by RETOS, and ReGIS commands not supported by RETOS. This chapter also gives information about the RETOS environment, including the converter's handling of ANSI text and escape sequences, screen addressing, color mapping, compatibility with other ReGIS devices, and examples of the ReGIS to sixel graphics conversion.

NOTE

RETOS supports VT240/241 and VT330/340 ReGIS. For complete information about VT240/241 ReGIS, see the *VT240 Series Programmer Reference Manual*. For more information on VT330/340 ReGIS, see the *VT330/340 Programmer Reference Manual*.

3.1 ReGIS Definition

ReGIS is a set of internal commands used by subroutines to draw pictures and plot data. It works by treating an image as a group of graphic objects. Each graphic object is a standard geometric form: dot, line, curve, circle, and arc. ReGIS lets you describe each form with a few commands. ReGIS also allows you to create text.

ReGIS commands are encoded as ASCII character strings. The ReGIS interpreter processes the ReGIS data serially, which allows the commands to be transmitted across serial communications lines. In general, a ReGIS string consists of a command keyletter followed by arguments.

ReGIS is a graphics descriptor protocol rather than a programming language. It has no algorithmic structure or arithmetic functions. However, high-level programming languages can use ReGIS strings to generate graphic images. Languages such as BASIC, FORTRAN, and Pascal can use ReGIS strings in PRINT or WRITE statements.

3.2 ReGIS Display Structure

RETOS uses a default coordinate system of 800 horizontal by 480 vertical logical units (for compatibility with the VT240/241 and VT330/340 terminals). RETOS scales these units to map them to the physical pixels in the output device. Depending on your choice of coordinate system and the resolution of the output device, several logical coordinates may map to one physical pixel. Conversely, two adjacent logical coordinates may map to two nonadjacent physical pixels.

Coordinate units in ReGIS commands refer to the logical coordinate system. Most ReGIS commands use X/Y coordinates to specify where to move or where to draw an image. Some commands can use pixel vectors, an alternative way of specifying a position in the image.

3.2.1 [X,Y] Coordinate System

The ReGIS coordinate system lets you access each logical pixel by using an X/Y coordinate value for the specific pixel. The X coordinate specifies the horizontal position value. The Y coordinate specifies the vertical position value. The pixel is located at the intersection of the X and Y values.

The upper left corner of the image, known as the origin, is location [0,0]. The ReGIS current position is initially [0,0]. The default X coordinates range from 0 (the left edge) to 799 (the right edge). Default Y coordinates range from 0 (the top) to 479.

Coordinates in ReGIS commands must be enclosed in brackets. The X coordinate must be first. X and Y coordinates must be separated by a comma.

You need not specify X and Y values in all cases. In fact, you have to specify an X or Y value only when that value is different from the current value. If you want to change only the X value, you specify only the new X value. ReGIS recognizes [X] as meaning the Y value is unchanged. If you want to change only the Y value, use a comma before the new Y value in the brackets. ReGIS recognizes [,Y] as meaning the X value is

unchanged. The comma identifies the numeric coordinate value as a Y value; no comma identifies a single numeric value as an X coordinate

You specify coordinate values by using the numeric values assigned to the display addressing, whether that addressing is done at the default value or in embedded decimal or exponential values. (See the display addressing option to the Screen Control command in Section 3.4.)

Coordinate values can be absolute, which refers to a numerically specific point; relative, which refers to a point as it relates to the current position; or a combination of the two. You can also use a null position, [ ] or [+0,+0], which does not change the current position. The following list shows some examples of coordinate values:

Coordinate	Meaning
[10,86]	Absolute values for X and Y
[52]	Absolute value for X with Y unchanged
[,121]	Absolute value for Y with X unchanged
[+10,100]	Relative value for X, absolute value for Y
[+15,–10]	Relative values for X and Y
[100,–25]	Absolute value for X, relative value for Y
[6.25,10.4]	Absolute embedded decimal values for X and Y
[.1E3,1000E–2]	Absolute exponential values for X and Y
[ ] or [+0,+0]	Current values for X and Y unchanged

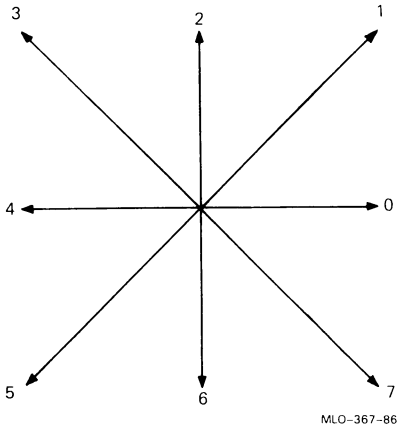
3.2.2 Pixel Vector System

Several ReGIS commands can use pixel vectors (PVs). The PV system provides for relative positioning or movement from one logical pixel to another.

The size of each logical pixel is determined by the screen addressing command S(A), which determines the extent and orientation of the image area. The default values are S(A[0,0][799,479]); this makes each logical pixel 1/800 of the image width. For example, if the screen addressing range were changed to S(A[0,0][499,499]), each logical pixel would be 1/500 of the image height or width.

As Figure 3–1 shows, PV movement can occur in eight different directions, each direction at 45-degree intervals. Each direction has an assigned number. If you specify the number associated with the direction desired, drawing or moving occurs in that direction in proportion to the number of times the PV value is specified.

Figure 3–1: Pixel Vector Directions



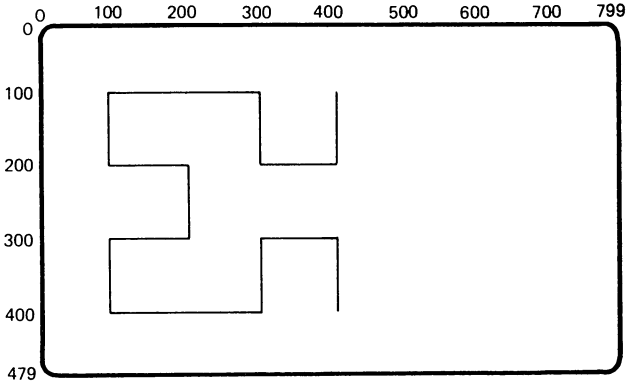
3.2.3 Pixel Vector Multiplier

In some PV applications, entering all of the required PV values is tedious. In such cases, you can use a PV multiplier command to simplify the task.

The PV multiplier command lets you specify a multiplication value for each PV entered in a command. For example, if you specify a multiplication value of 10, then each PV entered in later commands will cause moving or drawing for 10 logical pixels, not just 1.

Figure 3–2 shows an image drawn by using PV multiplication. In this figure, a Write command (W) sets a PV multiplication factor of 100 (M100). The image is then drawn by Vector commands (V), with each specified PV value multiplied by 100, providing the Vector commands for drawing the figure.

Figure 3–2: Pixel Vector Multiplication Example



NOTE:
THE CURRENT POSITION AT START OF COMMAND WAS (400,100); THE CURRENT POSITION AT THE END OF THE COMMAND IS (400,400).

COMMANDS
W(M100) V642446064600206

MLO-368-86

3.3 ReGIS Command Structure

The ReGIS data stream consists of standard ASCII characters, including letters, numbers 0 through 9, the at character (@), the space, and the following punctuation characters: semicolon (;); colon (:); decimal point (.); quotation marks, both single (') and double ("); parentheses (( )); brackets ([ ]); and comma (,). In quoted strings, ReGIS also accepts the following control characters: carriage return, backspace, linefeed, and horizontal tab.

Because the ReGIS interpreter processes ReGIS data serially, the order of the characters and the punctuation required to identify arguments, options, and suboptions are vital to accurate ReGIS processing. Except in quoted strings, ReGIS treats uppercase and lowercase letters the same. In general, a ReGIS string consists of a command keyletter followed by arguments.

ReGIS ignores inappropriate ReGIS commands and punctuation. RETOS ignores ReGIS commands that it does not support, as well as escape sequences, control sequences, device control sequence introducers, and unrecognized control characters. See Section 3.5 for the Report command and Screen Control command options not supported by RETOS.

3.3.1 ReGIS Commands

ReGIS has nine command types, which are represented by command keyletters. In addition, the macrograph (@) and resynchronization (;) characters affect the processing in a manner similar to the command keyletters. The macrograph character temporarily passes control from the current command to a macrograph. The semicolon terminates the current command. Table 3–1 briefly describes the command types, the macrograph, and the resynchronization character.

Table 3–1: ReGIS Command Summary

Command Keyletter	ReGIS Command	Description
S	Screen Control	Specifies screen controls, such as erasing the image.
P	Position	Positions the graphics cursor without performing any writing.
W	Write Control	Specifies writing controls, such as writing patterns.
V	Vector	Draws vectors (straight lines) between specified coordinate locations.
C	Curve	Draws circles, arcs, and curves, using coordinate locations specified in the command.
F	Polygon Fill	Fills in single closed figures, such as circles and squares.

Table 3–1 (Cont.): ReGIS Command Summary

Command Keyletter	ReGIS Command	Description
T	Text	Controls display of graphics text strings and lets you specify characters to display.
L	Load	Controls definition and loading of alternate characters that you can display by using the text command.
R	Report	Reports information, such as active position and error codes. RETOS does not support the Report command.
@	Macrograph	Defines a command string as a macrograph. You use macrographs to store and recall other ReGIS command strings. Macrographs let you store a complex figure that you may use more than once in a graphic image and select that figure with a single command.
;	Resynchronization	The semicolon serves as a resynchronization character for ReGIS command strings.

ReGIS command keyletters require no punctuation. When the ReGIS interpreter encounters a command keyletter, it assumes that all subsequent data is an argument to the command. The interpreter continues to process all data relative to the command currently in effect, until it encounters one of the following:

- A new command keyletter
- A semicolon, which is the resynchronization character that terminates the current command whether completed or not

3.3.2 ReGIS Command Arguments

ReGIS commands can have four types of arguments:

- Bracketed extents
- Quoted strings
- Digit strings
- Options (parenthesized extents)

Not all argument types apply to each command. Each argument type has punctuation that identifies it in the ReGIS stream. Using the proper punctuation is vital to accurate processing. The following sections describe each argument type and its accompanying punctuation.

3.3.2.1 Bracketed Extents

A bracketed extent is numeric data enclosed in brackets ([]). In ReGIS, brackets enclose the following types of numeric values:

- Coordinate position values
- Height and width values

Coordinate position values serve as arguments to commands, options, and suboptions. The values can represent an absolute value, a relative value, or a combination of the two.

Height and width values are arguments to Text and Load commands and represent relative displacement values for Text options.

3.3.2.2 Quoted Strings

Quoted strings can be any series of ASCII characters enclosed in quotation marks. ReGIS treats all characters in quoted strings as literals, including punctuation that normally functions as part of ReGIS syntax (semicolon and brackets, for example) and macrographs (@). Quoted strings can be any of the following:

- Text characters to process for display on the screen during Text command activity
- A printable character to use for shading

- A name given to a character set selected by a Text command
- A single ASCII character used as a call letter for Load command load cell arguments

In all cases, you can use double (") or single quotation marks ('); however, you must use matched pairs. The first quotation mark defines the start of the argument, while the second defines the end. If you begin a text string with a double quotation mark, ReGIS does not recognize a single quotation mark as the end of the argument but continues processing all data as a quoted string until it encounters a double quotation mark.

If you need to use quotation marks inside a quoted string, use the type not currently used as the delimiter. For example, ReGIS recognizes single quotation marks as a literal when they occur in a quoted string delimited by double quotation marks. To include a literal that is the same type of quotation mark as the delimiter, you can enter the character twice with no intervening spaces. The following examples clarify how ReGIS interprets quotes.

```
"A"             refers to the string A
'"'             refers to the string "
"'"             refers to the string '
'a''C'          refers to the string a'C
''''            refers to the string '
"A'""B"         refers to the string A'"B
"" or ''        refers to the empty string
```

While commas do not have explicit meaning in ReGIS syntax, they determine how arguments are interpreted. For example, ReGIS interprets consecutive string arguments 'ABC''DEF' as a single text string ABC'DEF. However, ReGIS interprets 'ABC','DEF' as ABCDEF.

3.3.2.3 Digit Strings

Numeric values not enclosed in brackets or quotation marks are digit strings. Most often these are pixel vectors, explained in Section 3.2.2. Digit strings that are not pixel vectors represent numeric values that may be signed. Most of these are forced to the nearest integer before use.

3.3.2.4 Options

Options are arguments that modify the action of the command keyletter. Parentheses define the boundaries of options and suboptions. The left parenthesis "(" defines the beginning of the option, suboption, or argument; the right parenthesis ")" defines the end.

ReGIS considers any letter not enclosed by parentheses, quotation marks, or brackets to be a command keyletter. Parentheses define the enclosed information as an option. ReGIS processes options in much the same way as it processes command keyletters. Once an option is introduced by a left parenthesis, ReGIS processes all subsequent data as arguments to that option until ReGIS receives a closing parenthesis.

Therefore, you must enclose suboptions with additional sets of parentheses. Otherwise, ReGIS assumes that the suboption is an option and tries to process it as such.

The following examples show ReGIS commands that use parentheses:

- S(E) — Erase option to the Screen Control command.
- W(I0,P3) — Foreground intensity and pattern select options to the Write Control command. The 0 and 3 are arguments to the options, which are separated by a comma.
- P(W(M100)) — Pixel vector (PV) multiplication temporary write option to the Position command. It uses the Write Control command as an option and the PV multiplication option of the Write Control command as a suboption.
- V(W(I(R))) — Temporary write option affecting the value of the foreground intensity to be used by a Vector command. It uses the Write command as an option. The foreground intensity option to the Write Control command is a suboption to the Vector command. The (R) argument to the I suboption is enclosed in a third set of parentheses, since this argument is a letter value.

As these examples show, you must use matching parentheses to control the levels of nesting of options and suboptions. The command V(W(I(R))) demonstrates this. The first parenthesis defines the start of option values. The second parenthesis defines the start of suboption values. The third parenthesis defines the start of sub-suboption values. The fourth parenthesis defines the end of sub-suboption values. The fifth parenthesis defines the end of suboption values. The sixth parenthesis defines the end of option values.

3.3.3 Punctuation Significant to ReGIS Syntax

3.3.3.1 Commas and Spaces

Commas separate position values in bracketed extents; commas and spaces separate option values in ReGIS commands. In most other cases, commas and spaces merely increase readability. However, in two cases besides bracketed extents, commas are necessary for correct processing. You need commas to separate two or more quoted strings, as ReGIS interprets consecutive quotation marks as a literal.

Another case involves any command identified by an E (such as a screen erase). If the E follows a numeric value, ReGIS interprets it as an exponential value, unless you insert a comma between the numeric value and the E command letter.

Commas and spaces are not part of a graphic image, unless specified in a quoted string.

3.3.3.2 Semicolon

ReGIS recognizes a semicolon (;) as a command for resynchronization. A semicolon in a command string causes ReGIS instructions to resynchronize to the top-level command state. For example, you would use the semicolon between command strings when transmission errors may be occurring. The semicolon cannot fix a garbled message, but it may reduce the effect of a single transmission error. You may want to include a semicolon at selected intervals. You should use a semicolon at the end of a Load command. The semicolon is not recognized as a resynchronization character when included in a quoted text string or when used as part of the Macrograph command syntax.

3.3.4 Control Characters

ReGIS recognizes four control characters *only* when used in a quoted string: carriage return (CR), linefeed (LF), backspace (BS), and horizontal tabs (HT). Because ReGIS ignores all control characters not in a quoted string, you can use linefeeds and carriage returns to define how command strings are displayed or printed. This makes your command strings easier to read without affecting the image.

3.3.5 ReGIS Default Values Summary

ReGIS commands have default values that apply when you invoke the converter. When you change these values, the new values remain in effect until you redefine them or exit from the converter. Table 3–2 summarizes ReGIS default values. The specified RETOS command qualifiers determine the default values; the default values in Table 3–2 are based on /DEVICE=LJ250. See Chapter 2 for a description of the effect of each RETOS command qualifier.

NOTE

Since /REVERSE_VIDEO is in effect for the LJ250 printer, the converter maps Lightness in reverse order. Lightness 100% is printed as black. Lightness 0% is printed as white.

Table 3–2: ReGIS Default Values

Command	Default Command	Default Description
Screen Control	S(A[0,0] [799,479])	Defines the screen as having coordinate values of [0,0] for upper left corner and [799,479] for lower right corner.
Screen Control	S(M0(H0L0S0)1(H0L50H60) 2(H120L46S72)3(H240L50S60) 4(H60L50S60)5(H300L50S60) 6(H180L50S60)7(L53)8(L26) 9(H0L46S29)10(H120L43S39) 11(H240L46S29)12(H60L46S29) 13(H300L46S29) 14(H180L46S29)15(L80))	Output map values are black for M0, blue for M1, red for M2, green for M3, etc. See Table 3–5.
Screen Control	S(I0)	Output map location 0 is used for background intensity value, with white background (default value for M0).
Write Control	W(P1)	Solid line selected for writing pattern.
Write Control	W(P(M2))	Pattern multiplication factor of 2.

Table 3–2 (Cont.): ReGIS Default Values

Command	Default Command	Default Description
Write Control	W(M1)	Pixel vector (PV) multiplication factor of 1.
Write Control	W(N0)	Negative pattern control disabled.
Write Control	W(I3)	Output map location 3 selected for write tasks. This results in green, since this is the default translator value for M3.
Write Control	W(V)	Overlay writing in effect.
Write Control	W(S0)	Shading disabled.
Text	T(A0)	Character set containing standard DEC Multinational characters is selected for text processing.
Text	T(A0(L"(B"))	Load ASCII into the left half of the alphabet 0 code table.
Text	T(A0(R"(<"))	Load DEC Supplemental (right half of DEC Multinational) into the right half of the alphabet 0 code table.
Text	T(S1)	Standard character cell size 1 is selected for text processing.
Text	T(S[9,20])	Display cell size associated with standard character cell size 1.
Text	T(U[8,20])	Unit cell size associated with standard character cell size 1.
Text	T[+9,+0]	Character escapement associated with standard character cell size 1.

Table 3–2 (Cont.): ReGIS Default Values

Command	Default Command	Default Description
Text	T(H2)	Height multiplication factor of 2.
Text	T(D0 S1 D0)	String and character tilt disabled.
Text	T(I0)	Italics disabled.
Load	L(A1)	Alphabet 1 selected for loading.

3.3.6 Conventions Used in ReGIS Commands

The following conventions apply to the explanations of ReGIS commands:

- Angle brackets (< >) indicate that you can select different values. The <values> in the brackets define the type of information you can use, but the brackets are not part of ReGIS syntax.
- [X,Y] indicates you can select coordinate position values. The brackets are part of the ReGIS syntax. X and Y are variables for a coordinate position. This position can have both X and Y values, just an X value, or just the Y value. (See Section 3.2.1.)
- This manual uses uppercase letters for clarity. However, you can use either uppercase or lowercase letters with ReGIS commands. Except in quoted strings, ReGIS treats both cases the same.

3.3.7 Conventions Used in ReGIS Examples

The following conventions apply to the examples in this chapter:

- Examples of the Vector and Curve commands show the position of the cursor. This is for information only; the cursor does not appear when the example is converted from ReGIS to sixel graphics and printed.
- Examples of shaded figures show the shading when the examples are converted from ReGIS to sixel graphics and printed. This is the reverse of how figures are shaded when they are displayed on the screen.

3.4 ReGIS Commands Supported by RETOS

The ReGIS to sixel graphics converter, RETOS, supports the following ReGIS commands:

- Screen Control
- Position
- Write Control
- Vector
- Curve
- Polygon Fill
- Text
- Load
- Macrograph

See Section 3.5 for ReGIS commands that RETOS does not support.

Screen Control

Screen Control command arguments either set parameters and attributes for the whole image or execute actions affecting the whole image. RETOS supports five Screen Control command arguments:

- Display addressing
- Background intensity
- Output mapping
- Screen erase
- Page eject

Refer to Section 3.5 for unsupported Screen Control commands.

Format

S *option*

Command Arguments	Description
(A[X,Y][X,Y])	Display Addressing
(I(...)) or (I <n>)	Background intensity select
(M <n> (...))	Output mapping
(E)	Erase (used with I to set background)
(F)	Page eject option

Command Arguments

(A[X,Y][X,Y])

The display address option defines the addressable extent of the image area. This lets you run ReGIS code written for ReGIS devices with different address ranges, without having to convert the coordinates.

The first pair of bracketed extents indicates the coordinate values for the upper left corner. The second pair indicates the coordinate values for the lower right corner. If either position specifier is missing, ReGIS ignores the command.

The default coordinate system is [0,0] for the upper left corner and [799,479] for the lower right for compatibility with the VT240. This means that RETOS scales pictures so that 800 logical units correspond horizontally to 720 or 1440 pixels in the output file for /DEVICE=LJ250. However, if you design a picture for display on the LJ250 printer and a one-to-one correspondence between a logical unit and a pixel is important, then begin your file with the following Screen Control command arguments:

- S(A[0,0][719,944]) for low resolution (90 dots/inch)
- S(A[0,0][1439,1889]) for high resolution (180 dots/inch)

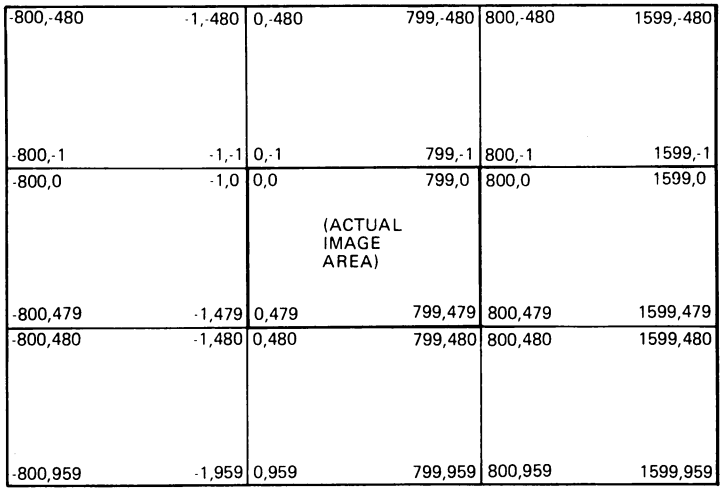
You can use exponential numbers as well as decimal numbering. The ratio of the defined area should be as close as possible to the aspect ratio of the presentation area.

In mapping a specified display addressing into the image area, ReGIS maintains the picture aspect ratio. Squares are always square, and angles are correctly drawn, regardless of the addressing parameters.

Figure 3–3 shows the effective address range when default values are in place.

Negative addresses are valid, although they may not be addressed directly. You cannot specify an absolute negative address. However, you can specify a relative value that results in a negative address; the negative address is valid, provided it does not exceed the address range. If it exceeds the address range, wraparound may occur.

Figure 3–3: Effective Default Address Range



MLO-369-86

There is no restriction on the relative values of the left, right, top, and bottom margins. If the right margin value is less than the left margin value, the X coordinate increases to the left instead of to the right, as it would in the default coordinate value system. If the bottom margin value is less than the top margin value, the Y coordinate increases upward instead of downward.

NOTE

Pixel vector (PV) magnitude values depend on the display addressing values and the PV multiplier. However, PV directions are independent of addressing orientation. For example, 0 is always to the right.

RETOS uses the screen addressing in effect at the end of the ReGIS file to determine what part of the page to write to the output file. If you changed the screen addressing during your drawing, part of the picture may be outside the final addressable area and will not appear in the output file.

(I <n>)
(I(RGB))
(I(H <n> L <n> S <n>))

The background intensity option lets you select the shade of the background color. Used alone, this command does not change the appearance of the image. It sets up the color to be used in (1) screen erase and (2) replace and erase writing modes. You can use two methods for this selection:

- Provide an RGB or HLS specifier value
- Provide the output map location number (0 to 15), which selects the shade stored in that location

The first method explicitly selects a color. Use this method when portability to other ReGIS devices is a consideration.

The second method selects the intensity stored in a specific output map location. This method is provided for compatibility for devices with limited output maps.

The RGB (red/green/blue) specifier system in Table 3–3 uses a single letter to specify any one of eight different colors or shades:

Table 3–3: RGB Color Specifier System

RGB Specifier	HLS Equivalent H	L	S	Color
D	0	0	0	Dark (black)
R	120	50	100	Red
G	240	50	100	Green
B	0	50	100	Blue
C	300	50	100	Cyan (a secondary color made from an equal mixture of blue and green)
Y	180	50	100	Yellow (a secondary color made from an equal mixture of red and green)

Table 3–3 (Cont.): RGB Color Specifier System

RGB Specifier	HLS Equivalent H	L	S	Color
M	60	50	100	Magenta (a secondary color made from an equal mixture of red and blue)
W	0	100	0	White

If your output device is /MONOCHROME, such as the VT240, RETOS converts the RGB color specifiers to appropriate gray values. If your device is /NOCOLOR or black and white, such as the LN03, RETOS sends a 1 to the output file for all colors except the background color. The background color becomes a 0, and therefore no color information reaches the sixel graphics output file.

The HLS (hue/lightness/saturation) specifier system provides more colors. It uses different values of hue (H), lightness (L), and saturation (S). RETOS passes HLS colors unchanged to the output file. Table 4 lists HLS colors and shades, along with the H, L, and S values corresponding to each shade.

Table 3–4: HLS Color Specifier System

Color	H	L	S
Aquamarine	260	65	60
Aquamarine, medium	280	50	60
Black (dark)	0	0	0
Blue	0	50	100
Blue, cadet	300	50	25
Blue, cornflower	0	35	25
Blue, dark slate	40	35	60
Blue, light	300	80	25
Blue, light steel	0	65	25
Blue, medium	0	50	60
Blue, medium slate	30	50	100

Table 3–4 (Cont.): HLS Color Specifier System

Color	H	L	S
Blue, midnight	0	25	25
Blue, navy	0	35	60
Blue, sky	320	50	60
Blue, slate	330	50	100
Blue, steel	320	35	60
Coral	150	50	100
Cyan	300	50	100
Firebrick	120	35	60
Gold	150	50	60
Goldenrod	180	65	60
Goldenrod, medium	180	80	60
Green	240	50	100
Green, dark	240	25	25
Green, dark olive	180	25	25
Green, forest	240	35	60
Green, lime	240	50	60
Green, medium forest	200	35	60
Green, medium sea	240	35	25
Green, medium spring	210	50	100
Green, pale	240	65	25
Green, sea	280	35	60
Green, spring	270	50	100
Green, yellow	200	50	60
Gray, dark slate	300	25	25
Gray, dim	0	33	0
Gray, light	0	66	0
Khaki	180	50	100
Magenta	60	50	100
Maroon	80	35	60

Table 3–4 (Cont.): HLS Color Specifier System

Color	H	L	S
Orange	120	50	60
Orchid	60	65	60
Orchid, dark	40	50	60
Orchid, medium	20	65	60
Pink	120	65	25
Plum	60	80	60
Red	120	50	100
Red, indian	120	25	25
Red, medium violet	100	65	60
Red, orange	90	50	100
Red, violet	80	50	60
Salmon	120	35	25
Sienna	160	35	60
Tan	140	65	60
Thistle	60	80	25
Turquoise	300	80	60
Turquoise, dark	340	65	60
Turquoise, medium	300	65	60
Violet	60	25	25
Violet, blue	60	50	25
Wheat	180	80	25
White	0	99	0
Yellow	180	50	100
Yellow, green	220	65	60

NOTE

The color names are only rough approximations of the specific shades. The actual appearance of each color depends on the output device.

(M <n> (L <n>))
(M <n> (<RGB>))
(M <n> (H <n> L <n> S <n>))

The output mapping option lets you change the values in the entries of the output map. The output map for this converter has 16 entries. Each entry stores a color value. This option is provided for compatibility with devices that provide an output map. RETOS chooses HLS values for each color so that the LJ250 output matches terminal colors as closely as possible.

Output map locations are numbered 0 through 15. The default values for the 16 color entries are listed in Table 3–5.

Table 3–5: Default Color Output Map Values

Output Map Entry	Default Converter Value	HLS Value		
		H	L	S
0	Black	0	0	0
1	Blue	0	50	60
2	Red	120	46	72
3	Green	240	50	60
4	Magenta	60	50	60
5	Cyan	300	50	60
6	Yellow	180	50	60
7	Gray 50%	0	53	0
8	Gray 25%	0	26	0
9	Pale blue	0	46	29
10	Pale red	120	43	39
11	Pale green	240	46	29
12	Pale magenta	60	46	29
13	Pale cyan	300	46	29
14	Pale yellow	180	46	29
15	Gray 75%	0	80	0

Although the LJ250 printer supports up to 256 colors, the ReGIS color map provides only 16 entries. When an S(I(<hls>)) or W(I(<hls>)) command requests a color not in the color map, and the /COLORS qualifier value is greater than 16, RETOS uses the first available color

map entry above 15 so that it can specify the exact color requested in the output file. If no unused color map entries exist or if the output device supports 16 or fewer colors, RETOS approximates the requested color with the closest color already in the color map.

When you select a device with the /MONOCHROME qualifier, RETOS initializes the color map as a gray scale and writes only the lightness component of colors to the output file. With the /NOMONOCHROME qualifier, RETOS initializes the color map with entries as shown in Table 3–5 and writes full HLS colors to the output file.

With the /NOCOLOR qualifier, RETOS produces black and white rather than colored sixels. The converter writes a 0 to the output file for pixels drawn in the background color and a 1 for pixels drawn in any other color. The output file contains no color information. Devices such as the LN03 PLUS and LA75 printers require the /NOCOLOR qualifier.

Command Structure for Changing Value

Follow these steps to change the value of an output map location:

1. Specify the output map location: 0 to 15.
2. Specify a new color value. Use either a single letter of the RGB color specifier system (see Table 3–3) or the hue, lightness, and saturation values of the HLS color specifier system (see Table 3–4).

The following example shows the command syntax to change an output map's value:

```
S(M1(L25)2(L99)3(L50))
```

(E)

The screen erase option lets you erase the screen by setting the whole screen to the display background color. This option does not change either the current position or the values in the output map.

To change the background color, combine the screen erase option with the background intensity option.

RETOS does not interpret the screen erase option to mean that a page should be printed.

(F)

The page eject option prints the current image. An implied S(E) command occurs after each S(F) command.

The end of a file implies an S(F) command; however, an S(F) command at the end of a file does not produce a blank page. Successive S(F) commands do not eject blank pages.

3.4.1 Screen Control Command Summary

Table 3–6 summarizes the Screen Control command arguments that RETOS supports, including default values associated with the arguments (assuming /DEVICE=LJ250 is in effect).

Table 3–6: Screen Control Command Summary

Argument	Default	Description
(A[X,Y][X,Y])	[0,0][799,479]	Display addressing. Lets you define addressing at a different size or orientation than the default.
(I <n>)	(I0)	One of three background intensity select options.
(M <n> (HLS))	0(H0L0S0) 1(H0L50S60) 2(H120L46S72) 3(H240L50S60) 4(H60L50S60) 5(H300L50S60) 6(H180L50S60) 7(H0L53S0) 8(H0L26S0) 9(H0L46S29) 10(H120L43S39) 11(H240L46S29) 12(H60L46S29) 13(H300L46S29) 14(H180L46S29) 15(H0L80S0)	Output mapping option for changing color values. You can change any or all values in a given option. Defines the color value to store in selected <n> output map location.
(E)	None	Screen erase option. Rewrites the whole image at current background intensity.
(F)	None	Page eject option. Prints the current image.

Position

Position commands let you select a new current position without writing. The three basic command arguments are as follows:

- Coordinate or pixel vector arguments
- Position stack options
- Temporary write control option

Format

P *option*

Command Arguments	Description
[X,Y]	Position argument
<PV>	Position argument, using PV values
(B)	Begin bounded sequence option
(S)	Begin unbounded sequence option
(E)	End of sequence option
(W(M <n>))	Temporary write control option

Command Arguments

[X,Y]
<PV>

These arguments let you select a current position before performing other ReGIS functions. You can use four types of positioning:

- Absolute
- Relative
- Absolute/relative
- PV offset

Absolute positioning uses absolute X and Y coordinate values to define a new current position. You can specify absolute positioning in three ways:

- Specify new X and Y coordinates
- Specify only a new X coordinate (with the Y coordinate unchanged)
- Specify only a new Y coordinate (with the X coordinate unchanged)

The three formats for the absolute positioning argument are as follows:

```
P[X,Y]
P[X]
P[,Y]
```

NOTE

Position commands do not cause drawing. The lines in the diagrams represent only the movement that occurs.

Relative positioning uses negative and positive values to define a new current position relative to its current position. You can specify relative positioning in three ways:

- Specify relative positioning on both X and Y axes
- Specify relative positioning on the X axis only
- Specify relative positioning on the Y axis only

Relative position values always start with a plus (+) or minus (–) sign. A positive value is added to the value of the current position coordinate to be affected; the resulting value becomes the absolute value of the new location. A negative value is subtracted to arrive at the new absolute value. However, the direction of change depends on the screen addressing orientation.

The relative positioning argument can take the following eight forms:

```
P[+X,+Y]
P[+X,-Y]
P[-X,+Y]
P[-X,-Y]
P[+X]
P[-X]
P[,+Y]
P[,-Y]
```

You can define a new current position with a combination of absolute and relative X and Y coordinate values. This combination of Position command values can take two basic forms:

- An absolute X value with a relative Y value
- A relative X value with an absolute Y value

The pixel vector (PV) positioning form of the Position command uses PV values to define a new current position. PV moves are relative to the old current position.

PV moves use the current PV multiplication factor. If you want a different multiplication factor, you can use a Write Control command to change the current PV multiplication or a PV multiplication temporary write control option. The value defined by the temporary write control option is in effect only until you use a new keyletter (including a new P command keyletter) or another temporary write control option.

The format for the PV positioning argument is as follows:

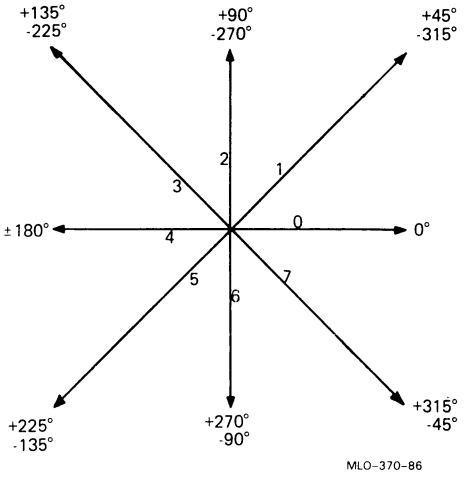
```
P<pv value>
```

The format for the PV multiplication temporary write control option is as follows:

```
P(W(M<multiplication value>))<pv value>
```

Figure 3–4 shows the directions associated with each of the PV offset numeric values (0 through 7).

Figure 3–4: Pixel Vector Direction Values



(B)

Both bounded and unbounded sequences let you group sets of position specifiers into position blocks that are processed as units. Both consist of either a begin bounded (B) or begin unbounded (S) option and an end (E) option. Usually, other commands, such as Vector (V) or Curve (C) commands, are embedded between the sequence begin and end options. As such, these sequences are useful for such ReGIS tasks as polygon definition and shading. The format for a bounded sequence is as follows:

```
P(B)<embedded command>(E)
```

A bounded sequence returns the current position to a specific starting point at the end of the sequence.

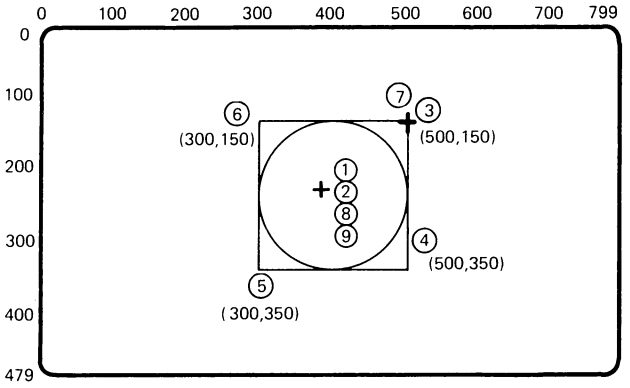
A bounded sequence consists of a minimum of one begin (B) option and one end (E) option. You should repeat the Position command keyletter (P) before the final (E) option, because the embedded commands usually contain other command keyletters. If you do not repeat P, the last command keyletter in the embedded commands sequence becomes the current command. You can save up to 16 positions. For each (B) option, there must be an (E) option. For example, if you use five (B) options in a graphic image, you need five (E) options to return the active position to the original saved position.

NOTE

Position values are also saved during Position command unbounded sequences, as well as Vector command bounded and unbounded sequence options. The limit on the number of unended, saved position values (including all save commands) is 16. However, for transportability, use a maximum of eight.

Figure 3–5 shows an example of how to build a simple graphic image with a Position command bounded sequence. The example includes Vector (V) and Curve (C) commands.

Figure 3–5: Bounded Sequence Example



COMMANDS	
(1)	P[400,250]
(2)	(B)
(3)	[+100,-100]
(4)	V[,+200]
(5)	[-200]
(6)	[,-200]
(7)	[+200]
(8)	P(E)
(9)	C[+100]

NOTE:
ONCE A COMMAND LETTER IS SPECIFIED, IT DOES NOT NEED TO BE RESPECIFIED, UNLESS ANOTHER TYPE OF COMMAND COMES BETWEEN COMMAND OPTION VALUES.

MLO-371-86

(S)

The difference between the bounded and unbounded sequences is the start (S) option. In bounded sequences, the (B) option tells ReGIS to save the current position and return to that position after a corresponding (E) option. In the unbounded sequence, the (S) option tells ReGIS to save a dummy, or nonexistent position. When ReGIS comes to a corresponding (E) option, the position does not change from the last specified current position. The unbounded sequence is provided primarily for symmetry

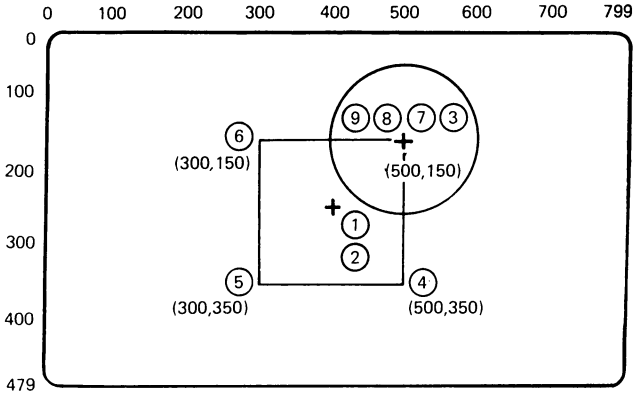
with other command types, such as Curve commands, that can use bounded and unbounded sequences.

The format for an unbounded sequence is as follows:

```
P(S)<embedded commands>(E)
```

With an unbounded sequence, you should repeat the P command keyletter before the final (E) option to ensure that the command keyletter remains P. Figure 3–6 shows an unbounded sequence with the same Vector (V) and Curve (C) commands used in the bounded sequence in Figure 3–5. Comparing these figures shows the different results obtained by using bounded and unbounded sequences.

Figure 3–6: Unbounded Sequence Example



COMMANDS	
(1)	P[400,250]
(2)	(S)
(3)	[+100,-100]
(4)	V[,+200]
(5)	[-200]
(6)	[,-200]
(7)	[+200]
(8)	P(E)
(9)	C[+100]

NOTE:
ONCE A COMMAND LETTER IS SPECIFIED, IT DOES NOT NEED TO BE RESPECIFIED, UNLESS ANOTHER TYPE OF COMMAND COMES BETWEEN COMMAND OPTION VALUES.

MLO-372-86

(E)

The end of sequence option is used to end both bounded and unbounded sequences. This option refers back to the most recently stored (B) or (S) option value. If the last value was stored by a (B) option, the active position is defined by the stored value. If the last value was stored by an (S) option, the active position remains at its current location.

For both bounded and unbounded sequences, you should repeat the P command keyletter before the final (E) option to ensure that the command keyletter remains P.

(W(M <n>))
The temporary write control option defines the multiplication factor for PV values. It defines the number of coordinates affected by PV values specified by a PV move argument.

3.4.2 Position Command Summary

Table 3–7 summarizes the Position command arguments.

Table 3–7: Position Command Summary

Argument	Description
[X,Y]	Cursor position argument using [X,Y] values to define a new active position. The [X,Y] values can be absolute, relative, or absolute/relative.
<PV>	Cursor positioning argument using PV values to define a relative repositioning of the active position.
(B)	Begin a bounded sequence option. Stores the current active position for reference at the end of the sequence.
(S)	Start an unbounded sequence option. Stores a dummy position for reference at the end of the sequence.
(E)	End of sequence option. Selects last stored (B) or (S) option value for reference.
(W(M <n>))	Temporary write control option defining multiplication factor for PV values. Defines number of coordinates affected by PV values specified by a PV move argument.

Write Control

Write Control command options let you set attributes and parameters used at the pixel level during writing tasks. RETOS supports the following tasks performed by the Write Control command options:

- PV multiplication
- Foreground intensity selection
- Writing modes: erase, overlay, replace, complement, and negate
- Pattern control
- Shading control

You can set write controls by using other commands, for example, Vector, Curve, Screen Control, and Position commands, as temporary write control options. For more information, see the sections on these commands.

Format

W *option*

Command Arguments	Description
(M <n>)	PV multiplication
(I(...)) or (I <n>)	Foreground intensity select
(E/R/V/C)	Type of writing: Erase, Replace, Overlay, Complement
(P <pattern>)	Pattern
(P(M <n>))	Pattern multiplication
(N <0 or 1>)	Negative pattern control
(S <0 or 1>) or (S" <char> ")	Shading

Command Arguments

(M <n>)

The PV multiplication option lets you define a multiplication factor for PV values used in moving and drawing. PV values are then multiplied by the defined factor. The format for the PV multiplication option is as follows:

```
W(M<n>)
```

<n> is the numeric value defining the multiplication factor.

You can also use the PV multiplication option as a temporary write control option with other commands, such as Position, Screen Control, Vector, and Curve commands. In these cases, you can leave the overall PV multiplication value unchanged but select a temporary multiplication value for a specific task.

(I <n>)
(I(<RGB>))
(I(H <n> L <n> S <n>))

The foreground intensity option is identical in form to the Screen Control background intensity option, except the options start with different command keyletters (W for write control, S for screen control). However, the options have different functions. The screen control option selects the shade for background, while the write command option selects the shade you use for writing on that background.

The foreground intensity option affects only the shade/color of writing done after the option is invoked. This feature lets you select different colors for different parts of a graphic image without affecting other parts of the same image.

You must select a writing shade that differs from the selected background to make sure that the foreground is visible.

The foreground intensity option can select shades from the output map only if you use the form W(I <n>). Otherwise, the actual specified color is sent to the output file.

(E)

You can use the erase writing option (1) by itself, (2) with negative pattern control on or off, (3) with a foreground intensity value, or (4) in any combination of these options. The function of erase writing depends on all of these options. If you use erase writing by itself, it sets any pixels that were written. In that case, erase writing changes the erased area to the currently selected background color/shade value (assuming negative pattern control is off, which is the default).

The format for a basic erase writing option is as follows:

```
W(E)
```

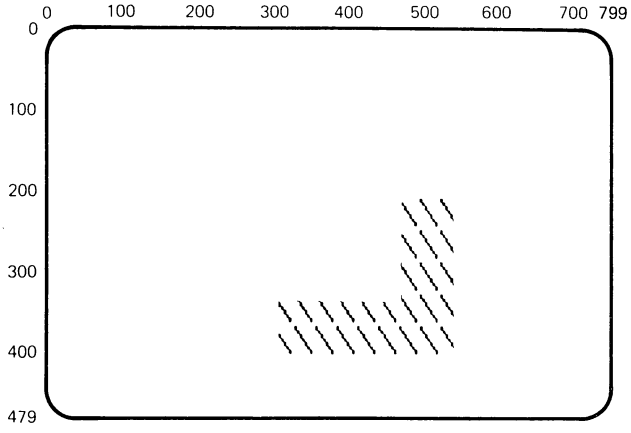
If you specify negative pattern control as on (N1), erase writing changes the erased area to the currently selected foreground color value. Figure 3–7 shows the effect of negative pattern control on erase writing. Example A shows how the square is erased to the background color

when negative pattern control is off. Example B shows how the square is erased to the foreground color when negative pattern control is on.

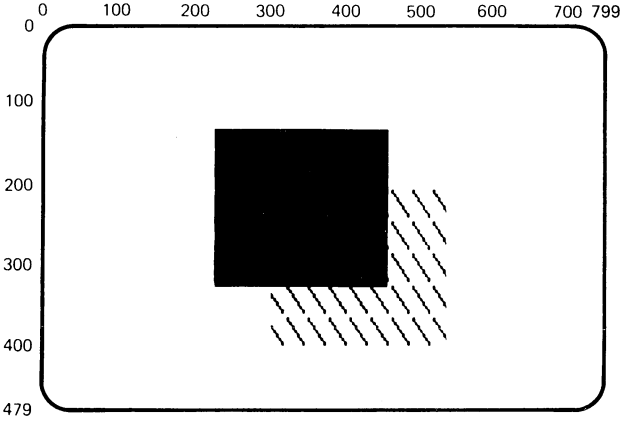
When you use erase writing with the foreground intensity option, you can write at the newly defined foreground value — as long as negative pattern control is on. If negative pattern control is off, the foreground intensity option changes the foreground value for later writing activity. However, the erase writing option still uses the background shade value. Figure 3–8 shows the effect that the foreground select option can have on erase writing when negative pattern control is on.

Erase writing ignores the pattern selections, back line pattern, and shading pattern.

Figure 3–7: Erase Writing with Negative Pattern Control



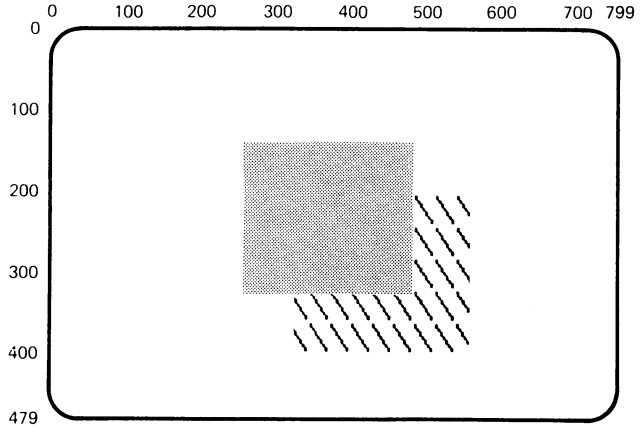
A) PATTERNED SQUARE PARTIALLY ERASED USING W(N0,E), OR W(E) IF NEGATIVE WRITING OFF (DEFAULT OR PREVIOUS N0)



B) PATTERNED SQUARE PARTIALLY ERASED USING W(N1,E), OR W(E) IF NEGATIVE WRITING ON (PREVIOUS N1)

MLO-373-86

Figure 3–8: Erase Writing with Foreground Specification



(R)

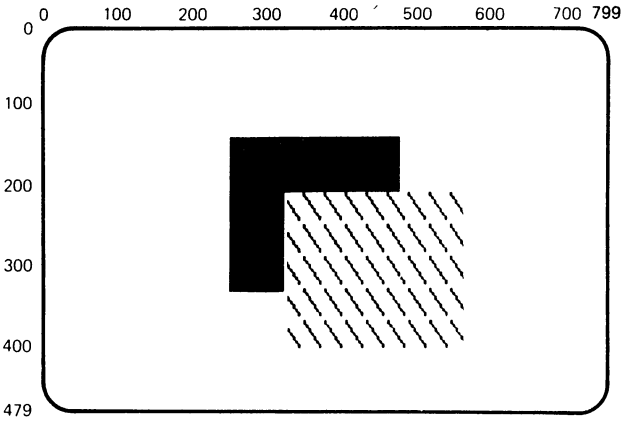
In replace writing, 1s in the bit pattern memory write the foreground intensity; 0s in the bit pattern memory write the background intensity.

The format for the replace writing option is as follows:

```
W(R)
```

Figure 3–9 shows an example of a graphic image created by using replace writing.

Figure 3–9: Replace Writing Example



BLACK SQUARE IS DRAWN FIRST. THEN PATTERNED SQUARE IS DRAWN IN REPLACE MODE.

MLO-375-86

(V)

Bit map values do not change for those parts of the new image defined by 0s in pattern memory. A change occurs only for those parts of the new image defined by 1s in pattern memory. The foreground intensity replaces the old bit map value for all pixels defined as 1s in the new image.

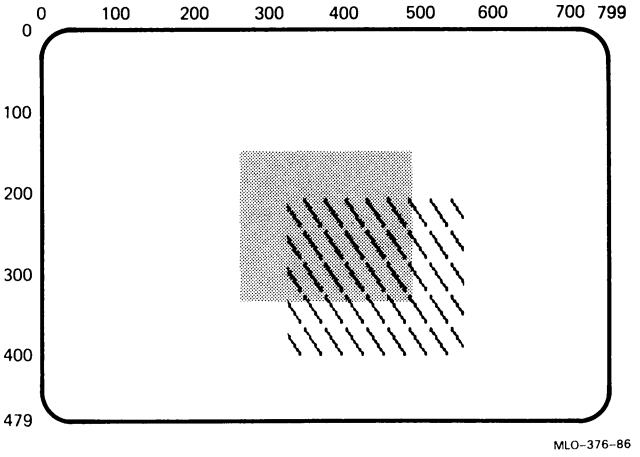
Because overlay is the default, you do not have to use the overlay option, unless erase or replace writing has occurred. If you use one of those forms of writing control, the overlay writing option lets you return to the default mode. However, it is good practice to specify overlay writing, as you cannot always be sure of the current writing mode.

The format for the overlay writing option is as follows:

```
W(V)
```

Figure 3–10 shows an example of overlay writing. Figure 3–10 uses the same basic graphic image used for the erase and replace writing examples; however, the square is shaded light gray, rather than dark, so that the overlay is visible.

Figure 3–10: Overlay Writing Example



(C)
Complement writing allows writing over another image in the opposite shade.

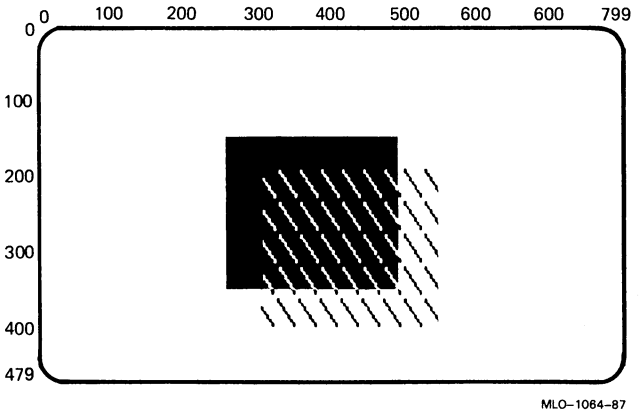
Figure 3–11 shows a simple example of complement writing. In this example, a pattern written in black partially overwrites a shaded black square. Where the pattern does overwrite, the pattern is displayed as white; where the pattern does not overwrite, the pattern remains black. However, complement writing occurs even where the pattern does not overwrite the square. The white background is being complemented, resulting in the black pattern.

Complement writing is NOT recommended for color output printers. Complement writing affects bits stored in a bit map. This effect does not necessarily produce the complement (opposite) of the selected shade/color.

If the output map contains 16 colors, the bit map provides four bits of memory for each pixel. These four bits provide a code identifying one of 16 output map locations. That output map location defines the shade/color for the associated pixel.

During complement writing, each bit of a code that is overwritten changes to its opposite value. The code 0100 (an address for output map location 4) changes to 1011 (the address for output map location 11). The resulting shade/color depends on the shade/color stored in the complemented output location that is addressed.

Figure 3–11: Complement Writing Example



(P <pattern>)

RETOS uses an 8-bit wide pattern memory. The contents of this memory let you define the appearance of lines and shaded areas. This memory is read to control the appearance of the pixels in a graphic object. For example, a vector command draws a line. As the line is drawn, the pattern memory is read, bit by bit, to determine if a pixel should be on (1) or off (0). In replace writing mode, a 1 value sets the pixel to the foreground shade value, and a 0 value sets the pixel to the background

shade value. In the case of negative pattern control, settings are reversed. See the description of (N <0 or 1>) in this section.

The writing cycles through the 8-bit pattern, unless you use a new command keyletter. If you want successive Vector or Curve commands to start at the first position of pattern memory, start them with the command keyletter.

The default for pattern memory is all 1s. Therefore, during a typical drawing process, the line is defined by having all pixels turned on to the foreground shade. Pattern control consists of options that let you change the pattern in four ways:

- Select standard pattern
- Specify binary pattern
- Pattern multiplication
- Negative pattern control

Select standard pattern and specify binary pattern both use the pattern select command option to define a pattern. However, they specify different values in the pattern select option.

Ten standard write patterns are available: 0 through 9. The format for the standard pattern select option is as follows:

```
W(P<pattern number>)
```

You can select any of the 10 standard patterns by using the standard pattern select option. Table 3–8 identifies the bit configurations for the standard patterns. Figure 3–12 shows how the various standard patterns appear on the screen. Figure 3–13 shows how these patterns are invoked in a vector that is 24 pixels long.

Table 3–8: Standard Pattern Memory Descriptions

Pattern Number	Binary Pattern	Description
0	00000000	All-off write pattern
1	11111111	All-on write pattern
2	11110000	Dash pattern
3	11100100	Dash-dot pattern
4	10101010	Dot pattern
5	11101010	Dash-dot-dot pattern
6	10001000	Sparse dot pattern
7	10000100	Asymmetrical sparse dot pattern
8	11001000	Sparse dash-dot pattern
9	10000110	Sparse dot-dash pattern

Figure 3–12: Standard Patterns Display

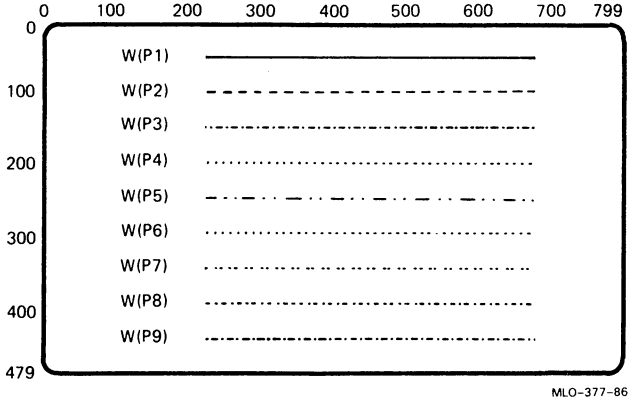
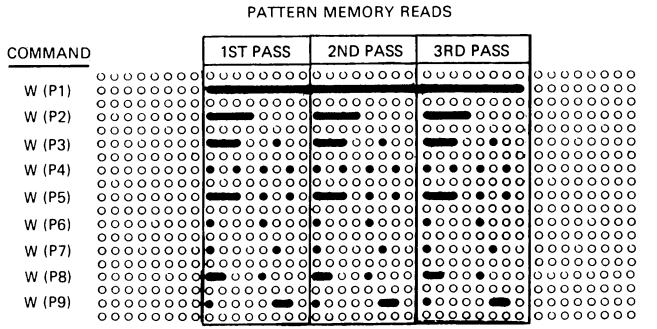


Figure 3–13: Standard Patterns



NOTES

1. P0 IS NOT SHOWN; ITS VALUE IS ALL 0'S.
2. ALL PATTERNS ARE SHOWN WITH MULTIPLICATION VALUE OF 1, WITH NEGATIVE PATTERN CONTROL OFF.
3. ALL PATTERNS ARE SHOWN FOR THREE READS OF PATTERN MEMORY, WITH FIRST READ STARTING AT THE FIRST BIT.

MLO-378-86

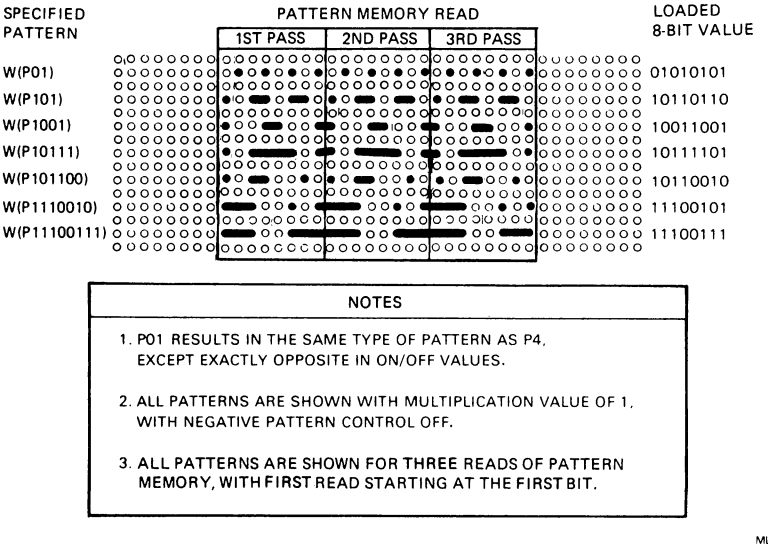
You can select unique patterns not available as standard patterns by using a specified pattern select option. The format of this option is similar to that of the standard pattern select option, except that the value you specify is a specific binary pattern instead of a standard pattern number. The format used for the specified pattern select option is as follows:

```
W(P<binary pattern>)
```

The specified binary pattern can be up to 8 bits long, the maximum size of the pattern memory. If you specify a pattern that is greater than 8 bits, only the last 8 bits are used. Pattern cells of 1, 2, 4, and 8 bits are repeated as full subunits in the 8-bit pattern memory. However, patterns of 3, 5, 6, and 7 bits are repeated only as far as possible within the 8-bit limitation.

Figure 3–14 shows examples of patterns you can create by using the specified pattern select option. The figure shows how these patterns are invoked in a vector 24 pixels long and how patterns of 3, 5, 6, and 7 bits do not repeat as complete subunits.

Figure 3–14: Examples of Binary Patterns



(P(M <n>))

Pattern multiplication lets you change the appearance of a pattern by specifying the number of pixels to be affected by each bit in the 8-bit pattern memory. The minimum value is 1. The default value is 2. For portability to other ReGIS devices, you should use a maximum value of 8. However, the converter supports higher values.

The two basic forms of the pattern multiplication suboption are as follows:

- Standard pattern:

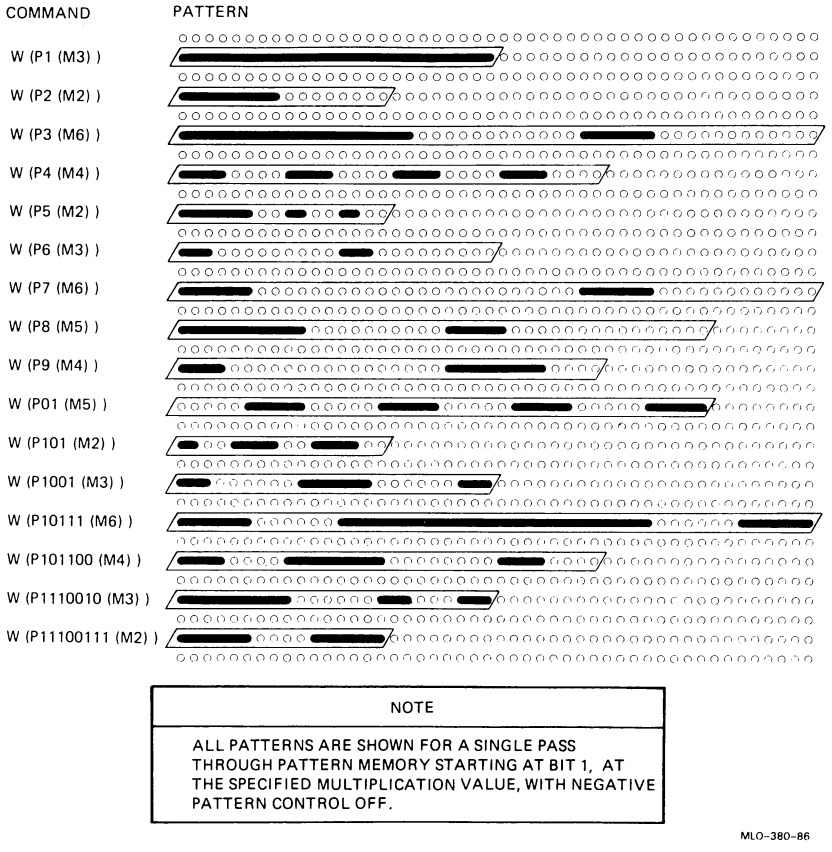
```
W(P4(M5))
```

- Specified binary pattern:

```
W(P11000011(M3))
```

Figure 3–15 shows how the pattern examples from Figure 3–13 and Figure 3–14 are affected by multiplication values.

Figure 3–15: Pattern Multiplication



(N <0 or 1>)

Negative pattern control lets you reverse the effect of pattern memory. The default value for negative pattern control is off. The format for the negative pattern control option is as follows:

```
W(N<0 or 1>)
```

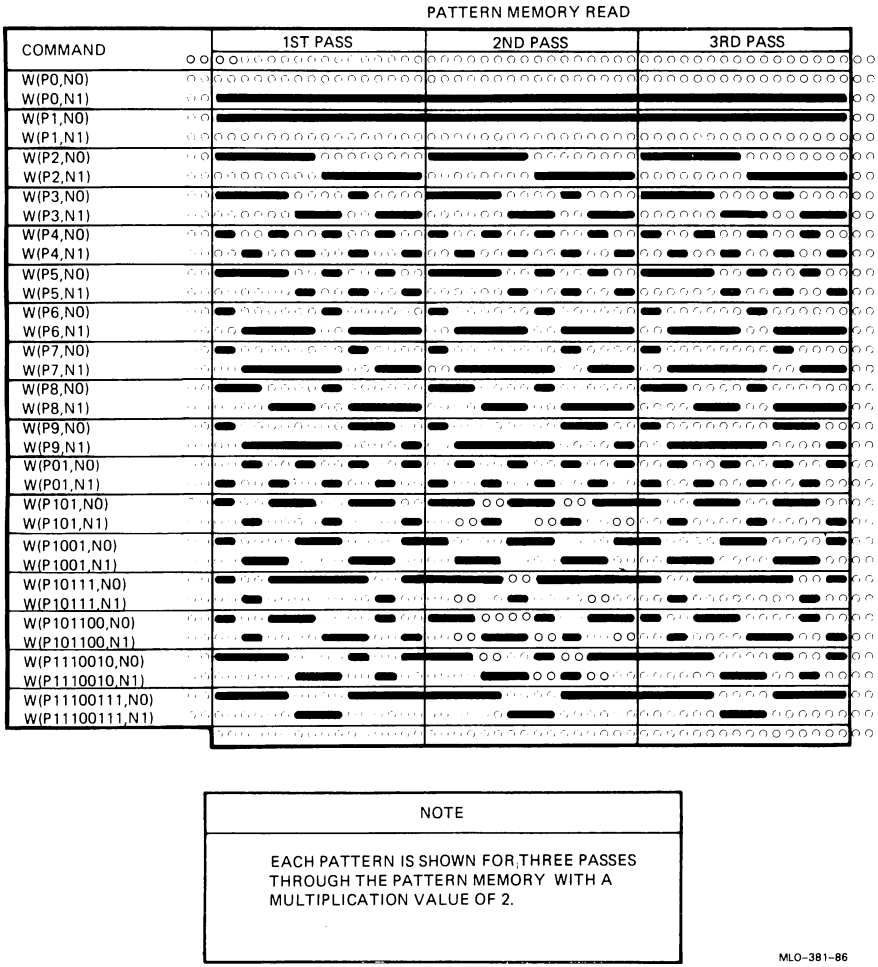
During normal writing conditions in replace mode, 1s in the pattern memory define the pixels as having the foreground shade; 0s define the pixels as having the background shade. With negative pattern control on, the reverse is true: 1s select background; 0s select foreground. You can use negative pattern control with all writing modes.

NOTE

Negative pattern control functions differently with erase mode writing. See the description of (E) in this section.

Figure 3–16 shows how the negative pattern control on and off conditions affect various patterns. The patterns shown are the same standard patterns from Figure 3–13 and the specified binary patterns from Figure 3–14.

Figure 3–16: Negative Pattern Control



NOTE

EACH PATTERN IS SHOWN FOR THREE PASSES THROUGH THE PATTERN MEMORY WITH A MULTIPLICATION VALUE OF 2.

MLO-381-86

(S <0 or 1>)
(S" <char> ")

The shading control option lets you shade the inside of a graphic object as it is drawn. During shading commands, Vector and Curve commands operate as usual. However, as each point in a vector or curve is drawn,

shading occurs from that point to a shading reference line. The shading includes the point being drawn, as well as the point on the reference line.

The default value for the shading reference line is the horizontal line defined by the Y coordinate value of the current position when shading is turned on. You can select a different reference line with a position argument to the shading control option.

Figure 3–17 shows how shading occurs. This figure shows phases of a circle being drawn while shading is enabled and demonstrates the use of the reference line in shading.

You can shade an object by using either patterns or text characters with the shading control option. You define both types of shading by foreground intensity, background intensity, negative writing, and any overlay, erase, or replace writing in effect. In addition, you define pattern shading by the pattern you use and the multiplication factor for the pattern. Similarly, you define character shading by any text options that affect the selected character.

There are three types of shading controls:

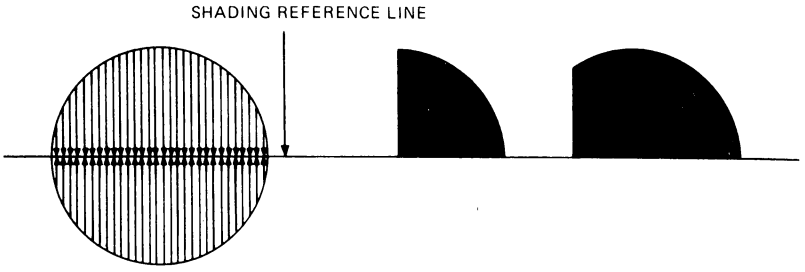
- Shading on/off controls
- Shading reference line select
- Shading character select

The following sections cover the shading controls and the use of multiple shading reference lines.

NOTE

Polygon Fill eliminates the need for multiple reference lines. Use multiple reference lines only if you plan to transport your application to a device that does not support Polygon Fill.

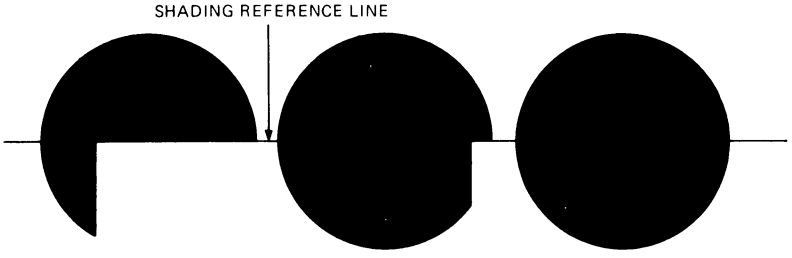
Figure 3–17: Shading Examples



(A)
DIRECTION OF SHADING TO OCCUR AS CIRCLE IS DRAWN

(B)
APPROX. 90 DEGREES OF CIRCLE DRAWN

(C)
APPROX. 130 DEGREES OF CIRCLE DRAWN



(D)
APPROX. 220 DEGREES OF CIRCLE DRAWN

(E)
APPROX. 340 DEGREES OF CIRCLE IS DRAWN

(F)
COMPLETED CIRCLE

NOTE
THE SHADING REFERENCE LINE WOULD NOT BE VISIBLE ON THE SCREEN OR PAGE.

MLO-382-86

Shading On/Off Controls

When shading is enabled, the Write Control command uses the pattern and foreground intensity. If the selected pattern is a solid line (P1), the graphic image area is completely shaded at the currently selected intensity (I0 through I3). No outline appears for the shaded graphic image, other than the difference in contrast between the background and foreground intensity.

The format for the shading on/off control option is as follows:

```
W(S <0 or 1>)
```

Figure 3–18 shows three circles shaded with different foreground intensities. This figure shows that the outline for each circle is formed by the contrast between the background and foreground values. If you want an outline, you can simply repeat the circle command with shading off.

Figure 3–19 shows the circles drawn in Figure 3–18 with shading off and with a different foreground intensity from that used in shading.

Figure 3–18: Circle Shading Examples: Without Outlines

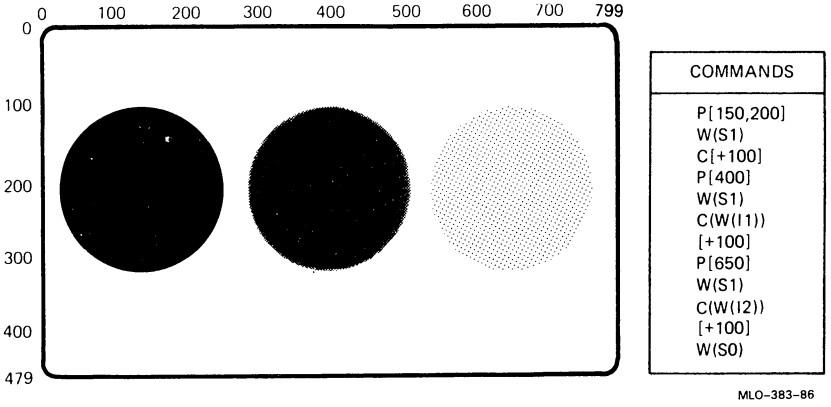
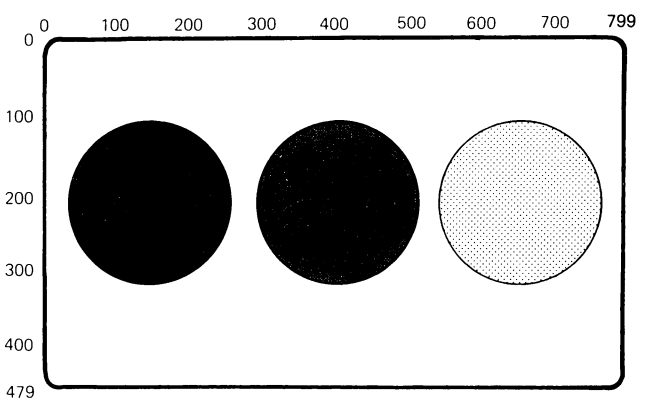


Figure 3–19: Circle Shading Examples: with Outlines



Figures in this section on shading on/off controls use the default value for the shading reference line: the Y coordinate value of the current position when shading is turned on. When you use the default shading line, remember to redefine shading each time the current position is moved for a new shading task. Otherwise, shading occurs to the previously defined reference line.

Remember that shading includes the shading reference line, regardless of whether the line is the default line or a line selected by the shading reference line option.

Figure 3–20 shows a graph in which the reference line is the same as the graph baseline. By repositioning the current position up one pixel row before enabling shading, you can keep the baseline intact, as shown in Figure 3–21. However, another more device-independent technique is available. Shade to and include the graph baseline, then redraw the graph baseline.

Figure 3–20: Shading Through the Graph Baseline

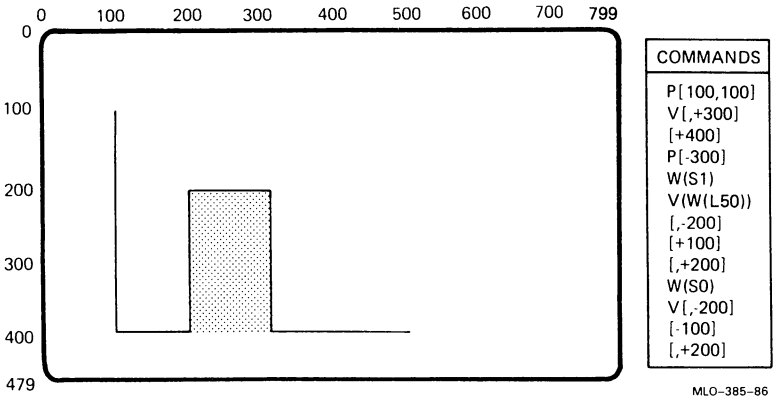
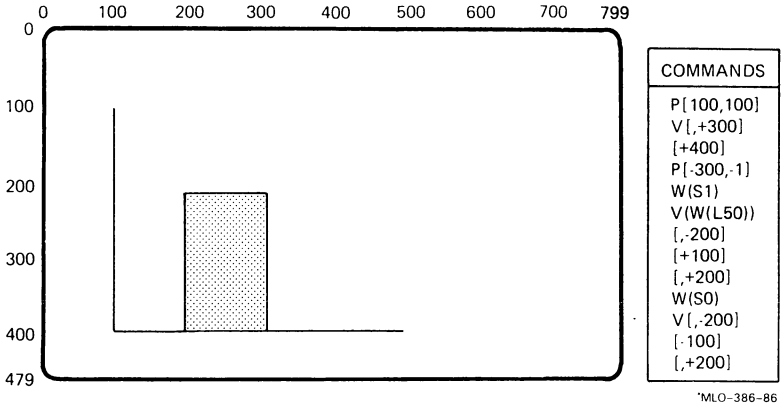


Figure 3–21: Shading to the Graph Baseline



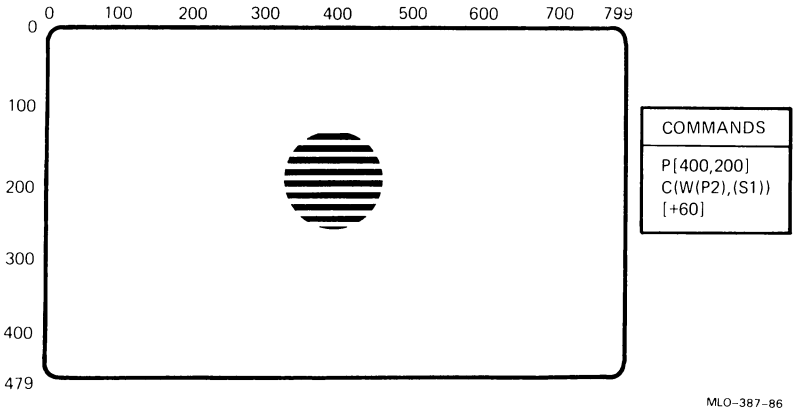
You can change the effect of shading by selecting a writing pattern other than a solid (P1). Figure 3–22 shows an example. In this figure, the circle

is shaded with a dash line pattern (P2). As shown, this pattern defines the circle with horizontal bars.

NOTE

If you want to change the currently selected pattern for shading, you must specify the new pattern before you turn on shading.

Figure 3–22: Circle Shading Example

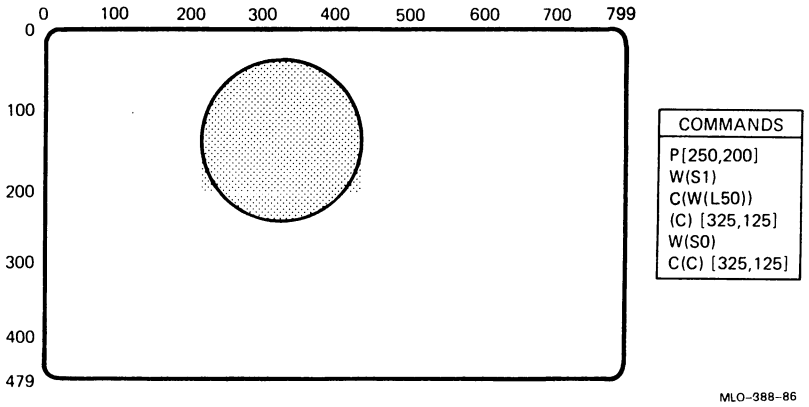


Shading Reference Line Select

The default value for the shading reference line is a horizontal line defined by the Y coordinate of the current position when shading is turned on. For most shading tasks, the default shading value shades the graphic object correctly. (See Figures 3–18 and 3–19.) For some graphic objects, however, the default value produces incorrect shading. An example is a circle with a center at a specified position.

Figure 3–23 shows the shading that results if the default value is used. In this figure, the circle is first invoked for shading at a foreground intensity of dim gray (I1); then the circle is invoked again (with shading off, and with the foreground intensity at I0) to define the shading area. As shown, the default shading line produces shading outside the intended area.

Figure 3–23: Incorrect Shading Example



The shading reference line argument lets you define a reference line value other than the default value selected by the shading on/off control option. The position coordinate used can be absolute or relative.

The format for selecting a specific horizontal (Y position) shading reference line is as follows:

```
W(S[<position>])
```

<position> provides the position value of the horizontal (Y axis) shading reference line. You can use either [X,Y], with the X value being ignored, or [,Y].

Figure 3–24 shows how to avoid the incorrect shading shown in Figure 3–23 by using the shading reference line select argument.

Figure 3–24: Correct Shading Example

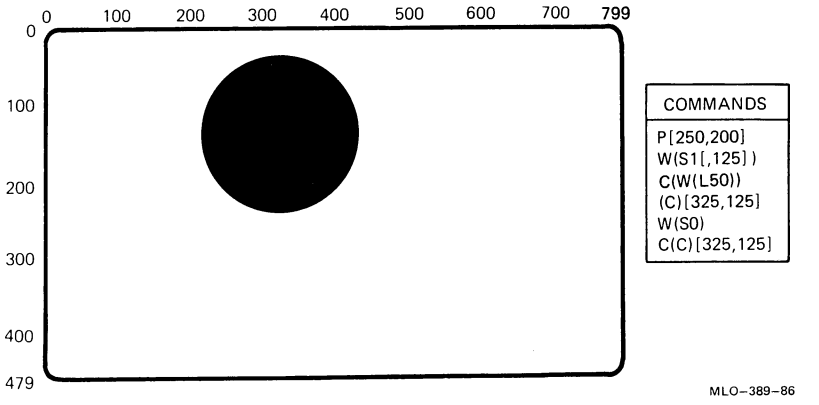
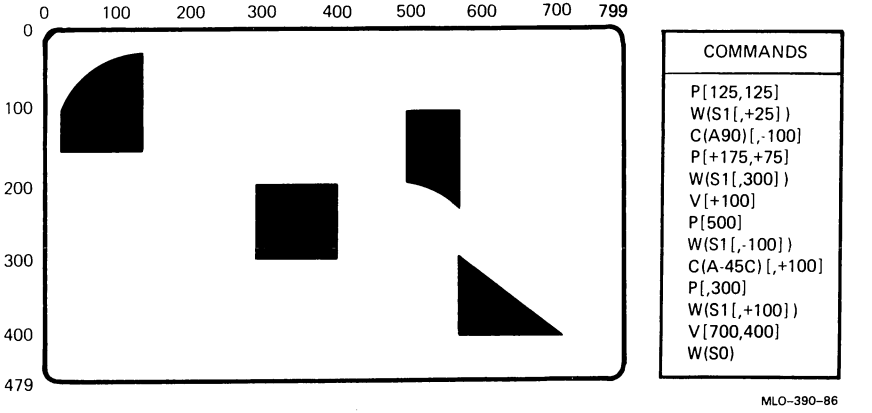


Figure 3–25 shows examples of shaded images drawn with the horizontal (Y coordinate) shading reference line.

Figure 3–25: Horizontal Shading Reference Line Examples



You can also use a vertical (X coordinate) shading reference line. If you use the vertical shading reference line, you have two options:

- You can use the default shading value, which is defined by the X coordinate of the current position when shading is turned on.
- You can specify the shading reference value with the vertical reference line select option.

Just as with the horizontal shading reference option, you may need to specify the reference line value to ensure proper shading.

The syntax for both of these options is shown here:

```
W(S(X))
```

or

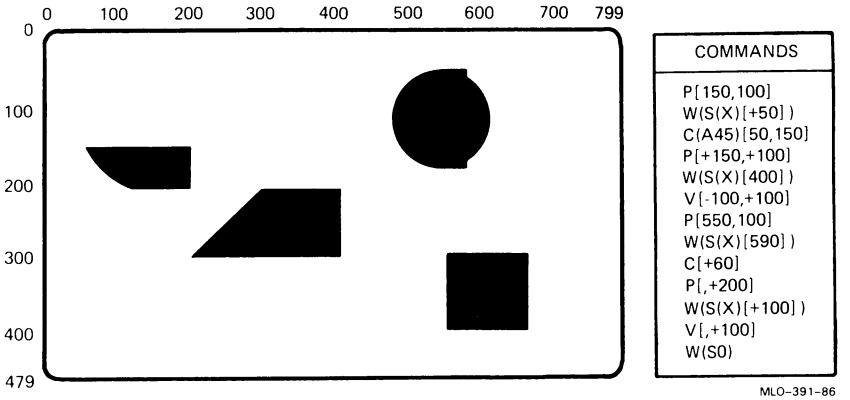
```
W(S(X)<position>)
```

(X) defines the shading control option as selecting a vertical (X axis) shading reference line.

<position> provides the position value of the vertical (X axis) shading reference line. You can use either [X,Y], with the Y value being ignored, or [X]. If no value is given, ReGIS uses the X value of the current position.

Figure 3–26 shows examples of simple shaded images drawn with the vertical (X coordinate) shading reference line.

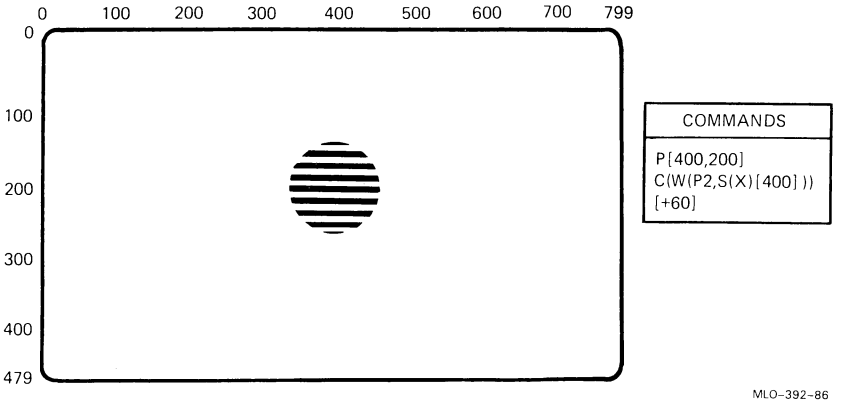
Figure 3–26: Vertical Shading Reference Line Examples



By comparing Figure 3–25 and Figure 3–26, you can see how selecting either a horizontal or vertical shading reference line produces different effects.

For example, Figure 3–27 shows a circle shaded with a dashed pattern (P2) while using a vertical reference line value. The circle that results is identical to the circle in Figure 3–22, where the default horizontal value for the shading line was used. Thus, regardless of the reference line orientation, you can maintain the pattern orientation while shading complex objects.

Figure 3–27: Vertical Shading Reference Line Example



Shading Character Select

This argument lets you shade objects by using text characters instead of patterns.

The format for the shading character select argument is as follows. You must use single or double quotes to enclose the character selected for shading.

```
W(S"<character>")
```

When you use character shading, text command options define the character set the shading character comes from and the unit cell size of the character. If you do not define these parameters, the standard character set is used. The character size is the last size specified during a Text command or the default value of S1, if no other size is specified.

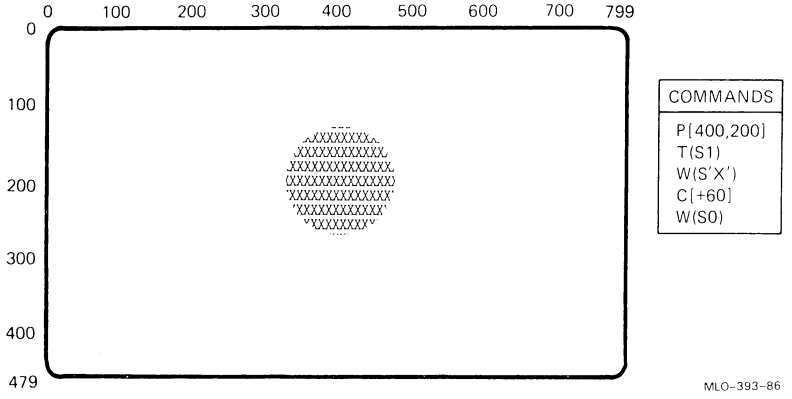
Shading with a character can provide half-tone effects. This feature is useful when designing graphic images for a device that has only two intensity values, such as a dot-matrix printer. In such applications, gray scale effects are achieved by shading with different density characters. You can use load character cell controls to define a set of characters that have different numbers of pixels as dark; then you can use those characters for shading.

When you shade with a character, only the top 8 x 8 matrix of an 8 x 10 cell's storage is used. Remember this when selecting shading characters or when creating characters with load cell commands.

The shading character is oriented in the same way for either horizontal or vertical shading reference lines. The shading pattern remains consistent when shading complex objects.

Figure 3–28 shows a circle shaded with Xs. In this example, only the size of the character has been specified. Therefore, the X from the standard character set is used. You can use the shading character select argument to shade any graphic image.

Figure 3–28: Shading Character Select Argument Example



When used alone, the shading character select argument uses the Y component of the current position to define a horizontal reference line for the character shading. You can also specify a horizontal or vertical shading reference line or a point when shading with a character.

The formats for combining a shading character with specified shading reference lines are as follows:

```
W(S"<character>"(X)[<position>])

W(S"<character>"[<position>])
```

<character> identifies the character to be used for shading.

(X) defines the shading control option as selecting a vertical (X axis) shading reference line.

<position> defines either the X axis or Y axis value of the line to be used for shading reference. The Y axis is the default; the X axis must be explicitly selected.

Multiple Shading Reference Line Use

You can use the ReGIS Polygon Fill command to shade complex areas that are difficult to shade with reference lines. (See the description of Polygon Fill.) Using Polygon Fill is easier and more efficient than other methods. However, if your application was written for either a VT125, a VT240 earlier than Version 2.1, or a version of Pro/Communications earlier than Version 3.0, it does not use Polygon Fill. In those cases, refer to the following information.

One shading reference line is not enough for graphic images that have unshaded areas between the point on the graphic image being drawn and the reference line. You can use the following method to shade such graphic images:

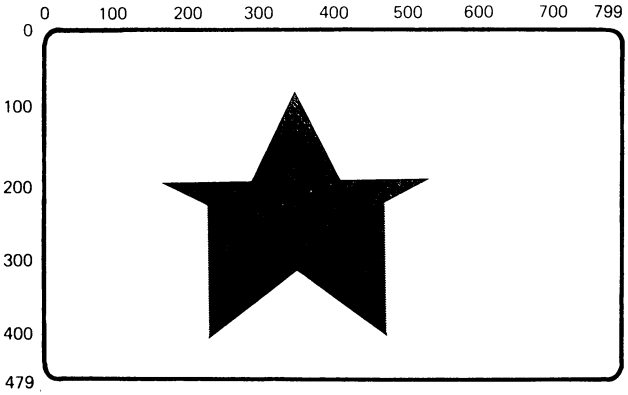
- Build the shaded graphic image in two or more sections. Use different shading reference lines for each section, including both horizontal and vertical shading reference lines.
- Reshade areas of the graphic image with a shade equal to the background intensity.
- Define the graphic image by using both procedures above. Use two or more sections with reshading.

Figure 3–29 shows an attempt to shade a star with only one shade value and one reference line. First, the star is defined as shading at dim gray (I1). Then the star is drawn with shading off, to outline the area selected for shading. Figure 3–30 takes the same example and breaks it down into stages, adding commands that define a second reference line and a second shade value. This figure shows a process for building a correct star graphic image.

NOTE

The commands used to build the star shown in Figure 3–29 and Figure 3–30 are not the only commands you can use. They are used in these figures to show how you can combine more than one reference line with more than one shading value to produce a correctly shaded image.

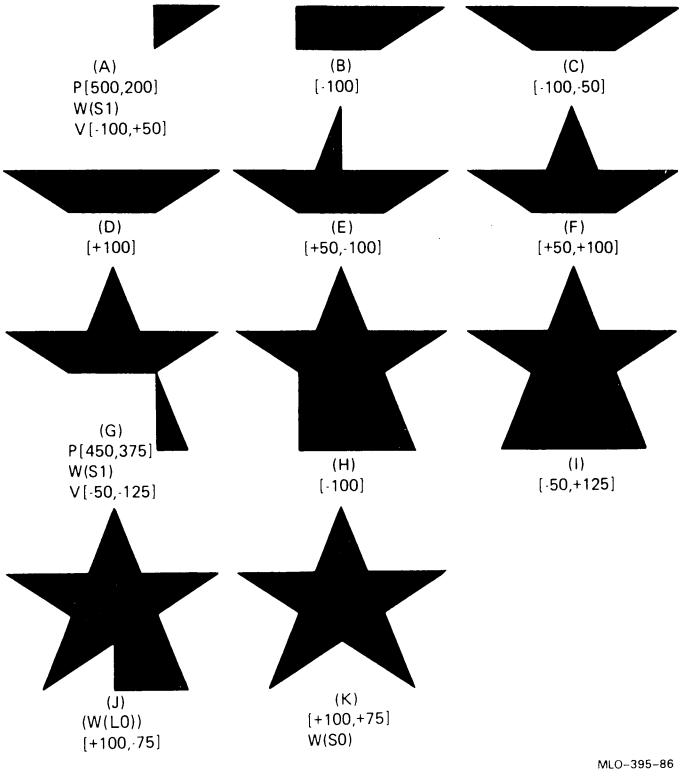
Figure 3–29: Incorrect Shading of Complex Graphic Object



COMMANDS
P[500,200]
W(S1)
V(W(I(L33)))
[-100]
[-50,-100]
[-50,+100]
[-100]
[+100,+50]
[-50,+125]
[+100,-75]
[+100,+75]
[-50,-125]
[+100,-50]
W(S0)
V[-100]
[-50,-100]
[-50,+100]
[-100]
[+100,+50]
[-50,+125]
[+100,-75]
[+100,+75]
[-50,-125]
[+100,-50]

MLO-394-86

Figure 3–30: Complex Graphic Shading Example



3.4.3 Write Control Command Summary

Table 3–9 summarizes the Write Control command options, including any default values.

Table 3–9: Write Control Command Summary

Argument	Default	Description
(M <n>)	(M1)	PV multiplication option. Defines multiplication factor (<n>) for PV values specified by a later PV positioning argument. Can serve as temporary write control for other types of commands.
(I <0–15>)	(I3)	One of three foreground intensity select options.
(E,R,V, or C)	(V)	Three argument letters available to define type of writing to occur. (E) for erase writing; (R) for replace writing; (V) for overlay writing; (C) for complement writing.
(P <0–9>)	(P1)	Select standard pattern option. Selects 1 of 10 stored writing patterns.
(P <binary>)	None	Specify binary pattern option. Lets you specify unique writing pattern for write tasks. The specified pattern can be up to 16 bits long.
(P(M <1–16>))	(M2)	Pattern multiplication option. Used to define the number of times each bit of the pattern memory is processed. You can use this option with the select standard pattern option or the specify binary pattern option, or by itself to define a multiplication factor for the last specified pattern.

Table 3–9 (Cont.): Write Control Command Summary

Argument	Default	Description
(N <0 or 1>)	(N0)	Negative pattern control option. (N1) reverses currently selected write pattern for all writing modes except erase writing. N0 turns off negative pattern control. In the case of negative writing, this option affects only whether picture erases to foreground or background color. N1 erases to foreground color; N0 to background color.
(S <0 or 1>)	(S0)	Shading on/off control. (S1) enables shading at currently selected pattern. The shading reference line is defined by the Y axis value of the active position when (S1) is selected. S0 turns off shading.
(S[,y])	None	Horizontal shading reference line select option. Selects a horizontal shading reference line defined by [,Y], which can be either an absolute or relative value.
(S(X)[x])	None	Vertical shading reference line select option. Selects a vertical shading reference line defined by [X], which can be either an absolute or relative value.
(S" <character> ")	None	Shading character select option. Lets you fill graphic objects with the specified character.

Vector

The Vector command lets you draw lines between the current position and a specified new position. The form the lines take is determined by write controls in effect when the Vector command is issued.

The four basic arguments for the Vector command are as follows:

- Draw dot arguments
- Draw line arguments
- Sequence of coordinates options
- Temporary writing controls

Format V *option*

Command Arguments	Description
[ ]	Null position argument
[X,Y]	Position argument to draw a dot or line
<PV>	PV value that defines endpoint for a line
(B)	Begin bounded sequence option
(S)	Begin unbounded sequence option
(E)	End of sequence option
(W)	Temporary write control option

Command Arguments

[ ]

The draw dot argument uses a null position argument to write to a single pixel. The format of the draw dot option is as follows:

```
V[]
```

[X,Y]

The draw line arguments for the Vector command are identical in form to the move arguments for the Position command. However, instead of moving the current position, draw line arguments draw a line from the current position to a new current position that you specify. You can specify the new current position in four ways:

- Absolute — Specifies the actual [X,Y] address of the line's endpoint.
- Relative positioning — Specifies the line's endpoint relative to the current location.
- Absolute/relative positioning — Specifies the line's endpoint by using a relative value for one coordinate and an absolute value for the other.
- PV positioning — Uses the PV system to specify the line's endpoint relative to the current position. You can use PV positioning with a temporary write control for PV multiplication.

The formats for the absolute argument form are as follows:

```
V[X,Y]
V[X]
V[,Y]
```

The formats for relative argument forms are as follows:

- Positive X and Y displacement:

  ```
  V[+X,+Y]
  ```

- Positive X and negative Y displacement:

  ```
  V[+X,-Y]
  ```

- Negative X and positive Y displacement:

  ```
  V[-X,+Y]
  ```

- Negative X and Y displacement:

  ```
  V[-X,-Y]
  ```

- Positive X displacement only:

  ```
  V[+X]
  ```

- Negative X displacement only:

  ```
  V[-X]
  ```

- Positive Y displacement only:

  ```
  V[,+Y]
  ```

- Negative Y displacement only:

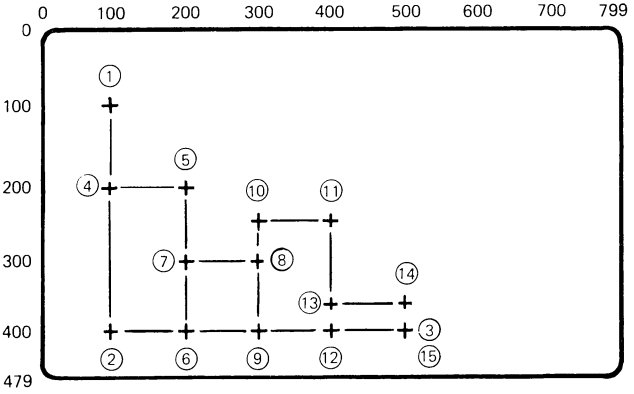
  ```
  V[,-Y]
  ```

The format for absolute/relative positioning combines parts of the forms shown for absolute positioning and relative positioning.

The draw line argument draws a straight line from the old current position to the new current position. Lines are drawn with the pattern mask in effect, with the pattern repeated cyclically. When using patterns other than P0 (all 0s) or P1 (all 1s), you may want to repeat the V keyletter to reset writing to the first position of the pattern. Otherwise, the pattern continues where it finished. This could result, for example, in a blank at a point where two vectors intersect.

Figure 3–31 shows a bar graph drawn by using absolute, relative, and absolute/relative arguments to the Vector command.

Figure 3–31: Bar Graph Using Vector Draw Line Arguments



COMMANDS	
(1)	P[100,100]
(2)	V[,400]
(3)	[500]
(4)	P[-400,200]
(5)	V[200]
(6)	[,400]
(7)	[,-100]
(8)	[+100]
(9)	[,+100]
(10)	[,-150]
(11)	[+100]
(12)	[,+150]
(13)	[,-20]
(14)	[+100]
(15)	[,+20]

NOTE:
THE STARTING POSITION [0,0] FOR THE CURSOR IS ARBITRARY. THE CIRCLED NUMBERS, AND CURSORS, ARE SHOWN ONLY TO ILLUSTRATE THE POSITION OF THE CURSOR AT THE END OF EACH COMMAND.

MLO-396-86

<PV>

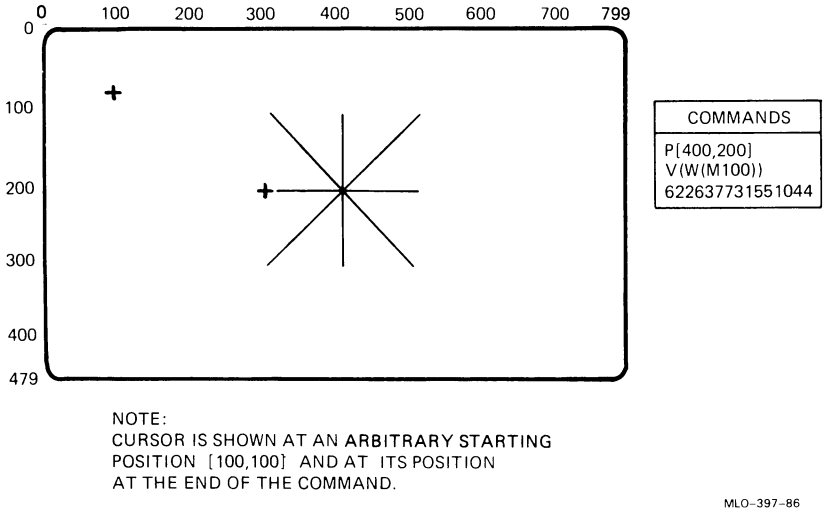
The PV value defines an endpoint for a line to be drawn, relative to the current position. The line is drawn in the direction defined by the PV value. You can use PV positioning with a temporary write control for PV multiplication.

The format for PV offset and PV multiplication temporary write options are as follows:

```
V(W(M<multiplication value>)) <pv value>
```

Figure 3–32 shows an image of the PV directions built by using PV positioning with a temporary write control for PV multiplication.

Figure 3–32: PV Directions Graphic Image



(B)

This is the bounded sequence option.

A sequence option lets you group sets of vectors into blocks that can be processed as units. A sequence option consists of a start (or begin) option and an end option. You can embed Position (P) and Curve (C) commands in the sequences.

The format for a Vector command bounded sequence is as follows:

```
V(B)<embedded options>V(E)
```

The bounded sequence is useful to connect the last vector of a sequence to the starting position of the sequence, thus generating a closed figure.

A bounded sequence consists of at least one begin (B) option and one end (E) option. Each (B) option stores the coordinate value of the active position in effect when the option is invoked. A sequence can consist of up to 16 (B) options. Each (B) option must have a corresponding (E) option.

Figure 3–33 shows an image drawn by using multiple (B) options, with (C) commands embedded in the sequence. Figure 3–34 shows images drawn by using Vector command bounded sequences.

Figure 3–33: Vector Command Bounded Sequence Example

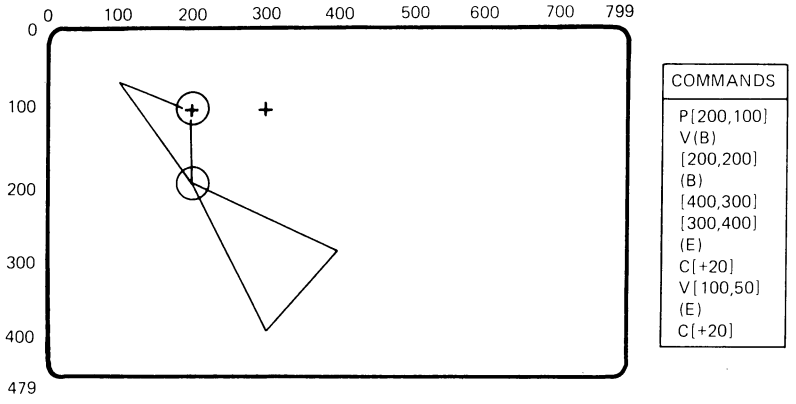
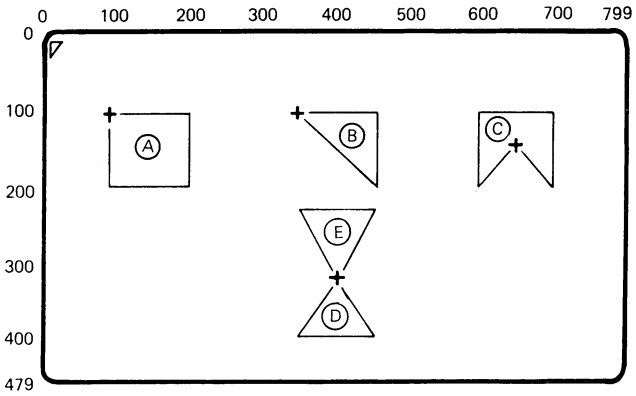


Figure 3–34: Bounded Sequence Examples



NOTE:
CURSOR IS SHOWN IN ARBITRARY STARTING POSITION OF [0,0] AND AT STARTING AND ENDING POINTS OF EACH SEQUENCE; CIRCLED LETTERS IN FIGURE AND COMMAND DESCRIPTIONS ARE FOR EASE OF DESCRIPTION ONLY.

COMMANDS				
(A)	(B)	(C)	(D)	(E)
P[100,100] V(B) [+100] [,+100] [-100] (E)	P[350,100] V(B) [+100] [,+100] (E)	P[650,150] V(B) [700,200] [,-100] [-100] [,+100] (E)	P[400,340] V(B) [450,400] [-100] (E)	V(B) [350,250] [+100] (E)

MLO-399-86

(S)

This is the unbounded sequence option.

A sequence option lets you group sets of vectors into blocks that can be processed as units. A sequence option consists of a start (or begin) option and an end option. You can embed Position (P) and Curve (C) commands in the sequences.

The format for a Vector command unbounded sequence is as follows:

```
V(S)<embedded options>V(E)
```

The difference between a bounded and unbounded sequence is the role of the start option. In bounded sequences, the (B) option tells ReGIS to store the current position and to return to that position after a corresponding end (E) option. In unbounded sequences, the (S) option tells ReGIS to store a dummy, or nonexistent, location. In this case, a corresponding (E) option does not change the current position.

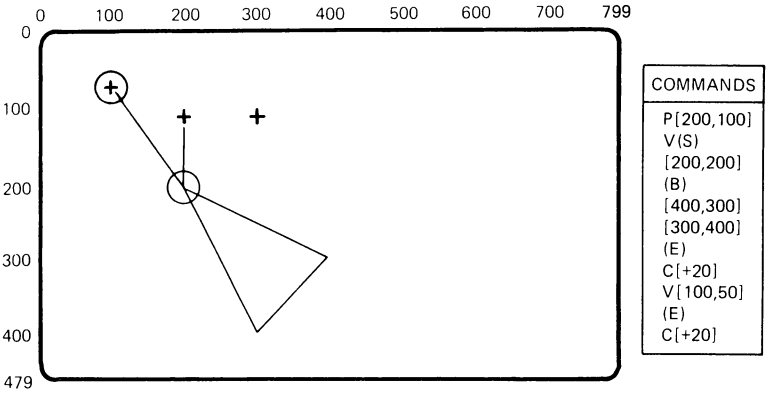
NOTE

Coordinate values are saved during Vector command unbounded sequences and during Position (P) command sequence options. The limit for all unended, saved values (including all save commands) is 16.

The unbounded sequence serves little purpose for images drawn with a Vector command. This sequence provides symmetry with the unbounded sequence of the Curve command.

Figure 3–35 shows the image produced if the same set of commands used in the bounded sequence in Figure 3–33 were placed in an unbounded sequence.

Figure 3–35: Vector Command Unbounded Sequence Example



NOTE:
CURSOR IS SHOWN AT ARBITRARY STARTING POSITION OF [300,100] AND AT STARTING [200,100] AND ENDING [100,50] POSITIONS FOR SEQUENCE.

MLO-400-86

(E)

The end of sequence option ends a bounded or unbounded sequence. It references the last stored (B) or (S) option value. If the value was stored by a (B) option, a line is drawn from the active position where (E) is sensed to the location stored by (B). If the value was stored by an (S) option, no line is drawn, and the active position remains at the current position.

(W)

All Vector command options are done with the write control values currently in effect. The temporary write control option lets you use different values in a specific Vector command without changing the write control values. The format for the temporary write control option is as follows:

```
V(W(<suboptions>))<options>
```

You can use a temporary write control to affect any of the following:

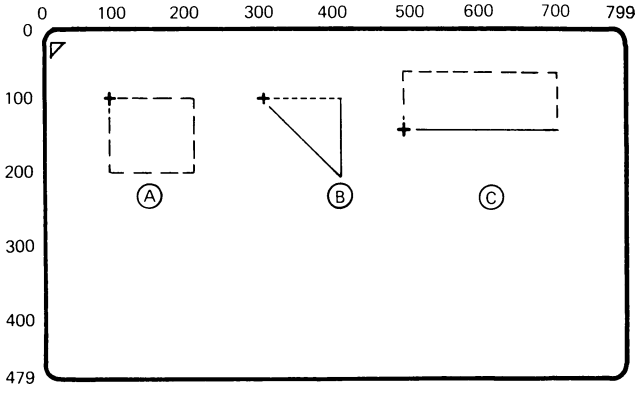
- PV multiplication
- Pattern control
- Foreground intensity
- Type of writing (overlay, erase, replace)

The temporary write control values remain in effect only until one of the following conditions occurs:

- A new temporary write control option is used.
- A nonvector command is performed, such as a Curve command.
- A new Vector command is defined with the Vector command keyletter.

When any one of these conditions occurs, writing returns to the current write control values. Figure 3–36 shows images drawn with temporary write control values affecting only the pattern used.

Figure 3–36: Temporary Write Control Option Example



NOTE:
CURSOR IS SHOWN AT ARBITRARY STARTING POSITION OF [0,0] AND AT STARTING AND ENDING POSITIONS FOR EACH GRAPHIC.

COMMANDS		
(A)	(B)	(C)
P[100,100] P[] V(W(P2(M2))) [+100] [,+100] [-100] [,-100]	P[300,100] P[] V(W(P4(M2))) [+100] V[,+100] [300,100]	P[500,150] P[] V(W(P2(M2))) [,-75] [+200] [,+75] V[-200]

MLO-401-86

3.4.4 Vector Command Summary

Table 3–10 summarizes the Vector command arguments. There are no default values for these arguments.

Table 3–10: Vector Command Summary

Argument	Description
[]	Draw dot argument. Used to write to a single pixel defined by the current active position. Does not move the cursor.
[X,Y]	Draw line argument. [X,Y] value defines the endpoint of a line to be drawn from the current active position. The [X,Y] value can be absolute, relative, or absolute/relative.
<PV>	Draw line argument. PV value defines an endpoint for a line to be drawn, relative to the current active position, in the direction defined by the PV value.
(B)	Begin a bounded sequence option. Stores the current active position for reference at the end of the sequence.
(S)	Begin an unbounded sequence option. Stores a dummy position for reference at the end of the sequence.
(E)	End of sequence option. References last stored (B) or (S) option value.
(W(<suboptions>))	Temporary write control option. Lets you select temporary Write Control values without changing the current Write Control values. Temporary Write Control values remain in effect only for the command controlling them.

Curve

The Curve command draws circles, arcs, and other curved images. The form of the curve is determined by the write control values in effect when the Curve command is invoked.

There are three basic types of Curve commands:

- Circles
- Arcs
- Curve interpolation sequence

You can use the temporary write control option with all three types.

Format

C *option*

Command Arguments	Description
<position>	Circle with center at the current position
(C) <position>	Circle with center at specified position
(A <angle>)[X,Y]	Arc with center at the current position
(A <angle> C)[X,Y]	Arc with center at specified position
(B)	Closed curve sequence
(S)	Open curve sequence
(E)	End of an open or closed curve sequence
(W(...))	Temporary write control option

Command Arguments

C <position>

This command includes a point through which the circumference of a circle will be drawn with center at the current position. The current position at the end of the command is the same as it was at the start. This command can use the same absolute, absolute/relative, relative, and PV positioning value arguments used with the Position (P) and Vector (V) commands.

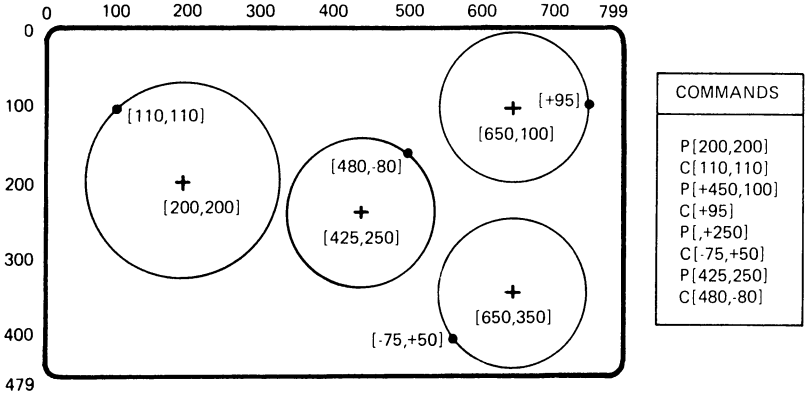
The format for the circle with center at the current position is as follows:

```
C<position>
```

The position value used with this command depends on whether the circumference passes through a specific point or the circumference has a specific radius. If the circumference has a specific radius, you can specify either a PV value or a single relative position value. For the circumference to pass through a specific point, you can use absolute positioning, relative positioning, or absolute/relative positioning.

Figure 3–37 shows circles drawn by using the various position arguments available for drawing a circle with a center at the current position.

Figure 3–37: Circle with Center at Current Position Example



NOTE:
ONLY CIRCLES WOULD BE SHOWN; DOTS, POSITION VALUES, AND CURSORS ARE FOR REFERENCE.

MLO-402-86

(C) <position>

This option defines the center of a circle, using the current position as the point through which the circumference of the circle will be drawn. The current position at the end of the command is the same as it was at the start. This option can use the same absolute, absolute/relative, relative, and PV positioning value arguments used with the Position (P) and Vector (V) commands.

The format for the circle with center at a specified position is as follows:

```
C(C)<position>
```

Although this option uses the same position values used with the circle with center at the current position command, the results are different. In both cases, the diameter of the circle differs, depending on the specified position. But with the current position command, the circle is always drawn an equal distance around the current position. With the specified position command, the placement of the circle relative to the current position depends on the specified position.

Figure 3–38 shows an example. In this figure, two circles with centers at specified positions are drawn, each with the same current position. As shown, the circles are drawn in a direction relative to the direction of the specified position from the current position.

Figure 3–38: Varying Circle Direction

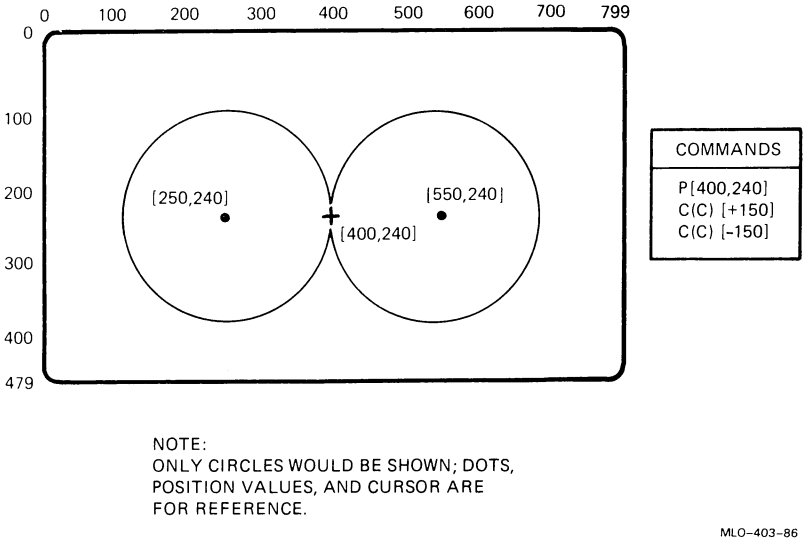
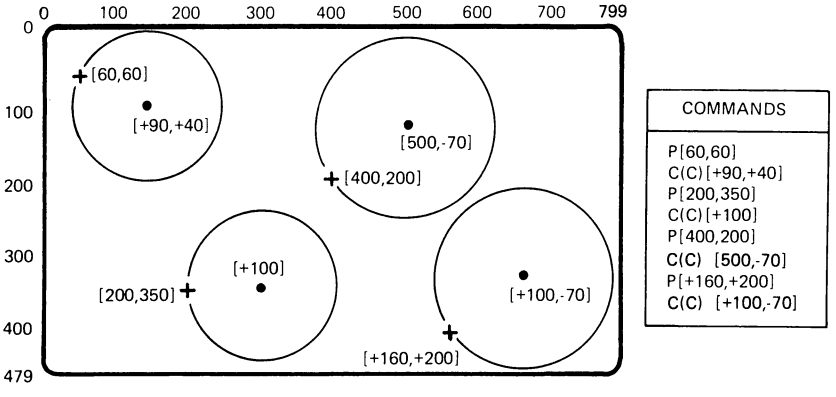


Figure 3–39 shows examples of circles drawn with the various position values available for drawing a circle with a center at a specified position.

Figure 3–39: Circle with Center at Specified Position Example



NOTE:
ONLY CIRCLES WOULD BE SHOWN; DOTS, POSITION VALUES, AND CURSORS ARE FOR REFERENCE.

MLO-404-86

(A <angle>) <position>

Arcs are sections of a circle. You can draw arcs in either of two ways:

- Arc with center at the current position
- Arc with center at a specified position

Both options can use the relative, absolute/relative, absolute, and PV positioning value arguments used with Position, Vector, and Curve commands.

Arc-drawing is at 1-degree resolution. If you specify a degree value greater than 360 in an arc option, ReGIS draws 360 degrees.

This command defines an arc drawn from a specified point. The current position is considered as the center of a circle, of which the arc is a part. The current position at the end of this command is the same as the current position at the start of the option.

The format for the arc with center at the current position command is as follows:

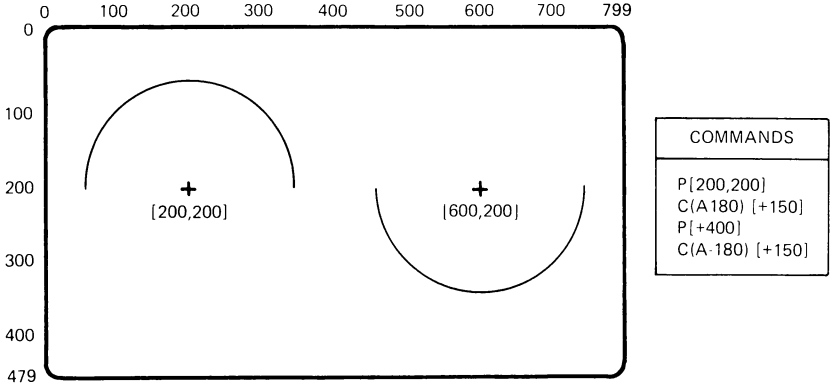
```
C(A<degrees>)<position>
```

<degrees> provides the number of degrees to be drawn for the arc and the direction the arc is to be drawn. With no sign or a positive sign (+), the arc is drawn counterclockwise from the specified position. With a negative (–) sign, the arc is drawn clockwise.

<position> provides the value of the position at which arc-drawing is to start. Value is either absolute, relative, absolute/relative, or a PV value (as defined by the current PV multiplication factor).

Figure 3–40 shows two arcs drawn with the same basic arc with the center at the current position. One option uses a positive (+) degree value, and the other uses a negative (–) value. Figure 3–41 shows arcs drawn with this same command, using different forms of position values.

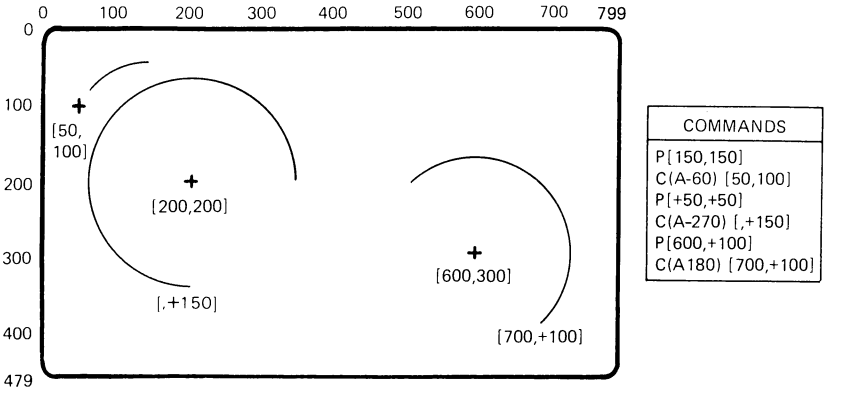
Figure 3–40: Effect of Signed Degree Values on Arc



NOTE:
ONLY ARCS WOULD BE SHOWN; LOCATION VALUES AND CURSORS ARE FOR REFERENCE.

MLO-405-86

Figure 3–41: Effect of Position Values on Arc



NOTE:
ONLY ARCS WOULD BE SHOWN; LOCATION VALUES AND CURSORS ARE FOR REFERENCE.

MLO-406-86

(A <degrees> C) <position>

This option defines an arc drawn from the current position. The specified position is the center of a circle, of which the arc is a part. The current position changes as the arc is drawn. At the end of the command, the current position is the same as the end of the arc drawn. This is particularly useful for linking the endpoint of one arc with the starting point of another.

NOTE

Due to limitations in the accuracy of the curve algorithm for arcs on some devices, the end position of an arc and the current position may not be where you would expect. When you chain arcs, use absolute positioning between them.

The format for the arc with center at specified position command is as follows:

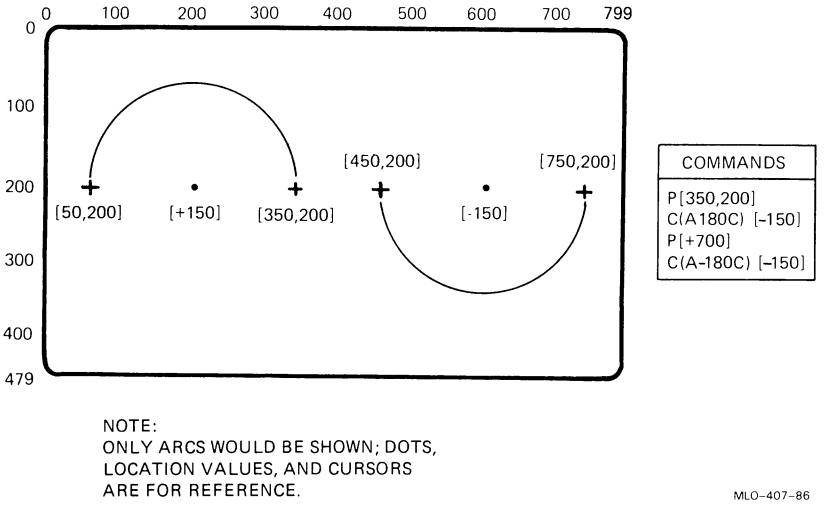
```
C(A<degrees>C)<position>
```

<degrees> provides the amount of degrees to be drawn for the arc and the direction the arc is to be drawn. With no sign or a positive sign (+), the arc is drawn counterclockwise from the specified position. With a negative (–) sign, the arc is drawn clockwise.

<position> provides the value of the position at which arc-drawing is to start. Value is either absolute, relative, absolute/relative, or a PV value (as defined by the current PV multiplication factor).

Figure 3–42 shows a positive or negative sign affects the way an arc is drawn. Figure 3–43 shows the chaining of arcs, using the arc with center at specified position option.

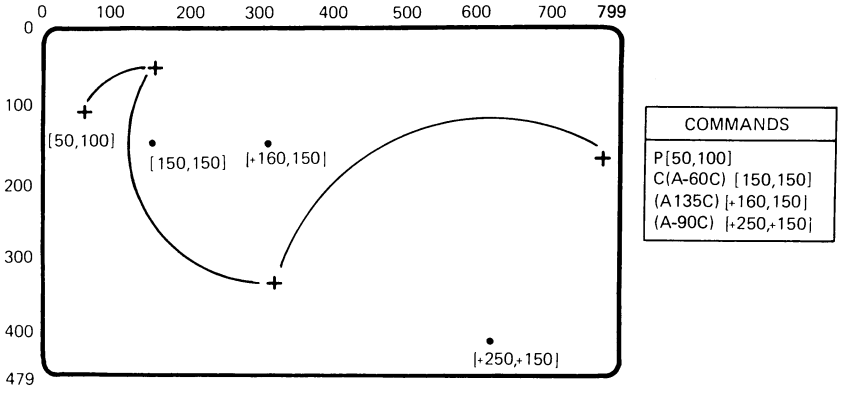
Figure 3–42: Effect of Signed Degree Values on Arc



NOTE:
ONLY ARCS WOULD BE SHOWN; DOTS, LOCATION VALUES, AND CURSORS ARE FOR REFERENCE.

MLO-407-86

Figure 3–43: Effect of Specified Positions on Arc



NOTE:
ONLY ARCS WOULD BE SHOWN; DOTS, LOCATION VALUES, AND CURSORS ARE FOR REFERENCE.

MLO-408-86

Curve Interpolation

Curve interpolation uses bounded and unbounded sequences to define a set of positions used for interpolation.

A curve drawn during a sequence option is not the result of the function used to specify points for the curve. It is instead the result of a graphic technique that produces a reasonable imitation of a function-type curve, such as those used in graphs. The curve represents a generalized, non-linear function intersecting all specified positions. The curve indicates the presence of a nonlinear function, rather than the function itself.

You must use a minimum of four positions to ensure that the ReGIS curve generator is following the function being represented. The positions should include the current position at the start of the sequence and at least three specified positions within the sequence, because the curve generator uses four positions at a time to perform its interpolation. As each interpolation is performed, the curve generator moves to the next position in the sequence. The curve generator then performs a new interpolation, using that position and the previous three. This action continues until the curve generator uses all positions in a sequence.

There are two types of interpolation sequence options:

- Closed curve sequence option
- Open curve sequence option

The closed curve sequence uses the same option syntax as the bounded sequence options for the Position and Vector commands. The open curve sequence uses the same unbounded sequence options as the Position and Vector commands.

You can also use a null position argument with closed and open curve sequences. This argument causes the position value immediately preceding the null position to be used twice in the interpolation. The effects of a null position depend on how it is used in the sequence. The following sections provide greater detail concerning the effect of the null position, including examples.

(B)

This option uses the bounded sequences (used with the Position and Vector commands) to define the set of positions for interpolating a closed curve. While the bounded sequences in Position and Vector commands can save up to 16 begin options, a closed curve sequence uses only one begin and end option. Also, bounded sequences in Position and Vector commands can contain other commands. However, the Curve (C) command bounded sequence must follow one C command, with no intervening commands. Any keyletter, including another C, aborts the curve.

The format for the closed curve sequence is as follows:

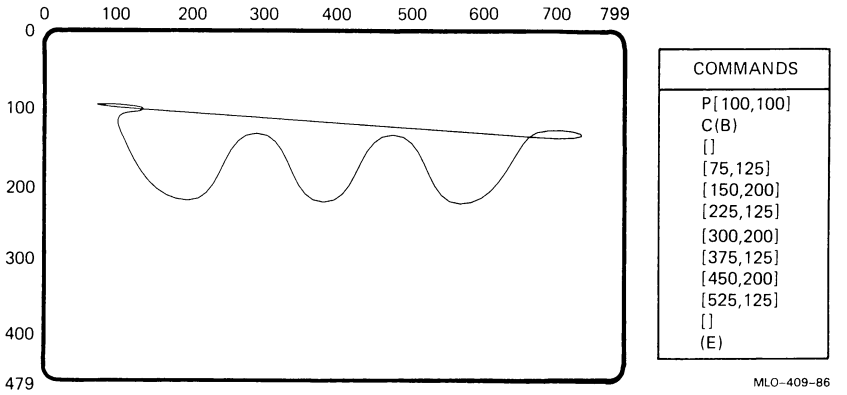
```
C(B)<positions>(E)
```

The positioning used in the sequence can be absolute, relative, absolute/relative, or PV values. When you use absolute values, the specified [X], [,Y], or [X,Y] location is used for the interpolation. When you use relative values (including PV values), the value used in the interpolation is defined as relative to the last current location before the relative position value (whether that was the current position at the start of the sequence, or one of the values specified in the sequence). The current position at the end of the closed curve sequence is the same as the current position when the sequence started.

You can use the null position argument, [ ], with the closed curve sequence to produce two results:

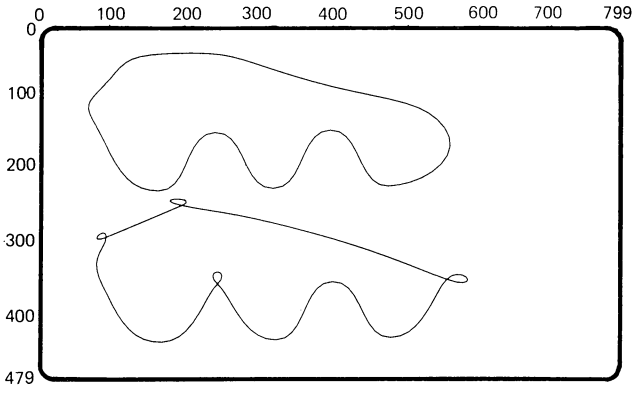
- Close the curve with a straight line. A null position argument at the start and end of the sequence causes the values of the first and last positions to be used twice in the interpolation. Figure 3–44 shows a closed curve, using the null position argument at the start and end of the sequence.

Figure 3–44: Closed Curve Sequence with Null Position Argument



- Create a sharper change in the interpolated curve form. A null position argument during the sequence uses the value of the preceding position twice in the interpolation. Figure 3–45 shows the same figure drawn both with and without null position arguments. (Note the Y values of positions used in the bottom figure have been increased uniformly by 200, so that you can see the two figures in one grid. To clarify the process, numbers and circles identify in order the positions used to interpolate the curve. The ReGIS code listed does *not* generate the circles or numbers.) The sequence that creates the top figure contains no null position arguments, which accounts for the smoothness of the curves. The sequence that creates the bottom figure contains three null position arguments after the fourth, eighth, and tenth position, which causes the curves to become discontinuous.

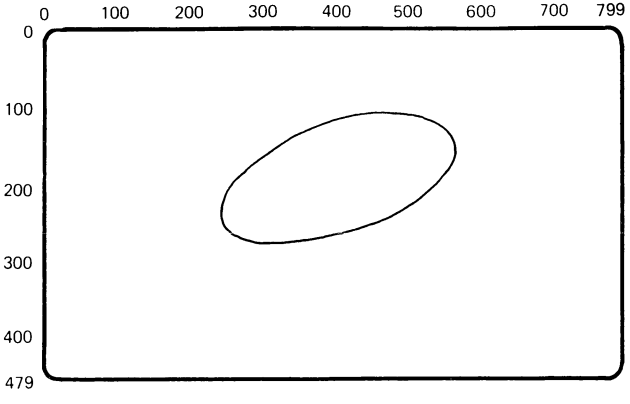
Figure 3–45: Closed Figure with and Without Null Position Argument



COMMANDS	
TOP FIGURE	BOTTOM FIGURE
S(E)P[80,60] C(B) [75,125] [150,200] [225,125] [300,200] [375,125] [450,200] [525,125] [350,60] [180,20] (E)	P[80,260] C(B)[] [75,325] [150,400] [225,325] [] [300,400] [375,325] [450,400] [525,325] [] [350,260] [180,220] [] (E)

Figure 3–46 shows another example of a curve generated by a closed curve sequence without the null argument.

Figure 3–46: Closed Curve Sequence Without Null Position Argument



COMMANDS
P[230,240] C(B) [320,160] [480,120] [570,160] [480,240] [320,280] (E)

(S)

This option uses the unbounded sequences (used with Position and Vector commands) to define a set of positions for interpolation of an open curve. However, the unbounded sequences available to Position and Vector commands can save up to 16 start options; an open curve sequence uses only one start and end option.

The format for an open curve sequence is as follows:

```
C(S)<positions>(E)
```

You use the null position argument, [ ], with the open curve sequence argument, when you are drawing a curve from the current position through to the last specified location. Without the null argument, the curve interpolation still considers all the position values for the actual interpolation. However, the curve is drawn from the position specified immediately following (S) to the second to last position. The null argument duplicates the first and last positions, extending the drawing of the curve through those locations, if desired. The current position at the end of an open curve sequence is the last position specified in the argument list.

You can also use the null position argument to use any specified value twice during interpolation. This method creates a sharper change in the interpolated curve form. Figure 3–47 shows an example of an open curve generated without using null position arguments.

Figure 3–48 shows the curve generated when the same command is invoked with null arguments.

Figure 3–47: Open Curve Sequence Without Null Position Arguments

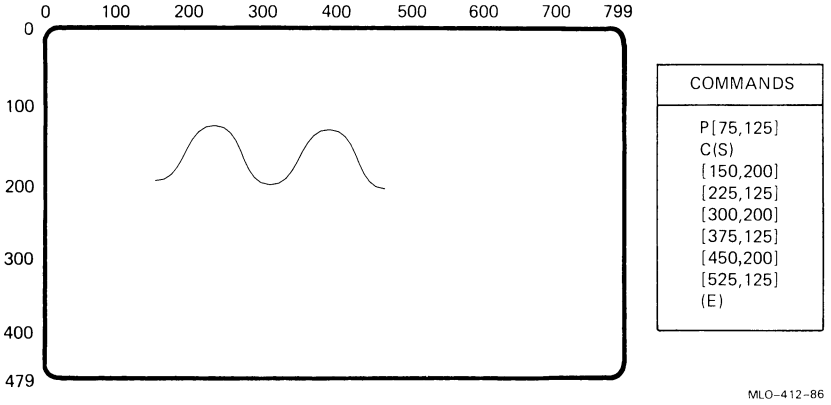
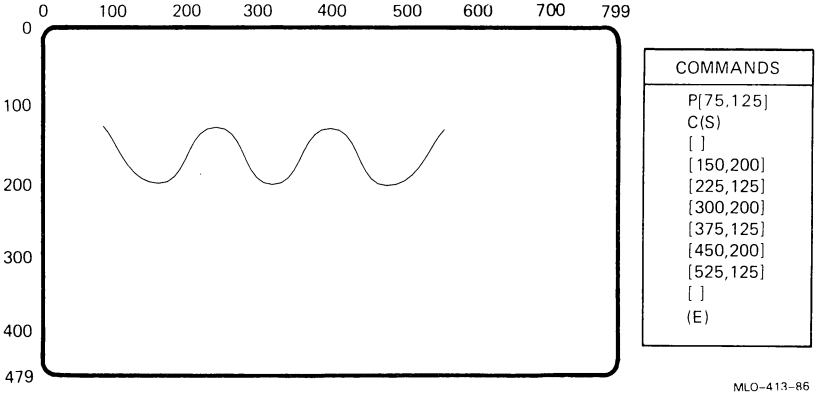


Figure 3–48: Open Curve Sequence with Null Position Arguments



(W)

Curve commands use the write control values currently in effect. The temporary write control option lets you use different values in a specific Curve command without changing the current values.

You can use a temporary write control to affect any of the following:

- PV multiplication
- Pattern control
- Foreground intensity
- Type of writing (overlay, erase, replace)
- Shading control

The format for this option is as follows:

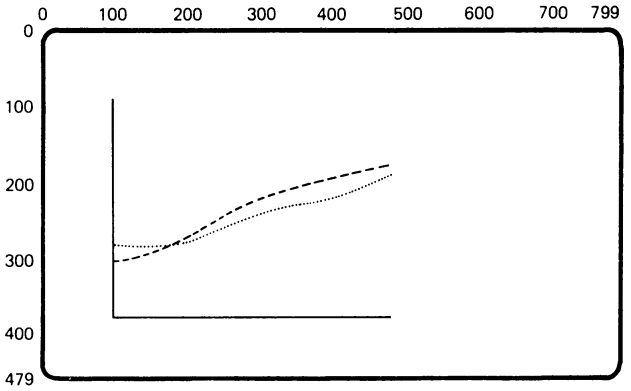
```
C(W(<suboptions>))<options>
```

The temporary write control values remain in effect only until one of the following conditions occurs:

- A new temporary write control option is used (only values specified change).
- A new command keyletter is encountered (including another Curve command).

When one of these conditions occurs, writing returns to the current write control values. Figure 3–49 shows a graph using a temporary write control option to change open curve sequences. In this example, only the pattern is affected.

Figure 3–49: Temporary Write Control Option Example




```
COMMANDS

P[100,100]
V[,+300]
[+400]
P[100,-80]
C(W(P2))
(S)
[]
[+100,-30]
[+100,-50]
[+100,-30]
[+100,-20]
[]
(E)
P[100,300]
C(W(P4))
(S)
[]
[+100]
[+100,-40]
[+100,-20]
[+100,-40] []
(E)
```

MLO-414-86

3.4.5 Curve Command Summary

Table 3–11 summarizes the Curve command arguments. There are no default values for these arguments.

Table 3–11: Curve Command Summary

Argument	Description
[X,Y]	Circle with center at the current position. [X,Y] defines a point on the circumference of the circle. The [X,Y] value can be absolute, relative, or absolute/relative.
(C)[X,Y]	Circle with center at specified position. [X,Y] defines the center of the circle, while the current active position defines a point on the circumference. The [X,Y] value can be absolute, relative, or absolute/relative.
(A <degrees>)[X,Y]	Arc with center at the current position. [X,Y] defines the starting point for drawing the arc, while the signed value of the <degrees> determines which direction the arc is drawn from that point: plus sign (+) for counterclockwise, and minus sign (–) for clockwise. The [X,Y] value can be absolute, relative, or absolute/relative.
(A <degrees> C)[X,Y]	Arc with center at specified position. [X,Y] defines the center, while the current active position is the point from which the arc is drawn. The signed value for <degrees> determines which direction the arc is drawn: plus sign (+) for counterclockwise, and minus sign (–) for clockwise. The [X,Y] value can be absolute, relative, or absolute/relative.
(B) <positions> (E)	Closed curve sequence. Defines a closed curve graphic image built from interpolation of [X,Y] positions specified within the sequence. The [X,Y] values can be absolute, relative, or absolute/relative.
(S) <positions> (E)	Open curve sequence. Defines an open curve graphic image built from interpolation of [X,Y] positions specified within the sequence. The [X,Y] values can be absolute, relative, or absolute/relative.

Table 3–11 (Cont.): Curve Command Summary

Argument	Description
[]	Null position argument. Used with either sequence option to affect interpolation. The null argument stores a position equal to the last specified active position as part of the positions to interpolate. When used at the beginning of a sequence, the value stored is the current active position.
(W(<suboptions>))	Temporary write control option. Lets you select temporary write control values without changing the current write control values. Temporary write control values remain in effect only for the command controlling them.

Polygon Fill

You use the Polygon Fill command (F) to draw filled, closed figures, such as circles, ellipses, triangles, and squares. There are four basic options to the Polygon Fill command:

- Vector option
- Curve and arc option
- Position option
- Temporary write control option

Format

F *option*

Command Arguments	Description
(V)	Vector option
(C)	Curve and arc option
(P)	Position option
(W <suboptions>)	Temporary write control option

Command Arguments

(V)

The Polygon Fill command accepts all Vector command options and arguments, which allows you to draw filled figures, such as squares, rectangles, and diamonds.

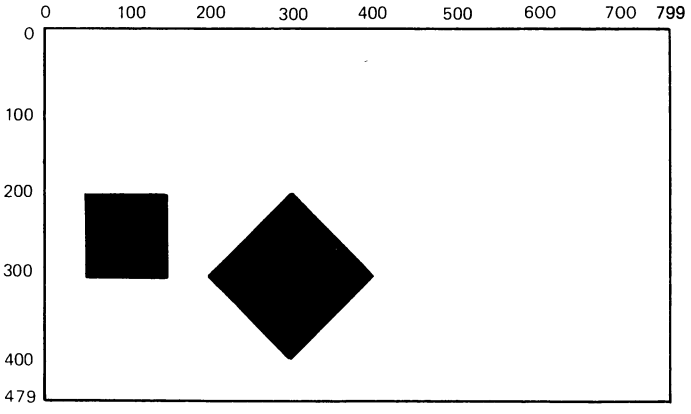
The basic format of the Polygon Fill command using a vector option is as follows:

```
F(V<positions>)
```

<positions> identifies the positions of the vertices.

Figure 3–50 shows a filled square and a filled diamond drawn by using the Polygon Fill command with the vector option and the bounded sequence (B) suboption.

Figure 3–50: Vector Option Example



(C)

The Polygon Fill command accepts all Curve command options and arguments, which allows you to draw filled, curved figures, such as circles and ellipses.

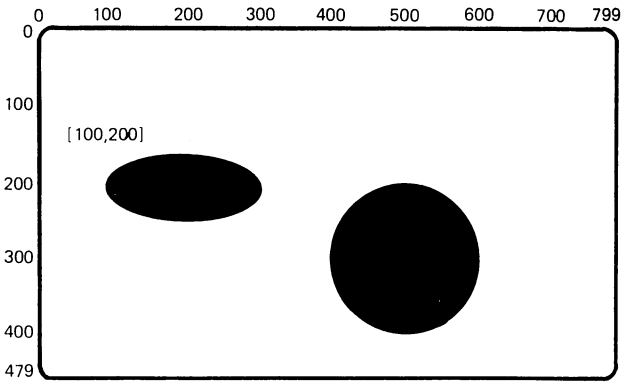
The basic format of the Polygon Fill command using a curve option is as follows:

```
F(C<position>)
```

<position> provides coordinate values for the circle's center and radius.

Figure 3–51 shows a filled ellipse and a filled circle. The circle is drawn by using the Polygon Fill command and the curve option. The ellipse is drawn by using the Polygon Fill command and the curve option with the closed curve (B) suboption.

Figure 3–51: Curve Option Example



COMMANDS
P [100,200] F (C (B) [+200] [,+100] [−200] (E))

FILLED ELLIPSE

COMMANDS
P [500,300] F (C[+100])

FILLED CIRCLE

MLO-416-86

(P)

The Polygon Fill command accepts all Position command options and arguments. The position option does not generate graphic images, as do the curve and vector options. You can use the P option with the open curve function of the curve option to set the slope at an open curve's endpoints. You can also use the P option to reset the position before and after an arc with its center at the current position.

The format of a Polygon Fill command using the position option with the curve option is as follows:

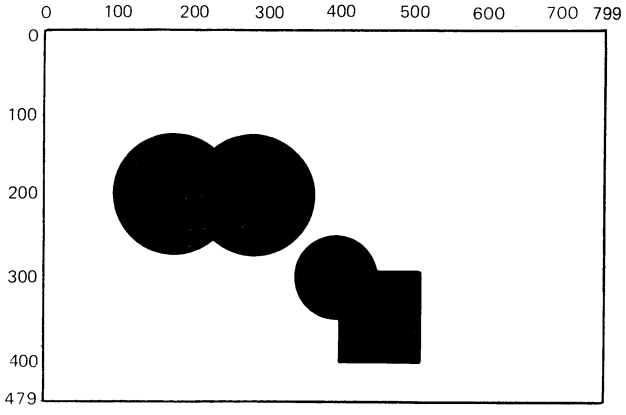
```
F(C(A + <degrees>)<position>P<position>)
```

<degrees> provides the amount of degrees to be drawn for the arc and the direction the arc is to be drawn.

<position> first provides the value of the position at which the arc drawing is to start. The second value for <position> is the new position.

Figure 3–52 shows filled, connected arcs and a filled, connected arc and rectangle. The connected arcs are drawn with the Polygon Fill command, the position option, and the open curve option. The connected arc and rectangle are drawn with the Polygon Fill command, the position option, the vector option, and the open curve option.

Figure 3–52: Position Option Example



COMMANDS
P [200,200] F (C (A + 270) [+50,−50] P[+100] C (A + 270) [−50,+50])

FILLED CONNECTED ARCS

COMMANDS
P [450,300] F (C(C,A + 270) [−50] V [,+50] [+100] [,−100] [−50])

FILLED CONNECTED ARC AND RECTANGLE

MLO-417-86

(W)

The Polygon Fill command accepts all Write command options and arguments. You can use temporary write control options as options of the Polygon Fill command or suboptions of the C and V options.

The format of a Polygon Fill command using the temporary write control command as an option is as follows:

```
F(W(<suboptions>)<options>)
```

The format of a Polygon Fill command using the temporary write control command as a suboption is as follows:

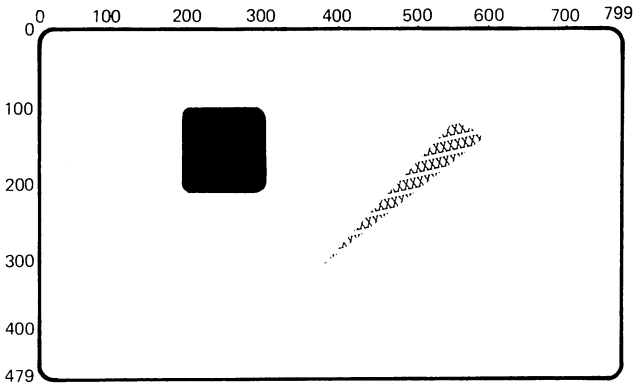
```
F(C(W(<sub-suboptions>)<suboptions>)<options>)
```

NOTE

Only the last W option in a Polygon Fill command affects the graphic image. Other W options have no effect, because no drawing takes place in a Polygon Fill command, until the complete command is read. The one exception is when you use pixel vector multiplication as a suboption of the W option.

Figure 3–53 shows a pie segment filled with Xs and a filled box with rounded corners. The pie segment is drawn with the Polygon Fill command, the temporary write control option with the shading character select (S) suboption, and the vector option. The box with rounded corners is drawn with the Polygon Fill command, temporary write control option with the foreground intensity suboption, the vector option, and the open curve option.

Figure 3–53: Temporary Write Control Option Example



COMMANDS
P [200,100] F (W (I (W)), V [+100] C (C, A −90) [,+50] V[,+100] C (C, A −90) [−50] V[−100] C (C, A −90) [,−50] V[,−100] C (C, A −90) [+50])

FILLED BOX WITH ROUNDED CORNERS

COMMANDS
P [400,300] F (W (S "X"), V (B) [+200,−100] C (C, A +30) [−200,+100] V (E))

PIE SEGMENT FILLED WITH Xs

MLO-418-86

Filling Complex Polygons

You fill complex polygons just as you would simple polygons. However, for more complex polygons, you must use a structured, logical method. The following method is one example:

1. Build a ReGIS command string that draws the outline of the polygon. This command string may use Vector, Curve, and Position commands. The outline should be a single, closed figure and must not have any gaps or cross over itself.

2. Enclose the command string from step 1 in a Polygon Fill command, as follows:

```
F(<ReGIS command string>)
```

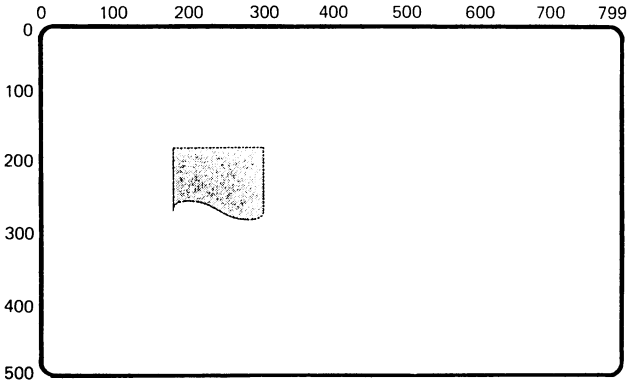
If a sequence of ReGIS commands (C, V, and P) draws a line drawing, then enclosing the same commands in F(C, V, and P) draws and fills-in the same shape.

If you want your polygon to have a contrasting outline, you can use a macrograph in the following way:

```
@:A <ReGIS commands> @;    ;"Load macrograph"
F(@A)                      ;"Fill polygon"
@A                         ;"Draw outline"
```

Figure 3–54 shows a filled paper icon with a dotted outline drawn with the Polygon Fill command, the C, V, and W options, and macrographs.

Figure 3–54: Filling a Complex Polygon



COMMANDS I

```
P [200,200]
@:X
V(B)
  [,+70]
P [−25,−10]
C (S) [+25, +10]
      [−25,+10]
      [−50,−20]
      [−25,+10]
      [+25,+10]
P [−25,−10]
V[,−70] (E) @;

F (W (I 1), @ X)
```

FILLED PAPER ICON

COMMANDS II

```
P [200,200]
W (P4) @X W(P1)
```

PAPER ICON OUTLINE

MLO-419-86

Using the Polygon Fill Command

Consider the following points when you use the Polygon Fill command:

- *Vertices.* You must specify at least three different vertices, or no drawing takes place. If you specify more than 1450 vertices, ReGIS ignores the additional vertices. Two consecutive vertices that map to the same physical pixel are counted as one vertex.

NOTE

For the V option, each argument generates one vertex. For the C option, each argument can generate more than one vertex.

- *Closed Figures.* If the commands for creating a polygon do not represent a true closed figure with all vertices given in the same direction, the Polygon Fill command acts as if consecutive vertices are connected by straight lines. The results may be unexpected.
- *Perimeter.* In some cases when you use the Polygon Fill command, the outline of the filled area may not line up with the vectors that connect the same vertices. The reason is an algorithmic restriction, which implies that you should draw a border after the filled area.
- *Single Closed Figures.* Use the Polygon Fill command to fill single closed figures only. The Polygon Fill command is not designed to fill polygons made of intersecting groups of single closed figures. Although the Polygon Fill command can fill these polygons, the results may be unexpected.
- *Current Position.* The current drawing position is saved at the beginning of a Polygon Fill command and restored at the end of the command. The Polygon Fill command saves and restores the position whether or not any drawing takes place. This feature provides some compatibility with devices that do not have the Polygon Fill command.
- *Sequence of Coordinates Options.* Any Polygon Fill command string that changes the arrangement of positions stored by sequence of coordinates options (B and E options) is not compatible with ReGIS devices that do not have the Polygon Fill command. Therefore, do not use unmatched B, S, or E options within the Polygon Fill command.

3.4.6 Polygon Fill Command Summary

Table 3–12 summarizes the Polygon Fill command options. These arguments do not have default values.

Table 3–12: Polygon Fill Command Summary

Argument	Description
(V <positions>)	Vector option. Draws filled figures, such as squares, rectangles, and diamonds.
(C <position>)	Curve and arc option. Draws filled figures, such as circles and ellipses.
(P <position>)	Position option. Can be used to reset position before and after an arc with center at the current position. Can be used with the open curve function of the curve option to set the slope at an open curve's endpoints.
(W(<suboptions>) <options>)	Temporary write control option. Lets you select temporary write control values without changing the current write control values. Temporary write control values remain in effect only for the command controlling them.

Text

The Text command lets you draw characters in many combinations of size, position, and orientation. You can use characters from one of the standard character sets or from a user-loadable character set. The standard character sets include: DEC Multinational (including ASCII), ISO Latin-1, DEC Technical, DEC Special Graphics, and 12 National Replacement Character Sets — United Kingdom, French, French Canadian, Norwegian/Danish, Finnish, German, Dutch, Italian, Swiss, Swedish, Spanish, and Portuguese. (Refer to charts in Appendix A.)

The following are options and arguments to the Text command:

- Text strings
- Character set
- Character positioning
- Size options
- Height multiplier
- Size multiplier
- String/character tilt
- Italics
- PV spacing
- Temporary text control
- Temporary write control

Specified character set, positioning, size, height, tilt, italics, and PV spacing values remain in effect until you define new values. Temporary write control option values remain in effect only for the text command controlling them. The temporary text control option has specific start and end options. Following the start option, all values are considered part of the temporary text control until the end option.

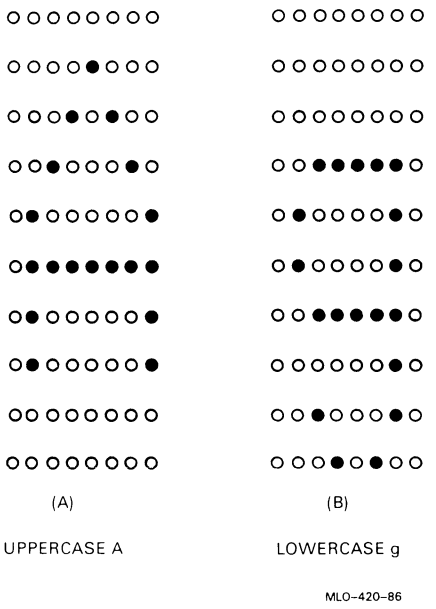
Character Drawing

Text command options determine the form of characters that are drawn while you use the Text command. However, all characters are drawn in basically the same manner. A character is taken from a stored character set, scaled according to multiplication value, positioned at the proper tilt and italic angles, then drawn at the current position. The size of the loaded alphabets is 8 x 10 bits.

The current position at the start of each character is the pivotal point for drawing the character on the screen. The starting current position is always the pixel value that is the upper left point of the stored character form. Pivoting occurs around that point. For example, a character drawn by using normal orientation (text drawn left to right on a straight line, with no tilt to the characters) appears to the right and down from the current position. If the character were tilted 180 degrees, it would be drawn to the left and up from the current position. The character escapement value (the relative displacement of the current position after each character is drawn) is then used for positioning additional characters.

Figure 3–55 shows examples of a loaded alphabet. The top row, first column, and the final two columns are blank to allow for spacing between characters. The upper left pixel of each 8 x 10 format is positioned at the current position.

Figure 3–55: Stored Character Format Examples



Format T *option*

Command Arguments	Description
" <text> "	Text string
(A <n>)	Character set option
(A0(L"designator"))	Left half selection option
(A0(R"designator"))	Right half selection option
[X,Y]	Character positioning argument
(S <n>)	Standard character cell size option
(S[<width,height>])	Display cell size option
(U[<width,height>])	Unit cell size option
(H <n>)	Height multiplier option
(M[<width,height>])	Size multiplication option
(D <angle>)	Character tilt option
(D <angle> ,S <0–16>)	String tilt option
(D <angle> ,S <0–16> , D <angle>)	String/character tilt option
(I <angle>)	Italics option
<PV>	PV spacing argument
(B) <options> (E)	Temporary text control option
(W(<options>))	Temporary write control option

Command Arguments

"text string"

Text strings define the text characters to be drawn or printed. You can use any 8-bit graphic character in the text string. This includes characters that ReGIS would recognize as command instructions, if the characters were not part of the text string: the semicolon (;), the resynchronization character, and the at sign (@) used with macrographs. In addition, you can use four control characters as part of a text string:

- Carriage return (CR). Returns the current position back to the position in effect when the current text writing command started.
- Linefeed (LF). Moves the current position down from the current baseline (the reference line from which characters are drawn), to a position equal in distance to the current vertical cell size (the amount of screen area to be written for each character).

- Backspace (BS). Moves the current position back one character position, using the current character escapement value, and provides a simple means of generating an overstrike.
- Horizontal Tab (HT). Moves the current position forward one character position, using the current text escapement value.

The format for a text string in its simplest form, that is, with all options at previously defined values, is as follows:

```
T"<text string>"
```

A text string is enclosed by a set of quotation marks, either single or double (refer to Section 3.3.2.2).

(A <n>)

The character set option is used to select which set is to be used for drawing or printing a text string. You can select any of four character sets.

Set 0 is the standard character set containing either the DEC Multinational Character Set, which is a superset of ASCII, or other predefined character sets. Set 0 contains up to 192 printable characters. Sets 1, 2, and 3 are sets you can load. These sets contain up to 96 printable characters.

If you select a loadable set (1, 2, or 3) that contains no characters, a solid rectangle appears on the page for each text string character. The same result occurs when a specified text character is not present in a selected character set.

The format for selecting a character set is as follows:

```
T(A<n>)
```

<n> is 0, 1, 2, or 3.

(A0(L"designator"))
(A0(R"designator"))

Using the "L" and "R" suboptions for alphabet 0, you can specify which character sets load into the left half (characters 32 to 127) and the right half (characters 160 to 255) of the code table.

"L" indicates the left half. "R" specifies the right half.

The variable "designator" is a 2- or 3-character sequence enclosed in quotation marks that specifies the character set to be loaded into the left or right half of the alphabet 0 code table. The first character is any of the following (which are equivalent):

- $ (dollar sign)
- & (ampersand)
- ((open parenthesis)
-) (close parenthesis)
- * (asterisk)
- + (plus sign)
- – (dash)
- . (period)
- / (slash)
- 0, 1, 2, 3, 4, or 5

RETOS includes this first character for compatibility with the ANSI character set selection process. The second and third (when applicable) characters specify the character set to be loaded. (See Table 3–13.)

Table 3–13: Character Set Designators

Character Set	Designator
ASCII	B
DEC Special Graphics (line drawing)	0
DEC Supplemental (right half of DEC Multinational)	< or %5
DEC Technical	>
United Kingdom	A
French	R
French Canadian	Q or 9
Norwegian/Danish	E or 6
Finnish	C or 5
German	K
Dutch	4
Italian	Y
Swiss	=
Swedish	H or 7

Table 3–13 (Cont.): Character Set Designators

Character Set	Designator
Spanish	Z
Portuguese	%6

NOTE

By default, ASCII is in the left half of the code table and DEC Supplemental is in the right half.

For example, to print the diamond pattern from the DEC Special Graphics character set, use either of the following commands:

```
T(A0(L"(B"))"b"
```

```
T(A0(R"(B"))"a"
```

The first command loads DEC Special Graphics into the left half of alphabet 0, and the second loads DEC Special Graphics into the right half.

[X,Y]

This argument specifies a character escapement value that defines the new current position after each character is drawn. This is one of the two ways to select character positioning, which affects spacing between characters.

The other way to select character positioning is to select a standard cell size which selects the character escapement value associated with that standard size. (See the description of (S <n>).)

Usually, the character positioning argument has only a positive X value. This produces a text string drawn across the screen from left to right, at whatever baseline orientation is in effect for the string (tilt and italics options), with equal spacing between characters. However, you can use a negative X value to draw a string backwards. You can also use Y values (+ and –) with different X values (+ or –) for a staircase effect.

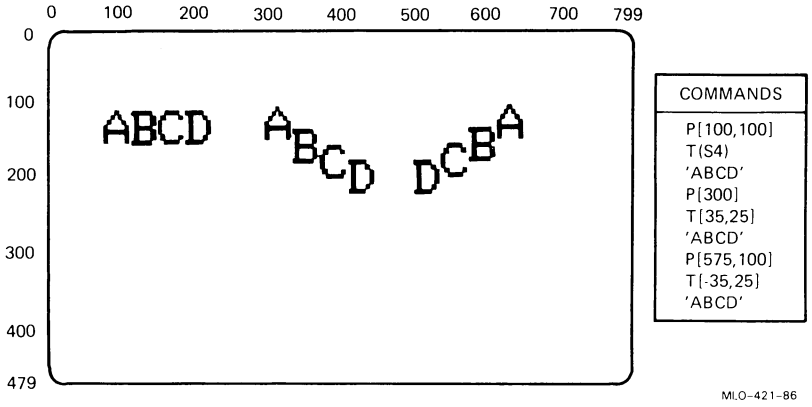
The format for the character positioning argument is as follows:

```
T<position>
```

<position> provides a relative positioning value to define character spacing ([+X,+Y], [–X,+Y], [–X, –Y], [+X, –Y], [+X], [–X], [,+Y], or [,–Y]).

Figure 3–56 shows how different character escapement values can affect how a text string is drawn.

Figure 3–56: Character Positioning Argument Example



(S <n>)

Standard character cell size is one of the three size options. It is also one of the two ways to select character escapement, which affects spacing between characters. (The other method of selecting character escapement, [X,Y], is described in this section.)

Seventeen standard character cell sizes are available: size 0 through size 16. Each standard character cell size has assigned values. These values define a display cell size (amount of display area used for each character in a text string), unit cell size (height and width values for the characters to be drawn within the display cell), and character escapement (relative displacement of the current position after each character is drawn).

The format for the standard character cell size option is as follows:

```
T(S<n>)
```

<n> is a number 0 through 16.

NOTE

Values are in ReGIS logical coordinates with default addressing mode S(A[0,0][799,479]).

Table 3–14 defines the values associated with each standard character cell size.

Table 3–14: Standard Character Cell Size Values

Standard Size	Unit Cell Size	Display Cell Size	Character Escapement
S0	[8,10]	[9,10]	[9,0]
S1	[8,20]	[9,20]	[9,0]
S2	[16,30]	[18,30]	[18,0]
S3	[24,45]	[27,45]	[27,0]
S4	[32,60]	[36,60]	[36,0]
S5	[40,75]	[45,75]	[45,0]
S6	[48,90]	[54,90]	[54,0]
S7	[56,105]	[63,105]	[63,0]
S8	[64,120]	[72,120]	[72,0]
S9	[72,135]	[81,135]	[81,0]
S10	[80,150]	[90,150]	[90,0]
S11	[88,165]	[99,165]	[99,0]
S12	[96,180]	[108,180]	[108,0]
S13	[104,195]	[117,195]	[117,0]
S14	[112,210]	[126,210]	[126,0]
S15	[120,225]	[135,225]	[135,0]
S16	[128,240]	[144,240]	[144,0]

(S[<width,height>])

Display cell size is one of the three size options. It lets you define the height and width of a display cell that differs from those in the standard character cell sizes. This display cell represents the amount of image area for each character of text.

The format for the display cell size option is as follows:

```
T(S[width,height])
```

The width and height values provide the size of the display cell in ReGIS logical coordinates.

No specific unit cell sizes are associated with display cell size option values. The display cell size option does not change the size of the printed character; the unit size option changes character size.

(U[<width,height>])

Unit cell size is one of the three size options. It lets you define the size of the characters.

The format for the unit cell size option is as follows:

```
T(U[width,height])
```

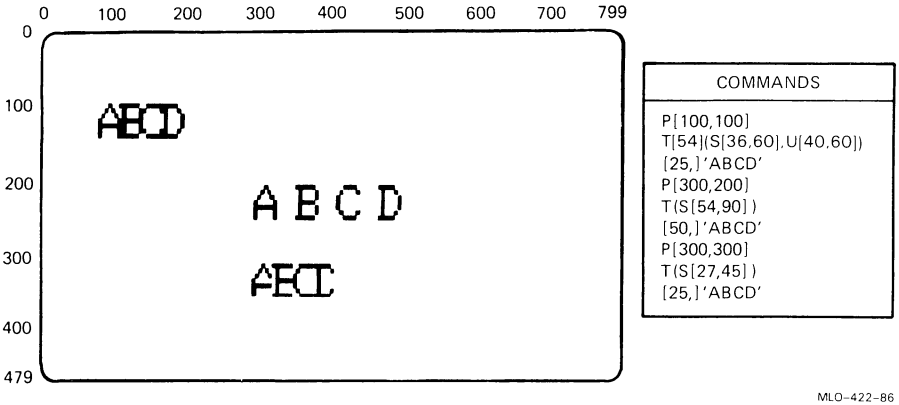
The width and height values provide the size of the unit cell in ReGIS logical coordinates.

Unless you want special effects, the unit cell size should be as close as possible to the display cell size. ReGIS uses all of the display cell for each character, filling any unused space with the appropriate background intensity. ReGIS also uses only the amount of defined display area, regardless of the unit cell size.

All characters are justified at the upper left corner in the display cell, relative to the current character baseline orientation. When the unit cell is smaller than the display cell, the whole character appears on the page, with the unused part of the display cell at the background value. When the unit cell is larger than the display cell, only the part of the character that can fit into the display area appears on the page.

Figure 3–57 shows what happens when the same unit cell size and different display cell size values are used for printing the same text string.

Figure 3–57: Display Cell and Unit Cell Size Options Example



(H <n>)

The height multiplier option lets you change the height of characters without affecting the width. The height multiplier option changes the height value of both the display and unit cells.

The format for the height multiplier option is as follows:

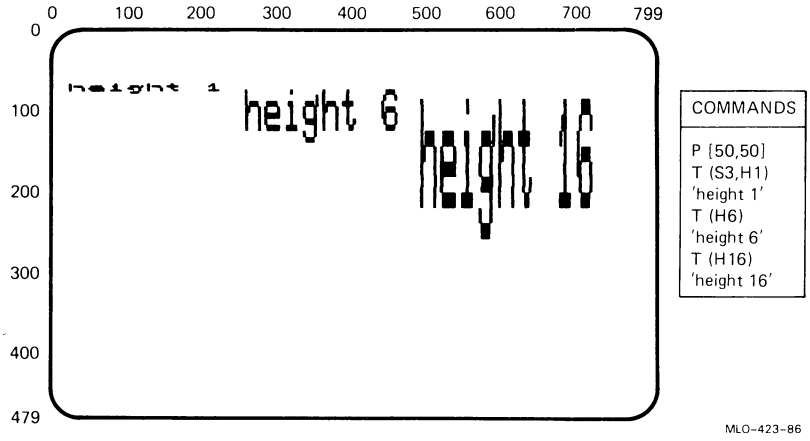
```
T(H<n>)
```

<n> provides a multiplication value.

Multiplication is done against the standard character sizes, with H2 being the height of standard size 1. Therefore, an option value of 8 changes the height components of the display and unit cells to S4, while a value of 16 changes the same components to S8. The change in display cell and unit cell height values occurs regardless of differences that may exist in those values before the height multiplier option is invoked.

Figure 3–58 shows the effect of the height multiplier option. As shown, only the height values change; character positioning and width values remain the same.

Figure 3–58: Height Multiplier Option Example



(M[<width,height>])

The size multiplier option is an alternative way of specifying the unit cell size, provided for VT125 compatibility. You can specify different multiplication factors, including fractional values, for width and height.

The format of the size multiplier option is as follows:

```
T(M[width,height])
```

Width and height values provide multiplication values.

With this option, the unit width equals the width multiplier you specify, multiplied by the standard size unit width (S1). The unit height equals the height multiplier you specify, multiplied by the standard unit height divided by 2.

String/Character Tilt Options

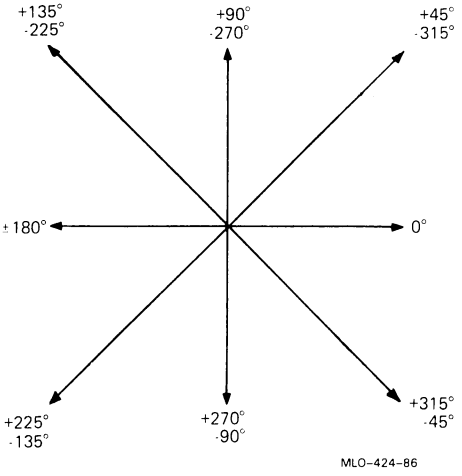
The normal orientation of text is along a horizontal baseline, with characters drawn from left to right. However, in some graphic applications, you may want to write the text at an angle. The string/character tilt options let you tilt text strings and the characters within text strings, in 1-degree increments for 360 degrees. There are three types of tilt options:

- Character tilt option — Defines the tilt value for the characters in the string.
- String tilt option — Defines the orientation of the text string to the horizontal baseline.
- String/character tilt option — Defines two tilt values: one for the text string as a unit, and one for the characters in the string.

These tilt options are separate from italic tilting.

Figure 3–59 is a tilt compass that shows the direction of tilt for some tilt values you can use with the tilt options.

Figure 3–59: Tilt Compass



(D <angle>)

The character tilt option defines the tilt for the characters in the string.

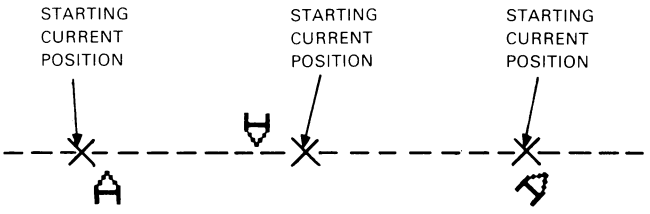
The format for the character tilt option is as follows:

```
T(D<angle>)
```

<angle> provides the value of the character tilt, in degrees.

Figure 3–60 shows how different angles affect how characters are drawn.

Figure 3–60: Character Tilt Option Directions



(D <angle> ,S <0–16>)

The string tilt option defines a baseline. ReGIS draws the characters in a text string along this baseline. When you use this option, the baseline of each character in the string is at the defined tilt.

The format for the string tilt option is as follows:

```
T(D<angle>, S<0-16>)
```

<angle> provides the value of the string tilt, in degrees.

S <0–16> identifies one of the 17 standard sizes. Escapement associated with that size determines spacing between characters during the tilt option.

Some devices, such as the VT240, cannot accurately rotate text. When a character is drawn at a diagonal orientation, distortion occurs (relative to the size of that same character drawn at an orthogonal orientation). In some graphic applications, you may want to keep this distortion. If not, you can partially correct the distortion by adjusting the size value. To adjust the distortion, define a size one-half the desired width for diagonal characters. Note that diagonal characters always have a slight distortion of size. This distortion is more apparent when drawing larger characters at 45, 135, 225, and 315 degrees. You may need to perform an additional adjustment.

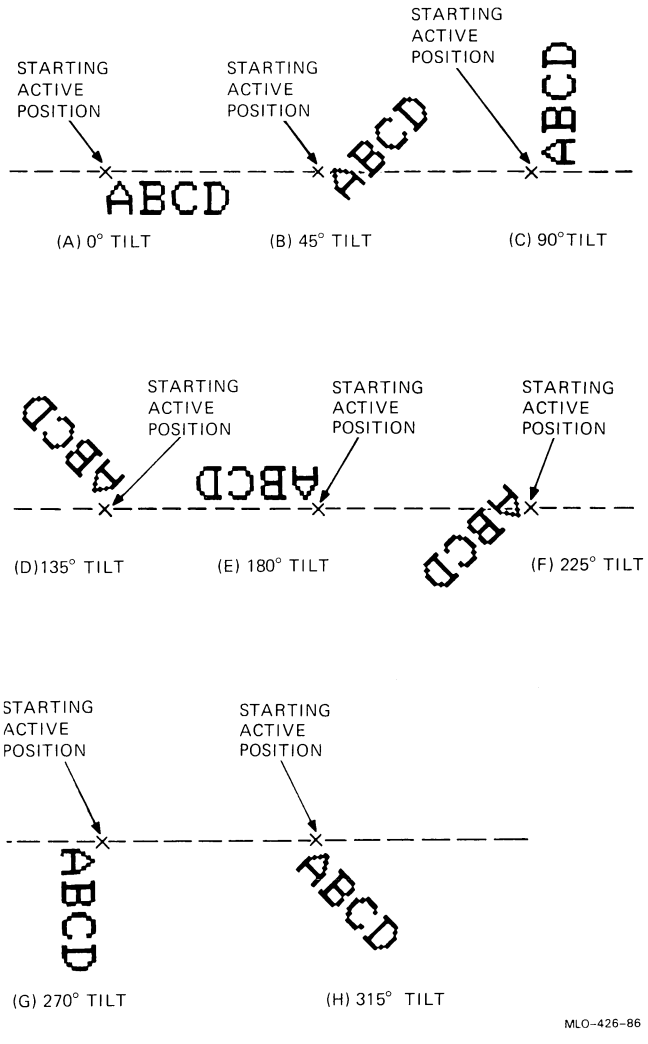
For example, if you are drawing a size 8 character (64 x 120 screen coordinates) at 45 degrees, you would define the character size as a 4 (32 x 60 coordinates), with an H factor of 6 (60 pixels, 120 coordinates). However, in this case, you could get a better result by defining set 3 (24 x 45 coordinates) with an H factor of 5 (50 pixels, 100 screen coordinates).

RETOS spaces characters in a tilted string further apart than the characters in a horizontal string. (See Figure 3–61.) You can explicitly adjust the character spacing by using the T[X,Y] command.

RETOS rotates text in increments of 45 degrees, which is compatible with the VT200 and VT300 Series terminals.

Figure 3–61 shows how each string tilt value affects a text string drawn on the screen.

Figure 3–61: String Tilt Directions



(D <angle> , S <0–16> , D <angle>)

The string/character tilt option first defines a tilt orientation for a text string, then a separate orientation for the characters in the string.

The format for the string/character tilt option is as follows:

```
T(D<angle>, S<0-16>, D<angle>)
```

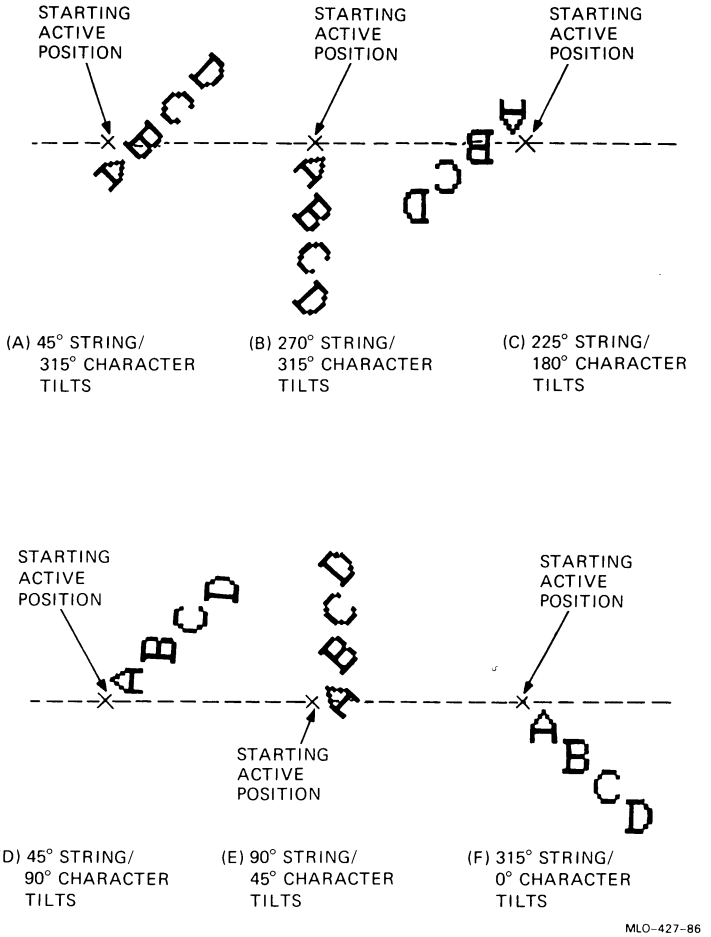
<angle> provides the value of the string tilt, in degrees.

S <0–16> identifies one of the 17 standard sizes. Escapement associated with that size determines spacing between characters during the tilt option.

The VT240 cannot accurately rotate text; to adjust for this, VT240 ReGIS increases the size of the rotated characters. RETOS spaces characters in a tilted string further apart than the characters in a horizontal string. (See Figure 3–62.) You can explicitly adjust the character spacing by using the T[X,Y] command.

Figure 3–62 shows different effects produced by the string/character tilt option. This figure does not show character distortion. The size value was adjusted. For example, to match a size 1 character drawn at 90 degrees on a string tilted at 45 degrees, use the command T(D45, S0H2, D90).

Figure 3–62: String/Character Tilt Option Directions



(I <angle>)

The italics option lets you tilt characters without changing their orientation to the baseline, giving you slanted text.

The format for the italics option is as follows:

```
T(I<angle>)
```

<angle> identifies the degree of italic slant and the direction of the slant (to the left, if no sign; to the right, if negative sign).

When drawing italic characters, ReGIS displaces each horizontal slice of the characters. However, italic slants do not significantly distort the basic width and height values of a character. You can use italic slants with the tilt option to create slant/tilt effects not available with either tilt or italics options alone.

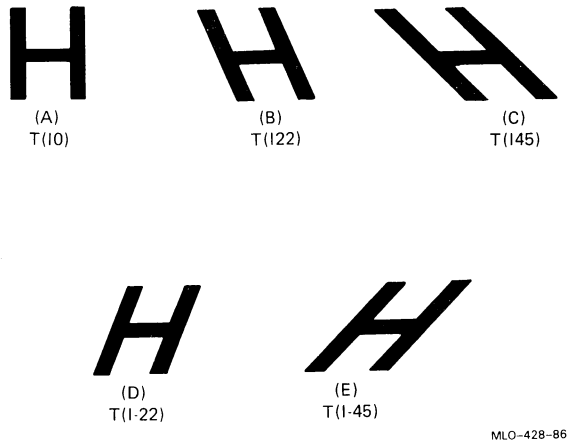
RETOS uses the values in Table 3–15 to represent italic angles.

Table 3–15: RETOS Italic Angles

Angle Requested (degrees)	Angle Drawn (degrees)
Less than –30	–45
–30 to –1	–26.5
0	0
1 to 30	26.5
Greater than 30	45

Figure 3–63 shows an H character with different italic slant values.

Figure 3–63: Italic Option Slant Values



<PV value>

The Text command uses PV spacing arguments to define overstrike, superscript, and subscript functions. The direction specified by the PV value is relative to the character rotation.

In Text commands, each PV value defines a movement equal to one half of the defined display cell, in the specified direction. The PV multiplication factor does not affect this movement.

The format for the PV spacing argument is as follows:

```
T<PV value>
```

<PV value> defines the offset to occur with each PV value specified equal to an offset of 1/2 of the currently defined display cell size.

The PV spacing argument can use any of the eight pixel vector direction values. The following are the most useful:

Value	Function
1	Superscripts. Displaces the character up and to the right of the baseline.
2	Superscripts. Displaces the character straight up from the baseline.
4	Overstrikes. A 44 displaces the character back over a previously drawn cell.
6	Subscripts. Displaces the character straight down from the baseline.
7	Subscripts. Displaces the character down and to the right of the baseline.

You can use PV offset values of 3, 5, and 0, but they partially overwrite the previous character (using normal escapement).

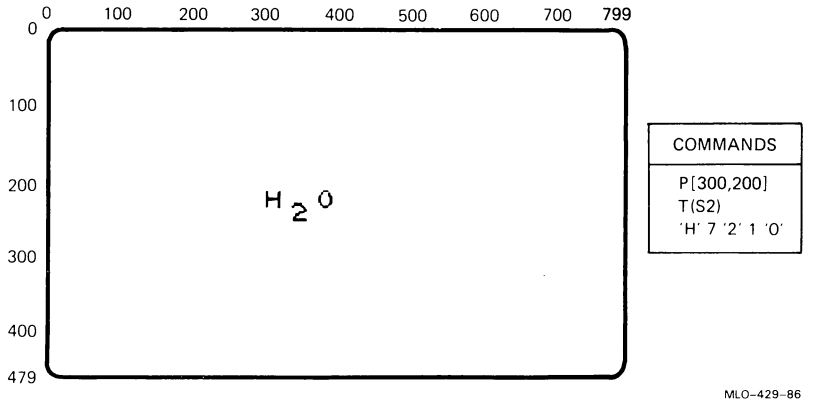
A specified PV value offsets the following text strings from the original baseline, until you correct the offset. You correct the offset by specifying the opposite PV value. For example, 6 corrects 2, and 2 corrects 6. For an overstrike (44), use the PV value of 00.

NOTE

Text PV spacing action is in relation to the baseline. This action rotates with the baseline, if the baseline is tilted. PV spacing is done in terms of display cell size. If the escapement value is set differently (by the character positioning argument), a PV 44 does not produce the desired overstrike. The backspace code (0/8) moves backwards one character space as set by the character positioning argument.

Figure 3–64 is an example of subscripting with the PV spacing argument.

Figure 3–64: PV Spacing Argument Example



(B) <options> (E)

This is the temporary text control option. Text command option values you specify remain in effect until you change them. You can use temporary text controls to draw text strings with new Text command option values, without affecting the current values.

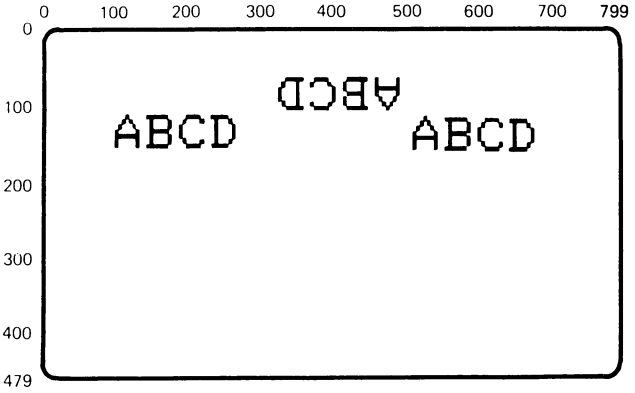
The format for the temporary text control is as follows:

```
T(B)<options>(E)
```

The temporary text controls work as a bounded sequence. Options in the sequence remain in effect until the sequence ends with an end (E) option. A new command (position, vector, etc.) does *not* terminate the temporary text control. Only an (E) ends the sequence. Temporary text controls cannot be nested, because ReGIS does not recognize a second (B) option until the first (B) option has been terminated by an (E). Values specified between the begin (B) option and end (E) option are temporary. After the end (E) option, Text command option values return to the values previously in effect.

Figure 3–65 shows an example of a temporary text control option.

Figure 3–65: Temporary Option Example



(W(<suboptions>)) <options>

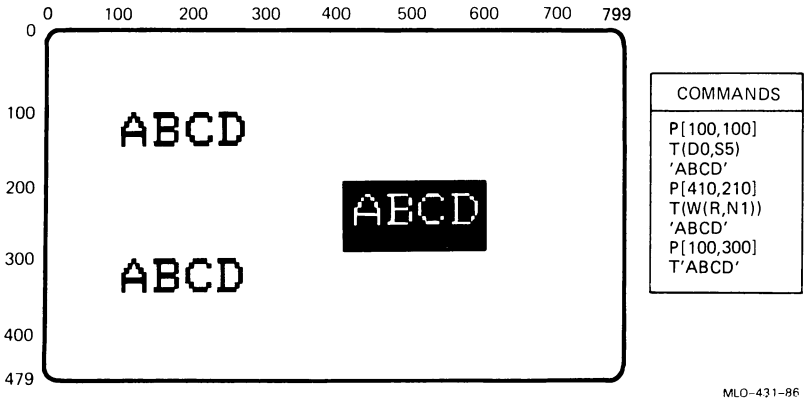
The temporary write control option lets you change the write control values for one Text command, without affecting the current Write Control command option values. The temporary write control values affect only the Text command controlling the option. At the next command keyletter, even if it is another Text command keyletter, the write control values return to the current Write Control command option values. You can use this option to change colors, as well as writing mode (overlay, replace, or erase).

The format for the temporary write control option is as follows:

```
T(W(<suboptions>))<options>
```

Figure 3–66 shows an example of the temporary write control option.

Figure 3–66: Text Command Temporary Write Control Option Example



3.4.7 Text Command Summary

Table 3–16 summarizes the Text command arguments and their default values.

Table 3–16: Text Command Summary

Argument	Default	Description
'text'	None	Text string. Includes text to be displayed. Text string characters must be delimited by either single quotes ('text') or double quotes ("text").
(A <0–3>)	(A0)	Character set option. Selects which of four possible character sets (<0–3>) to use for processing text string characters.
(A0(L"designator"))	(A0(L"(B"))	Selects which standard character set to load into the left half of alphabet 0.
(A0(R"designator"))	(A0(R"(<"))	Selects which standard character set to load into the right half of alphabet 0.
[X,Y]	[+9,+0]	Character positioning argument. Lets you vary positioning between text string characters. Default value comes from the character escapement of standard cell size (S1). [X,Y] values are relative.
(S <0–16>)	(S1)	Standard character cell size option. Defines a set of display cell, unit cell, and character escapement values to be used in processing text string characters. There are 17 different sizes (<0–16>) available. The character tilt is used to set the positioning.

Table 3–16 (Cont.): Text Command Summary

Argument	Default	Description
(S[<width,height>])	(S[9,20])	Display cell size option. Lets you change size of screen area written for each character. Default value comes from standard cell size (S1).
(U[<width,height>])	(U[8,20])	Unit cell size option. Lets you change scaling of characters. Default value comes from standard cell size (S1).
(H <number>)	(H2)	Height multiplier option. When selected, this option changes the display cell and unit cell size height values to a value equal to S1 multiplied by the specified multiplier without affecting width values or positioning. The default value comes from standard cell size.
(M[width,height])	(M[1,2])	Size multiplication option. Provides multiplication factors for the height and width values of the unit cell size associated with the standard cell size (S1).
(D <a>)	(D0)	Character tilt option. Defines tilt value for the characters in the text string. <a> defines the degrees of the tilt for the characters.
(D <a> S <0–16>)	(D0 S1)	String tilt option. Defines tilt of text string, as a whole, relative to the horizontal. <a> defines the degrees of the tilt; <0–16> provides a standard size value used to compute positioning during the tilt.

Table 3–16 (Cont.): Text Command Summary

Argument	Default	Description
(D <a> S <0–16> D <a>)	(D0 S1 D0)	String/Character tilt option. Defines separate tilt values for the string and the characters in the text string. The first <a> defines the degrees of tilt for the string; the second <a> defines the degrees of tilt for the characters in the string; <0–16> provides a standard size value used to compute positioning during the tilt.
(I <a>)	(I0)	Italics option. Defines a degree of tilt (<a>) for characters without changing their orientation to the current baseline.
<PV>		PV spacing argument. Uses PV values to select superscript, subscript, and overstrike functions.
(B) <options> (E)	None	Temporary text control option. Lets you select temporary Text command options, without changing the current values. Temporary values remain in effect until you use (E).
(W(<options>))	None	Temporary write control option. Lets you select temporary Write Control command values, without changing the current Write Control command values. Temporary write control values remain in effect only for the command controlling them.

Load

RETOS can store up to four character sets at one time: a standard character set, stored as character set 0, and three loadable sets stored as character sets 1 through 3. Each set contains up to 95 characters. You use the Load command to select, load, or reload sets 1, 2, and 3. You cannot load character set 0, the standard character set.

There are three arguments to the Load command:

- Select character set
- Specify name
- Load character cell

Format

L *option*

Command Arguments	Description
(A <n>)	Select character set option
(A" <name> ")	Specify name option
" <character> " <hex numbers> ;	Load character cell argument

Command Arguments

(A <n>)

The select character set option lets you select which of the three optional character sets to load: set 1, 2, or 3.

The format for the select character set option is as follows:

```
L(A<n>)
```

<n> is either 1, 2, or 3.

After you define a select character set value, it remains in effect until you use a new select character set option. Other ReGIS commands can execute without affecting the character set selected for loading.

The select character set option only defines which character set to load. You load characters into the character set by using the load character cell argument. You can load characters into the character set as needed. You do not have to load the complete set at one time.

(A" <name> ")

The specify name option lets you define a name for a loaded character set. You do not need this option for set load tasks. You can select a name up to 10 characters long.

The format for the specify name option is as follows:

```
L(A"<name>")
```

" <name> " is the name you choose for the currently selected character set.

You can use the specify name and select character set options together. That is, you can define the name of the character set when you select that character set. Use the select character set option first. Otherwise, ReGIS applies the name to the character set already selected, not the character set you are selecting.

" <character> " <hex numbers> ;

The load character cell argument lets you build a character you want to store. Each character cell consists of 80 pixels in an 8 x 10-pixel array. The load character cell argument uses hexadecimal numbers to define the on/off pixel configuration for each row of pixels. You can draw up to 10 hexadecimal pairs to define the contents of a character cell.

The format for the load character cell argument is as follows:

```
L"<character>"<hex numbers> ;
```

<character> is the single ASCII character to serve as the call letter for the character cell being loaded.

<hex numbers> provides the hexadecimal numbers, with one pair of values supplied for each of the 10 rows of the character cell that can be defined. Pairs are separated by commas.

A semicolon (;) is used to terminate the load character cell argument. If more than one character is being defined, the argument, up to the semicolon, is repeated.

A call letter provides a way to select the stored character in Text commands. You can use any single ASCII character for the call letter, including numerals or a space. The call letter does not have to match the character you are storing.

You build the character cell from top to bottom. Two hexadecimal digits describe each row in the character cell, starting with the top row. If you list less than 20 digits (10 rows), ReGIS pads the rest of the cell with 0s. If you give only one digit for a row, ReGIS assumes the first digit is 0. If you give more than two digits for a row, ReGIS uses the extra codes for the next row(s).

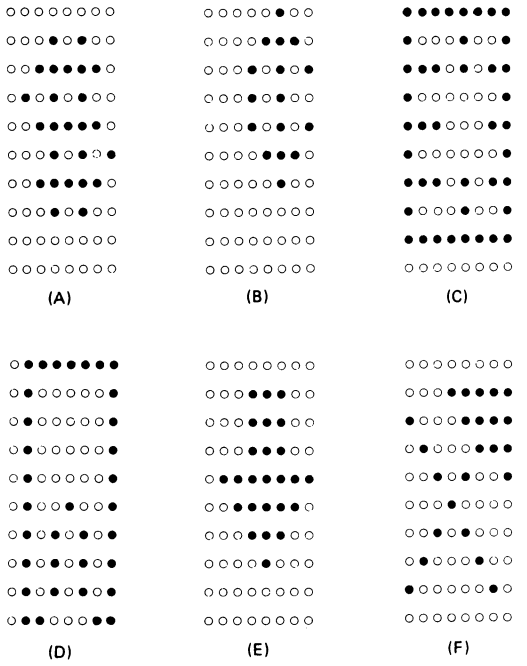
Each of the two hexadecimal digits describing a row of the character cell represents four bits in the pattern. The first digit represents the leftmost four bits in the row, and the second digit represents the rightmost four bits. Table 3–17 lists the bit pattern associated with each hexadecimal code. In each hexadecimal code, bit 3 represents the leftmost bit, and bit 0 represents the rightmost bit. You specify two hexadecimal code values for each row.

Table 3–17: Bit Patterns Associated with Hexadecimal Codes

Hexadecimal Code	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

Figure 3–67 shows examples of characters you can load and the hexadecimal codes required.

Figure 3–67: Load Character Cell Argument Example



COMMANDS
L(A3"alpha")
(A) "S" 00,14,3E,54,3E,15,3E ,14;
(B) 'c'04,0E,15,14,15,0E,04;
(C) 'C'FF,89,EB,81,E3,81,EB,89,FF;
(D) "b"7F,41,41,41,41,49,55,55,55,63;
(E) '1'0,1C,1C,1C,7F ,3E,1C,8;
(F) "."0,1F,8F,47,29,10,28,44,82;

NOTES:

1. ● INDICATES BIT ON; o BIT OFF.

2. LETTERS IN PARENTHESES ARE FOR DESCRIPTION ONLY: THEY ARE NOT PART OF THE COMMAND.

MLO-432-86

3.4.8 Load Command Summary

Table 3–18 summarizes the Load command arguments and their default values, if any.

Table 3–18: Load Command Summary

Argument	Default	Description
(A <1–3>)	(A1)	Select character set option. Selects one of the three loadable character sets to use for any following load character cell activity.
(A" <name> ")	None	Specify name option. Provides a name (<name>) of up to 10 characters for the currently selected character set. You can use this option with the select character set option: (A <1–3> " <name> ").
" <ASCII> " <hex numbers> ;	None	Load character cell argument. Used to generate characters to store in the selected set. <ASCII> is a single ASCII character that identifies the character cell. The variable <hex numbers> defines the bit pattern of the character to store on a line-by-line basis.

Macrograph (@)

The macrograph facility lets you define, store, and display graphic images. For example, you can store a logo as a macrograph, then use the logo in different displays. You do not have to rebuild the logo each time you need it.

A macrograph may consist of complete sets of command strings or any arbitrary string of characters.

You can use uppercase or lowercase characters to identify macrographs (a or A identify the same macrograph). You can select a macrograph as part of another macrograph, with up to 26 macrographs nested together. However, you cannot use a macrograph as part of itself. For example, if "A" is the first macrograph of a set of nested macrographs, none of the other macrographs can be "A".

You can define macrographs at any point in a ReGIS stream, without affecting the interpretation of that stream, except as follows:

- As part of a quoted string. ReGIS does not recognize commands in a quoted text string. If you try to define a macrograph in a text string, ReGIS interprets the commands and definition as simple text.
- In another macrograph. You can nest up to 26 macrographs. However, you must define macrographs separately. You include only the desired invoke macrograph operation in another macrograph definition, not the contents of the nested macrograph.

There are three types of macrograph operations:

- Define macrograph
- Invoke macrograph
- Clear macrograph

Format

@: <character> <definition> @;

@ <character>

@.

@: <character> @;

Macrograph Commands

@: <character> <definition> @;
This operation defines and stores the contents of a selected macrograph.

The definition cannot contain the following character sequences:

@:
@;
@.

No characters, including CR, LF, BS, HT, and SPACE, are allowed between the first at sign (@) and the colon (:); between the colon (:) and <character> ; or between the last at sign (@) and the semicolon (;).

@ <character>
This operation executes the contents of a selected macrograph. The contents of the selected macrograph are inserted in the ReGIS command stream. If the macrograph includes invocations of other macrographs, they are inserted into the command stream as well.

No characters, including CR, LF, BS, HT, and SPACE, are permitted between the at sign (@) and the macrograph <character> .

ReGIS uses the current values for command information in a macrograph, such as Write Control, Screen Control, or Text command values, unless you change the values. You can specify new values in the definition, by using temporary options or through Text, Screen Control, or Write Control commands.

@.
This operation clears the macrograph definitions stored in all 26 macrograph locations.

No characters, including CR, LF, BS, and HT, are allowed between the at sign (@) and the period (.).

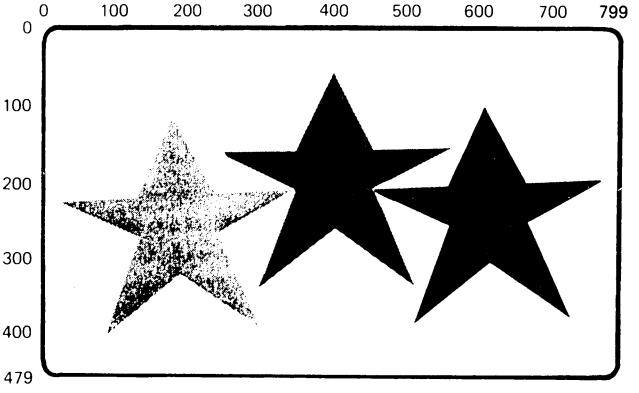
@: <character> @;
This operation clears only the contents of the specified macrograph.

No characters, including CR, LF, BS, and HT, are allowed in any of the following locations:

- Between the opening at sign (@) and the colon (:)
- Between the colon (:) and <character>
- Between the closing at sign (@) and the semicolon (;)

Figure 3–68 shows how the macrograph for a shaded star is defined, stored, and then invoked.

Figure 3–68: Macrograph Example




```
COMMANDS
@:S
W(S1)
P[+50,+50]
V[-100]
[-100,-50]
[+100]
[+50,-100]
[+50,+100]
[+100]
[-100,+50]
P[+50,+125]
W(S1)
V[-50,-125]
[-100]
[-50,+125]
W(I0)
V[+100,-75]
[+100,+75]
@;
P[200,200]
W(I3)
@S
W(I2)
P[400,150]
@S
W(I1)
P[600,200]
@S
```

MLO-433-86

3.4.9 Macrograph Command Summary

Table 3–19 summarizes macrograph operations, which have no default values.

Table 3–19: Macrograph Operation Summary

Operator	Description
@: <character> <definition> @;	Define macrograph. Defines the single letter used to identify a macrograph and the macrograph definition to store. The letter <character> is not case sensitive.
@ <character>	Invoke macrograph. Executes contents of the macrograph specified by <character> . The variable <character> is a single letter and is not case sensitive.
@.	Clear all macrographs. Deletes stored macrograph descriptions from all 26 macrograph storage locations.
@: <character> @;	Clear specified macrograph. Clears the contents of a single macrograph storage location. This operator also defines a macrograph with no definition.

3.5 ReGIS Commands Not Supported by RETOS

RETOS supports all VT240/241 and VT330/340 terminal commands except the Report command (R) and the following Screen Control command (S) option:

Command Options	Description
[ ]	Scrolling option
(C)	Graphic cursor control option
(D)	Data movement control option
(H)	Hardcopy control option
(T)	Time delay option
(W)	Temporary write control option

3.6 Screen Addressing and Resolution

For compatibility with the LJ250 printer, the default screen addressing is S(A[0,0][799,479]). This means that the ReGIS to sixel graphics converter scales pictures so that 800 logical units correspond horizontally to 720 or 1440 pixels in the output file.

However, if you design a picture for display on the LJ250 printer and a one-to-one correspondence between a logical unit and a pixel is important, then begin your file with the following:

- S(A[0,0][719,944]) for low resolution (90 dots/inch)
- S(A[0,0][1439,1889]) for high resolution (180 dots/inch)

RETOS supports ReGIS files created for display on video terminals. However, dot-to-dot compatibility between the RETOS and the VT241 terminal generally does not exist.

3.7 Color Mapping

RETOS provides a 16-entry ReGIS color map for compatibility with programs using the S(I) and W(I) commands. In order that LJ250 colors closely match video colors, RETOS selects HLS values for each color. Table 3–20 lists default color output map values.

Table 3–20: Default Color Output Map Values

Output Map Entry	Default Converter Value	HLS Values		
		H	L	S
0	Black	0	0	0
1	Blue	0	50	60
2	Red	120	46	72
3	Green	240	50	60
4	Magenta	60	50	60
5	Cyan	300	50	60
6	Yellow	180	50	60
7	Gray 50%	0	53	0
8	Gray 25%	0	26	0
9	Pale blue	0	46	29
10	Pale red	120	43	39
11	Pale green	240	46	29
12	Pale magenta	60	46	29
13	Pale cyan	300	46	29
14	Pale yellow	180	46	29
15	Gray 75%	0	80	0

Usually, ReGIS applications specify colors in HLS or RGB form. RETOS converts colors to HLS form before writing them to the output file.

3.8 Handling of ANSI Text and Escape Sequences

Ideally, RETOS input files contain pure ReGIS graphics. However, to provide partial support for files that mix text and graphics, RETOS parses and ignores ANSI escape sequences, control strings, device control string introducers, and control characters, except the following:

- BS (backspace)
- HT (horizontal tab)
- LF (line feed)
- CR (carriage return)

ReGIS syntax includes BS, HT, LF, and CR. If interpretation of ANSI text as ReGIS commands still occurs, it usually creates no problem. Text is unlikely to include valid ReGIS commands affecting the output.

Chapter 4

Sixel Graphics Protocol and Output

Sixel graphics is a method of defining and transmitting encoded bit map data. This chapter discusses printing of sixels, the structure of the sixel graphics protocol, sixel private control commands, selecting sixel mode, and RETOS's sixel output.

4.1 Terminology

This section defines terms relating to the conversion and printing of the sixels protocol.

DOT — Smallest displayable unit, a light dot on a screen, an ink dot on the paper. Dots can be round, oval, square, rectangular, and small or large.

DOT/PIXEL/PIXEL–SPOT relationship — Imaging devices use several dots to represent a single pixel through a pixel-spot. Multiple dots can be used to cover an area larger than a dot size or to produce darkness or scaling.

GRID — Positions on the page where devices place pixel-spots. Dimensionless points represent these positions. Pixel-spot size can exceed the grid size. A horizontal grid-size parameter defines the horizontal distance between two positions. The horizontal grid-size parameter and the aspect ratio parameter define vertical distance between two positions.

Pixel-spot size can be larger or smaller than the distance between two positions. This relationship of pixel-spot and grid size varies from device to device and from one set of parameters to the next on the same device.

Grid sizes on the device may not match the grid-size specification. In that case, the device selects a grid that best represents the specified grid. This "best fit" grid often becomes the "actual grid" or "actual grid size." Most imaging operations use the actual grid and not the exact specified grid.

OVERLAP — Percentage of pixel-spot that is larger than the grid.

PICTURE DEFINITION — Data describing the image, including colors, size, pixel aspect ratio, and encoded rasters. The picture definition excludes formatting information, such as position or actual presentation size.

PIXEL — Logical rectangular image area defin by each bit of sixels data, as intended by the generation software. An aspect ratio defines the shape of a pixel. The pixel has no size.

PIXEL ASPECT RATIO — Shape of the pixel as a ratio of the vertical side of the rectangle and the horizontal side. For example, a square pixel has an aspect ratio of 1 to 1 (or 10 to 10), and a pixel twice as high and wide has an aspect ratio of 2 to 1 (or 20 to 10).

PIXEL–SPOT — Area imaged (printed) for each pixel. Pixel-spots have shape (round, oval, square) and size. The shape and size are device dependent and do not necessarily relate to the grid size.

RASTER — All pixels defining a single image. For purposes of this book, a raster contains pixels defined in a single sixels control string.

RASTER ASPECT RATIO — Relative size of horizontal pixels to vertical pixels in a raster. No direct relationship exists between the raster aspect ratio and the pixel aspect ratio.

RASTER SIZE — Resultant size of the raster after printing, based on the grid size.

When you use the context of pixels not yet printed, raster size is the number of horizontal and vertical pixels of the raster.

SIXEL — A group of six vertical pixels represented by six bits in a character code of seven or eight bits.

SIXELS CONTROL CODES — Codes in the picture definition that provide additional information beyond the encoded raster, such as color and line breaking.

SIXELS DATA — This term includes only the encoded raster portion of the picture definitions.

4.2 Sixel Printing

Sixel graphics imaging (printing) consists of setting context and attributes for the pixels and then imaging each sixel in received order on adjacent grid positions.

A pixel represents the individual dots of ink you see on a printed page. A bit value of 1 means print a dot (pixel). A bit value of 0 means leave a space.

In sixel mode, printing starts at the top left corner of the bit map (the origin). Printing of each sixel advances the sixel active position to the next horizontal grid position. The distance between sixels is equal to the horizontal grid size selected by parameters of the device control string.

Positioning is relative to the active position. A graphic carriage return or a next line command moves the active position to the left margin. (For a summary of graphic control codes, refer to Section 4.5.2.) Sixel drawing proceeds from left to right, top to bottom.

The color currently selected by the color specifier control code determines pixel-spot color. Optional color specifier parameters define color maps. The one bits of a sixel print in the selected color. If your picture has pixels of different colors in the same sixel, send a new line of sixels for each color with a graphic carriage return between them.

The string terminator (ST) causes the exit from sixel graphics mode. Other characters causing the exit from sixel graphics mode include:

- ANSI control characters ESC and CAN
- C1 control codes

4.3 Structure of the Protocol

The structure of the sixel graphics protocol supports a layered system approach, where several independent processes define or interpret portions of the total data. That allows describing the size of an image that is independent of the actual image definition.

The data and commands separate into three primary fields:

- *Picture definition* — Used by creation software, editing software, imaging devices. This is the body of the protocol. Picture-generating

devices create files that contain the image definition, but no formatting information (or default formatting information).

- *Formatting information* — Added to the picture definition by the page composition software to define the size of the picture data. Including size in the *protocol selector,* allows you to add parameters without affecting the *picture definition* data.
- *Positioning data* — Used by page composition software. Application software determines the first pixel position. Other pixel positions are relative to the first, based on the grid size and aspect ratio.

4.4 Protocol Selector — Formatting Information

Page composition software adds formatting information to the sixel protocol selector. The DCS introducer and the protocol selector place the converter in Sixels mode. The protocol does not have an initial state. Following is the format of the protocol selector:

```
Ps1  ;  Ps2  ;  Pn3  q
```

Ps1 selects the horizontal grid size, vertical grid size, and pixel aspect ratio. The grid size defines the size of the area where you can place a single pixel. You select the Ps1 value that most closely matches the device developing the sixel data.

Ps1 exists only for compatibility with older devices (LA12, LA34, LA50, LA100, or LA210). With new software, set the macro parameters for the older devices, then override them with explicit parameters to get best results from a new machine. Unless you require compatibility with older devices, do not use macro parameters.

Table 4–1 lists the *Ps1 parameter* macro values. You can override the Ps1 value with the Pn3 parameter.

Table 4–1: Macro Parameter Selections

Ps1 (Parameter)	Horizontal Grid Size	Aspect ratio Vert pix : Horiz pix	Vertical Grid Size
0 (default)	.0075″	200:100	.0150″
1	.0075″	200:100	.0150″
2	.0030″	450:100	.0142″

Table 4–1 (Cont.): Macro Parameter Selections

Ps1 (Parameter)	Horizontal Grid Size	Aspect ratio Vert pix : Horiz pix	Vertical Grid Size
3	.0045″	300:100	.0150″
4	.0060″	250:100	.01425″
5	.0075″	183:100	.0150″
6	.0090″	150:100	.0150″
7	.0105″	130:100	.0144″
8	.0120″	112:100	.0144″
9	.0135″	100:100	.0150″

Ps2 selects a background color. The default background depends on the imaging device.

Select a horizontal grid size other than the standard sizes for Ps1, by using **Pn3**. Any Pn3 value other than 0 overrides the Ps1 value. The Pn3 value is in decipoints or pixels, depending on the destination device. The maximum horizontal grid size is 99 current units—pixels or decipoints.

The Pn3 value and the pixel aspect ratio together define the grid size (including the vertical grid size). Vertical grid size equals the horizontal grid size times the pixel aspect ratio. This ratio defines the pixel as a ratio of the vertical side of a rectangle and the horizontal side. A square pixel has an aspect ratio of 1 to 1 (or 100 to 100). A pixel twice as high and wide has an aspect ratio of 2 to 1 (or 200 to 100).

4.5 The Picture Definition

Sixel data and sixel control codes, including the aspect ratio, form the *picture definition*. Sixel data includes the encoded graphic image raster. Sixel control codes tell how to interpret the raster or pixels defining the image.

4.5.1 Sixel Data

Codes in the range 3/15 through 7/14 (refer to Figure A–1, ASCII Character Set) interpret as sixel data. The six pixels to image derive by subtracting the offset 63(decimal) from the code and assigning each of the low-order six bits to a grid position. The six pixels are arranged vertically as follows:

Bit 0 (LSB)	Top pixel
Bit 1	
Bit 2	
Bit 3	
Bit 4	
Bit 5 (MSB)	Bottom pixel

For instance, for the character code 4/3 (ASCII C), the offset value 63(decimal) subtracts from the code value 67(decimal). The resulting value of 4 maps into the horizontal scan as follows:

	MSB					LSB
Data Bits:	5	4	3	2	1	0
4 =	0	0	0	1	0	0

Scan:

1	o (top)
2	o
3	x
4	o
5	o
6	o (bottom)

The "x" indicates that the pixel spot prints, and "o" indicates that the pixel spot does not print.

Table 4–2 shows the printable dot patterns for selected character codes in the 3/15 (63 decimal) through 7/14 (126 decimal) range. Subtract 63 from the decimal value of the code to create the dot pattern. For the rest of the printable dot patterns, refer to Appendix C.

Table 4–2: Printable Dot Patterns for Sixel Mode

Character	Decimal Value	Dot Pattern	Action
?	63	o	Advance
		o	by a sixel
		o	space
		o	
		o	
		o	
@	64	x	Print only
		o	top pixel
		o	
		o	
		o	
		o	
A	65	o	Print second
		x	from top
		o	pixel
		o	
		o	
		o	
y	89	o	Print second
		x	from top
		o	pixel and
		x	bottom three
		x	pixels
		x	

4.5.2 Control Codes

Descriptions of specific control codes (commands) and parameters, which make up the remainder of the picture definition data, follow. Table 4–3 summarizes these commands.

Table 4–3: Sixel Graphics Private Control Characters

Name	Abbreviation	Code	Function
Graphics Repeat Introducer	DECGRI	! 2/1	Begins repeat sequence Maximum value is 65,536
Raster Attributes	DECGRA	" 2/2	Set raster attributes First parameter — pixel aspect ratio numerator Second parameter — pixel aspect ratio denominator
Graphics Color Introducer	DECGCI	# 2/3	Specifies color First parameter — color number (others optional) Second parameter — color coordinate system Parameters 3–5 — specify colors
Graphics Carriage Return	DECGCR	$ 2/4	Returns active position to graphics left margin
Graphics New Line	DECGNL	- 2/13	Returns active position to graphics left margin and increments to next line
Parameter Separator		; 3/11	Separates parameters

Set Raster Attributes (DECGRA — ")

The Set Raster Attributes command defines raster attributes that affect the display of sixel data. This command must precede picture-definition information requiring an aspect ratio: sixel printable characters (sixel data) and the Graphic New Line (DECGNL) command.

Format

" *Pn1 ; Pn2 ; Pn3 ; Pn4*

Command Parameters	Description
"	Command control character
Pn1	Pixel aspect ratio numerator
;	Parameter delimiter
Pn2	Pixel aspect denominator
Pn3	Horizontal extent
Pn4	Vertical extent

Command Code and Parameters

"

The character " is the Set Raster Attributes control character (DECGRA).

Pn1 ; Pn2

Pn1 and Pn2 set the pixel aspect ratio, which defines the shape of the pixels needed to reproduce the picture without distortion. This ratio is defined by two numbers:

- A numerator (Pn1), which is the number of vertical pixels for the distance unit
- A denominator (Pn2), which is the number of horizontal pixels for the same distance unit

If a pixel were to be half as wide as tall, the pixel aspect ratio would be 2:1 or 100:50 as it is for the VT240 terminal.

The pixel aspect ratio times the horizontal grid size (the third parameter of the sixel DCS yields the vertical grid size.

Pn3 ; Pn4

Pn3 and Pn4 define the horizontal and vertical extent, respectively.

Repeat Introducer (DECGRI — !)

The Repeat Introducer code followed by a numeric value repeats the next pixel the specified number of times. A repeat count of 0 implies a repeat count of 1. The maximum value for the repeat count is 65,536. If no sixel data character follows the repeat count, the repeat count is ignored.

Format

! *Pn sixel_data_character*

Command Parameters	Description
!	Command control character
Pn	Character string representing a decimal number
sixel_data_character	Repeated Sixels dot pattern

Command Code and Parameters

!

The character ! is the Repeat Introducer control character (DECGRI).

Pn

Pn is a string of characters evaluating to a decimal number (positions 3/0 to 3/9 in the Standard 8-Bit Character Set).

sixel_data_character

The sixel_data_character is a repeated dot pattern. See Table 4–2 above.

The following examples illustrate repeat sequences:

- *! 1 0 ?* — repeats 10 graphic spaces
- *! 6 @* — repeats six patterns of top dot

Graphics Carriage Return (DECGCR — $)

The Graphics Carriage Return command moves the active position to the graphic left margin. This control code is the only code that allows rewriting of a sixels position.

Format

$

Command Parameters	Description
$	Command control character

Command Code and Parameters

$

The character $ is the Graphics Carriage Return control character (DECGCR).

Graphics Next Line (DECGNL — -)

The Graphics Next Line command moves the active position to the left margin and down one row of sixels (six actual grid units).

Format

-

Command Parameters	Description
-	Command control character

Command Code and Parameters

-

The character (-) is the Graphics Next Line control character (DECGNL).

Color Introducer (DECGCI — #)

The Color Introducer command starts a color selection sequence. Follow the "#" (pound sign) with a color number selected from the color map, or use a universal color coordinate system to select a new definition for the color number.

Format

Pc ; Pu ; Px ; Py ; Pz

Command Parameters	Description
#	Command control character
Pc	Color number parameter
Pu	Universal coordinate system selector
Px	System color coordinate
Py	System color coordinate
Pz	System color coordinate

Command Code and Parameters

#

The character # is the Color Introducer control character.

Pc

Pc selects the color number for the following Sixels data.

Pu (optional)

Pu names the universal color coordinate system as follows:

0 — Invalid
1 — HLS (hue/lightness/saturation)
2 — RGB (red/green/blue)

Px ; Py ; Pz (optional)

Px, Py, and Pz select the color coordinates in the specified system:

Parameter	HLS	RGB
Px	Hue angle, 0–360	Red, 0–100
Py	Lightness, 0–100	Green, 0–100
Pz	Saturation, 0–100	Blue, 0–100

Table 4–3 (see Section 4.5.2) summarizes sixel control codes and functions. Specific sixel control codes (commands) consist of a code in the 2/0 through 3/14 range, except parameters and parameter separators, followed by zero or more parameters. Separate parameters with a semicolon. Terminate sixel commands by using any nonparameter character, that is, not 0–9 or ;.

4.5.3 Sixel Character Coding

This section describes the action on different groups of codes within the picture definition.

Table 4–4 describes the sixel graphics mode response to selected C0 control characters. The protocol considers other codes in the range of 0/0 through 1/15 as errors and ignores them.

Table 4–4: Graphics ANSI Control Characters

Name	Abbreviation	Function
Bell	BEL	Ignored
Cancel	CAN	Causes exit from sixel graphics mode
Enquire	ENQ	Same action as in ANSI text mode
Escape	ESC	Causes exit from sixel graphics mode; Processed as the start of a new sequence
Substitute	SUB	Processed as a blank sixel — 3/15 or ?

GL Codes form two groups: the control codes and the sixel column codes.

- Control codes in the range 2/0 through 3/14 define commands and parameters.
- Codes 3/0 through 3/9 are for parameters. Consecutive digits form a single decimal numeric parameter.
- Code 3/11 is a parameter separator for commands with more than one parameter.
- Codes 3/15 to 7/14 process as sixel data.

Other codes in this group specify commands. Ignore undefined control codes.

C1 control codes (8\0 through 9\15) transfer code from sixel graphics mode to ANSI text mode for processing.

Codes 10/0 through 15/15 (*GR codes*) are errors.

4.6 Selecting Sixel Mode

An ANSI-defined device control string (DCS) envelope contains the sixel graphics protocol. Initiate this envelope by using the string introducer (DCS) control code and terminate the envelope with the string terminator (ST) control code. This is the only method to enter the sixel graphics mode.

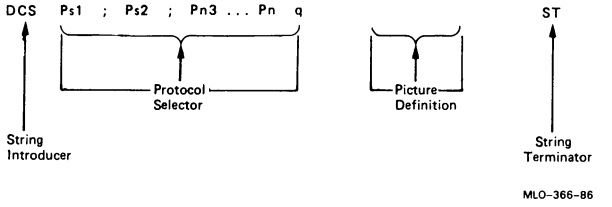
Sixel_DCS

The following components make up the DCS for the sixel graphics protocol:

- String introducer (DCS)
- Protocol selector
- Picture definition
- String terminator (ST)

Figure 4–1 shows the format of the control string.

Figure 4–1: Sixels Device Control String Envelope



Format

DCS *Ps1 ; Ps2 ; Pn3* **q** *picture_definition* **ST**

Control String Parameters	Description
DCS	String introducer character
Ps1	Macro parameter
; (semicolon)	Parameter delimiter
Ps2	Background select parameter
; (semicolon)	Parameter delimiter
Pn3	Horizontal grid size parameter
q	Protocol selector final character
picture_definition	Image description
ST	String terminator character

Control String Parameters

DCS

DCS, the device control string introducer character, is the ANSI C1 control (9/0) in 8-bit mode. *ESC P* (1/11, 5/0) is the sixel graphics introducer character in 7-bit mode.

Ps1, selective parameter

Ps1 is a selective parameter that sets the horizontal grid size, vertical grid size, and pixel aspect ratio according to a list of defined macro parameters. Naming a value other than 0 for the horizontal grid size parameter (Pn3) overrides the Ps1 value. See Section 4.4 for more information.

; (semicolon)

The semicolon acts as a delimiter to separate the parameters, Ps1, Ps2, and Pn3.

Ps2

Ps2, the second selective parameter, selects a background color.

Pn3

Pn3 is a numeric parameter naming the horizontal grid size. Pn3 is valid only in Level 2 sixels. Level 1 sixel devices omit this parameter. In Level 2 sixel devices, the Pn3 value and the pixel aspect ratio (Set Raster command) together define the grid size (including vertical grid size). For more information, see Section 4.4.

q

The character q is the protocol selector final character indicating that sixel data follows. Any other code indicates the remaining data is not sixel data.

picture_definition

Sixel graphics data and sixel graphics control codes, which describe the image, make up the picture definition. Sixel graphics data includes the encoded graphic image raster. Sixel graphics control codes tell how to interpret the raster or pixels defining the image.

ST

ST is the ANSI C1 control (9/12) in 8-bit mode or *ESC* \ (1/11, 5/12) in 7-bit mode. ST terminates the sixel graphics DCS, and leaves sixel graphics mode.

4.7 Sixels Output

RETOS converts ReGIS graphics to the sixel graphics protocol for display on sixel terminals and printers. The remainder of this chapter describes the RETOS sixel graphics interaction and output.

4.7.1 Producing the Output Header — The Sixel Graphics DCS

RETOS sixel graphics output begins with a device control string (DCS) introducer sequence. Section 4.6 describes the sixel DCS. The DCS introducer, along with a protocol selector, places a device, printer, or terminal in sixel graphics mode. The protocol selector consists of three parameters: a macro parameter (Ps1), a screen background parameter (Ps2), and a horizontal grid size parameter (Pn3).

RETOS defines the protocol selector as follows:

- RETOS sets the Ps1 parameter only if /LEVEL=1 is in effect; otherwise, RETOS uses the Pn3 parameter and the Set Raster Attributes command (DECGRA) to define the horizontal grid size, the vertical grid size, and the pixel aspect ratio.
- RETOS does not specify the Ps2 parameter.
- Pn3 is valid only in Level 2 Sixels. If /LEVEL=1 is in effect, RETOS omits this parameter; otherwise, RETOS sets this parameter to produce the desired horizontal grid size by using the horizontal resolution specified in the RETOS command line.

With /LEVEL=1 in effect, RETOS begins the output file with the DCS introducer sequence:

```
<ESC>       P       Ps1       q
1/11        5/0               7/1
```

Ps1 is the macro parameter, which depends on the aspect ratio (horizontal resolution divided by vertical resolution), defined by the following table:

Minimum Aspect Ratio	Macro Parameter
3.5	2
2.7	3
2.2	4
1.9	5
1.7	1
1.4	6
1.2	7
1.1	8
0	9

When /LEVEL=2 is in effect, RETOS does not use the macro parameter to set grid size and aspect ratio. Instead, RETOS uses the Horizontal Grid Size parameter and the Set Raster Attributes command to set the exact grid size and required aspect ratio.

If /GRID_UNITS is set to PIXELS, RETOS writes the following control sequence at the start of the output file:

```
<ESC>     [     7     space     I
1/11      5/11  3/7   2/0       4/9
```

This is the ANSI Set Size Unit (SSU) command to set the size unit to pixels rather than decipoints. If /GRID_UNITS is set to DECIPOINTS, RETOS does not write an SSU sequence at the start of the file.

Next, RETOS writes the DCS introducer sequence:

```
<ESC>     P     ;     ;     Pn3     q
1/11      5/0   3/11  3/11        7/1
```

Pn3 is the horizontal grid size in decipoints (/GRID_UNITS=DECIPOINTS) or pixels (/GRID_UNITS=PIXELS). Horizontal grid size is the inverse of the horizontal resolution, converted to decipoints or pixels.

RETOS now issues the Set Raster Attributes command to establish the aspect ratio:

```
"     Pn1     ;     Pn2     ;     Pn3     ;     Pn4
2/2         3/11        3/11        3/11
```

Pn1 and Pn2 are the numerator and denominator of the aspect ratio. RETOS sets Pn1 to the horizontal resolution and Pn2 to the vertical resolution. For example, the aspect ratio for the LJ250 printer, from the /RESOLUTION=(90,90) command qualifier, is 90:90 or 1:1. RETOS sets Pn3 and Pn4, the horizontal and vertical extent parameters, to the bit map size in pixels. This is 720;945 for the LJ250, using default parameters.

At this point, RETOS writes the bit map, with or without color information, by using the sixel protocol described earlier in this chapter.

At the end of the file, after the string terminator, if /GRID_UNITS=PIXELS is in effect, RETOS writes the following SSU control sequence to set the size unit back to decipoints:

<ESC>	[	2	space	I
1/11	5/11	3/2	2/0	4/9

4.7.1.1 The Default Header

If you did not specify qualifiers in the command line, RETOS produces the following header:

```
<ESC>P;;8q"1;1;720;945
```

This sets the horizontal grid size as 8 decipoints (90 dots/inch), the aspect ratio to 1:1, and the paper size to 8 x 10.5 inches, which produces low resolution output on the LJ250.

4.7.1.2 Headers Using the Command Line

If you specify /RESOLUTION=(180,180) in the command line, RETOS produces the following header:

```
<ESC>P;;4q"1;1;1440;1890
```

If you specify /DEVICE=LA50 in the command line, RETOS produces the following header:

```
<ESC>P5q
```

If you specify /DEVICE=LN03 in the command line, RETOS produces the following header:

```
<ESC>[7 I<ESC>P;;3q"1;1;800;1000
```

4.7.2 The Sixels Picture

RETOS scales the sixels picture so that the area defined by the ReGIS screen addressing command S(A) fills the output area as completely as possible without changing the aspect ratio. (Aspect ratio defines the shape of a pixel as the ratio of the vertical side of the rectangle and the horizontal side. A square pixel has an aspect ratio of 1:1.) This scaling insures that circles drawn in ReGIS remain circular when converted to sixels.

The origin of the converted picture is always the top left corner of the bit map, except in landscape mode. If necessary, RETOS adds space to the right and bottom of the picture. If you specify the /MARGIN qualifier, RETOS also adds blank space to the top and left of the picture.

Appendix A

Fonts and Character Sets

A.1 Supported Fonts

RETOS provides fonts in nine sizes for each character set. In pixels *width* × *height*, these sizes include the following:

- 7 × 17
- 8 × 10
- 8 × 20
- 8 × 38
- 12 × 14
- 12 × 29
- 14 × 35
- 15 × 38
- 16 × 40

RETOS uses these fonts for:

- LJ250 low resolution portrait
- VT240 portrait
- LJ250 low resolution landscape
- LA50 landscape
- LA50 portrait
- LA75 portrait

- LJ250 high resolution portrait
- LA75 landscape
- LJ250 high resolution landscape

If a character is not available in the requested size, RETOS finds the font that is closest in size and scales the character.

A.2 Supported Character Sets

RETOS supports the following character sets:

- ASCII
- DEC Multinational
- ISO Latin–1
- DEC Special Graphics (VT100 line drawing)
- DEC Technical
- National Replacement Character Sets:

  British
  French
  DEC French-Canadian
  DEC Norwegian/Danish
  DEC Finnish
  German
  DEC Dutch
  ISO Italian
  DEC Swiss
  DEC Swedish
  ISO Spanish
  DEC Portuguese

NOTE

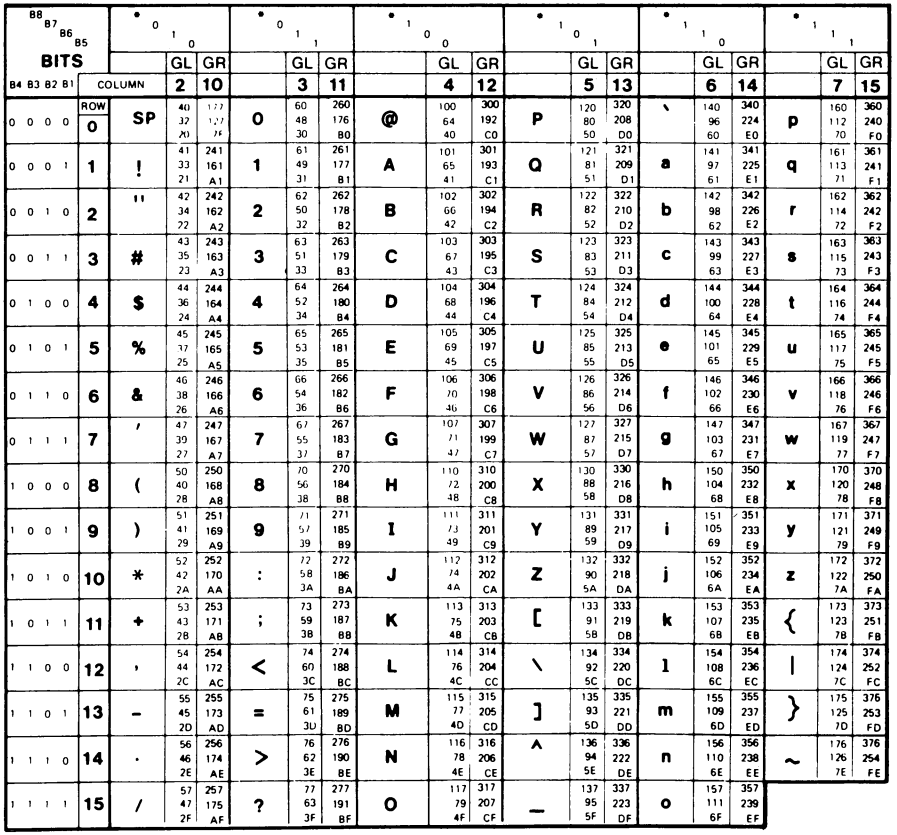
DEC Multinational is the User Preference Character Set (DEC Supplemental in GR).

The following pages show each character set supported by RETOS. ISO 646 is the basis for ISO Italian, ISO Spanish, and ISO Latin–1. Character sets with a DEC prefix indicate DIGITAL private character sets. Character sets with no prefix are country standards. Table A–1 lists the source standard for country standard character sets.

Table A–1: Character Set Source Standards

Name of Set	Source Standard
British	BS 4370
ASCII	ANSI X3.4–1986
French	AFNOR NF Z 62–010 (1973)
German	DIN 66 003
Norwegian/Danish	NS 4551–1, DS 2089

Figure A–1: ASCII Character Set (DEC Multinational Set — Left Half)



BITS B8 B7 B6 B5		0 0		0 1		1 0		1 1	
		GL	GR	GL	GR	GL	GR	GL	GR
B4 B3 B2 B1	COLUMN	2	10	3	11	4	12	5	13	6	14	7	15
0 0 0 0	0	SP 40 32 20	177 127 7F	0 60 48 30	260 176 B0	@ 100 64 40	300 192 C0	P 120 80 50	320 208 D0	` 140 96 60	340 224 E0	p 160 112 70	360 240 F0
0 0 0 1	1	! 41 33 21	241 161 A1	1 61 49 31	261 177 B1	A 101 65 41	301 193 C1	Q 121 81 51	321 209 D1	a 141 97 61	341 225 E1	q 161 113 71	361 241 F1
0 0 1 0	2	" 42 34 22	242 162 A2	2 62 50 32	262 178 B2	B 102 66 42	302 194 C2	R 122 82 52	322 210 D2	b 142 98 62	342 226 E2	r 162 114 72	362 242 F2
0 0 1 1	3	# 43 35 23	243 163 A3	3 63 51 33	263 179 B3	C 103 67 43	303 195 C3	S 123 83 53	323 211 D3	c 143 99 63	343 227 E3	s 163 115 73	363 243 F3
0 1 0 0	4	$ 44 36 24	244 164 A4	4 64 52 34	264 180 B4	D 104 68 44	304 196 C4	T 124 84 54	324 212 D4	d 144 100 64	344 228 E4	t 164 116 74	364 244 F4
0 1 0 1	5	% 45 37 25	245 165 A5	5 65 53 35	265 181 B5	E 105 69 45	305 197 C5	U 125 85 55	325 213 D5	e 145 101 65	345 229 E5	u 165 117 75	365 245 F5
0 1 1 0	6	& 46 38 26	246 166 A6	6 66 54 36	266 182 B6	F 106 70 46	306 198 C6	V 126 86 56	326 214 D6	f 146 102 66	346 230 E6	v 166 118 76	366 246 F6
0 1 1 1	7	' 47 39 27	247 167 A7	7 67 55 37	267 183 B7	G 107 71 47	307 199 C7	W 127 87 57	327 215 D7	g 147 103 67	347 231 E7	w 167 119 77	367 247 F7
1 0 0 0	8	(50 40 28	250 168 A8	8 70 56 38	270 184 B8	H 110 72 48	310 200 C8	X 130 88 58	330 216 D8	h 150 104 68	350 232 E8	x 170 120 78	370 248 F8
1 0 0 1	9	) 51 41 29	251 169 A9	9 71 57 39	271 185 B9	I 111 73 49	311 201 C9	Y 131 89 59	331 217 D9	i 151 105 69	351 233 E9	y 171 121 79	371 249 F9
1 0 1 0	10	* 52 42 2A	252 170 AA	: 72 58 3A	272 186 BA	J 112 74 4A	312 202 CA	Z 132 90 5A	332 218 DA	j 152 106 6A	352 234 EA	z 172 122 7A	372 250 FA
1 0 1 1	11	+ 53 43 2B	253 171 AB	; 73 59 3B	273 187 BB	K 113 75 4B	313 203 CB	[133 91 5B	333 219 DB	k 153 107 6B	353 235 EB	{ 173 123 7B	373 251 FB
1 1 0 0	12	, 54 44 2C	254 172 AC	< 74 60 3C	274 188 BC	L 114 76 4C	314 204 CC	\ 134 92 5C	334 220 DC	l 154 108 6C	354 236 EC	| 174 124 7C	374 252 FC
1 1 0 1	13	- 55 45 2D	255 173 AD	= 75 61 3D	275 189 BD	M 115 77 4D	315 205 CD	] 135 93 5D	335 221 DD	m 155 109 6D	355 237 ED	} 175 125 7D	375 253 FD
1 1 1 0	14	. 56 46 2E	256 174 AE	> 76 62 3E	276 190 BE	N 116 78 4E	316 206 CE	^ 136 94 5E	336 222 DE	n 156 110 6E	356 238 EE	~ 176 126 7E	376 254 FE
1 1 1 1	15	/ 57 47 2F	257 175 AF	? 77 63 3F	277 191 BF	O 117 79 4F	317 207 CF	_ 137 95 5F	337 223 DF	o 157 111 6F	357 239 EF		

LEGEND

CHARACTER — A — CODES: 101 (OCTAL), 65 (DECIMAL), 41 (HEX); 301 (OCTAL), 193 (DECIMAL), C1 (HEX)

• NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1

ASCII

MLO–438–86

Figure A–2: DEC Supplemental Character Set (DEC Multinational Set — Right Half)

B8 B7 B6 B5 BITS		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
B4 B3 B2 B1	COLUMN	GL 2	GR 10	GL 3	GR 11	GL 4	GR 12	GL 5	GR 13	GL 6	GR 14	GL 7	GR 15
0 0 0 0	ROW 0	NBSP 40 240 32 160 20 A0	°	60 260 48 176 30 B0	À	100 300 64 192 40 C0	Ð	120 320 80 208 50 D0	à	140 340 96 224 60 E0	ð	160 360 112 240 70 F0
0 0 0 1	1	¡ 41 241 33 161 21 A1	±	61 261 49 177 31 B1	Á	101 301 65 193 41 C1	Ñ	121 321 81 209 51 D1	á	141 341 97 225 61 E1	ñ	161 361 113 241 71 F1
0 0 1 0	2	¢ 42 242 34 162 22 A2	2	62 262 50 178 32 B2	Â	102 302 66 194 42 C2	Ò	122 322 82 210 52 D2	â	142 342 98 226 62 E2	ò	162 362 114 242 72 F2
0 0 1 1	3	£ 43 243 35 163 23 A3	3	63 263 51 179 33 B3	Ã	103 303 67 195 43 C3	Ó	123 323 83 211 53 D3	ã	143 343 99 227 63 E3	ó	163 363 115 243 73 F3
0 1 0 0	4	¤ 44 244 36 164 24 A4	´	64 264 52 180 34 B4	Ä	104 304 68 196 44 C4	Ô	124 324 84 212 54 D4	ä	144 344 100 228 64 E4	ô	164 364 116 244 74 F4
0 1 0 1	5	¥ 45 245 37 165 25 A5	µ	65 265 53 181 35 B5	Å	105 305 69 197 45 C5	Õ	125 325 85 213 55 D5	å	145 345 101 229 65 E5	õ	165 365 117 245 75 F5
0 1 1 0	6	¦ 46 246 38 166 26 A6	¶	66 266 54 182 36 B6	Æ	106 306 70 198 46 C6	Ö	126 326 86 214 56 D6	æ	146 346 102 230 66 E6	ö	166 366 118 246 76 F6
0 1 1 1	7	§ 47 247 39 167 27 A7	·	67 267 55 183 37 B7	Ç	107 307 71 199 47 C7	×	127 327 87 215 57 D7	ç	147 347 103 231 67 E7	÷	167 367 119 247 77 F7
1 0 0 0	8	¨ 50 250 40 168 28 A8	¸	70 270 56 184 38 B8	È	110 310 72 200 48 C8	Ø	130 330 88 216 58 D8	è	150 350 104 232 68 E8	ø	170 370 120 248 78 F8
1 0 0 1	9	© 51 251 41 169 29 A9	1	71 271 57 185 39 B9	É	111 311 73 201 49 C9	Ù	131 331 89 217 59 D9	é	151 351 105 233 69 E9	ù	171 371 121 249 79 F9
1 0 1 0	10	ª 52 252 42 170 2A AA	º	72 272 58 186 3A BA	Ê	112 312 74 202 4A CA	Ú	132 332 90 218 5A DA	ê	152 352 106 234 6A EA	ú	172 372 122 250 7A FA
1 0 1 1	11	« 53 253 43 171 2B AB	»	73 273 59 187 3B BB	Ë	113 313 75 203 4B CB	Û	133 333 91 219 5B DB	ë	153 353 107 235 6B EB	û	173 373 123 251 7B FB
1 1 0 0	12	¬ 54 254 44 172 2C AC	¼	74 274 60 188 3C BC	Ì	114 314 76 204 4C CC	Ü	134 334 92 220 5C DC	ì	154 354 108 236 6C EC	ü	174 374 124 252 7C FC
1 1 0 1	13	SHY 55 255 45 173 2D AD	½	75 275 61 189 3D BD	Í	115 315 77 205 4D CD	Ý	135 335 93 221 5D DD	í	155 355 109 237 6D ED	ý	175 375 125 253 7D FD
1 1 1 0	14	® 56 256 46 174 2E AE	¾	76 276 62 190 3E BE	Î	116 316 78 206 4E CE	Þ	136 336 94 222 5E DE	î	156 356 110 238 6E EE	þ	176 376 126 254 7E FE
1 1 1 1	15	— 57 257 47 175 2F AF	¿	77 277 63 191 3F BF	Ï	117 317 79 207 4F CF	ß	137 337 95 223 5F DF	ï	157 357 111 239 6F EF	ÿ	177 377 127 255 7F FF

LEGEND

CODES

CHARACTER — A | 101 65 41 | 301 193 C1 — OCTAL DECIMAL HEX

MLO–449–86

• NOTE
WHEN SET IS MAPPED INTO GR, BIT B8 IS 1

DEC SUPPLEMENTAL

Figure A–3: ISO Latin–1 Character Set — Left Half

B8 B7 B6 B5 / BITS		• 0 0 0		• 0 0 1		• 0 1 0		• 0 1 1		• 1 0 0		• 1 0 1		• 1 1 0		• 1 1 1	
		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN	2	10		3	11		4	12		5	13		6	14		7	15
0 0 0 0	ROW 0	SP 40 177 32 127 20 7F		0 60 260 48 176 30 B0		@ 100 300 64 192 40 C0		P 120 320 80 208 50 D0		` 140 340 96 224 60 E0		p 160 360 112 240 70 F0	
0 0 0 1	1	! 41 241 33 161 21 A1		1 61 261 49 177 31 B1		A 101 301 65 193 41 C1		Q 121 321 81 209 51 D1		a 141 341 97 225 61 E1		q 161 361 113 241 71 F1	
0 0 1 0	2	" 42 242 34 162 22 A2		2 62 262 50 178 32 B2		B 102 302 66 194 42 C2		R 122 322 82 210 52 D2		b 142 342 98 226 62 E2		r 162 362 114 242 72 F2	
0 0 1 1	3	# 43 243 35 163 23 A3		3 63 263 51 179 33 B3		C 103 303 67 195 43 C3		S 123 323 83 211 53 D3		c 143 343 99 227 63 E3		s 163 363 115 243 73 F3	
0 1 0 0	4	$ 44 244 36 164 24 A4		4 64 264 52 180 34 B4		D 104 304 68 196 44 C4		T 124 324 84 212 54 D4		d 144 344 100 228 64 E4		t 164 364 116 244 74 F4	
0 1 0 1	5	% 45 245 37 165 25 A5		5 65 265 53 181 35 B5		E 105 305 69 197 45 C5		U 125 325 85 213 55 D5		e 145 345 101 229 65 E5		u 165 365 117 245 75 F5	
0 1 1 0	6	& 46 246 38 166 26 A6		6 66 266 54 182 36 B6		F 106 306 70 198 46 C6		V 126 326 86 214 56 D6		f 146 346 102 230 66 E6		v 166 366 118 246 76 F6	
0 1 1 1	7	' 47 247 39 167 27 A7		7 67 267 55 183 37 B7		G 107 307 71 199 47 C7		W 127 327 87 215 57 D7		g 147 347 103 231 67 E7		w 167 367 119 247 77 F7	
1 0 0 0	8	(50 250 40 168 28 A8		8 70 270 56 184 38 B8		H 110 310 72 200 48 C8		X 130 330 88 216 58 D8		h 150 350 104 232 68 E8		x 170 370 120 248 78 F8	
1 0 0 1	9	) 51 251 41 169 29 A9		9 71 271 57 185 39 B9		I 111 311 73 201 49 C9		Y 131 331 89 217 59 D9		i 151 351 105 233 69 E9		y 171 371 121 249 79 F9	
1 0 1 0	10	* 52 252 42 170 2A AA		: 72 272 58 186 3A BA		J 112 312 74 202 4A CA		Z 132 332 90 218 5A DA		j 152 352 106 234 6A EA		z 172 372 122 250 7A FA	
1 0 1 1	11	+ 53 253 43 171 2B AB		; 73 273 59 187 3B BB		K 113 313 75 203 4B CB		[133 333 91 219 5B DB		k 153 353 107 235 6B EB		{ 173 373 123 251 7B FB	
1 1 0 0	12	, 54 254 44 172 2C AC		< 74 274 60 188 3C BC		L 114 314 76 204 4C CC		\ 134 334 92 220 5C DC		l 154 354 108 236 6C EC		| 174 374 124 252 7C FC	
1 1 0 1	13	- 55 255 45 173 2D AD		= 75 275 61 189 3D BD		M 115 315 77 205 4D CD		] 135 335 93 221 5D DD		m 155 355 109 237 6D ED		} 175 375 125 253 7D FD	
1 1 1 0	14	. 56 256 46 174 2E AE		> 76 276 62 190 3E BE		N 116 316 78 206 4E CE		^ 136 336 94 222 5E DE		n 156 356 110 238 6E EE		~ 176 376 126 254 7E FE	
1 1 1 1	15	/ 57 257 47 175 2F AF		? 77 277 63 191 3F BF		O 117 317 79 207 4F CF		_ 137 337 95 223 5F DF		o 157 357 111 239 6F EF			

LEGEND

CODES

CHARACTER — A | 101 301 OCTAL | 65 193 DECIMAL | 41 C1 HEX

• NOTE:
WHEN SET IS MAPPED INTO GR, BIT B8 IS 1

ASCII

MLO–438–86

Figure A–4: ISO Latin–1 Character Set — Right Half

BITS																			
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
	ROW		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
0 0 0 0	0	NBSP	40 32 20	240 160 A0	•	60 48 30	260 176 B0	À	100 64 40	300 192 C0	Ð	120 80 50	320 208 D0	à	140 96 60	340 224 E0	ð	160 112 70	360 240 F0
0 0 0 1	1	¡	41 33 21	241 161 A1	±	61 49 31	261 177 B1	Á	101 65 41	301 193 C1	Ñ	121 81 51	321 209 D1	á	141 97 61	341 225 E1	ñ	161 113 71	361 241 F1
0 0 1 0	2	¢	42 34 22	242 162 A2	2	62 50 32	262 178 B2	Â	102 66 42	302 194 C2	Ò	122 82 52	322 210 D2	â	142 98 62	342 226 E2	ò	162 114 72	362 242 F2
0 0 1 1	3	£	43 35 23	243 163 A3	3	63 51 33	263 179 B3	Ã	103 67 43	303 195 C3	Ó	123 83 53	323 211 D3	ã	143 99 63	343 227 E3	ó	163 115 73	363 243 F3
0 1 0 0	4	¤	44 36 24	244 164 A4	´	64 52 34	264 180 B4	Ä	104 68 44	304 196 C4	Ô	124 84 54	324 212 D4	ä	144 100 64	344 228 E4	ô	164 116 74	364 244 F4
0 1 0 1	5	¥	45 37 25	245 165 A5	µ	65 53 35	265 181 B5	Å	105 69 45	305 197 C5	Õ	125 85 55	325 213 D5	å	145 101 65	345 229 E5	õ	165 117 75	365 245 F5
0 1 1 0	6	¦	46 38 26	246 166 A6	¶	66 54 36	266 182 B6	Æ	106 70 46	306 198 C6	Ö	126 86 56	326 214 D6	æ	146 102 66	346 230 E6	ö	166 118 76	366 246 F6
0 1 1 1	7	§	47 39 27	247 167 A7	·	67 55 37	267 183 B7	Ç	107 71 47	307 199 C7	×	127 87 57	327 215 D7	ç	147 103 67	347 231 E7	÷	167 119 77	367 247 F7
1 0 0 0	8	¨	50 40 28	250 168 A8	¸	70 56 38	270 184 B8	È	110 72 48	310 200 C8	Ø	130 88 58	330 216 D8	è	150 104 68	350 232 E8	ø	170 120 78	370 248 F8
1 0 0 1	9	©	51 41 29	251 169 A9	1	71 57 39	271 185 B9	É	111 73 49	311 201 C9	Ù	131 89 59	331 217 D9	é	151 105 69	351 233 E9	ù	171 121 79	371 249 F9
1 0 1 0	10	ª	52 42 2A	252 170 AA	º	72 58 3A	272 186 BA	Ê	112 74 4A	312 202 CA	Ú	132 90 5A	332 218 DA	ê	152 106 6A	352 234 EA	ú	172 122 7A	372 250 FA
1 0 1 1	11	«	53 43 2B	253 171 AB	»	73 59 3B	273 187 BB	Ë	113 75 4B	313 203 CB	Û	133 91 5B	333 219 DB	ë	153 107 6B	353 235 EB	û	173 123 7B	373 251 FB
1 1 0 0	12	¬	54 44 2C	254 172 AC	¼	74 60 3C	274 188 BC	Ì	114 76 4C	314 204 CC	Ü	134 92 5C	334 220 DC	ì	154 108 6C	354 236 EC	ü	174 124 7C	374 252 FC
1 1 0 1	13		55 45 2D	255 173 AD	½	75 61 3D	275 189 BD	Í	115 77 4D	315 205 CD	Ý	135 93 5D	335 221 DD	í	155 109 6D	355 237 ED	ý	175 125 7D	375 253 FD
1 1 1 0	14	®	56 46 2E	256 174 AE	¾	76 62 3E	276 190 BE	Î	116 78 4E	316 206 CE	Þ	136 94 5E	336 222 DE	î	156 110 6E	356 238 EE	þ	176 126 7E	376 254 FE
1 1 1 1	15	¯	57 47 2F	257 175 AF	¿	77 63 3F	277 191 BF	Ï	117 79 4F	317 207 CF	ß	137 95 5F	337 223 DF	ï	157 111 6F	357 239 EF	ÿ	177 127 7F	377 255 FF

LEGEND

CHARACTER — A

CODES: 101 / 65 / 41 — OCTAL / DECIMAL / HEX

• NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1

ISO LATIN-1 SUPPLEMENTAL

MLO–1053–87

Figure A–5: DEC Special Graphics Character Set

B8 B7 B6 B5 / BITS		0 1 0		0 1 1			1 0 0			1 0 1			1 1 0			1 1 1		
		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN	2	10		3	11		4	12		5	13		6	14		7	15
0 0 0 0	0			0	60 48 30	260 176 B0	@	100 64 40	300 192 C0	P	120 80 50	320 208 D0		140 96 60	340 224 E0	SCAN 3	160 112 70	360 240 F0
0 0 0 1	1	!	41 33 21	241 161 A1	1	61 49 31	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1		141 97 61	341 225 E1	SCAN 5	161 113 71	361 241 F1
0 0 1 0	2	"	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2		142 98 62	342 226 E2	SCAN 7	162 114 72	362 242 F2
0 0 1 1	3	#	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3		143 99 63	343 227 E3	SCAN 9	163 115 73	363 243 F3
0 1 0 0	4	$	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4		144 100 64	344 228 E4	┼	164 116 74	364 244 F4
0 1 0 1	5	%	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5		145 101 65	345 229 E5	┤	165 117 75	365 245 F5
0 1 1 0	6	&	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	°	146 102 66	346 230 E6	┴	166 118 76	366 246 F6
0 1 1 1	7	'	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	±	147 103 67	347 231 E7	┬	167 119 77	367 247 F7
1 0 0 0	8	(	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	N L	150 104 68	350 232 E8	┤	170 120 78	370 248 F8
1 0 0 1	9	)	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	¶	151 105 69	351 233 E9	≤	171 121 79	371 249 F9
1 0 1 0	10	*	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	J	152 106 6A	352 234 EA	≥	172 122 7A	372 250 FA
1 0 1 1	11	+	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	[	133 91 5B	333 219 DB	┐	153 107 6B	353 235 EB	π	173 123 7B	373 251 FB
1 1 0 0	12	,	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	\	134 92 5C	334 220 DC	┌	154 108 6C	354 236 EC	≠	174 124 7C	374 252 FC
1 1 0 1	13	-	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	]	135 93 5D	335 221 DD	└	155 109 6D	355 237 ED	£	175 125 7D	375 253 FD
1 1 1 0	14	.	56 46 2E	256 174 AE	>	76 62 3E	276 190 BE	N	116 78 4E	316 206 CE	^	136 94 5E	336 222 DE	┼	156 110 6E	356 238 EE	·	176 126 7E	376 254 FE
1 1 1 1	15	/	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	(BLANK)	137 95 5F	337 223 DF	SCAN 1	157 111 6F	357 239 EF			

LEGEND

CHARACTER: A — CODES: 101 (OCTAL), 65 (DECIMAL), 41 (HEX)

• NOTE
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1

DEC SPECIAL GRAPHICS

MLO–1057–87

Figure A–6: DEC Technical Character Set

BITS			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
0 0 0 0	0				⊦	60 48 30	260 176 B0	•⋰•	100 64 40	300 192 C0	Π	120 80 50	320 208 D0	⌐	140 96 60	340 224 E0	π	160 112 70	360 240 F0
0 0 0 1	1	⌟	41 33 21	241 161 A1	⌜	61 49 31	261 177 B1	∝	101 65 41	301 193 C1	Ψ	121 81 51	321 209 D1	α	141 97 61	341 225 E1	ψ	161 113 71	361 241 F1
0 0 1 0	2	⌠	42 34 22	242 162 A2	∠	62 50 32	262 178 B2	∞	102 66 42	302 194 C2		122 82 52	322 210 D2	β	142 98 62	342 226 E2	ρ	162 114 72	362 242 F2
0 0 1 1	3	⎯	43 35 23	243 163 A3	\	63 51 33	263 179 B3	÷	103 67 43	303 195 C3	Σ	123 83 53	323 211 D3	χ	143 99 63	343 227 E3	σ	163 115 73	363 243 F3
0 1 0 0	4	⌡	44 36 24	244 164 A4	/	64 52 34	264 180 B4	Δ	104 68 44	304 196 C4		124 84 54	324 212 D4	δ	144 100 64	344 228 E4	τ	164 116 74	364 244 F4
0 1 0 1	5	⌡	45 37 25	245 165 A5	⌝	65 53 35	265 181 B5	∇	105 69 45	305 197 C5		125 85 55	325 213 D5	ε	145 101 65	345 229 E5		165 117 75	365 245 F5
0 1 1 0	6	⎮	46 38 26	246 166 A6	⌟	66 54 36	266 182 B6	Φ	106 70 46	306 198 C6	√	126 86 56	326 214 D6	φ	146 102 66	346 230 E6	ƒ	166 118 76	366 246 F6
0 1 1 1	7	⌠	47 39 27	247 167 A7	⟩	67 55 37	267 183 B7	Γ	107 71 47	307 199 C7	Ω	127 87 57	327 215 D7	γ	147 103 67	347 231 E7	ω	167 119 77	367 247 F7
1 0 0 0	8	⌊	50 40 28	250 168 A8		70 56 38	270 184 B8	~	110 72 48	310 200 C8	Ξ	130 88 58	330 216 D8	η	150 104 68	350 232 E8	ξ	170 120 78	370 248 F8
1 0 0 1	9	⌝	51 41 29	251 169 A9		71 57 39	271 185 B9	≃	111 73 49	311 201 C9	Υ	131 89 59	331 217 D9	ι	151 105 69	351 233 E9	υ	171 121 79	371 249 F9
1 0 1 0	10	⌡	52 42 2A	252 170 AA		72 58 3A	272 186 BA	Θ	112 74 4A	312 202 CA	⊂	132 90 5A	332 218 DA	θ	152 106 6A	352 234 EA	ζ	172 122 7A	372 250 FA
1 0 1 1	11	⌠	53 43 2B	253 171 AB		73 59 3B	273 187 BB	×	113 75 4B	313 203 CB	⊃	133 91 5B	333 219 DB	κ	153 107 6B	353 235 EB	←	173 123 7B	373 251 FB
1 1 0 0	12	⌈	54 44 2C	254 172 AC	≤	74 60 3C	274 188 BC	Λ	114 76 4C	314 204 CC	∩	134 92 5C	334 220 DC	λ	154 108 6C	354 236 EC	↑	174 124 7C	374 252 FC
1 1 0 1	13	⌉	55 45 2D	255 173 AD	≠	75 61 3D	275 189 BD	⇔	115 77 4D	315 205 CD	∪	135 93 5D	335 221 DD		155 109 6D	355 237 ED	→	175 125 7D	375 253 FD
1 1 1 0	14	⌋	56 46 2E	256 174 AE	≥	76 62 3E	276 190 BE	⇒	116 78 4E	316 206 CE	∧	136 94 5E	336 222 DE	ν	156 110 6E	356 238 EE	↓	176 126 7E	376 254 FE
1 1 1 1	15	⌉	57 47 2F	257 175 AF	∫	77 63 3F	277 191 BF	≡	117 79 4F	317 207 CF	∨	137 95 5F	337 223 DF	∂	157 111 6F	357 239 EF			

LEGEND

CHARACTER: α — CODES: 101 (OCTAL), 301 (DECIMAL), 65 (HEX), 193 (HEX), 41 (HEX), C1 (HEX)

• NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1

DEC TECHNICAL

MLO–452–86

Figure A–7: British Character Set

B8 B7 B6 B5 / BITS		• 0 1 0 GL	• 0 1 0 GR	• 0 1 1 GL	• 0 1 1 GR	• 1 0 0 GL	• 1 0 0 GR	• 1 0 1 GL	• 1 0 1 GR	• 1 1 0 GL	• 1 1 0 GR	• 1 1 1 GL	• 1 1 1 GR
B4 B3 B2 B1	COLUMN	2	10	3	11	4	12	5	13	6	14	7	15
0 0 0 0	ROW 0			0 60 48 30	260 176 B0	@ 100 64 40	300 192 C0	P 120 80 50	320 208 D0	` 140 96 60	340 224 E0	p 160 112 70	360 240 F0
0 0 0 1	1	! 41 33 21	241 161 A1	1 61 49 31	261 177 B1	A 101 65 41	301 193 C1	Q 121 81 51	321 209 D1	a 141 97 61	341 225 E1	q 161 113 71	361 241 F1
0 0 1 0	2	" 42 34 22	242 162 A2	2 62 50 32	262 178 B2	B 102 66 42	302 194 C2	R 122 82 52	322 210 D2	b 142 98 62	342 226 E2	r 162 114 72	362 242 F2
0 0 1 1	3	£ 43 35 23	243 163 A3	3 63 51 33	263 179 B3	C 103 67 43	303 195 C3	S 123 83 53	323 211 D3	c 143 99 63	343 227 E3	s 163 115 73	363 243 F3
0 1 0 0	4	$ 44 36 24	244 164 A4	4 64 52 34	264 180 B4	D 104 68 44	304 196 C4	T 124 84 54	324 212 D4	d 144 100 64	344 228 E4	t 164 116 74	364 244 F4
0 1 0 1	5	% 45 37 25	245 165 A5	5 65 53 35	265 181 B5	E 105 69 45	305 197 C5	U 125 85 55	325 213 D5	e 145 101 65	345 229 E5	u 165 117 75	365 245 F5
0 1 1 0	6	& 46 38 26	246 166 A6	6 66 54 36	266 182 B6	F 106 70 46	306 198 C6	V 126 86 56	326 214 D6	f 146 102 66	346 230 E6	v 166 118 76	366 246 F6
0 1 1 1	7	' 47 39 27	247 167 A7	7 67 55 37	267 183 B7	G 107 71 47	307 199 C7	W 127 87 57	327 215 D7	g 147 103 67	347 231 E7	w 167 119 77	367 247 F7
1 0 0 0	8	(50 40 28	250 168 A8	8 70 56 38	270 184 B8	H 110 72 48	310 200 C8	X 130 88 58	330 216 D8	h 150 104 68	350 232 E8	x 170 120 78	370 248 F8
1 0 0 1	9	) 51 41 29	251 169 A9	9 71 57 39	271 185 B9	I 111 73 49	311 201 C9	Y 131 89 59	331 217 D9	i 151 105 69	351 233 E9	y 171 121 79	371 249 F9
1 0 1 0	10	* 52 42 2A	252 170 AA	: 72 58 3A	272 186 BA	J 112 74 4A	312 202 CA	Z 132 90 5A	332 218 DA	j 152 106 6A	352 234 EA	z 172 122 7A	372 250 FA
1 0 1 1	11	+ 53 43 2B	253 171 AB	; 73 59 3B	273 187 BB	K 113 75 4B	313 203 CB	[133 91 5B	333 219 DB	k 153 107 6B	353 235 EB	{ 173 123 7B	373 251 FB
1 1 0 0	12	, 54 44 2C	254 172 AC	< 74 60 3C	274 188 BC	L 114 76 4C	314 204 CC	\ 134 92 5C	334 220 DC	l 154 108 6C	354 236 EC	| 174 124 7C	374 252 FC
1 1 0 1	13	- 55 45 2D	255 173 AD	= 75 61 3D	275 189 BD	M 115 77 4D	315 205 CD	] 135 93 5D	335 221 DD	m 155 109 6D	355 237 ED	} 175 125 7D	375 253 FD
1 1 1 0	14	. 56 46 2E	256 174 AE	> 76 62 3E	276 190 BE	N 116 78 4E	316 206 CE	^ 136 94 5E	336 222 DE	n 156 110 6E	356 238 EE	~ 176 126 7E	376 254 FE
1 1 1 1	15	/ 57 47 2F	257 175 AF	? 77 63 3F	277 191 BF	O 117 79 4F	317 207 CF	_ 137 95 5F	337 223 DF	o 157 111 6F	357 239 EF		

LEGEND

CHARACTER: A — CODES: 101 (OCTAL), 65 (DECIMAL), 41 (HEX); 301 (OCTAL), 193 (DECIMAL), C1 (HEX)

• NOTE
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1

BRITISH

MLO-1041-87

Figure A–8: French Character Set

B8 B7 B6 B5 / BITS			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
0 0 0 0	ROW 0				0	60 48 30	260 176 B0	à	100 64 40	300 192 C0	P	120 80 50	320 208 D0	`	140 96 60	340 224 E0	p	160 112 70	360 240 F0
0 0 0 1	1	!	41 33 21	241 161 A1	1	61 49 31	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1	a	141 97 61	341 225 E1	q	161 113 71	361 241 F1
0 0 1 0	2	"	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2	b	142 98 62	342 226 E2	r	162 114 72	362 242 F2
0 0 1 1	3	£	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3	c	143 99 63	343 227 E3	s	163 115 73	363 243 F3
0 1 0 0	4	$	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4	d	144 100 64	344 228 E4	t	164 116 74	364 244 F4
0 1 0 1	5	%	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5	e	145 101 65	345 229 E5	u	165 117 75	365 245 F5
0 1 1 0	6	&	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	f	146 102 66	346 230 E6	v	166 118 76	366 246 F6
0 1 1 1	7	'	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	g	147 103 67	347 231 E7	w	167 119 77	367 247 F7
1 0 0 0	8	(	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	h	150 104 68	350 232 E8	x	170 120 78	370 248 F8
1 0 0 1	9	)	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	i	151 105 69	351 233 E9	y	171 121 79	371 249 F9
1 0 1 0	10	*	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	j	152 106 6A	352 234 EA	z	172 122 7A	372 250 FA
1 0 1 1	11	+	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	°	133 91 5B	333 219 DB	k	153 107 6B	353 235 EB	é	173 123 7B	373 251 FB
1 1 0 0	12	,	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	ç	134 92 5C	334 220 DC	l	154 108 6C	354 236 EC	ù	174 124 7C	374 252 FC
1 1 0 1	13	-	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	§	135 93 5D	335 221 DD	m	155 109 6D	355 237 ED	è	175 125 7D	375 253 FD
1 1 1 0	14	.	56 46 2E	256 174 AE	>	76 62 3E	276 190 BE	N	116 78 4E	316 206 CE	^	136 94 5E	336 222 DE	n	156 110 6E	356 238 EE	¨	176 126 7E	376 254 FE
1 1 1 1	15	/	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	_	137 95 5F	337 223 DF	o	157 111 6F	357 239 EF			

LEGEND

CODES

CHARACTER — A — 101 (OCTAL) / 65 (DECIMAL) / 41 (HEX)

* NOTE
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1

FRENCH CHARACTER SET

MLO–1044–87

Figure A–9: DEC French-Canadian Character Set

BITS			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
0 0 0 0	0				0	60 48 30	260 176 B0	à	100 64 40	300 192 C0	P	120 80 50	320 208 D0	ô	140 96 60	340 224 E0	p	160 112 70	360 240 F0
0 0 0 1	1	!	41 33 21	241 161 A1	1	61 49 31	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1	a	141 97 61	341 225 E1	q	161 113 71	361 241 F1
0 0 1 0	2	"	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2	b	142 98 62	342 226 E2	r	162 114 72	362 242 F2
0 0 1 1	3	#	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3	c	143 99 63	343 227 E3	s	163 115 73	363 243 F3
0 1 0 0	4	$	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4	d	144 100 64	344 228 E4	t	164 116 74	364 244 F4
0 1 0 1	5	%	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5	e	145 101 65	345 229 E5	u	165 117 75	365 245 F5
0 1 1 0	6	&	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	f	146 102 66	346 230 E6	v	166 118 76	366 246 F6
0 1 1 1	7	'	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	g	147 103 67	347 231 E7	w	167 119 77	367 247 F7
1 0 0 0	8	(	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	h	150 104 68	350 232 E8	x	170 120 78	370 248 F8
1 0 0 1	9	)	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	i	151 105 69	351 233 E9	y	171 121 79	371 249 F9
1 0 1 0	10	*	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	j	152 106 6A	352 234 EA	z	172 122 7A	372 250 FA
1 0 1 1	11	+	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	â	133 91 5B	333 219 DB	k	153 107 6B	353 235 EB	é	173 123 7B	373 251 FB
1 1 0 0	12	,	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	ç	134 92 5C	334 220 DC	l	154 108 6C	354 236 EC	ù	174 124 7C	374 252 FC
1 1 0 1	13	-	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	ê	135 93 5D	335 221 DD	m	155 109 6D	355 237 ED	è	175 125 7D	375 253 FD
1 1 1 0	14	.	56 46 2E	256 174 AE	>	76 62 3E	276 190 BE	N	116 78 4E	316 206 CE	î	136 94 5E	336 222 DE	n	156 110 6E	356 238 EE	û	176 126 7E	376 254 FE
1 1 1 1	15	/	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	_	137 95 5F	337 223 DF	o	157 111 6F	357 239 EF			

LEGEND

CODES

CHARACTER — A | 101 (OCTAL) | 65 (DECIMAL) | 41 (HEX)

* NOTE:
WHEN SET IS MAPPED INTO GR, BIT B8 IS 1.

DEC FRENCH-CANADIAN CHARACTER SET MLO–1045–87

Figure A–10: DEC Norwegian/Danish Character Set

B8 B7 B6 B5 / BITS		* 0 0 0		* 0 0 1		* 0 1 0		* 0 1 1		* 1 0 0		* 1 0 1		* 1 1 0		* 1 1 1	
		GL	GR	GL	GR	GL	GR	GL	GR	GL	GR	GL	GR	GL	GR	GL	GR
B4 B3 B2 B1	COLUMN	2	10	3	11	4	12	5	13	6	14	7	15
0 0 0 0	ROW 0			0	60 30	Ä	100 64 40	P	120 80 50	ä	140 96 60	p	160 112 70
0 0 0 1	1	!	41 33 21 A1	1	61 49 31 B1	A	101 65 41 C1	Q	121 81 51 D1	a	141 97 61 E1	q	161 113 71 F1
0 0 1 0	2	"	42 34 22 A2	2	62 50 32 B2	B	102 66 42 C2	R	122 82 52 D2	b	142 98 62 E2	r	162 114 72 F2
0 0 1 1	3	#	43 35 23 A3	3	63 51 33 B3	C	103 67 43 C3	S	123 83 53 D3	c	143 99 63 E3	s	163 115 73 F3
0 1 0 0	4	$	44 36 24 A4	4	64 52 34 B4	D	104 68 44 C4	T	124 84 54 D4	d	144 100 64 E4	t	164 116 74 F4
0 1 0 1	5	%	45 37 25 A5	5	65 53 35 B5	E	105 69 45 C5	U	125 85 55 D5	e	145 101 65 E5	u	165 117 75 F5
0 1 1 0	6	&	46 38 26 A6	6	66 54 36 B6	F	106 70 46 C6	V	126 86 56 D6	f	146 102 66 E6	v	166 118 76 F6
0 1 1 1	7	'	47 39 27 A7	7	67 55 37 B7	G	107 71 47 C7	W	127 87 57 D7	g	147 103 67 E7	w	167 119 77 F7
1 0 0 0	8	(	50 40 28 A8	8	70 56 38 B8	H	110 72 48 C8	X	130 88 58 D8	h	150 104 68 E8	x	170 120 78 F8
1 0 0 1	9	)	51 41 29 A9	9	71 57 39 B9	I	111 73 49 C9	Y	131 89 59 D9	i	151 105 69 E9	y	171 121 79 F9
1 0 1 0	10	*	52 42 2A AA	:	72 58 3A BA	J	112 74 4A CA	Z	132 90 5A DA	j	152 106 6A EA	z	172 122 7A FA
1 0 1 1	11	+	53 43 2B AB	;	73 59 3B BB	K	113 75 4B CB	Æ	133 91 5B DB	k	153 107 6B EB	æ	173 123 7B FB
1 1 0 0	12	,	54 44 2C AC	<	74 60 3C BC	L	114 76 4C CC	Ø	134 92 5C DC	l	154 108 6C EC	ø	174 124 7C FC
1 1 0 1	13	-	55 45 2D AD	=	75 61 3D BD	M	115 77 4D CD	Å	135 93 5D DD	m	155 109 6D ED	å	175 125 7D FD
1 1 1 0	14	.	56 46 2E AE	>	76 62 3E BE	N	116 78 4E CE	Ü	136 94 5E DE	n	156 110 6E EE	ü	176 126 7E FE
1 1 1 1	15	/	57 47 2F AF	?	77 63 3F BF	O	117 79 4F CF	_	137 95 5F DF	o	157 111 6F EF		

LEGEND

CHARACTER: A — CODES: 101 (OCTAL), 65 (DECIMAL), 41 (HEX); 301 (OCTAL), 193 (DECIMAL), C1 (HEX)

* NOTE:
WHEN SET IS MAPPED INTO GR, BIT B8 IS 1

DEC NORWEGIAN/DANISH CHARACTER SET

MLO–1049–87

Figure A–11: DEC Finnish Character Set

BITS B8 B7 B6 B5			0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
0 0 0 0	ROW 0				0	60 48 30	260 176 B0	@	100 64 40	300 192 C0	P	120 80 50	320 208 D0	é	140 96 60	340 224 E0	p	160 112 70	360 240 F0
0 0 0 1	1	!	41 33 21	241 161 A1	1	61 49 31	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1	a	141 97 61	341 225 E1	q	161 113 71	361 241 F1
0 0 1 0	2	"	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2	b	142 98 62	342 226 E2	r	162 114 72	362 242 F2
0 0 1 1	3	#	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3	c	143 99 63	343 227 E3	s	163 115 73	363 243 F3
0 1 0 0	4	$	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4	d	144 100 64	344 228 E4	t	164 116 74	364 244 F4
0 1 0 1	5	%	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5	e	145 101 65	345 229 E5	u	165 117 75	365 245 F5
0 1 1 0	6	&	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	f	146 102 66	346 230 E6	v	166 118 76	366 246 F6
0 1 1 1	7	'	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	g	147 103 67	347 231 E7	w	167 119 77	367 247 F7
1 0 0 0	8	(	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	h	150 104 68	350 232 E8	x	170 120 78	370 248 F8
1 0 0 1	9	)	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	i	151 105 69	351 233 E9	y	171 121 79	371 249 F9
1 0 1 0	10	*	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	j	152 106 6A	352 234 EA	z	172 122 7A	372 250 FA
1 0 1 1	11	+	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	Ä	133 91 5B	333 219 DB	k	153 107 6B	353 235 EB	ä	173 123 7B	373 251 FB
1 1 0 0	12	,	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	Ö	134 92 5C	334 220 DC	l	154 108 6C	354 236 EC	ö	174 124 7C	374 252 FC
1 1 0 1	13	–	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	Å	135 93 5D	335 221 DD	m	155 109 6D	355 237 ED	å	175 125 7D	375 253 FD
1 1 1 0	14	.	56 46 2E	256 174 AE	>	76 62 3E	276 190 BE	N	116 78 4E	316 206 CE	Ü	136 94 5E	336 222 DE	n	156 110 6E	356 238 EE	ü	176 126 7E	376 254 FE
1 1 1 1	15	/	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	_	137 95 5F	337 223 DF	o	157 111 6F	357 239 EF			

LEGEND

CHARACTER: A

CODES

101	OCTAL
65	DECIMAL
41	HEX

• NOTE:
WHEN SET IS MAPPED INTO GR, BIT B8 IS 1

DEC FINNISH CHARACTER SET

MLO–1043–87

Figure A–12: German Character Set

B8 B7 B6 B5 = BITS		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
		GL	GR	GL	GR	GL	GR	GL	GR	GL	GR	GL	GR
B4 B3 B2 B1	COLUMN	2	10	3	11	4	12	5	13	6	14	7	15
0 0 0 0	ROW 0			0 60 48 30	260 176 B0	§ 100 64 40	300 192 C0	P 120 80 50	320 208 D0	` 140 96 60	340 224 E0	p 160 112 70	360 240 F0
0 0 0 1	1	! 41 33 21	241 161 A1	1 61 49 31	261 177 B1	A 101 65 41	301 193 C1	Q 121 81 51	321 209 D1	a 141 97 61	341 225 E1	q 161 113 71	361 241 F1
0 0 1 0	2	" 42 34 22	242 162 A2	2 62 50 32	262 178 B2	B 102 66 42	302 194 C2	R 122 82 52	322 210 D2	b 142 98 62	342 226 E2	r 162 114 72	362 242 F2
0 0 1 1	3	# 43 35 23	243 163 A3	3 63 51 33	263 179 B3	C 103 67 43	303 195 C3	S 123 83 53	323 211 D3	c 143 99 63	343 227 E3	s 163 115 73	363 243 F3
0 1 0 0	4	$ 44 36 24	244 164 A4	4 64 52 34	264 180 B4	D 104 68 44	304 196 C4	T 124 84 54	324 212 D4	d 144 100 64	344 228 E4	t 164 116 74	364 244 F4
0 1 0 1	5	% 45 37 25	245 165 A5	5 65 53 35	265 181 B5	E 105 69 45	305 197 C5	U 125 85 55	325 213 D5	e 145 101 65	345 229 E5	u 165 117 75	365 245 F5
0 1 1 0	6	& 46 38 26	246 166 A6	6 66 54 36	266 182 B6	F 106 70 46	306 198 C6	V 126 86 56	326 214 D6	f 146 102 66	346 230 E6	v 166 118 76	366 246 F6
0 1 1 1	7	' 47 39 27	247 167 A7	7 67 55 37	267 183 B7	G 107 71 47	307 199 C7	W 127 87 57	327 215 D7	g 147 103 67	347 231 E7	w 167 119 77	367 247 F7
1 0 0 0	8	(50 40 28	250 168 A8	8 70 56 38	270 184 B8	H 110 72 48	310 200 C8	X 130 88 58	330 216 D8	h 150 104 68	350 232 E8	x 170 120 78	370 248 F8
1 0 0 1	9	) 51 41 29	251 169 A9	9 71 57 39	271 185 B9	I 111 73 49	311 201 C9	Y 131 89 59	331 217 D9	i 151 105 69	351 233 E9	y 171 121 79	371 249 F9
1 0 1 0	10	* 52 42 2A	252 170 AA	: 72 58 3A	272 186 BA	J 112 74 4A	312 202 CA	Z 132 90 5A	332 218 DA	j 152 106 6A	352 234 EA	z 172 122 7A	372 250 FA
1 0 1 1	11	+ 53 43 2B	253 171 AB	; 73 59 3B	273 187 BB	K 113 75 4B	313 203 CB	Ä 133 91 5B	333 219 DB	k 153 107 6B	353 235 EB	ä 173 123 7B	373 251 FB
1 1 0 0	12	, 54 44 2C	254 172 AC	< 74 60 3C	274 188 BC	L 114 76 4C	314 204 CC	Ö 134 92 5C	334 220 DC	l 154 108 6C	354 236 EC	ö 174 124 7C	374 252 FC
1 1 0 1	13	- 55 45 2D	255 173 AD	= 75 61 3D	275 189 BD	M 115 77 4D	315 205 CD	Ü 135 93 5D	335 221 DD	m 155 109 6D	355 237 ED	ü 175 125 7D	375 253 FD
1 1 1 0	14	. 56 46 2E	256 174 AE	> 76 62 3E	276 190 BE	N 116 78 4E	316 206 CE	^ 136 94 5E	336 222 DE	n 156 110 6E	356 238 EE	ß 176 126 7E	376 254 FE
1 1 1 1	15	/ 57 47 2F	257 175 AF	? 77 63 3F	277 191 BF	O 117 79 4F	317 207 CF	_ 137 95 5F	337 223 DF	o 157 111 6F	357 239 EF		

LEGEND

CODES

CHARACTER → A | 101 65 41 (OCTAL DECIMAL HEX) | 301 193 C1

- NOTE:
  WHEN SET IS MAPPED INTO GR, BIT B8 IS 1

GERMAN CHARACTER SET

MLO–1046–87

Figure A–13: DEC Dutch Character Set

B8 B7 B6 B5 / BITS		0 0		0 1		1 0		1 1		1 1		1 1	
		GL	GR	GL	GR	GL	GR	GL	GR	GL	GR	GL	GR
B4 B3 B2 B1	COLUMN	2	10	3	11	4	12	5	13	6	14	7	15
0 0 0 0	0	SP		0		3/4		P		`		p	
0 0 0 1	1	!		1		A		Q		a		q	
0 0 1 0	2	"		2		B		R		b		r	
0 0 1 1	3	£		3		C		S		c		s	
0 1 0 0	4	$		4		D		T		d		t	
0 1 0 1	5	%		5		E		U		e		u	
0 1 1 0	6	&		6		F		V		f		v	
0 1 1 1	7	'		7		G		W		g		w	
1 0 0 0	8	(		8		H		X		h		x	
1 0 0 1	9	)		9		I		Y		i		y	
1 0 1 0	10	*		:		J		Z		j		z	
1 0 1 1	11	+		;		K		ij		k		¨	
1 1 0 0	12	,		<		L		1/2		l		f	
1 1 0 1	13	-		=		M		¦		m		1/4	
1 1 1 0	14	.		>		N		^		n		´	
1 1 1 1	15	/		?		O		_		o			

	GL	GR	
5/11	120	320	ij
	80	208	
	50	D0	
7/11	134	334	¨
	92	220	
	5C	DC	
7/12	135	335	f
	93	221	
	5D	DD	
7/14	136	336	´
	94	222	
	5E	DE	

LEGEND

CHARACTER — A

CODES: 101 / 65 / 41 OCTAL; 193 / C1 DECIMAL; HEX

* NOTE
WHEN SET IS MAPPED INTO GR BIT B8 IS 1

HIGHLIGHTS DIFFERENCES FROM ASCII

NOTE:
THE FOLLOWING TABLE INDICATES THE APPROXIMATIONS THAT ARE USED TO REPRESENT THE DUTCH CHARACTERS THAT ARE NOT AVAILABLE IN THE DECMCS SET. (THESE APPROXIMATIONS ARE TO BE COMPATIBLE WITH THE VT220 AND VT240.) THE CHARACTER POSITION IN THE CHART IS LISTED BY COLUMN/ROW.

COLUMN/ ROW	CHARACTER SET NAME (SYMBOL)	APPROXIMATION NAME (SYMBOL)
4/0	THREE QUARTERS (3/4)	SUPERSCRIPT (3)
5/11	LOWERCASE ij LIGATURE (ij)	LOWERCASE y WITH DIAERESIS (ÿ)
7/11	DIAERESIS (¨)	QUOTATION MARKS (")
7/12	FLORIN SIGN (f)	LOWERCASE f (f)
7/14	ACUTE ACCENT (´)	APOSTROPHE, SINGLE QUOTATION MARK, ASCII ACUTE ACCENT (')

DEC DUTCH MLO–1042–87

Figure A–14: ISO Italian Character Set

B8 B7 B6 B5 / BITS		• 0 0 0		• 0 0 1		• 0 1 0		• 0 1 1		• 1 0 0		• 1 0 1		• 1 1 0		• 1 1 1	
		GL	GR	GL	GR	GL	GR	GL	GR	GL	GR	GL	GR	GL	GR	GL	GR
B4 B3 B2 B1	COLUMN	2	10	3	11	4	12	5	13	6	14	7	15				
0 0 0 0	ROW 0			0	60 260 48 176 30 B0	@	100 300 64 192 40 C0	P	120 320 80 208 50 D0	`	140 340 96 224 60 E0	p	160 360 112 240 70 F0				
0 0 0 1	1	!	41 241 33 161 21 A1	1	61 261 49 177 31 B1	A	101 301 65 193 41 C1	Q	121 321 81 209 51 D1	a	141 341 97 225 61 E1	q	161 361 113 241 71 F1				
0 0 1 0	2	"	42 242 34 162 22 A2	2	62 262 50 178 32 B2	B	102 302 66 194 42 C2	R	122 322 82 210 52 D2	b	142 342 98 226 62 E2	r	162 362 114 242 72 F2				
0 0 1 1	3	£	43 243 35 163 23 A3	3	63 263 51 179 33 B3	C	103 303 67 195 43 C3	S	123 323 83 211 53 D3	c	143 343 99 227 63 E3	s	163 363 115 243 73 F3				
0 1 0 0	4	$	44 244 36 164 24 A4	4	64 264 52 180 34 B4	D	104 304 68 196 44 C4	T	124 324 84 212 54 D4	d	144 344 100 228 64 E4	t	164 364 116 244 74 F4				
0 1 0 1	5	%	45 245 37 165 25 A5	5	65 265 53 181 35 B5	E	105 305 69 197 45 C5	U	125 325 85 213 55 D5	e	145 345 101 229 65 E5	u	165 365 117 245 75 F5				
0 1 1 0	6	&	46 246 38 166 26 A6	6	66 266 54 182 36 B6	F	106 306 70 198 46 C6	V	126 326 86 214 56 D6	f	146 346 102 230 66 E6	v	166 366 118 246 76 F6				
0 1 1 1	7	'	47 247 39 167 27 A7	7	67 267 55 183 37 B7	G	107 307 71 199 47 C7	W	127 327 87 215 57 D7	g	147 347 103 231 67 E7	w	167 367 119 247 77 F7				
1 0 0 0	8	(	50 250 40 168 28 A8	8	70 270 56 184 38 B8	H	110 310 72 200 48 C8	X	130 330 88 216 58 D8	h	150 350 104 232 68 E8	x	170 370 120 248 78 F8				
1 0 0 1	9	)	51 251 41 169 29 A9	9	71 271 57 185 39 B9	I	111 311 73 201 49 C9	Y	131 331 89 217 59 D9	i	151 351 105 233 69 E9	y	171 371 121 249 79 F9				
1 0 1 0	10	*	52 252 42 170 2A AA	:	72 272 58 186 3A BA	J	112 312 74 202 4A CA	Z	132 332 90 218 5A DA	j	152 352 106 234 6A EA	z	172 372 122 250 7A FA				
1 0 1 1	11	+	53 253 43 171 2B AB	;	73 273 59 187 3B BB	K	113 313 75 203 4B CB	o	133 333 91 219 5B DB	k	153 353 107 235 6B EB	à	173 373 123 251 7B FB				
1 1 0 0	12	,	54 254 44 172 2C AC	<	74 274 60 188 3C BC	L	114 314 76 204 4C CC	ç	134 334 92 220 5C DC	l	154 354 108 236 6C EC	ò	174 374 124 252 7C FC				
1 1 0 1	13	-	55 255 45 173 2D AD	=	75 275 61 189 3D BD	M	115 315 77 205 4D CD	é	135 335 93 221 5D DD	m	155 355 109 237 6D ED	è	175 375 125 253 7D FD				
1 1 1 0	14	.	56 256 46 174 2E AE	>	76 276 62 190 3E BE	N	116 316 78 206 4E CE	^	136 336 94 222 5E DE	n	156 356 110 238 6E EE	ì	176 376 126 254 7E FE				
1 1 1 1	15	/	57 257 47 175 2F AF	?	77 277 63 191 3F BF	O	117 317 79 207 4F CF	_	137 337 95 223 5F DF	o	157 357 111 239 6F EF						

LEGEND

CODES

CHARACTER — A | 101 65 41 | 301 193 C1 — OCTAL DECIMAL HEX

• NOTE
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1

ISO ITALIAN

MLO–445–86

Figure A–15: DEC Swiss Character Set

B8 B7 B6 B5 / BITS			0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
0 0 0 0	0				0	60 48 30	260 176 B0	`	100 64 40	300 192 C0	P	120 80 50	320 208 D0	é	140 96 60	340 224 E0	p	160 112 70	360 240 F0
0 0 0 1	1	!	41 33 21	241 161 A1	1	61 49 31	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1	a	141 97 61	341 225 E1	q	161 113 71	361 241 F1
0 0 1 0	2	"	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2	b	142 98 62	342 226 E2	r	162 114 72	362 242 F2
0 0 1 1	3	ù	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3	c	143 99 63	343 227 E3	s	163 115 73	363 243 F3
0 1 0 0	4	$	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4	d	144 100 64	344 228 E4	t	164 116 74	364 244 F4
0 1 0 1	5	%	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5	e	145 101 65	345 229 E5	u	165 117 75	365 245 F5
0 1 1 0	6	&	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	f	146 102 66	346 230 E6	v	166 118 76	366 246 F6
0 1 1 1	7	'	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	g	147 103 67	347 231 E7	w	167 119 77	367 247 F7
1 0 0 0	8	(	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	h	150 104 68	350 232 E8	x	170 120 78	370 248 F8
1 0 0 1	9	)	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	i	151 105 69	351 233 E9	y	171 121 79	371 249 F9
1 0 1 0	10	*	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	j	152 106 6A	352 234 EA	z	172 122 7A	372 250 FA
1 0 1 1	11	+	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	ć	133 91 5B	333 219 DB	k	153 107 6B	353 235 EB	ä	173 123 7B	373 251 FB
1 1 0 0	12	,	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	ç	134 92 5C	334 220 DC	l	154 108 6C	354 236 EC	ö	174 124 7C	374 252 FC
1 1 0 1	13	-	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	ß	135 93 5D	335 221 DD	m	155 109 6D	355 237 ED	ü	175 125 7D	375 253 FD
1 1 1 0	14	.	56 46 2E	256 174 AE	>	76 62 3E	276 190 BE	N	116 78 4E	316 206 CE	ì	136 94 5E	336 222 DE	n	156 110 6E	356 238 EE	à	176 126 7E	376 254 FE
1 1 1 1	15	/	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	ò	137 95 5F	337 223 DF	o	157 111 6F	357 239 EF			

LEGEND

CHARACTER: A — CODES: 101 (OCTAL) / 065 (DECIMAL) / 41 (HEX); 301 (OCTAL) / 193 (DECIMAL) / C1 (HEX)

HIGHLIGHTS DIFFERENCES FROM ASCII

* NOTE: WHEN SET IS MAPPED INTO GR BIT B8 IS 1

NOTE:
AT COLUMN/ROW 5/15 LOWERCASE e WITH GRAVE ACCENT REPLACES UNDERLINE (_) WHICH IS USED IN ASCII AND ALL OTHER NRC SETS.

DEC SWISS

MLO-451-86

Figure A–16: DEC Swedish Character Set

B8 B7 B6 B5 / BITS		• 0 1 0			0 1 1			1 0 0			1 0 1			1 1 0			1 1 1		
			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
0 0 0 0	ROW 0				0	60 48 30	260 176 B0	É	100 64 40	300 192 C0	P	120 80 50	320 208 D0	é	140 96 60	340 224 E0	p	160 112 70	360 240 F0
0 0 0 1	1	! ¡	41 33 21	241 161 A1	1	61 49 31	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1	a	141 97 61	341 225 E1	q	161 113 71	361 241 F1
0 0 1 0	2	" ¨	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2	b	142 98 62	342 226 E2	r	162 114 72	362 242 F2
0 0 1 1	3	# £	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3	c	143 99 63	343 227 E3	s	163 115 73	363 243 F3
0 1 0 0	4	$ ¤	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4	d	144 100 64	344 228 E4	t	164 116 74	364 244 F4
0 1 0 1	5	% ‰	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5	e	145 101 65	345 229 E5	u	165 117 75	365 245 F5
0 1 1 0	6	& ⅋	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	f	146 102 66	346 230 E6	v	166 118 76	366 246 F6
0 1 1 1	7	' ´	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	g	147 103 67	347 231 E7	w	167 119 77	367 247 F7
1 0 0 0	8	(¿	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	h	150 104 68	350 232 E8	x	170 120 78	370 248 F8
1 0 0 1	9	) ‘	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	i	151 105 69	351 233 E9	y	171 121 79	371 249 F9
1 0 1 0	10	* ¤	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	j	152 106 6A	352 234 EA	z	172 122 7A	372 250 FA
1 0 1 1	11	+ ±	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	Ä	133 91 5B	333 219 DB	k	153 107 6B	353 235 EB	ä	173 123 7B	373 251 FB
1 1 0 0	12	, ¸	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	Ö	134 92 5C	334 220 DC	l	154 108 6C	354 236 EC	ö	174 124 7C	374 252 FC
1 1 0 1	13	- ¯	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	Å	135 93 5D	335 221 DD	m	155 109 6D	355 237 ED	å	175 125 7D	375 253 FD
1 1 1 0	14	. ·	56 46 2E	256 174 AE	>	76 62 3E	276 190 BE	N	116 78 4E	316 206 CE	Ü	136 94 5E	336 222 DE	n	156 110 6E	356 238 EE	ü	176 126 7E	376 254 FE
1 1 1 1	15	/ ¿	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	_	137 95 5F	337 223 DF	o	157 111 6F	357 239 EF			

LEGEND

CODES

CHARACTER: A — 101 (OCTAL), 65 (DECIMAL), 41 (HEX)

• NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1

DEC SWEDISH CHARACTER SET

MLO–1054–87

Figure A–17: ISO Spanish Character Set

B8 B7 B6 B5		• 0 0 0			• 0 0 1			• 0 1 0			• 0 1 1			• 1 0 0			• 1 0 1			• 1 1 0			• 1 1 1		
BITS			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		**2**	**10**		**3**	**11**		**4**	**12**		**5**	**13**		**6**	**14**		**7**	**15**
0 0 0 0	ROW 0				0	60 48 30	260 176 B0	§	100 64 40	300 192 C0	P	120 80 50	320 208 D0	`	140 96 60	340 224 E0	p	160 112 70	360 240 F0
0 0 0 1	1	!	41 33 21	241 161 A1	1	61 49 31	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1	a	141 97 61	341 225 E1	q	161 113 71	361 241 F1
0 0 1 0	2	"	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2	b	142 98 62	342 226 E2	r	162 114 72	362 242 F2
0 0 1 1	3	£	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3	c	143 99 63	343 227 E3	s	163 115 73	363 243 F3
0 1 0 0	4	$	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4	d	144 100 64	344 228 E4	t	164 116 74	364 244 F4
0 1 0 1	5	%	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5	e	145 101 65	345 229 E5	u	165 117 75	365 245 F5
0 1 1 0	6	&	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	f	146 102 66	346 230 E6	v	166 118 76	366 246 F6
0 1 1 1	7	'	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	g	147 103 67	347 231 E7	w	167 119 77	367 247 F7
1 0 0 0	8	(	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	h	150 104 68	350 232 E8	x	170 120 78	370 248 F8
1 0 0 1	9	)	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	i	151 105 69	351 233 E9	y	171 121 79	371 249 F9
1 0 1 0	10	*	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	j	152 106 6A	352 234 EA	z	172 122 7A	372 250 FA
1 0 1 1	11	+	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	¡	133 91 5B	333 219 DB	k	153 107 6B	353 235 EB	o	173 123 7B	373 251 FB
1 1 0 0	12	,	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	Ñ	134 92 5C	334 220 DC	l	154 108 6C	354 236 EC	ñ	174 124 7C	374 252 FC
1 1 0 1	13	-	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	¿	135 93 5D	335 221 DD	m	155 109 6D	355 237 ED	ç	175 125 7D	375 253 FD
1 1 1 0	14	.	56 46 2E	256 174 AE	>	76 62 3E	276 190 BE	N	116 78 4E	316 206 CE	^	136 94 5E	336 222 DE	n	156 110 6E	356 238 EE	~	176 126 7E	376 254 FE
1 1 1 1	15	/	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	_	137 95 5F	337 223 DF	o	157 111 6F	357 239 EF			

LEGEND

CHARACTER — A

CODES: 101 (OCTAL) / 65 (DECIMAL) / 41 (HEX); 301 (OCTAL) / 193 (DECIMAL) / C1 (HEX)

• NOTE:
WHEN SET IS MAPPED INTO GR, BIT B8 IS 1

ISO SPANISH CHARACTER SET

MLO–448–86

Figure A–18: DEC Portuguese Character Set

B8 B7 B6 B5		• 0 0 0			• 0 0 1			• 0 1 0			• 0 1 1			• 1 0 0			• 1 0 1			• 1 1 0			• 1 1 1		
BITS			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		**2**	**10**		**3**	**11**		**4**	**12**		**5**	**13**		**6**	**14**		**7**	**15**
	ROW					60	260		100	300		120	320		140	340		160	360
0 0 0 0	**0**				**0**	48	176	**@**	64	192	**P**	80	208	`	96	224	**p**	112	240
						30	B0		40	C0		50	D0		60	E0		70	F0
			41	241		61	261		101	301		121	321		141	341		161	361
0 0 0 1	**1**	**!**	33	161	**1**	49	177	**A**	65	193	**Q**	81	209	**a**	97	225	**q**	113	241
			21	A1		31	B1		41	C1		51	D1		61	E1		71	F1
		"	42	242		62	262		102	302		122	322		142	342		162	362
0 0 1 0	**2**		34	162	**2**	50	178	**B**	66	194	**R**	82	210	**b**	98	226	**r**	114	242
			22	A2		32	B2		42	C2		52	D2		62	E2		72	F2
			43	243		63	263		103	303		123	323		143	343		163	363
0 0 1 1	**3**	**#**	35	163	**3**	51	179	**C**	67	195	**S**	83	211	**c**	99	227	**s**	115	243
			23	A3		33	B3		43	C3		53	D3		63	E3		73	F3
			44	244		64	264		104	304		124	324		144	344		164	364
0 1 0 0	**4**	**$**	36	164	**4**	52	180	**D**	68	196	**T**	84	212	**d**	100	228	**t**	116	244
			24	A4		34	B4		44	C4		54	D4		64	E4		74	F4
			45	245		65	265		105	305		125	325		145	345		165	365
0 1 0 1	**5**	**%**	37	165	**5**	53	181	**E**	69	197	**U**	85	213	**e**	101	229	**u**	117	245
			25	A5		35	B5		45	C5		55	D5		65	E5		75	F5
			46	246		66	266		106	306		126	326		146	346		166	366
0 1 1 0	**6**	**&**	38	166	**6**	54	182	**F**	70	198	**V**	86	214	**f**	102	230	**v**	118	246
			26	A6		36	B6		46	C6		56	D6		66	E6		76	F6
		'	47	247		67	267		107	307		127	327		147	347		167	367
0 1 1 1	**7**		39	167	**7**	55	183	**G**	71	199	**W**	87	215	**g**	103	231	**w**	119	247
			27	A7		37	B7		47	C7		57	D7		67	E7		77	F7
			50	250		70	270		110	310		130	330		150	350		170	370
1 0 0 0	**8**	**(**	40	168	**8**	56	184	**H**	72	200	**X**	88	216	**h**	104	232	**x**	120	248
			28	A8		38	B8		48	C8		58	D8		68	E8		78	F8
			51	251		71	271		111	311		131	331		151	351		171	371
1 0 0 1	**9**	**)**	41	169	**9**	57	185	**I**	73	201	**Y**	89	217	**i**	105	233	**y**	121	249
			29	A9		39	B9		49	C9		59	D9		69	E9		79	F9
			52	252		72	272		112	312		132	332		152	352		172	372
1 0 1 0	**10**	*	42	170	:	58	186	**J**	74	202	**Z**	90	218	**j**	106	234	**z**	122	250
			2A	AA		3A	BA		4A	CA		5A	DA		6A	EA		7A	FA
			53	253		73	273		113	313		133	333		153	353		173	373
1 0 1 1	**11**	+	43	171	;	59	187	**K**	75	203	**Ã**	91	219	**k**	107	235	**ã**	123	251
			2B	AB		3B	BB		4B	CB		5B	DB		6B	EB		7B	FB
			54	254		74	274		114	314		134	334		154	354		174	374
1 1 0 0	**12**	,	44	172	<	60	188	**L**	76	204	**Ç**	92	220	**l**	108	236	**ç**	124	252
			2C	AC		3C	BC		4C	CC		5C	DC		6C	EC		7C	FC
			55	255		75	275		115	315		135	335		155	355		175	376
1 1 0 1	**13**	–	45	173	=	61	189	**M**	77	205	**Õ**	93	221	**m**	109	237	**õ**	125	253
			2D	AD		3D	BD		4D	CD		5D	DD		6D	ED		7D	FD
			56	256		76	276		116	316		136	336		156	356		176	376
1 1 1 0	**14**	.	46	174	>	62	190	**N**	78	206	**^**	94	222	**n**	110	238	**~**	126	254
			2E	AE		3E	BE		4E	CE		5E	DE		6E	EE		7E	FE
			57	257		77	277		117	317		137	337		157	357			
1 1 1 1	**15**	/	47	175	?	63	191	**O**	79	207	**_**	95	223	**o**	111	239			
			2F	AF		3F	BF		4F	CF		5F	DF		6F	EF			

LEGEND

CODES

CHARACTER → **A** | 101 | 301 | OCTAL
65 | 193 | DECIMAL
41 | C1 | HEX

• NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1

DEC PORTUGUESE

MLO–1059–87

Appendix B

Compatibility with Other ReGIS Devices

RETOS is compatible with ReGIS applications, with the following exceptions:

- Device dependent output

  Exact fonts for characters

- Inappropriate operators for a printer

  Reports
  Rectangle operations

RETOS supports all ReGIS commands supported by the VT240, VT241, VT330, and VT340, with the following exceptions:

- R — Reports
- S( ) — Scrolling
- S(C) — Graphic cursor control
- S(D) — Data movement control
- S(H) — Hardcopy control
- S(T) — Time delay
- S(W) — Temporary write control

Appendix C

Printable Dot Patterns for Sixels Mode

Figure C–1 shows the 64 printable dot patterns used for each character code in the 3/15 (63 decimal) through 7/14 (126 decimal) range. The converter subtracts 63 from the decimal value of the received code to create the dot pattern.

The character x indicates that the pixel spot prints, and the character o indicates that the pixel spot does not print.

Figure C–1: Printable Dot Patterns for Sixels Mode

Character	?	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N
Value																
Octal	077	100	101	102	103	104	105	106	107	110	111	112	113	114	115	116
Decimal	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78
Hexadecimal	3F	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E
Dot Patterns	0	x	0	x	0	x	0	x	0	x	0	x	0	x	0	x
	0	0	x	x	0	0	x	x	0	0	x	x	0	0	x	x
	0	0	0	0	x	x	x	x	0	0	0	0	x	x	x	x
	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Character	_	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n
Value																
Octal	137	140	141	142	143	144	145	146	147	150	151	152	153	154	155	156
Decimal	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110
Hexadecimal	5F	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E
Dot Patterns	0	x	0	x	0	x	0	x	0	x	0	x	0	x	0	x
	0	0	x	x	0	0	x	x	0	0	x	x	0	0	x	x
	0	0	0	0	x	x	x	x	0	0	0	0	x	x	x	x
	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

MLO–1060–87

(Continued on next page)

Figure C–1 (Cont.): Printable Dot Patterns for Sixels Mode

Character	O	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^
Value																
Octal	117	120	121	122	123	124	125	126	127	130	131	132	133	134	135	136
Decimal	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94
Hexadecimal	4F	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E
Dot Patterns	0	x	0	x	0	x	0	x	0	x	0	x	0	x	0	x
	0	0	x	x	0	0	x	x	0	0	x	x	0	0	x	x
	0	0	0	0	x	x	x	x	0	0	0	0	x	x	x	x
	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x
	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Character	o	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~
Value																
Octal	157	160	161	162	163	164	165	166	167	170	171	172	173	174	175	176
Decimal	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126
Hexadecimal	6F	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E
Dot Patterns	0	x	0	x	0	x	0	x	0	x	0	x	0	x	0	x
	0	0	x	x	0	0	x	x	0	0	x	x	0	0	x	x
	0	0	0	0	x	x	x	x	0	0	0	0	x	x	x	x
	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x
	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

MLO–1061–87

Glossary

ANSI An acronym for American National Standards Institute, an organization that compiles and publishes computer industry standards.

ASCII An abbreviation/acronym for American Standard Code for Information Interchange. ASCII is a set of binary numbers representing the alphabet, punctuation, numerals, and other special symbols used in text representation and communications protocol.

bit map An image in digitized form that can be stored, transmitted, and reproduced precisely.

character set A set of codes that describe the general appearance of a set of characters. For example, a character set might contain the code for an uppercase A or the number 1. Character sets do not describe the style of a printed character. See *Font*.

command string A data record included in a device control string. Examples are a type family identification or font identification.

conformance level An architectural agreement between hardware and software developers. Each level has a fixed group of functions within a class of operations, which are met if a product claims to implement that level.

control characters Characters that do not print, but cause the printer or terminal to perform some action. For example, the HTS control character sets a horizontal tab. There are two groups of control characters, C0 and C1.

C0 (control 0) and C1 (control 1) codes C0 codes represent 7-bit ASCII control characters. C1 codes represent 8-bit control characters that let you perform more functions than possible with C0 codes. Only in an 8-bit environment can you use C1 codes.

control function A method of controlling how the printer or terminal processes characters. Control functions include control characters, control strings, and escape and control sequences.

control sequences Two or more bytes that define a specific function. Control sequences usually include variable parameters.

decipoint A unit of measure equal to 1/720 inch.

device control strings (DCS) Device control strings define specific control functions. The DCS format includes an introducer character, a protocol selector, a command string, and a terminator.

escape sequence Two or more bytes that define a specific function. Escape sequences do not include variable parameters, but may include intermediate characters.

font The artistic representation of a typeface that describes some set of characters rendered in a particular point size, weight, and style.

GL (graphic left) and GR (graphic right) codes Two code tables in memory, reserved for printable characters. You store the character sets you want to use in GL and GR.

Printers and terminals use the graphic left (GL) table in memory when the character code format is 7-bit, or when the character code format is 8-bit and the graphic characters are in the 2/1 through 7/14 range.

They use the graphic right (GR) table in memory when the character code format is 8-bit and the graphic characters are in the 10/0 through 15/15 range.

grid units A unit measuring horizontal grid size. Horizontal grid size is the distance between adjacent dots in the output file.

horizontal margin The left horizontal margin specifies the first printable position on a line. The right horizontal margin specifies the last printable position on a line.

HLS Hue, Lightness, Saturation. HLS is a 3-parameter system for describing a color, based on human perceptual description of color.

image area The printable part of a page. On most printers you cannot print to the physical edge of the page.

IVP Abbreviation for installation verification procedure. IVP verifies that RETOS has been installed properly.

landscape printing A method of printing characters parallel to the long edge of the paper.

level 1 A level of sixel conformance. Level 1 devices do not support the Set Raster Attribute, Background Select, Horizontal Grid Size, or Macro Parameter commands. Level 1 devices parse these commands, but print in only one grid size defined for Macro Parameter 1. The VT240 is an example of a level 1 sixel device.

level 2 A level of sixel conformance. Level 2 devices support the Set Raster Attribute, Background Select, Horizontal Grid Size, and Macro Parameter commands. The LJ250 is an example of a level 2 sixel device.

node A network addressable component having a unique data link identification.

origin The starting point for printing on the page.

parameter A character that modifies the action or interpretation of a control sequence. All parameters are unsigned, positive decimal integers, with the most significant digit sent first.

- A *numeric parameter* indicates a numeric value, such as a tab or margin location. In this manual, numeric parameters appear as actual values or Pn, Pn1, Pn2, and so on.
- A *selective parameter* selects an action associated with the specific parameter value. In this manual, selective parameters appear as Ps, Ps1, Ps2, and so on.

pixel The smallest displayable picture on a screen. The printer prints pixels as dots.

portrait printing A method of printing characters parallel to the short edge of the paper. This is the normal page orientation for printing. For example, this page is printed in a portrait orientation.

printable characters Characters from position 2/0 through 7/14 in 7-bit character sets and from position 10/10 through 15/15 in 8-bit character sets.

protocol A basic procedure or set of rules that controls the communication between computers. Also, a set of conventions between communicating processes regarding the format and contents of messages to be exchanged.

ReGIS An acronym for Remote Graphics Instruction Set. ReGIS is a set of graphics object description commands.

RETOS An acronym for the ReGIS-to-sixel converter.

RGB Red, Green, Blue. RGB is a way to express colors: red, green, blue, cyan, yellow, magenta, white, and dark (black).

resolution The number of dots in a defined area. The default resolution of the LJ250 is 90 dots/inch horizontally and 90 dots/inch vertically (90,90).

sixel A group of six vertical pixels represented by six bits in a character code of seven or eight bits. Sixels are an image transfer mechanism.

vectors Lines drawn with length, width, and direction.

vertical margin The top vertical margin specifies the first printable line on a page. The bottom vertical margin specifies the last printable line. These margins are called *hard margins*, because you cannot print outside the area defined by the margins.

INDEX

A

B

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

V

W

HOW TO ORDER

ADDITIONAL DOCUMENTATION

From	Call	Write
Alaska, Hawaii, or New Hampshire	603–884–6660	Digital Equipment Corporation P.O. Box CS2008 Nashua, NH 03061
Rest of U.S.A. and Puerto Rico*	800–258–1710	
* Prepaid orders from Puerto Rico must be placed with DIGITAL's local subsidiary (809–754–7575)		
Canada	800–267–6219 (for software documentation) 613–592–5111 (for hardware documentation)	Digital Equipment of Canada Ltd. 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6 Attn: Direct Order desk
Internal orders (for software documentation)	—	Software Distribution Center (SDC) Digital Equipment Corporation Westminster, MA 01473
Internal orders (for hardware documentation)	617–234–4323	Publishing & Circulation Serv. (P&CS) NR03–1/W3 Digital Equipment Corporation Northboro, MA 01532

READER'S COMMENTS

Note: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well organized? Please make suggestions for improvement.

_____

_____

_____

_____

_____

Did you find errors in this manual? If so, specify the error and the page number.

_____

_____

_____

_____

Please indicate the type of user/reader that you most nearly represent:

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify) _____

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code _____
or Country

Do Not Tear — Fold Here and Tape

digital

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
CORPORATE USER PUBLICATIONS
MLO5–5/E45
146 MAIN STREET
MAYNARD, MA 01754–2571

Do Not Tear — Fold Here

Cut Along Dotted Line

DIGITAL EQUIPMENT CORPORATION